

CLUB
Computer

**Работа с dBASE
для Windows
на примерах**

С.Каратыгин А.Тихонов

РАБОТА В dBASE для WINDOWS на примерах

Версии 5.0 и 5.5



БИНОМ

Москва 1995

**Издание выпущено
при содействии Комитета РФ по печати**

Каратыгин С.А., Тихонов А.Ф.

Работа в dBASE для Windows на примерах: М.: БИНОМ. — 464 с.: ил.

ISBN 5-7503-0075-7

Книга знакомит с версиями 5.0 и 5.5 dBASE для Windows — одной из наиболее популярных систем управления базами данных. Читателю предлагается наиболее эффективный способ обучения работе в интерактивном режиме с dBASE для Windows. Приведена информация о большинстве утилит dBASE, используемых в интерактивном режиме.

© Восточная Книжная Компания, 1995
© Каратыгин С.А, Тихонов А.Ф., 1995
© Художественное оформление.
Н. Лозинская, 1995

ISBN 5-7503-0075-7

ВВЕДЕНИЕ.....	15

Часть I. Основные средства dBASE

ГЛАВА 1. ЧТО ТАКОЕ DBASE?	21
Что такое база данных ?.....	21
Реляционные базы данных.....	22
Объекты dBASE.....	26
Запуск dBASE.....	27
Знакомство с экраном dBASE.....	29
Панель инструментов в главном окне dBASE	30
Система меню dBASE.....	32
Выбор команд меню	32
Отказ от выбора команды	33
Немного о клавиатуре.....	34
Специальные клавиши	34
Функциональные клавиши.....	35
Цифровая часть.....	35
Комбинации клавиш	35
Получение справочной информации	35
Инструктор	38
Окно навигатора.....	39
Свойства навигатора.....	40
Выбор каталогов.....	41
Упорядочение файлов.....	42
Управление объектами	43
Псевдонимы.....	45
Определение псевдонима	45
Изменение существующего псевдонима	46
Удаление псевдонима	46
Окно папки.....	46
Создание папки.....	49
Добавление объекта в папку	49
Удаление объектов	51
Свойства окна папки.....	51
Выход из dBASE.....	51
ГЛАВА 2. СОЗДАНИЕ ТАБЛИЦ	54
Структура таблиц в dBASE.....	54
Наименования полей	55
Типы данных dBASE и Paradox для Windows.....	56
Текстовые поля	57
Числовые поля фиксированного формата и числовые поля с плавающей точкой	58
Поля дат	58
Текстовые поля произвольной длины (Мемо-поля).....	59
Поля вложенных объектов	59
Логические поля	59
Поля двоичного типа.....	60

Индексы	60
Определение структуры таблицы	62
Сохранение структуры таблицы	66
Задание имени таблицы	67
Открытие вновь созданной таблицы	68
Отказ от структуры таблицы	68
Модификация структуры таблицы	69

ГЛАВА 3. ПРОСМОТР И РЕДАКТИРОВАНИЕ ДАННЫХ..... 72

Ввод данных в таблицы	72
Редактирование поля	77
Настройка параметров ввода	79
Ввод данных в поля дат	80
Ввод данных в числовые поля	81
Ввод данных в Метод-поле	81
Настройка редактора Метод-поля	83
Ввод данных в двоичные поля	84
Ввод данных в OLE-поле	88
Сохранение данных и отказ от сделанных изменений	88
Поиск требуемой записи	89
Ускоренный способ поиска требуемой величины	89
Редактирование данных в таблицах	94
Удаление записи	95
Удаление и восстановление группы записей	97
Очистка записи	98
Добавление новой записи	98
Редактирование полей	98
Замена данных	102
Параметры отображения записей	104
Редактирование данных в режиме бланка, перечня и формы	105

ГЛАВА 4. ДОПОЛНИТЕЛЬНЫЕ СРЕДСТВА ДЛЯ ПРОСМОТРА И ПРОВЕРКИ ДАННЫХ.....109

Изменение параметров экранного образа таблицы	109
Изменение расположения и размера окна	110
Перемещение столбцов	110
Изменение ширины столбцов	111
Изменение свойств таблицы	113
Выбор полей, отображаемых в окне таблицы	115
Работа с построителем выражений	116
Форматирование данных	119
Форматирование текстовых полей	120
Форматирование числовых полей	122
Форматирование полей дат	123
Изменение заголовка поля	123
Запрет изменения значения поля	124
Проверка достоверности данных	124
Установка минимального и максимального значений	126

ГЛАВА 5. УПОРЯДОЧЕНИЕ ТАБЛИЦ.....129

Сортировка таблиц	130
Определение результирующей таблицы	130
Определение порядка сортировки	130

Использование индексов для изменения упорядочивания	132
Управление индексами	134
Упорядочение таблицы при просмотре данных	135

ГЛАВА 6. ВЫБОР ДАННЫХ ИЗ ТАБЛИЦ С ПОМОЩЬЮ ЗАПРОСОВ

ЗАПРОСОВ	137
Что такое "Запрос по образцу"	137
Бланк запроса	138
Меню и панель инструментов при создании запроса	139
Результирующая таблица	141
Сохранение запроса	141
Выполнение запроса	141
Открытие и выполнение существующего запроса	141
Печать результирующей таблицы	142
Выбор полей для включения в результат запроса	142
Отбор полей	142
Изменение порядка сортировки записей	143
Изменение результирующей таблицы	145
Построение простых условий для выбора записей	147
Точное совпадение значений одного поля	147
Выбор пустых значений	151
Использование области условий	153
Поиск записей за текущий день	153
Функция DATE (Текущая дата)	153
Точное несовпадение значений одного из полей.	154
Оператор <> (Не совпадает с)	154
Запросы по значениям Метод-полей	155
Условие включения. Оператор \$	156
Условия совпадения с шаблонами	157
Символ шаблона *	157
Оператор шаблона ?	159
Использование шаблона в полях дат	160
Выбор записей по диапазону значений	161
Более сложные И/ИЛИ-выражения	164
Комбинация И/ИЛИ-запросов	167
Операторы .OR. и .AND.	170

ГЛАВА 7. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ ЗАПРОСОВ ПО ОБРАЗЦАМ

ПО ОБРАЗЦАМ	172
Использование вычисляемых полей	172
Выражение для вычисления	173
Условие выборки для вычисляемого поля	174
Заголовок вычисляемого поля	174
Результат вычисления над пустыми полями	177
Редактирование записей с помощью результирующей таблицы	178

ГЛАВА 8. СОЗДАНИЕ ФОРМ

СОЗДАНИЕ ФОРМ	181
Создание форм с помощью экспертов	182
Создание форм с помощью конструктора	184
Работа в окне конструктора форм	186
Панель инструментов	187
Меню окна конструктора форм	188
Панель элементов	190

Выбор объектов.....	192
Изменение свойств объектов.....	195
Настройка свойств конструктора форм.....	196
Размещение объектов интерфейса.....	198
Размещение линий.....	198
Размещение текста.....	199
Настройка цвета объектов.....	203
Размещение графических изображений.....	205
Размещение OLE-объекта.....	211
Размещение кнопки.....	211
Размещение специальных объектов интерфейса.....	211
Группировка объектов.....	213
Размещение полей таблицы.....	214
Определение полей типа Рисунок.....	219
Определение вычисляемых полей.....	226
Размещение объекта типа Список.....	229
Размещение вспомогательной информации.....	231
Удаление поля.....	232
Изменения форматов полей.....	232
Размещение объекта-таблицы.....	233
Управление группами объектов.....	233
Порядок расположения объектов по слоям и порядок обхода объектов формы.....	234
Копирование объектов.....	235
Выравнивание объектов.....	236
Дополнительные замечания по созданию форм.....	238
Цветовая схема.....	238
Вспомогательные тексты и заголовки полей.....	238
Формат.....	239
Тестирование форм.....	240
Сохранение форм.....	240
Печать формы.....	241
ГЛАВА 9. ОТЧЕТЫ.....	243
Рекомендации, предшествующие созданию отчета.....	244
Создание отчета.....	245
Создание полей.....	245
Определение параметров страницы отчета.....	247
Использование областей в отчетах.....	248
Раздел заголовка страницы.....	249
Область данных.....	250
Раздел примечания страницы.....	250
Область группы.....	250
Раздел итоговых значений.....	251
Изменение области.....	251
Контекстное меню разделов отчета.....	251
Форматирование раздела.....	252
Управление объектами интерфейса.....	254
Выбор объектов.....	255
Свойства объектов.....	256
Размещение прямоугольников, линий.....	256
Размещение текста.....	257
Размещение графических изображений.....	258
Размещения OLE-объекта.....	258

Размещение полей	258
Изменение размеров объектов	269
Отображение полей в двух или более строках	270
Печать отчетов и отображение отчетов на экране	271
Просмотр отчета на экране	271
Печать отчета	271
Сохранение формата отчета	273
Отказ от текущего формата отчета	273
Модификация созданных форматов отчетов	274
Примеры создания специальных отчетов	274
Дополнительные замечания по созданию отчетов	281

ГЛАВА 10. УПРАВЛЕНИЕ ТАБЛИЦАМИ DBASE.....284

Меню команды Таблица	284
Выбор основного индекса	285
Обновление индексов	286
Действия над записями в таблицах	286
Добавление записей в текущую таблицу	286
Добавление записей со случайными данными	288
Определение количества записей таблицы, удовлетворяющих заданному условию	289
Финансовые и статистические вычисления	290
Экспорт данных	291
Импорт данных из электронных таблиц	294
Выбор языкового драйвера	296

Часть II. Многотабличные базы данных

ГЛАВА 11. ВВЕДЕНИЕ В МНОГОТАБЛИЧНЫЕ БАЗЫ ДАННЫХ.....301

Реляционные базы данных	301
Отношение один-к-одному	303
Отношение один-ко-многим	304
Отношение много-к-одному	305
Отношение много-ко-многим	305
Проектирование нормализованных баз данных	306
Первая нормальная форма таблицы	308
Вторая нормальная форма таблицы	310
Третья нормальная форма таблицы	310

ГЛАВА 12. ЗАПРОСЫ К МНОГОТАБЛИЧНЫМ БАЗАМ ДАННЫХ 312

Связывание таблиц	312
Установка связи	312
Отображение полей в результирующей таблицы	313
Проверка целостности данных	314
Запросы к связанным таблицам с отношением	314
один-к-одному	314
Запросы к связанным таблицам с отношением	317
один-ко-многим	317
Использование логического И/ИЛИ в многотабличных запросах	318
SQL — язык управления данными	319
Формирование запросов в SQL	320
Простая выборка	322
Выборка уникальных значений	322

Выборка вычисляемых значений	323
Ограничение выборки	324
Выборка с упорядочением	325
Выборка с использованием оператора BETWEEN (между)	326
Выборка с использованием оператора IN (принадлежит)	327
Выборка с использованием шаблонов	327
Выборки из связанных таблиц	328
Вычисление групповых значений	330
Использование группировки данных	332
Использование квантора существования в запросах	334
Объединение множеств	335
Модификация данных в таблицах	336
Удаление данных	337
Добавление записей	338
ГЛАВА 13. МНОГОТАБЛИЧНЫЕ ФОРМЫ	340
Просмотр данных из нескольких таблиц	340
Построение запроса для многотабличной формы	341
Создание многотабличной формы с помощью эксперта	342
Доработка многотабличной формы	344
Определение свойств объекта-таблицы	345
Создание многотабличной формы	347
Определение источника данных формы	347
Использование объекта-таблицы	348
Создание многотабличных форм со сложными связями	352
Сохранение целостности данных в многотабличных базах данных	357
ГЛАВА 14. ОТЧЕТЫ ДЛЯ СВЯЗАННЫХ ТАБЛИЦ И ГРУППИРОВКА ДАННЫХ	359
Отчеты для многотабличных баз данных	359
Определение запроса для многотабличного отчета	359
Группировка данных в отчетах	362
Определение группы	363
Группировка по диапазону значений для полей типа даты	368
Удаление групп из отчета	371
Переопределение групп	372
Перекрестные таблицы	373
Создание перекрестных таблиц	373
Одномерные перекрестные таблицы	374
Типы групповых операций	377
Двумерные перекрестные таблицы	378
Группировка по двум полям	381
Использование групп отчета при создании перекрестной таблицы	383
Создание перекрестных таблиц с использованием запроса	383
Настройка элементов перекрестной таблицы	385
Экспорт перекрестной таблицы	385
ГЛАВА 15. СИСТЕМА МЕНЮ ПРИЛОЖЕНИЯ	387
Планирование приложения	387
Создание меню	389
Главное меню	390
Окно конструктора меню	390
Сохранение меню	392

Определение назначенных клавиш пункта меню	392
Определение клавиш быстрого доступа	393
Определение сообщения для пункта меню	393
Создание ниспадающего меню	394
Создание разделителей элементов меню	396
Применение меток для переключения пунктов меню	396
Определение действий для пунктов меню	398
Блокирование команд меню	401
ГЛАВА 16. ОБМЕН ДАННЫМИ С ДРУГИМИ ПРИЛОЖЕНИЯМИ	
WINDOWS	403
Работа с составными объектами (OLE)	403
Встраивание OLE-объектов	404
Связывание OLE-объектов	407
Удаление OLE-объекта	409
Диаграммы	409
ГЛАВА 17. СОЗДАНИЕ СПРАВОЧНОЙ СИСТЕМЫ	416
Использование строки состояния	416
Создание и использование справочной системы в стандарте Windows	417
Справочная система в стандарте Windows	417
Создание тем и ключей для их поиска	418
Трансляция справочной системы	421
Раздел [Files]	421
Раздел [Options]	421
Раздел [Map]	422
Пример трансляции справочной системы	422
Организация перекрестных ссылок	422
Создание пояснений к терминам	424
ГЛАВА 18. НОВЫЕ ВОЗМОЖНОСТИ DBASE ДЛЯ WINDOWS	
ВЕРСИИ 5.5.....	426
Новые эксперты	426
Эксперт по созданию формы	427
Эксперт по созданию таблицы	428
Эксперт по созданию отчетов	428
Эксперт по созданию этикеток	429
Новые классы объектов	430
Класс PopUp	430
Класс Shape	430
Класс PaintBox	431
Класс TabBox	431
Класс MenuBar	432
Новые свойства и события объектов	432
Свойство Icon	432
Свойства Hint и Showhint	432
Свойство DesignView	433
Событие Onchar	433
События Onkeydown и Onkeyup	433
Конструктор форм	433
Схема отображения данных	433
Использование специальных объектов интерфейса	433
Использование базовых форм	435

Новые возможности панели инструментов.....	436
Конфигурирование панели инструментов с помощью таблицы регистрации классов	437
Панель полей	437
Целостность данных.....	437
Встроенный SQL.....	439
Защита данных и ограничение доступа	440
NULL значение.....	442
Расширенные средства работы с удаленными базами данных.....	442
Уровни изоляции транзакции	442
Хранимые процедуры	443
Новые типы графических форматов	443
Компилятор dBASE.....	443
Создание файлов установки.....	445
Поддержка Windows 95	446
ПРИЛОЖЕНИЕ I. УСТАНОВКА И НАСТРОЙКА DBASE.....	449
ПРИЛОЖЕНИЕ II. ИСПОЛЬЗОВАНИЕ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА	455

Введение

dBASE для Windows представляет собой систему управления базами данных, доступную для пользователя любого уровня. Эта система обладает ориентированным на пользователя развитым диалоговым интерфейсом. В версии 5.0 этого продукта появилась возможность выполнение запроса по образцу. Запрос предоставляет пользователю возможность быстро и эффективно осуществить выборку требуемых данных и обработать их.

dBASE для Windows содержит как интерактивные средства так и интегрированную среду программирования на объектно-ориентированном языке dBASE. Многочисленные интерактивные средства отличает простота использования, которая достигается за счет интуитивного интерфейса. Эти средства предоставляют вам широчайшие возможности по обработке информации, не прибегая к программированию. Язык dBASE является объектно-ориентированным, управляемым по событиям, визуально-программируемым языком, с помощью которого вы сможете добавить программные коды к отдельным объектам форм и самим формам, а также создать отдельные библиотеки и законченные приложения (как сетевые, так и автономные).

Настоящая книга знакомит вас с интерактивными средствами dBASE для Windows 5.0. Поэтому она представляет интерес практически для всех групп пользователей, начиная от новичка, и заканчивая искушенными пользователями. При написании книги мы использовали подход "на примерах", при котором вам предлагается в процессе знакомства с новыми средствами самостоятельно выполнять ряд примеров. С нашей точки зрения такой подход к обучению является наиболее эффективным.

Те, кто не знакомы с современными системами управления базами данных, в первой главе смогут познакомиться с проблематикой баз данных. Однако, мы рекомендуем познакомиться с этой главой также тем, кто знаком с основными средствами dBASE для DOS, или имеет опыт работы с другими системами управления базами данных.

В книге вы найдете все необходимые сведения о средствах dBASE для Windows, используемых в интерактивном режиме. С этой книгой вы пройдете все шаги в изучении dBASE от создания собственной базы данных до средств, используемых при создании законченных приложений. Книга содержит большое количество иллюстраций, которые помогут вам эффективно ознакомиться с изложенным в ней материалом.

Тем пользователям, которые только приступают к изучению dBASE, мы рекомендуем последовательно знакомиться с материалом, изложенным в книге. Искушенный пользователь может пропустить отдельные главы и обратиться непосредственно к интересующему его материалу. В конце глав вы найдете заключение, содержащее краткие сведения о средствах, изложенных в данной главе.

В первой части (гл. 1 — 10) будут рассмотрены средства для создания баз данных, содержащих одну таблицу, ввод данных и обработка информации, содержащейся в таких базах данных. В ней вы познакомитесь с конструкторами форм и отчетов. Глава 1 является введением в реляционные базы данных. В ней также содержатся сведения о запуске dBASE и использовании панели управления, меню и справочной системы.

В главах 2, 3 мы покажем вам как создавать таблицу, вводить и редактировать данные в таблице, а также просматривать содержимое таблицы на экране. Глава 4 знакомит со средствами управления отображения информации на экране, а также со средствами проверки вводимых и редактируемых данных.

Главы 5 — 7 посвящены средствам для упорядочивания и выборки данных из таблиц. Вначале (глава 5) рассматривается простейшая из этих операций — сортировка данных в таблицах. Использование запросов по образцу обсуждается в главах 6 и 7. В них вы узнаете как создать запрос для выбора из таблицы данных, удовлетворяющих определенным критериям. Там же мы покажем как в процессе выполнения запроса выполнить вычисления, основанные на данных из исходной таблицы, и использовать запрос в качестве средства редактирования данных.

В главах 8 и 9 освещены средства отображения данных. В главе 8 вы познакомитесь с созданием и использованием специальных форм. Генерация отчетов, использующих данные из ваших таблиц, обсуждается в главе 9. Вы познакомитесь с мощнейшими средствами — конструкторами, которые используются при создании форм и отчетов.

Средства для управления таблицами dBASE будут рассмотрены в главе 10. Вы узнаете как добавить данные из одной таблицы в другую, экспортировать и импортировать данные.

Вторая часть включает в себя главы 11 — 18. Глава 11 является введением в многотабличные базы данных. В ней вы узнаете как проектировать базы данных, содержащие несколько таблиц, и эффективно распределить данные между таблицами.

Выполнение запросов для связанных таблиц будет обсуждаться в главе 12. Вы познакомитесь с использованием запросов по образцу для связанных таблиц, а также языком SQL, являющимся мощным средством выборки данных.

Глава 13 посвящена созданию форм для нескольких связанных таблиц. В ней рассмотрены средства создания многотабличных форм, а также их использования для просмотра, редактирования и ввода данных в связанные таблицы.

В главе 14 будет показано как в одном отчете напечатать связанные данные из нескольких таблиц. В ней также рассматривается группировка данных при создании отчетов. Кроме этого вы узнаете как построить перекрестные таблицы на основании данных из ваших таблиц.

Глава 15 посвящена созданию меню приложения, с помощью которого вы можете объединить отдельные формы, отчеты и запросы в законченное приложение.

В главе 16 введены средства, которые позволяют осуществить непосредственную связь между dBASE и другими приложениями, работающими под Windows. Будет рассмотрен механизм работы с составными документами (Object Linking and Embedding — OLE). Кроме этого вы узнаете как построить диаграммы с помощью электронных таблиц.

В главе 17 мы покажем вам как создавать справочную систему в стандарте Windows, а в главе 18 рассмотрим новые средства, которые добавлены в версию dBASE для Windows 5.5.

Книга содержит два приложения. В приложении I вы найдете сведения об установке и настройке dBASE для Windows. Приложение II посвящено основным принципам работы с графическим интерфейсом dBASE.

Используемые в книге обозначения

Объекты	Соглашение	Примеры
Файлы и каталоги DOS	Прописные буквы	DBASEWIN\WORK, A:\INSTALL
Наименования таблиц, операторов и резервных слов	Прописные буквы	ADDRESS, SUM, DELETE
Клавиши на клавиатуре или на панели управления	Курсив	<i>F2, Alt+A</i>
Пункты меню	Полужирный шрифт	Файл, Операции
Кнопки на панели управления	Полужирный шрифт	Открыть таблицу, Печать
Сообщения dBASE, отображаемые на экране	Шрифт равной ширины	Выбор категории файлов dBASE

При создании иллюстраций был использован метод "экранной копии". Все приведенные в книге рисунки и меню представляют собой экранные копии.

Авторы выражают благодарность российскому представительству компании Borland Int., любезно предоставившей нам локализованную версию dBASE для Windows 5.0 и бета-версию dBASE для Windows 5.5, что позволило подготовить эту книгу к выходу нового продукта.

Мы искренне надеемся, что данная книга будет полезна самому широкому кругу отечественных пользователей dBASE.

*Каратыгин С.А.,
Тихонов А.Ф.*

Часть 1

ОСНОВНЫЕ СРЕДСТВА dBASE

Глава 1. Что такое dBASE?

В настоящей главе вы познакомитесь с основными понятиями баз данных и получите общие сведения о системе dBASE. В ней вы найдете необходимые сведения о том, как запустить dBASE и получить помощь в процессе работы, познакомитесь с экраном dBASE и узнаете о том, как найти требуемые файлы.

Что такое база данных ?

В настоящее время термины *база данных* и *система управления базами данных* используются исключительно как относящиеся к компьютерам. В общем смысле термин база данных можно применить к любой совокупности связанной информации, объединенной вместе по определенному признаку. Например, в качестве базы данных можно рассматривать расписание движения поездов или книгу регистрации данных о заказах покупателей и выполнении заказов. При этом в качестве базы данных рассматривается только набор данных, организованных определенным образом. Поэтому ваша записная книжка, если только вы не записываете в нее информацию в виде таблицы, не может считаться базой данных.

Большинство баз данных, независимо от того, реализованы ли они на компьютере или нет, для хранения данных используют таблицы. Каждая таблица состоит из строк и столбцов, которые в компьютерных базах данных называются *записями* и *полями*, соответственно. Например, книга регистрации заказов покупателей может содержать следующие сведения:

Код покупателя	Код товара
Дата заказа	Количество заказанного товара
Дата поставки	Количество поставленного товара
Цена	Примечание

Записи о заказах и продажах товаров записываются в книгу регистрации в табличном виде в хронологическом порядке (рис. 1.1). Информация имеет постоянную структуру данных. Каждую строку можно рассматривать как единичную запись. Информация внутри записи состоит из полей. В рассматриваемом примере это поля "Код покупателя", "Код товара", "Цена" и т.п.

Номер заказа	Код покупателя	Код товара	Дата заказа	Заказано	Дата продажи	Продано	Цена
1	2	3	1-Янв-95	10	12-Янв-95	8	1,025.00
2	2	1	2-Янв-95	2	12-Янв-95	2	2,600.00
3	2	2	3-Янв-95	15	17-Янв-95	15	625.00
4	6	2	5-Янв-95	20	23-Янв-95	20	620.00
5	7	8	5-Янв-95	50	23-Янв-95	50	27.00
6	11	2	10-Янв-95	1	27-Янв-95	1	635.00
7	11	4	10-Янв-95	20	21-Янв-95	20	220.00
8	14	8	10-Янв-95	10	30-Янв-95	10	30.00
9	2	1	12-Янв-95	14			
10	16	3	15-Янв-95	15	27-Янв-95	15	1,250.00
11	18	5	20-Янв-95	10	29-Янв-95	10	275.00
12	18	1	21-Янв-95	10	31-Янв-95	10	2,400.00
13	2	2	1-Фев-95	7	11-Фев-95	7	630.00
14	2	2	5-Фев-95	10	20-Фев-95	10	645.00
15	10	5	10-Фев-95	20	23-Фев-95	15	270.00

Рис. 1.1. Книга регистрации заказов.

Все записи состоят из одинаковых полей. Данные для одного поля во всех записях имеют одинаковый тип, но разные поля могут иметь разный тип данных. На рис. 1.2 выделены одна запись и одно поле в книге учета заказов.

Номер заказа	Код покупателя	Код товара	Дата заказа	Заказано	
1	2	3	1-Янв-95	10	
2	2	1	2-Янв-95	2	
3	2	2	3-Янв-95	15	
4	6	2	5-Янв-95	20	Запись
5	7	8	5-Янв-95	50	
6	11	2	10-Янв-95	1	
7	11	4	10-Янв-95	20	
8	14	8	10-Янв-95	10	
			Поле		

Рис. 1.2. Запись и поле в книге регистрации заказов.

Основным назначением баз данных является быстрый поиск содержащейся в них информации. Если вам требуется найти номер телефона абонента, зная его фамилию, имя и отчество, вы быстро сможете получить эту информацию из телефонно-адресной книги. Однако, если записи в книге заказов упорядочены в хронологическом порядке, вам придется потратить значительное время на получение списка клиентов, проживающих в определенном городе, или на выяснение того, какая из моделей компьютеров имела наибольший спрос у покупателей за последние три месяца.

Ручные базы данных имеют еще один существенный недостаток — их очень трудно модифицировать. Например, добавление двух новых рейсов может потребовать от вас изменение расписания полетов всех самолетов.

Использование компьютеров позволило устранить многие проблемы, свойственные ручным базам данных. Компьютерные базы данных обеспечивают высокую скорость поиска информации. В таких базах данных пользователь может выполнить поиск как по фамилии, так и по адресу абонента.

Кроме того, компьютерные базы данных обладают высокой компактностью. База данных, содержащая несколько тысяч записей, может поместиться на одной дискете.

Существует значительное количество программ, которые позволяют организовать информацию и поместить ее в таблицу. Однако это не означает, что любая из программ предоставляет вам широкие возможности по эффективному управлению вашей базой данных. Например, программы подготовки текстов имеют очень ограниченные возможности при их использовании в качестве систем управления базами данных.

Реляционные базы данных

В последнее время наибольшее распространение получили *реляционные базы данных*. В реляционных базах данных информация хранится в одной или нескольких таблицах. Связь между таблицами осуществляется посредством значений одного или нескольких совпадающих полей.

Каждая строка таблицы в реляционных базах данных уникальна. Для обеспечения уникальности строк используются *ключи*, которые содержат одно или несколько полей таблицы. Ключи хранятся в упорядоченном виде, что обеспечивает прямой доступ к записям таблицы во время поиска.

dbASE 5.0 для Windows

Файл Правка Вид Таблица Свойства Окно ?

Таблица [OS.DBF]

Зан	КОД ПОКУП	КОД ТОВАРА	ДАТА ЗАКА	ЗАКАЗАН	ДАТА ПРОД	ПРОДАНО	ЦЕНА
6		2	2 13.02.95		500 28.02.95	450	605.00
7		2	3 01.01.95		10 12.01.95	8	1025.00
8		2	5 12.04.95		10 22.04.95		270.00
9		2	8 05.03.95		100 17.03.95	100	27.00
10		6	2 05.01.95		20 23.01.95	20	620.00
11		7	4 05.03.95		20 17.03.95	20	210.00
12		7	8 05.01.95		50 23.01.95	50	27.00
13		10	1 25.02.95		3 01.03.95	3	2570.00
14		10	4 26.02.95		20 05.03.95	15	220.00
15		10	5 10.02.95		20 23.02.95	15	270.00
16		11	1 10.02.95		1 17.02.95	1	2500.00
17		11	2 10.01.95		1 27.01.95	1	635.00
18		11	3 10.02.95		1 19.02.95	1	1100.00
19		11	4 10.01.95		20 21.01.95	20	220.00
20		12	1 05.03.95		2 16.03.95	2	2650.00
21		12	6 10.03.95		14 21.03.95		830.00
22		12	7 25.02.95		5 10.03.95	5	1100.00
23		12	8 16.04.95		25 20.04.95		28.00

Os.dbf Зан 6/42 Ins

Рис. 1.3. База данных OS (Регистрация заказов и продаж).

dbASE 5.0 для Windows - (Report (1.RPT))

Файл Вид Отчет Свойства Окно ?

СПИСОК ТЕЛЕФОНОВ И АДРЕСОВ

140785 Страница 1

ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ПРЕДПРИЯТИЕ	ТЕЛЕФОН
Алексеев	Сергей	Иванович	АО Интерсервис	3832243124
Ткаченко	Марина	Михайловна	АО Компьютерсервис	0480564634
Морозов	Юрий	Анатолевич	ТОО Интерком	0480345874
Петров	Михаил	Николаевич	Банк Программ	8160256128
Ильян	Филипп	Петрович	ИЧП Эланс	8160256156
Петров	Михаил	Николаевич	Bel-ABM	8160177124
Зайцева	Людмила	Николаевна	СП КомпСофт	8635212453
Загребский	Андрей	Анатолевич		8312564216

Ins

Рис. 1.4. Табличный отчет для таблицы ADDRESS.

Для взаимодействия пользователя с базами данных используются *системы управления базами данных (СУБД)*. Современные СУБД обеспечивают:

- набор средств для поддержки таблиц и отношений между связанными таблицами;

- развитый пользовательский интерфейс, который позволяет вам вводить и модифицировать информацию, выполнять поиск и представлять информацию в текстовом или графическом виде;
- средства программирования высокого уровня, с помощью которых вы можете создать собственные приложения.

На рис. 1.3 показана книга учета заказов, представленная в виде реляционной базы данных в dBASE. С помощью средств СУБД вы можете например:

- выбрать информацию, представляющую для вас интерес. Например, вы можете получить сведения о всех покупках определенного клиента за последние два месяца;
- напечатать всю таблицу или только выбранные записи и поля в различных форматах. Например, на рис. 1.4 показан табличный отчет, содержащий телефоны и адреса абонентов из базы данных ADDRESS.

На рис. 1.5 показан один из вариантов информационного письма, содержащего имена и фамилии клиентов из той же самой базы данных.

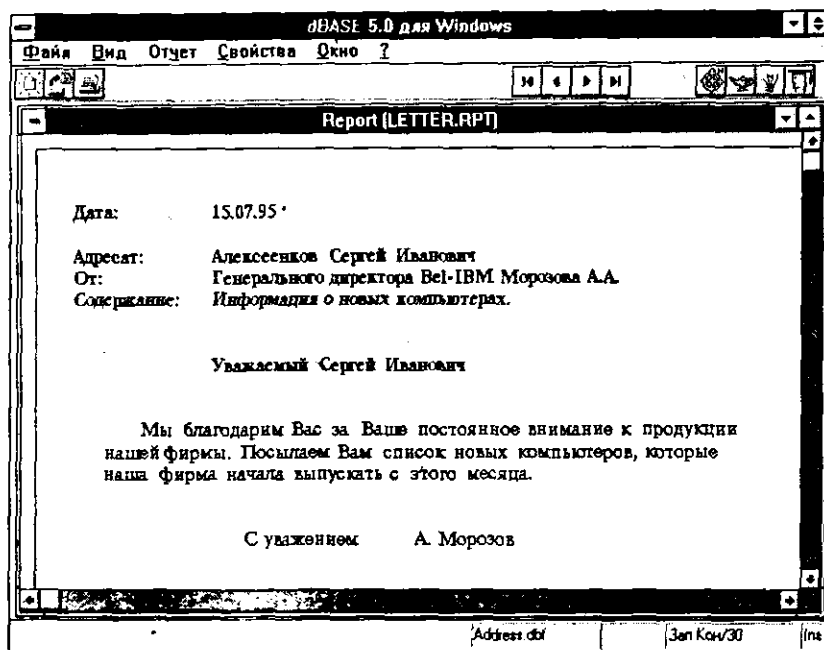


Рис. 1.5. Информационное письмо.

- переносить данные, хранимые в таблицах, в программы отображения результатов в графическом виде. В качестве примера на рис. 1.6 показано графическое представление информации из таблицы dBASE о продажах различных моделей компьютеров с помощью Microsoft Excel.
- выполнить различные вычисления в процессе подготовки отчетов или выбора данных из таблиц. Например, на рис. 1.7 для каждого клиента показана стоимость партии приобретенных им товаров.

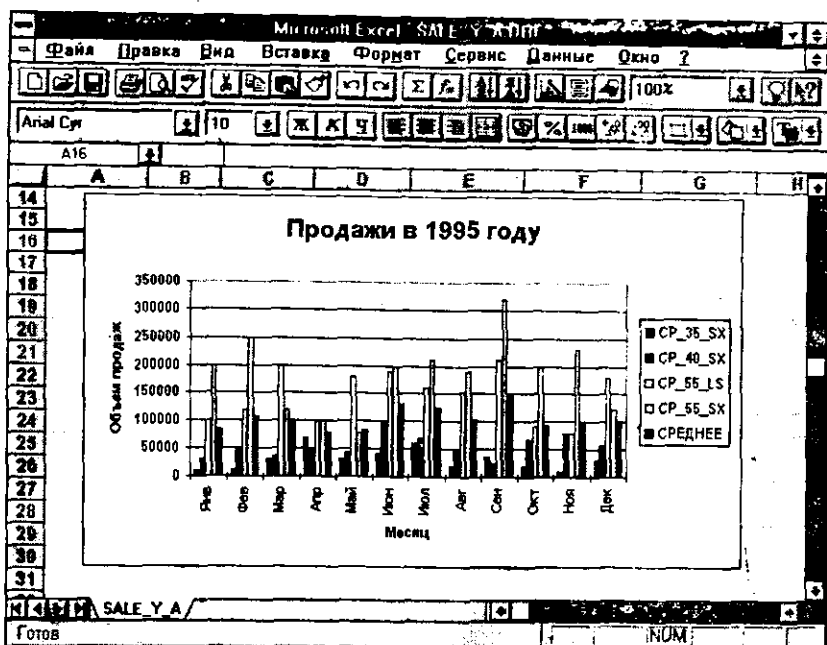


Рис. 1.6. Графическое представление информации о количестве проданных компьютеров.

dbase 5.0 для Windows

Файл Правка Вид Таблица Свойства Окно ?

Результаты запроса (ORDSALE.TOE)

Зам.	КОД ПОКУП	КОД ТОВАРА	ДАТА ПРОД	ПРОДАНО	ЦЕНА	ВСЕГО
1	1	2	12.04.95	5	510.0000	3050.00
2	2	1	12.01.95	2	2600.0000	5200.00
3	2	2	2.17.01.95	15	625.0000	9375.00
4	2	2	2.11.02.95	7	630.0000	4410.00
5	2	2	2.20.02.95	10	645.0000	6450.00
6	2	2	2.28.02.95	450	605.0000	272250.00
7	2	2	3.12.01.95	8	1025.0000	8200.00
8	2	2	5.22.04.95		270.0000	0.00
9	2	2	8.17.03.95	100	27.0000	2700.00
10	6	2	2.23.01.95	20	620.0000	12400.00
11	7	2	4.17.03.95	20	210.0000	4200.00
12	7	2	8.23.01.95	50	27.0000	1350.00
13	10	2	1.01.03.95	3	2670.0000	8010.00
14	10	2	4.05.03.95	15	220.0000	3300.00
15	10	2	5.23.02.95	15	270.0000	4050.00
16	11	2	1.17.02.95	1	2600.0000	2600.00
17	11	2	2.27.01.95	1	635.0000	635.00

Продано.doc Jan1/92

Рис. 1.7. Вычисление стоимости товаров, приобретенных клиентами.

В мире насчитывается более 50 типов СУБД для IBM PC и совместимых с ними компьютеров. Среди них наибольшее распространение получил dBASE, который был одним из первых и простых систем управления базами данных на персональных компьютерах.

Объекты dBASE

Отдельные компоненты баз данных, которые используются для хранения и представления информации, называются *объектами*. К объектам относятся: таблицы, формы, отчеты, запросы, программы и библиотеки. Все эти объекты являются *основными объектами dBASE*. Для создания форм и отчетов используются конструкторы, поэтому эти объекты получили название *конструкторских объектов*. Конструкторские объекты являются составными объектами, то есть состоят из более мелких объектов, таких как поля, кнопки, диаграммы, рамки и т.д., которые называются объектами интерфейса.

К объектам интерфейса относятся:

- текстовые объекты,
- прямоугольники и линии,
- поля,
- кнопки,
- графические объекты,
- OLE-объекты,
- страницы документов и т.д.

В dBASE вся информация содержится в таблицах. Каждая таблица имеет уникальное имя и хранится в отдельном файле, наименование которого совпадает с именем таблицы. При задании наименований таблиц действуют те же правила, что и при присвоении имен файлов в DOS. Более подробно эти правила будут рассмотрены в главе 2 при обсуждении создания таблиц. Файлам, содержащим таблицы dBASE, автоматически присваивается расширение .DBF.

Каждая создаваемая вами таблица может иметь связанные с ней индексы, а также файлы, содержащие значения Мемо, двоичных и OLE-полей. Перечисленные объекты всегда связаны с таблицей, для которой они созданы.

Каждый объект хранится в отдельном файле, причем имена файлов, содержащих основные объекты, вы задаете самостоятельно, а наименования файлов, содержащих объекты, связанные с таблицей, совпадают с именем таблицы. В зависимости от типа содержащегося в нем объекта, dBASE автоматически присваивает каждому файлу расширение, которое помогает в идентификации объекта. Список расширений файлов приведен в табл. 1.1.

Табл. 1.1. Расширения имен файлов объектов dBASE

Объекты	Расширения
Файл настройки IDAPI	CFG
Файл настройки dBASE	INI
Таблица dBASE	DBF
Мемо, OLE и двоичные поля таблицы dBASE	DBT
Непрерывно обновляемый индекс для таблицы dBASE	MDX
Обновляемый по запросу индекс для таблицы dBASE	NDX
Таблица Paradox	DB
Первичный индекс для таблицы Paradox	PX
Мемо, форматируемые, графические, двоичные и OLE поля таблицы Paradox	MB

Объекты	Расширения
Форма	WFM, FMT
Запрос	QBE, VUE, QRY
Отчет	RPT, RPC, FRG, FRM
Наклейка	RPL, LBG, LBL
Программа	PRG
Рисунок	BMP, PCX
Звукозапись	WAV
Папка	CAT

Формы используются для ввода и просмотра таблиц в окне формы. Формы позволяют ограничить объем информации, отображаемой на экране, и представить ее в требуемом виде. С помощью эксперта вы можете создать форму, помещая в нее поля исходной таблицы, расположенные в соответствии с одним из заранее созданных шаблонов. С помощью конструктора форм вы можете создавать формы любой степени сложности.

Отчеты используются для отображения информации, содержащейся в базе данных. С помощью конструктора отчетов вы можете разработать собственный отчет, включающий группировку данных, групповые и вычисляемые поля, оформив их соответствующим образом.

Запрос является средством для извлечения информации из базы данных, причем данные могут быть распределены среди нескольких таблиц. В dBASE для формирования запросов используется способ, получивший название *запроса по образцу*. Используя это средство, на основании визуальной информации вы можете не только извлечь нужную информацию из одной или нескольких таблиц, но даже изменить их значения.

Программы, написанные на языке dBASE, предназначены для выполнения различных операций. При использовании объектно-ориентированного программирования, как правило, программы связаны с объектами, размещаемыми в формах. Однако, программа может выступать и в роли независимого объекта. Такие программы выполняются независимо от наличия форм на экране.

Библиотеки предназначены для хранения программ, написанных на dBASE. Содержимое библиотек может совместно использоваться различными формами, программами и другими библиотеками.

Файл параметров настройки DBASEWIN.INI содержит параметры основного окна dBASE, настройки таблиц, параметры окон диалога и панели инструментов.

Индексы хранят ссылки, предназначенные для упорядочивания данных и ускорения поиска в таблице. Каждая таблица может иметь несколько индексов.

Мето поля содержат текстовую информацию, а двоичные поля могут содержать данные любого типа. OLE-поля используются при обмене данными с другими приложениями, работающими под Windows.

Запуск dBASE

Прежде чем запустить dBASE, его необходимо установить на вашем компьютере. Если вы еще не сделали этого, обратитесь к приложению I, в котором содержится подробное описание процедуры инсталляции.

Мы приступаем непосредственно к работе с Windows и dBASE. При этом мы предполагаем, что вы имеете представление о действиях, совершаемых с помощью мыши (нажать кнопку мыши, дважды нажать кнопку мыши, указать на объект и т.д.). Кроме того, в приложении II приведены необходимые сведения о работе с графичес-

ким интерфейсом. Далее мы считаем, что вы уже познакомились с графическим интерфейсом и описывая действия, не будем останавливаться на деталях, не являющихся специфическими для dBASE.

1. Если dBASE уже установлен на вашем компьютере, прежде всего вы должны запустить Windows. Перейдите в каталог, содержащий файлы Windows, напечатайте win и нажмите *Enter*. Расположение программы указывается при ее установке. По умолчанию Windows располагается на устройстве C: в каталоге WINDOWS. В этом случае для запуска вам необходимо ввести следующие команды:

```
C:
CD\WINDOWS
WIN
```

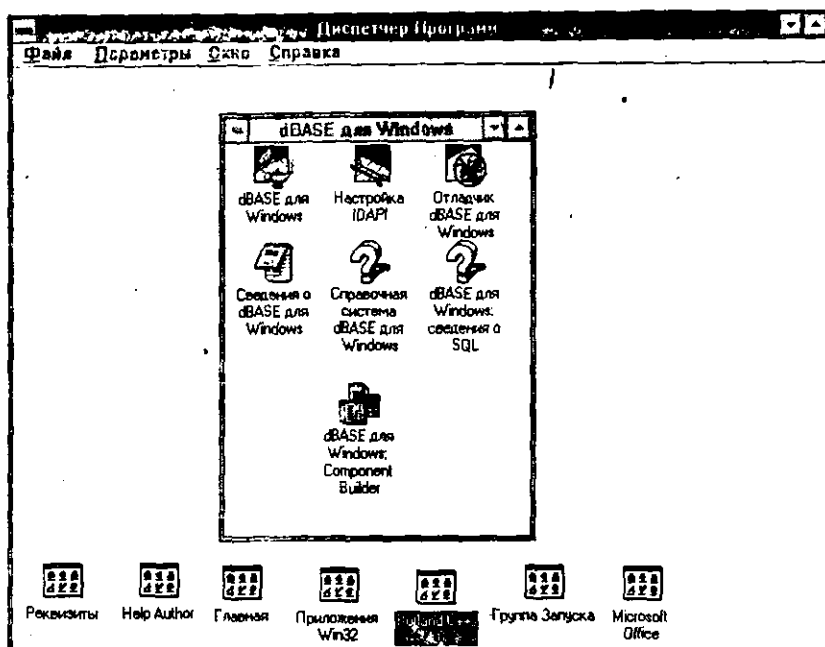


Рис. 1.8. Окно Диспетчера Программ с группой dBASE для Windows.

После запуска Windows на экране появится окно Диспетчера Программ, в котором отображаются все группы программ, установленных на вашем компьютере.

Если вы решили запустить dBASE во время работы с Windows, когда на экране отсутствует окно Диспетчера Программ, нажмите *Ctrl+Esc* и на экране появится окно диалога "Список задач", с помощью которого вы можете переключиться в Диспетчер Программ.

2. Если пиктограмма группы dBASE свернута, укажите на нее и дважды нажмите кнопку мыши. После раскрытия группы ваш экран будет выглядеть аналогично, показанному на рис. 1.8.
3. Наиболее просто запустить dBASE, указав на пиктограмму "dBASE для Windows" и дважды нажав кнопку мыши.



В качестве альтернативного способа запуска вы можете с помощью мыши или клавиатуры выбрать пиктограмму "dBASE для Windows", а затем выполнить команду **Файл | Открыть**. Независимо от того каким из способов вы воспользуетесь на экране появится главное окно dBASE (рис. 1.9).



Если каталог, содержащий системные файлы Windows, включен в PATH, вы можете запустить dBASE без предварительного запуска Windows. Для этого в ответ на приглашение DOS введите DBASEWIN, в результате чего вначале будет запущен Windows, а затем dBASE.

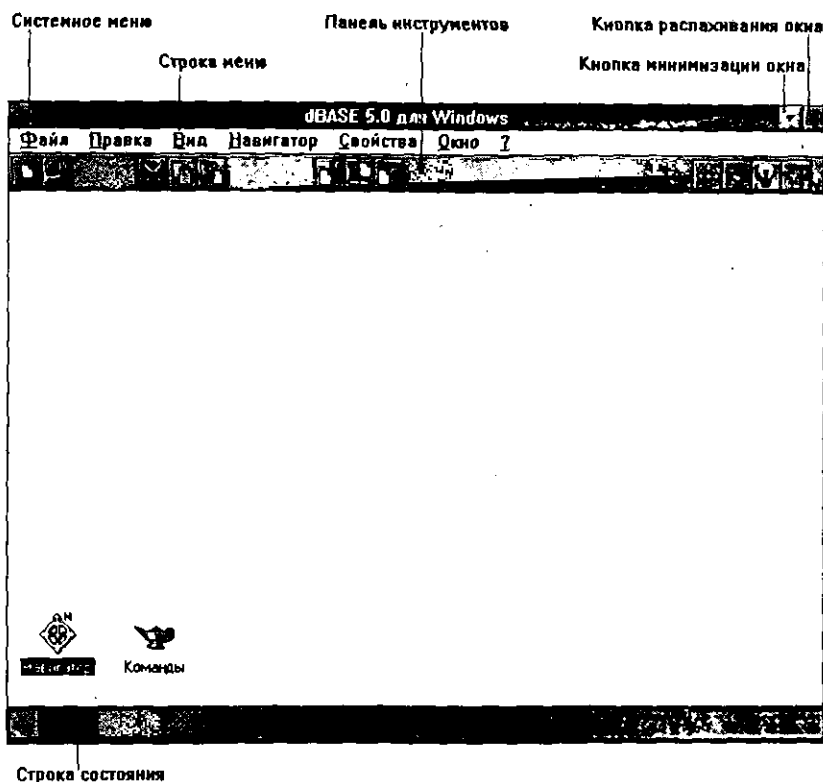


Рис. 1.9. Главное окно dBASE.

Знакомство с экраном dBASE

На рис. 1.9 показан экран dBASE, каким вы его увидите после запуска программы. Экран можно разделить на четыре основные области: строка меню, панель инструментов, строка состояния и рабочая область.

Строка меню содержит меню текущего окна. На рис. 1.9 вы видите меню главного окна, которое будет рассмотрено в следующем разделе.

Панель инструментов, расположенная под строкой меню, содержит набор кнопок, состав которых зависит от текущего окна. Большая часть кнопок предназначена для выполнения операций, которые обычно выполняются при выборе команды меню или нажатии сочетания клавиш. Остальные кнопки служат для быстрого перемещения по данным.

Строка состояния расположена в нижней части главного окна. В левой части строки состояния отображается наименование кнопки панели инструментов, на которую указывает курсор мыши. В правой части строки состояния вы можете видеть наименование таблицы и номер текущей записи, а также индикатор режима работы. Этот индикатор информирует, в каком из режимов вы работаете в настоящий момент. Например, на рис. 1.10 показана таблица ADDRESS и, как следствие, вы видите в строке состояния имя этой таблицы. В окне таблицы строка меню вместо команд основного меню содержит команды текущего окна, а на панели инструментов представлены кнопки, которыми вы можете манипулировать в данном окне.

Курсор расположен в первой записи, о чем свидетельствует сообщение Зап 1/30, где 30 — общее число записей в таблице. В строке состояния могут отображаться различные сообщения (например, о том что запись заблокирована).

Кроме главного окна в dBASE существует ряд окон, которые открываются и содержатся в главном окне. Для отображения объектов (например, таблица, форма или отчет) используются окна, специфические для каждого типа объектов. Например, на рис. 1.10 таблица ADDRESS отображается в окне таблицы.

Зап	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ТЕЛЕФОН	ИНДЕКС	СТРАНА
1	Алексеев	Сергей	Иванович	(383) 224-31-24	630090	Россия
2	Ткаченко	Марина	Михайловна	(048) 056-46-34	270005	Украина
3	Морозов	Дрий	Анатолевич	(048) 034-58-74	270029	Украина
4	Петров	Михаил	Николаевич	(816) 025-61-28	173024	Россия
5	Ильин	Филипп	Петрович	(816) 025-61-56	173003	Россия
6	Петров	Михаил	Николаевич	(816) 017-71-24	173003	Россия
7	Зайцева	Людмила	Николаевна	(863) 521-24-63	346415	Россия
8	Загребский	Андрей	Анатолевич	(831) 256-42-18	603006	Россия
9	Смешов	Филипп	Петрович	(383) 245-31-23	630056	Россия
10	Сухарев	Андрей	Анатолевич	(863) 573-45-43	357736	Россия
11	Зайцева	Дарья	Ивановна	(257) 212-24-53	142490	Россия
12	Фролов	Виктор	Михайлович	(095) 576-54-56	140160	Россия
13	Свешняков	Сергей	Иванович	(095) 453-59-85	142432	Россия
14	Иханоэ	Максим	Сергеевич	(812) 623-44-66	184230	Россия
15	Морозов	Андрей	Михайлович	(343) 232-45-42	620151	Россия
16	Райский	Максим	Сергеевич	(846) 253-45-42	443115	Россия
17	Антонов	Максим	Юрьевич	(095) 532-15-32	125167	Россия

Рис. 1.10. Таблица ADDRESS в окне таблицы.

Панель инструментов в главном окне dBASE

Каждому окну соответствует собственная панель инструментов. Набор кнопок меняется при переходе из одного окна в другое. При этом одни и те же кнопки могут присутствовать в различных окнах. На рис. 10.11 показана панель инструментов в главном окне dBASE.

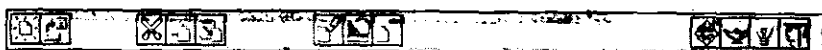


Рис. 1.11. Панель инструментов в главном окне dBASE.

Кнопки панели инструментов в главном окне имеют следующие назначения:

Кнопка	Название	Назначение
	Создать	Создает новый файл
	Открыть	Открывает существующий файл
	Удалить в буфер	Удаляет выделенный текст или выделенные объекты и помещает их в буфер
	Копировать в буфер	Копирует в буфер выделенный текст или выделенные объекты, не удаляя из таблицы
	Вставить из буфера	Копирует содержимое буфера. Конкретно действие этой кнопки зависит от того, в каком из режимов вы работаете
	Открыть выделенный элемент	Открывает выделенный в окне навигатора элемент
	Изменить выделенный элемент	Запускает соответствующий выделенному в окне навигатора элементу конструктор для его изменения
	Удалить выделенный элемент	Удаляет выделенный в окне навигатора элемент
	Навигатор	Открывает окно навигатора
	Окно команд	Открывает окно команд
	Эксперт	Осуществляет вызов эксперта, для создания специальных форм
	Инструктор	Открывает окно инструктора, в котором вы можете указать тему и урок для обучения



При работе в любом окне вы можете определить назначение кнопки на панели инструментов, указав на нее курсором мыши. В результате в левой части строки состояния появится название указанной кнопки.

Первые две кнопки предназначены для открытия существующего или создания нового объекта. В последующих главах мы подробно рассмотрим создание объектов, используемых при интерактивной работе с dBASE.

Три кнопки предназначены для работы с временным буфером Windows. Следующие за ними три кнопки используются для работы с элементами, выделенными в окне навигатора. Оставшиеся кнопки открывают окно команд, осуществляют вызов навигатора, инструктора или одного из экспертов, которые облегчают выполнение вами ряда стандартных операций.

Как и для большинства объектов dBASE, вы можете по своему усмотрению настроить внешний вид панели инструментов. Для этого выберите **Свойства | Среда dBASE**, в результате чего на экране появится окно диалога “Свойства среды dBASE”

(рис. 1.12). Это окно диалога содержит несколько карточек настройки. Выбрав из них карточку "Приложения" вы можете изменить положение панели инструментов, установить флажок отображения строки состояния и определить шрифт для сообщений.

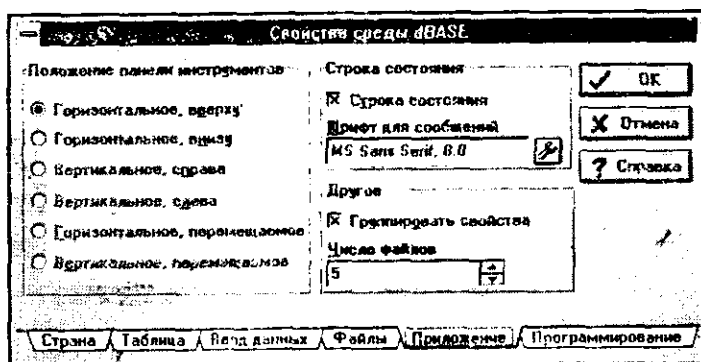


Рис. 1.12. Окно диалога "Свойства среды dBASE".

Система меню dBASE

Взаимодействие с dBASE осуществляется посредством выбора команд из текущего меню. Большинство команд являются иерархическими. В результате выбора одной из таких команд на экране появляется ее ниспадающее меню.

Окну каждого типа соответствует свой собственный набор команд и функций, применимых только к объекту данного типа. Например, в окне таблицы к командам основного меню добавлен пункт меню Таблица (рис. 1.10). Команды Файл, Правка, Окно, Свойства и ? присутствуют в меню окна любого типа и даже в том случае, когда на экране нет открытых окон.

Выбор команд меню

Для выбора команды из строки меню вы можете воспользоваться любым из следующих способов.

- Наиболее простой способ предполагает использование мыши. Для выполнения команды поместите курсор мыши на требуемую команду и нажмите кнопку мыши.
- Если вы предпочитаете пользоваться клавиатурой, вначале переместите курсор в строку меню. Для этого нажмите клавишу *F10* или *Alt*. Затем поместите световой указатель на требуемую команду и нажмите *Enter*.

Для перемещения светового указателя воспользуйтесь клавишами *←* и *→*. При нажатии на *←* вы переместитесь на команду, находящуюся левее текущей. Если же команда является самой левой, вы переместитесь в системное меню, последующее нажатие на *←* переместит вас на самую правую команду, т.е. как бы обойдете команды по кругу. Клавиша *→* имеет аналогичное действие.

- В качестве альтернативного способа выбора команды меню с помощью клавиатуры вы можете нажать *Alt*+подчеркнутый символ ее имени. При таком выборе команда будет выполняться сразу же после нажатия без последующего нажатия клавиши *Enter*.

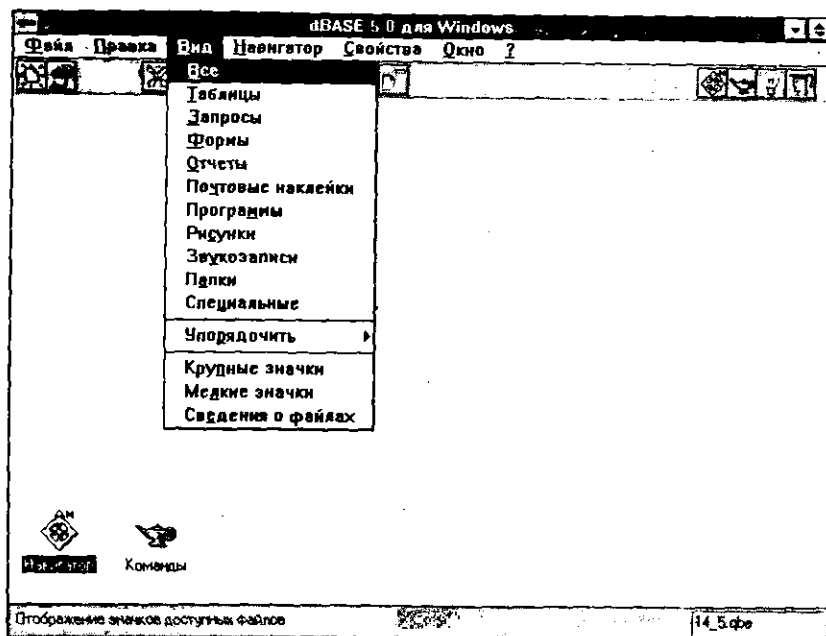


Рис. 1.13. Ниспадающее меню Вид.

На рис. 1.13 показан результат выбора команды **Вид**. Опции ниспадающего меню также могут иметь ниспадающее меню следующего уровня. Такие опции помечены символом ►. Кроме того, ряд опций после наименования содержат Это означает, что в результате выбора такой команды для продолжения операции dBASE откроет окно диалога, в котором вы должны указать значения конкретных параметров. И только опции, не имеющие символов ... , будут выполнены сразу в результате их выбора.

Процедура выбор опции из ниспадающего меню аналогична выбору команды меню, за исключением того, что вместо клавиш ← и → необходимо использовать клавиши ↑ и ↓. Если вы выбираете команду по первой букве ее наименования, вам не нужно нажимать клавишу **Alt**. На рис. 1.14 приведен результат выбора опции **Упорядочить** из меню **Вид**, которая также имеет свое ниспадающее меню.

Отказ от выбора команды

Для отказа от выбора опции ниспадающего меню после его появления на экране вы можете воспользоваться одним из следующих способов.

- Нажмите клавишу **Esc**, и вы попадете в меню предыдущего уровня. При этом в ниспадающем меню (или меню) световой указатель будет расположен на опции, выбор которой явился причиной вызова ниспадающего меню нижнего уровня.
- Нажмите клавишу **F10**, в результате все активные ниспадающие меню будут удалены с экрана.
- Нажмите мышью любую опцию в одном из ниспадающих меню предыдущего уровня. Выбранная вами команда будет выполнена, а все ниспадающие меню нижних уровней будут удалены с экрана.

- Нажмите мышью команду из строки меню. В результате все активные ниспадающие меню будут удалены с экрана, а выбранная команда будет выполнена.
- Нажмите мышью пустое место рабочей области.

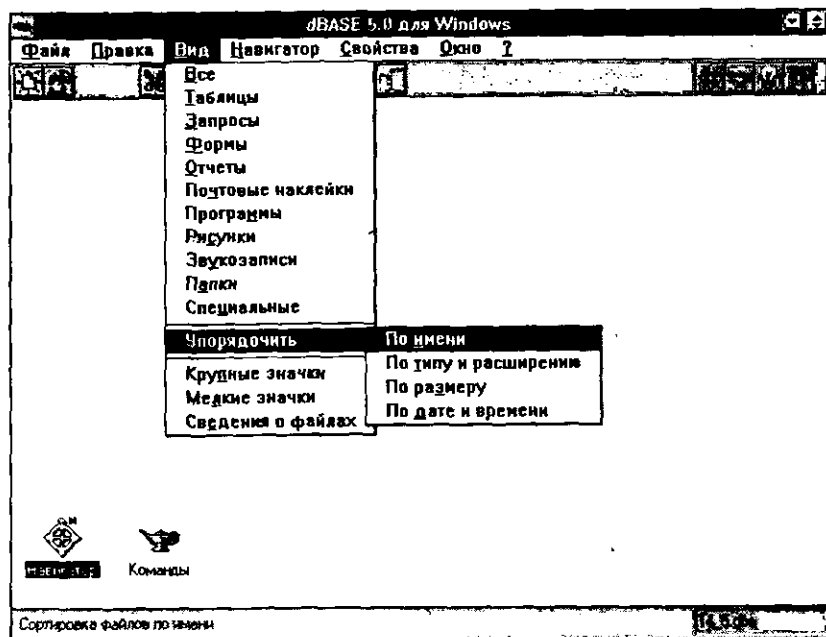


Рис. 1.14. Выбор опции Упорядочить из меню Вид.

Немного о клавиатуре

Обычно при работе с dBASE используется мышь. Однако вы можете выполнить большую часть операций с помощью клавиатуры. В процессе изучения средств dBASE мы будем описывать использование клавиатуры для выполнения операций или отмечать, что данная операция может быть выполнена только с помощью мыши.

Независимо от модификации любая IBM PC клавиатура содержит основную и цифровую части, специальные и функциональные клавиши. Сейчас мы кратко остановимся лишь на некоторых из наиболее важных клавиш. Клавиши, имеющие специальное назначение, будут рассмотрены при знакомстве с конкретными средствами dBASE.

Специальные клавиши

В зависимости от текущего режима работы вы можете использовать для перемещения курсора все или некоторые из следующих клавиш: ←, →, ↑, ↓, PgUp, PgDn, Home, End.

Клавиша Esc всегда используется в качестве средства, позволяющего возвратиться к предыдущему шагу. Например, вы можете воспользоваться клавишей Esc для перехода из ниспадающего меню нижнего уровня в меню предыдущего уровня, из которого был осуществлен его вызов, или для выхода из всплывающего меню, которое появилось на экране в результате ваших действий.

Клавиши *Alt* и *Shift* используются совместно с другими клавишами для выполнения специальных функций. Например, вы можете использовать клавишу *Alt* для быстрого доступа к пунктам меню.

Клавиша *Ctrl* также используется в сочетании с другими клавишами для выполнения специальных функций. Кроме того, эта клавиша имеет еще одно назначение — ее использование совместно с клавишами позволяет получить новые средства для перемещения курсора. Например, нажатие комбинации клавиш *Ctrl+End* в окне таблицы приведет к перемещению курсора не в конец поля, а в последнее поле.

Клавиши *Ins*, *Backspace* и *Del* выполняют свои стандартные функции.

Клавиша *Enter* является наиболее часто используемой клавишей, при этом результат ее нажатия зависит от конкретной ситуации.

Функциональные клавиши

dBASE использует функциональные клавиши от *F1* до *F12*. Вы уже познакомились с назначением одной из функциональных клавиш *F10*. Действие остальных клавиш будет обсуждаться при рассмотрении конкретных средств dBASE.

Цифровая часть

Правая часть клавиатуры содержит цифровую часть. dBASE использует клавиши этой части для двух целей. Во-первых, вы можете использовать ее клавиши для перемещения курсора.

Если вы нажмете клавишу *Num Lock*, вы можете использовать клавиши цифровой части для ввода чисел. В этом случае цифровая часть клавиатуры не может быть использована для перемещения курсора. Клавиша *Num Lock* является переключателем. При ее повторном нажатии клавиши цифровой части переключаются в режим управления курсором.

Цифровая часть также содержит клавиши *Enter*, *Ins* и *Del*, которые выполняют свои обычные функции.

Комбинации клавиш

Две или более клавиш, используемые совместно, называются комбинацией клавиш. Любая комбинация клавиш содержит одну из специальных клавиш: *Ctrl*, *Alt* или *Shift*.

Комбинация клавиш позволяет ускорить выполнение некоторых команд или получить совершенно новый эффект. Например, использование клавиши *Ctrl* в сочетании с клавишами *PgDn*, *PgUp*, *Home*, *End* позволяет вам более быстро перемещать курсор в окне таблицы. Комбинации клавиш *Ctrl+G* позволяет вам быстро перейти в таблице на требуемую запись.

Для ввода комбинации клавиш вы должны нажать клавишу, указанную в комбинации первой, и, не отпуская ее, нажать вторую клавишу. Например, для ввода *Ctrl+G* вы должны нажать клавишу *Ctrl* и, удерживая ее, нажать клавишу *G*.

Получение справочной информации

Подобно большинству наиболее популярных программ, dBASE имеет контекстно-зависимую справочную систему. Это означает, что вы можете получить справку по использованию средства, с которым работаете в настоящий момент. Для получения справочной информации вы можете воспользоваться одним из следующих способов:

➤ Выберите ? из меню.

- Нажмите *F1* в любой момент сеанса работы с dBASE. Справочная система считает, что вас интересует информация о текущей операции, и соответственно выбирает раздел справочника. Такая ситуация представлена на рис. 1.15.
- Некоторые из окон диалога содержат кнопку Справка на панели инструментов. Если вы нажмете кнопку Справка при работе в одном из окон диалога, dBASE снабдит вас информацией о том, как пользоваться этим окном.

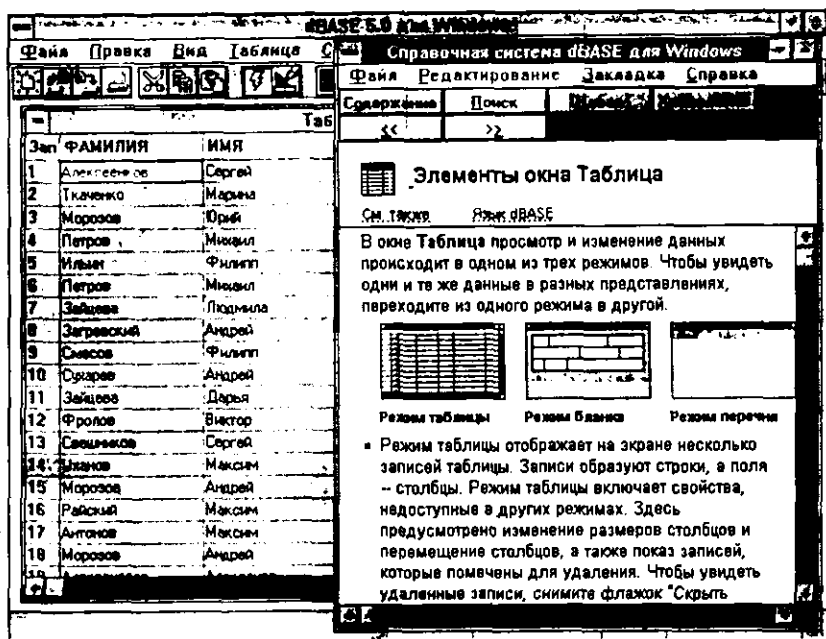


Рис. 1.15. Экран справочника dBASE.

Окно справочной системы имеет собственное меню и панель инструментов. Кнопки панели инструментов в главном окне справочной системы имеют следующее назначение:

Кнопка	Назначение
Содержание	Отображает на экране содержание справочной системы.
Поиск	Позволяет осуществить поиск требуемого раздела с помощью списка ключевых слов, определяющих содержимое раздела.
Назад	Выводит на экран последний просмотренный раздел Справки.
Хронология	Выводит на экран список 40 последних просмотренных в текущем сеансе работы разделов Справки.
<<	Выводит на экране предыдущий раздел последовательности подобных разделов до тех пор, пока не дойдет до первого раздела этой последовательности, после чего кнопка станет недоступной.
>>	Выводит на экране следующий раздел последовательности подобных разделов до тех пор, пока не дойдет до последнего раздела этой последовательности, после чего кнопка станет недоступной.

Для того чтобы выбрать кнопку Справки, укажите на требуемую кнопку и нажмите кнопку мыши или просто введите букву, подчеркнутую в названии кнопки Справки.

Если вы имеете представление в каком из разделов Справки может находиться интересующая вас информация наиболее эффективным средством ее поиска является кнопка Поиск. При ее нажатии на экране появляется окно диалога "Поиск" (рис. 1.16). Введите слово, по которому вы хотите осуществить поиск. В качестве примера было введено Вид, меню. Если справочная система обнаружит в списке требуемую тему, ее наименование будет выделено цветом или яркостью. Далее нажмите кнопку Показать Темы, и в нижней части окна появится список тем для данного раздела. Укажите на нужную вам тему и нажмите кнопку Перейти К. dBASE поместит информацию об указанной теме в окно справочной системы.

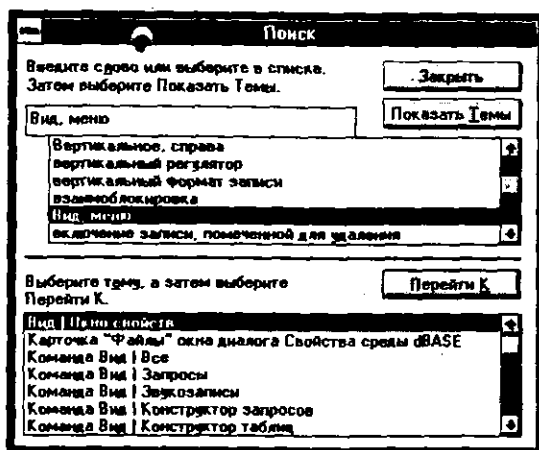


Рис. 1.16. Окно диалога "Поиск".

Вы можете последовательно получить справку по нескольким темам одного раздела, не прибегая к кнопкам Справки. Для этой цели в тексте справки слова, по которым может быть осуществлен переход выделены зеленым цветом. В справочной системе dBASE возможны переходы четыре типов:

Название	Признак	Характеристика
Полные переходы Временные переходы	Слова, соответствующие полным и временным переходам, выделяются зеленым цветом и подчеркнуты сплошной линией	Раздел, к которому вы перешли, остается на экране до тех пор, пока вы не перейдете к другому разделу
Всплывающие списки Всплывающие определения	Слова, соответствующие всплывающим спискам и определениям, выделяются зеленым цветом и подчеркнуты пунктирной линией	Информация остается на экране до тех пор, пока вы не нажмете кнопку мыши

dBASE фиксирует полные переходы, которые просматривались в течение текущего сеанса работы со справочной системой. Для того чтобы получить доступ к списку просмотренных разделов, нажмите кнопку Хронология, в результате чего dBASE от-

кроет окно диалога “Хронология Справки Windows”, показанное на рис. 1.17. Вы можете вернуться к любому из просмотренных разделов, для чего достаточно указать на его наименование и дважды нажать кнопку мыши.

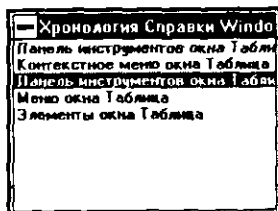


Рис. 1.17. Окно диалога “Хронология Справки Windows”.



Для получения более подробной информации о справочной системе нажмите *F1*, находясь в любом из окон справочной системы.

Закрытие окон справочной системы осуществляется стандартным способом. Для этого вы можете выбрать **Закрыть** из системного меню или нажать *Alt+F4*.

Инструктор



dBASE содержит инструктор, который может быть вызван с помощью команды **? | Инструктор** или кнопки **Инструктор**, расположенной на панели инструментов. Независимо от того, каким из средств вы воспользуетесь на экране появится окно диалога, содержащее список тем (рис. 1.18). В правой части окна отображается список уроков, которые вы можете пройти вместе с инструктором по выбранной теме.

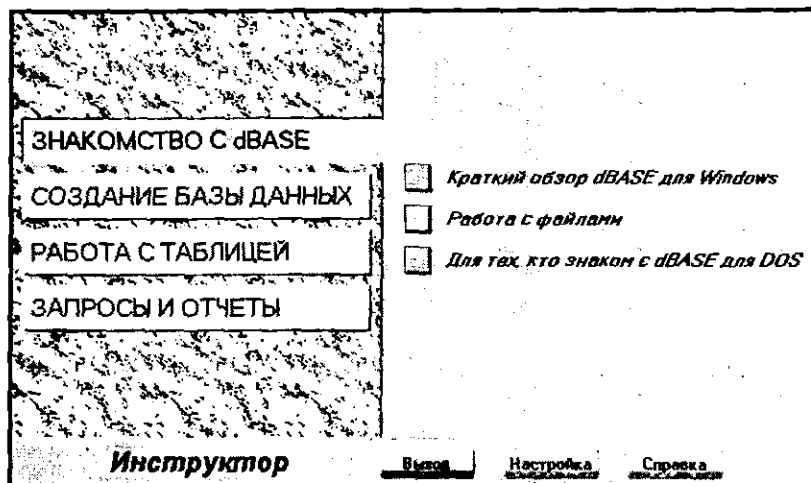


Рис. 1.18. Главное окно инструктора.

Инструктор содержит набор уроков, в процессе выполнения которых вы познакомитесь с dBASE и проблематикой баз данных, а также научитесь выполнять ряд стандартных задач. Инструктор внимательно следит за вашими действиями, отмечает

ошибочные действия и дает рекомендации по их исправлению. Вы можете воспользоваться в уроках файлами примеров, входящих в состав поставки dBASE. Однако инструктор допускает использование собственных объектов. Таким образом, вместе с инструктором вы можете создать собственные таблицы, запросы и отчеты. Если вы являетесь начинающим пользователем dBASE мы рекомендуем вам прибегнуть к помощи инструктора.

Для выбора урока укажите на интересующую вас тему в левой половине окна. В правой половине отображается список уроков по заданной теме. Нажмите кнопку рядом с названием интересующего вас урока.



Кроме инструктора в состав поставки dBASE входит эксперт по разработке форм. Эксперт помогает создавать формы, разбивая задачу на элементарные действия и снабжая каждое из них инструкциями. Далее мы познакомим вас с этим простым, но очень полезным средством.

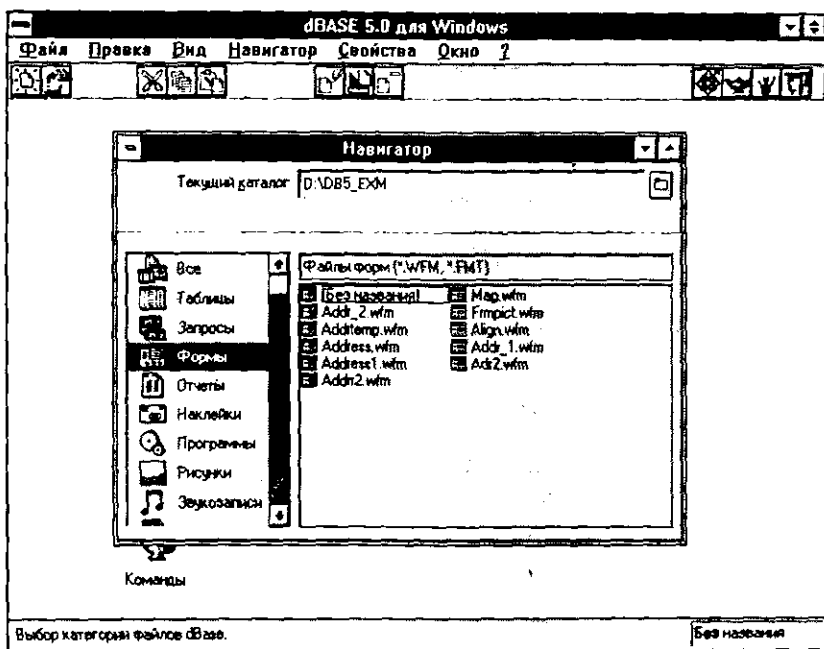


Рис. 1.19. Окно навигатора.

Окно навигатора

dBASE содержит новое средство, которое значительно ускоряет и облегчает управление объектами dBASE. Этим средством является окно навигатора системы dBASE для Windows. Окно навигатора использует двухуровневый доступ к объектам. Вы сначала выбираете тип объекта, а затем экземпляр данного типа. Не выходя из окна навигатора вы можете запустить на выполнение выбранный объект, модифицировать его с помощью соответствующего конструктора, удалить, изменить свойства объекта и т.д. Перечень допустимых действий над объектом определяется его типом.

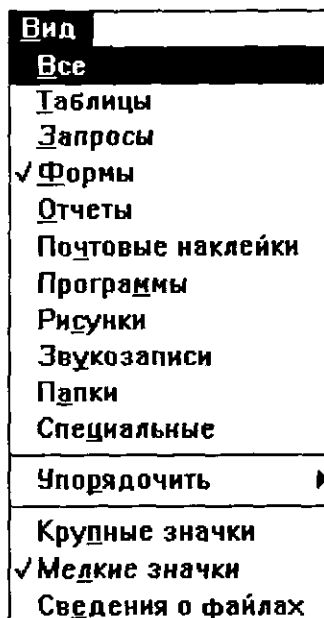
Чтобы открыть окно навигатора выполните любое из следующих действий:

- Нажмите кнопку



- Выберите команду **Окно | Навигатор**
- Раскройте пиктограмму навигатора в основном окне dBASE

На экране появится окно навигатора (рис. 1.19), которое содержит поле, содержащее текущий каталог, область типов объектов и область файлов выбранного типа из текущего каталога.



Для каждого типа файла определены наименование, пиктограмма и шаблон поиска файлов (например *.WFM для форм), который отображается над списком файлов. Любой файл, хранящийся в текущем каталоге и имеющий расширение, указанное в шаблоне имени файла, отображается в списке, расположенном в правой части окна навигатора. Файлы обозначаются пиктограммами, которые отражают тип файла. Имя текущего выбранного файла отображается в правой части строки состояния. При изменении типа объекта изменяется и список отображаемых файлов. Для выбора типа объектов вы можете выбрать требуемый тип в области типов объектов окна навигатора или же использовать для этого меню Вид.

Свойства навигатора

Как и все объекты dBASE, навигатор обладает рядом свойств. Эти свойства в основном определяют характеристики отображения информации в окне навигатора. Для определения свойств навигатора выполните команду **Свойства | Навигатор** основного меню или выберите **Свойства навигатора** из контекстного меню. На экране открывается окно диалога "Свойства навигатора" (рис. 1.20).

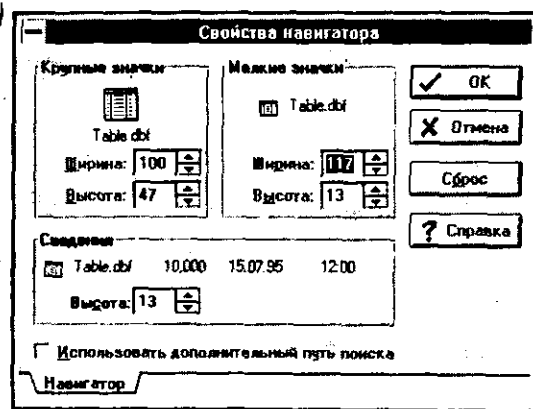


Рис. 1.20. Окно диалога "Свойства навигатора".



Значения параметров, заданные в окне диалога "Свойства навигатора", сохраняются в файле DBASEWIN.INI.

Области "Крупные значки" и "Мелкие значки" определяют расстояние между пиктограммами файлов. По умолчанию на экране отображаются мелкие пиктограммы. Для вывода крупных пиктограмм, выберите команду Вид | Крупные значки. На рис. 1.21 вы видите отображение крупных пиктограмм файлов. Если вы решили опять перейти к отображению мелких пиктограмм выберите команду Вид | Мелкие значки. Расстояния между пиктограммами задаются счетчиками **Ширина** и **Высота**. Счетчик **Ширина** задает расстояние между столбцами пиктограмм файлов, а счетчик **Высота** - между строками пиктограмм.

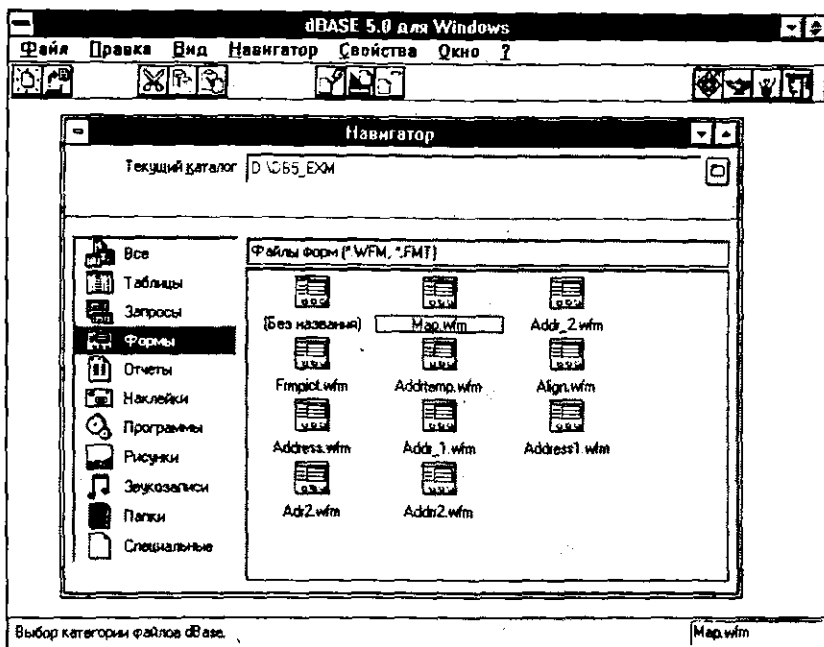


Рис. 1.21. Отображение крупных пиктограмм файлов.

В области "Сведения" задаются расстояния между пиктограммами файлов при отображении сведений о файлах. Для вывода на экран подробных сведений о файлах, выберите команду Вид | Сведения о файлах (рис. 1.22).

При установке флажка **Использовать дополнительный путь поиска** в окне навигатора добавляется новое поле **Путь поиска**. При вводе имени каталога в это поле в окне навигатора будут выводиться файлы из каталогов, указанных в обоих полях.

Кнопка **Сброс** возвращает стандартные значения, установленные в dBASE, счетчикам **Ширина** и **Высота**.

Выбор каталогов

Система dBASE для Windows осуществляет поиск файлов в одном или нескольких каталогах, указанных в полях **Текущий каталог** и **Путь поиска**. По умолчанию в окне навигатора всегда находится поле **Текущий каталог**. Для вывода на экран поля **Путь поиска** установите флажок **Использовать дополнительный путь поиска** в окне диалога "Свойства навигатора".

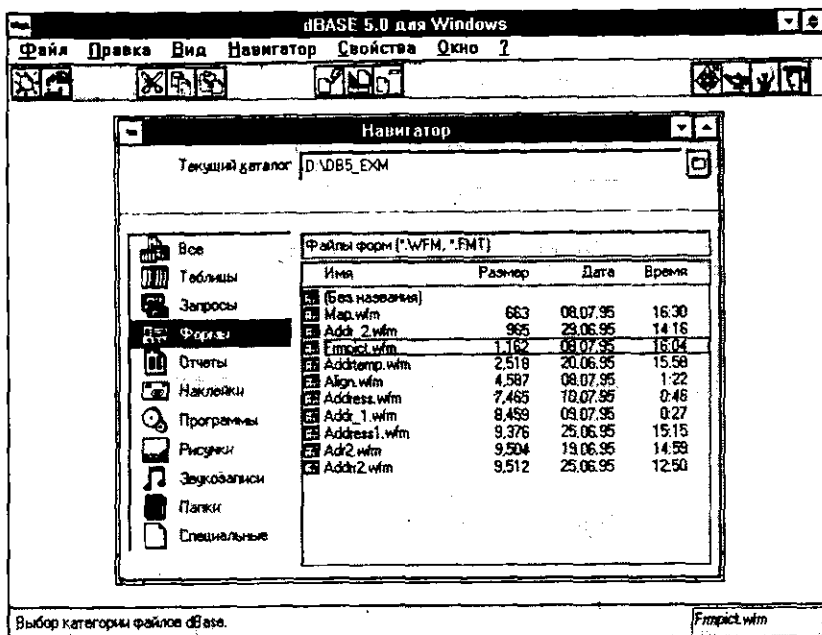


Рис. 1.22. Отображение сведений о файлах.

Вы можете задать только один текущий каталог и несколько каталогов в поле Путь поиска. При этом отделяйте наименования каталогов друг от друга знаками точки с запятой (;), запятой (,) или пробела. Например:

C:\DATABASEWIN\SAMPLES;C:\DB_EXM

Вы можете вводить имена каталогов непосредственно в поля Текущий каталог и Путь поиска с помощью клавиатуры. Если вы не помните точное название каталога, нажмите кнопку каталогов, расположенную рядом с этими полями, откройте окно диалога "Выбор каталога" и выберите в нем требуемый каталог.

При выборе типа данных "Таблицы" в окне навигатора появляется группа переключателей Таблицы из каталога и Таблицы из базы данных. По умолчанию выбран переключатель Таблицы из каталога и поиск таблиц осуществляется в каталогах, заданных в полях Текущий каталог и Путь поиска. При установке переключателя Таблицы из базы данных поле ввода Текущий каталог изменяется на раскрывающийся список Текущая база данных, а поле Путь поиска удаляется из окна навигатора. В этом случае обеспечивается поиск таблиц в базе данных, выбранной из раскрывающегося списка (рис. 1.23).



Для определения баз данных используется программа настройки IDAPI, которую вы можете запустить из группы программ dBASE для Windows.

Упорядочение файлов

Окно навигатора позволяет установить порядок расположения выбранных файлов в окне навигатора. Вы можете упорядочить файлы:

- по имени

- по типу и расширению
- по размеру
- по дате и времени

По имени

По типу и расширению

✓ По размеру

По дате и времени

Для выбора порядка расположения выполните команду Вид | Упорядочить и из подменю этой команды выберите требуемое упорядочивание файлов.

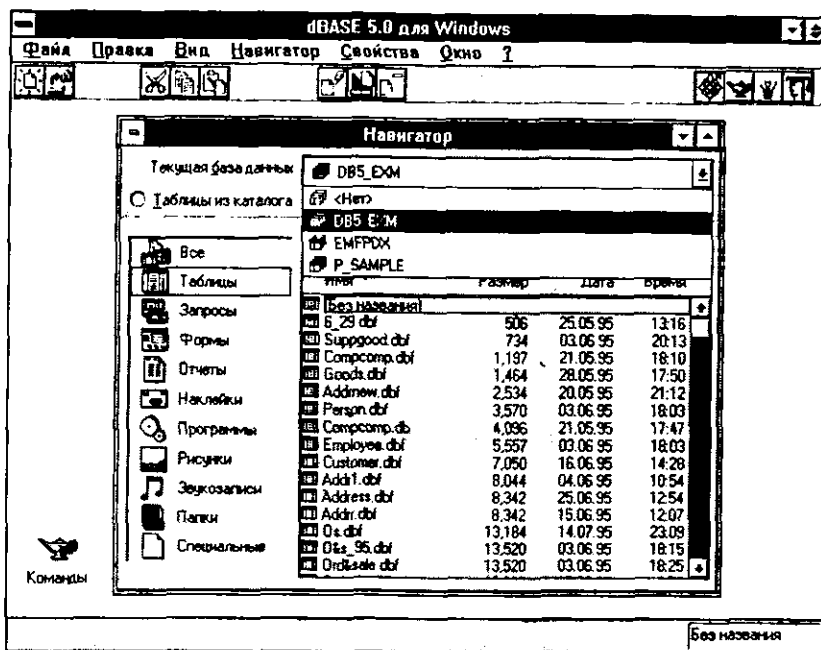


Рис. 1.23. Выбор текущей базы данных.

Управление объектами

Указав тип файла и каталог, вы можете выбрать любой из отображаемых в окне навигатора файлов. С выбранным файлом можно выполнить ряд доступных для данного типа действий: выполнить, изменить, удалить, определить свойства и т. д. Для выполнения этих действий вы можете использовать подменю команды Навигатор или контекстное меню. Список пунктов этих меню определяется типом объекта. Для открытия контекстного меню выделите пиктограмму файла и нажмите клавишу *Spacebar* или установите курсор мыши на пиктограмму файла и нажмите правую кнопку мыши. На рис. 1.24 вы видите вызов контекстного меню при выборе формы. Рассмотрим несколько действий, выполняемых над объектами любого типа.

Создание нового объекта

Чтобы создать новый объект, установите указатель на пиктограмму "Без названия" и выполните одно из следующих действий:

- Дважды нажмите кнопку мыши

- Выберите команду **Навигатор | Новая форма (отчет и т.д.)**
- Выберите команду **Новая форма (отчет и т.д.)** из контекстного меню
- Нажмите комбинацию клавиш **Shift+F2**

После выполнения любого из этих действий вызывается конструктор, соответствующий данному типу объекта (например, конструктор форм).

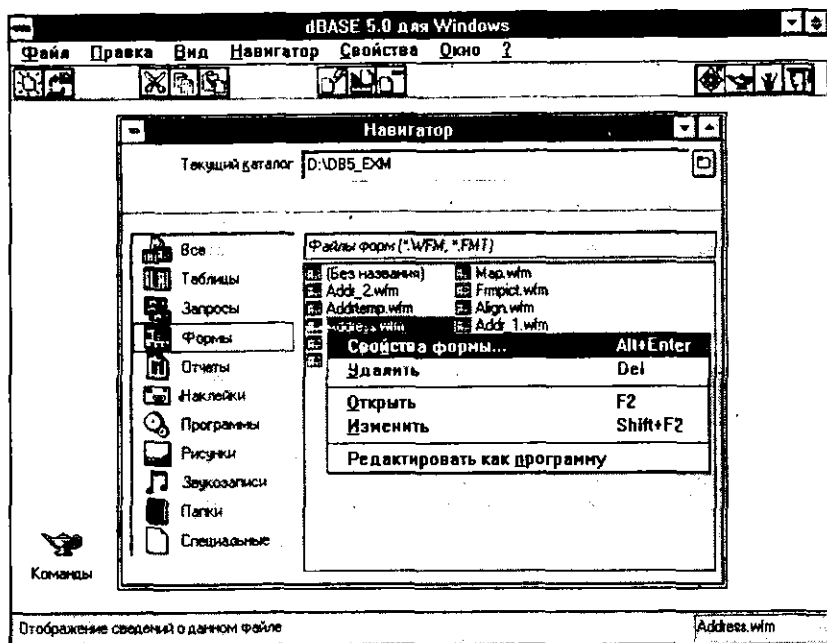


Рис. 1.24. Контекстное меню.

Выполнение объекта

Под выполнением объекта понимается открытие формы, отчета, запуск запроса и программы, проигрывание звука и просмотр изображения. Для выполнения выбранного объекта, выполните одно из следующих действий:

- Откройте меню **Навигатор** и выберите команду выполнения
- Откройте контекстное меню и выберите команду выполнения
- Нажмите кнопку 
- Нажмите клавишу **F2**
- Установите указатель на имя файла и дважды нажмите кнопку мыши
- С помощью мыши перенесите выбранный файл в рабочую область системы dBASE для Windows или в окно команд

Изменение объекта

Изменение объекта осуществляется с помощью соответствующего конструктора. Например, для изменения формы используется конструктор форм. Изменение вы-

бранного объекта осуществляется при выполнении любого из приведенных ниже действий:

- Откройте меню **Навигатор** и выберите команду изменения
- Откройте контекстное меню и выберите команду изменения
- Нажмите кнопку
- Нажмите клавишу *Shift+F2*
- Установите указатель на файл и дважды нажмите правую кнопку мыши



Удаление объекта

Для удаления выбранного объекта достаточно нажать клавишу *Del* или нажать кнопку на панели управления.



Псевдонимы

Псевдоним представляет собой специальное имя, присвоенное полному имени каталога. Псевдонимы используются в качестве наименования базы данных в окне навигатора. В результате вместо того чтобы вводить длинное имя каталога вы можете просто сослаться на его псевдоним.

Определение псевдонима

Для определения, изменения и удаления псевдонимов используется карточка "Псевдонимы" окна диалога "Программа настройки IDAPI", показанного на рис. 1.25. Для того чтобы открыть это окно диалога, запустите программу настройки IDAPI из группы программ dBASE для Windows.

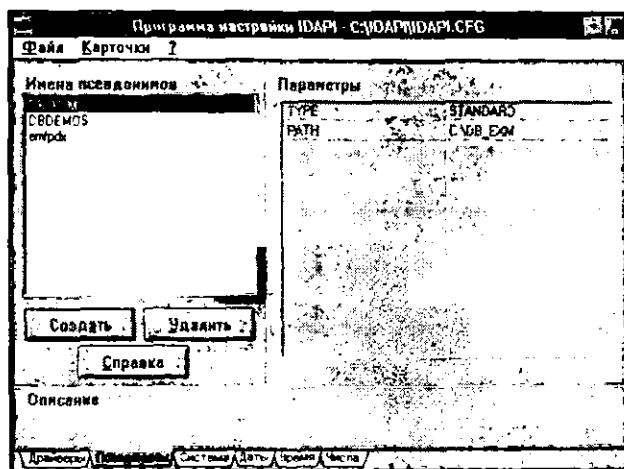


Рис. 1.25. Окно диалога определения псевдонима.

При определении нового псевдонима вам предстоит выполнить следующую последовательность действий:

1. Выберите **Создать**, в результате чего программа настройки откроет окно диалога ввода имени псевдонима и его типа (рис. 1.26).

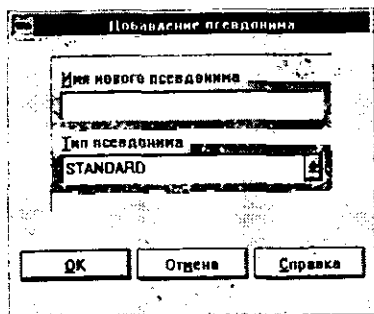


Рис. 1.26. Окно диалога ввода имени и типа псевдонима.

2. Введите в строку ввода **Имя нового псевдонима** псевдоним, который вы хотите присвоить каталогу.
3. На этом же этапе вам необходимо установить требуемый тип драйвера. По умолчанию dBASE предлагает вам воспользоваться стандартным драйвером (STANDARD). Если вы предполагаете создать базу данных, в которой будут использованы только таблицы dBASE и/или dBASE, воспользуйтесь стандартной установкой.
4. Нажмите кнопку **ОК** и имя псевдонима появится в списке псевдонимов. В области "Параметры" введите в строку ввода Path полное имя каталога, для которого вы предполагаете определить псевдоним.
5. Теперь вам необходимо сохранить созданный псевдоним, воспользовавшись командой **Файл | Сохранить**.

Изменение существующего псевдонима

Изменение псевдонимов также осуществляется в окне диалога "Программа настройки IDAPI". Выберите из списка **Имена псевдонимов** требуемый псевдоним и внесите необходимые изменения.



При изменении имени псевдонима dBASE считает, что создан новый псевдоним.

После внесения всех изменений, сохраните файл конфигурации IDAPI.

Удаление псевдонима

Для того чтобы удалить существующий псевдоним, откройте окно диалога "Программа настройки IDAPI". Укажите в списке **Имена псевдонимов** псевдоним, который вы хотите удалить, и выберите **Удалить**.

Окно папки

Окно папки предназначено для ускорения доступа к наиболее часто используемым объектам. В данном случае речь идет только об основных объектах (таблицы, формы, отчеты, наклейки, запросы и программы). При этом окно папки позволяет не только просматривать список объектов, но и выполнять над ними допустимые дейст-

вия. Вы можете создать несколько папок и использовать один и тот же объект в различных папках.

Для того чтобы открыть окно папки, выберите **Файл | Открыть** из основного меню dBASE и выберите файл с расширением .CAT или в окне навигатора установите тип объекта "Папки" и выберите файл папки в области объектов. Внешний вид окна папки показан на рис. 1.27.

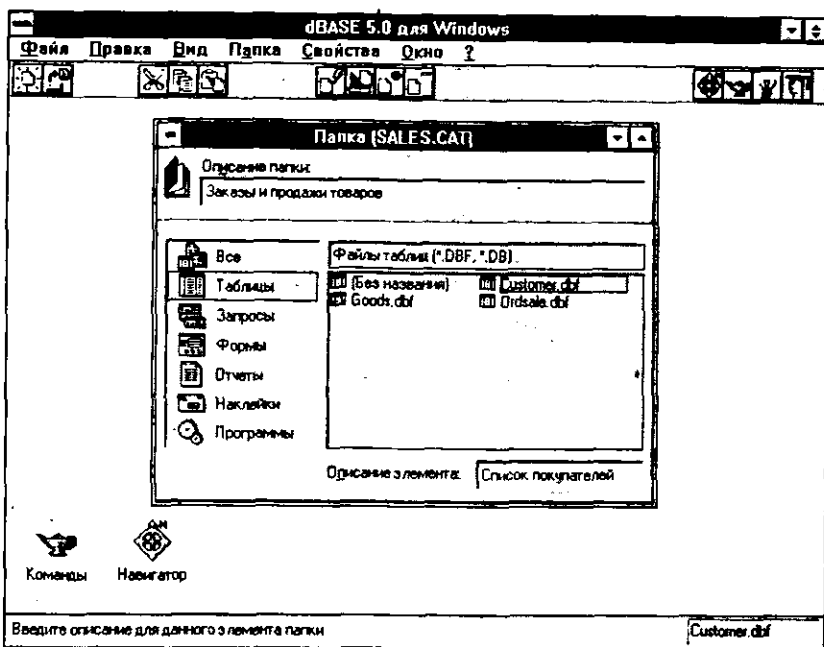


Рис. 1.27. Окно папки.

В верхней части окна папки содержится краткое описание папки, в левой части отображаются типы объектов, которые могут помещаться в папку. В правой части приведен список объектов выбранного типа. Каждый элемент этого списка может иметь краткое описание, которое выводится в правом нижнем углу.

Аналогично окну навигатора вы можете вызвать контекстное меню любого объекта, пиктограмма которого содержится в окне папки. Для этого укажите на пиктограмму требуемого объекта и нажмите правую кнопку мыши или выделите пиктограмму с помощью клавиш-стрелок и нажмите *Spacebar*. Состав команд меню определяется типом указанного объекта. Например, меню пиктограммы таблицы имеет вид, приведенный на рис. 1.28.

Для каждого объекта определено действие, которое является стандартным для объекта данного типа. Для большинства объектов таким действием является **Открыть**. Если вам требуется выполнить для объекта стандартное действие, для этого не нужно вызывать меню объекта. Вам достаточно указать на его наименование и дважды нажать кнопку мыши. Если вы проделаете эту операцию над формой, dBASE откроет указанную форму.

После открытия окна папки в главное меню добавляется команда **Папка**, а в меню **Вид** добавляется новая команда **Показать связи**. Команда **Вид** предоставляет в ваше распоряжение меню, дублирующее пиктограммы для выбора типов объекта, расположенные в левой части окна папки. Кроме того, в состав этого меню включена опция **Упорядочение**, предназначенная изменения порядка расположения файлов в окне папки. Эти опции меню аналогичны опциям меню **Вид** в окне навигатора. Но-

вая команда **Показать связи** устанавливает режим отображения не только файлов, но и связей между ними. Например, форма может быть создана на основе таблицы или запроса. Допустим, мы создали форму просмотра покупателей для таблицы CUSTOMER. В этом случае при просмотре как таблиц, так и форм будет показана связь между таблицей и формой (рис. 1.29).

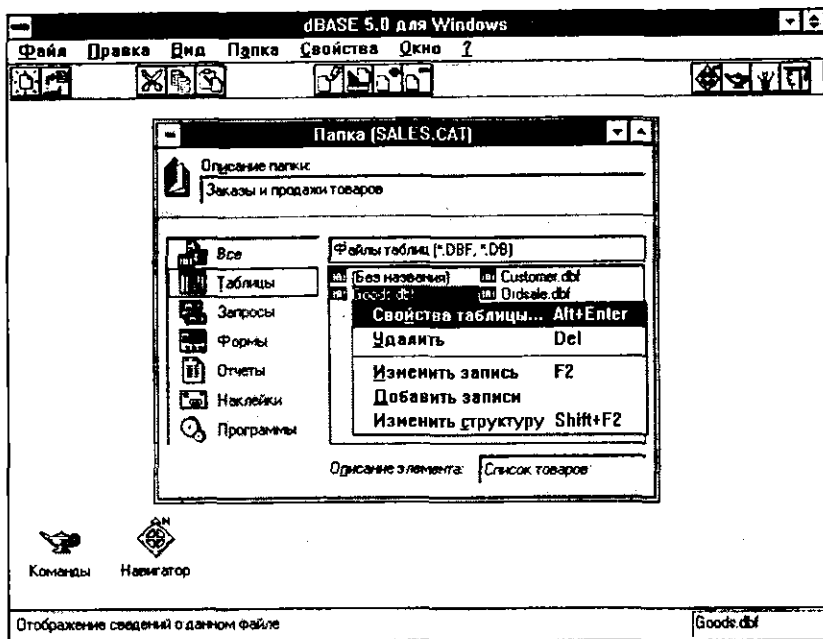


Рис. 1.28. Контекстное меню таблицы.

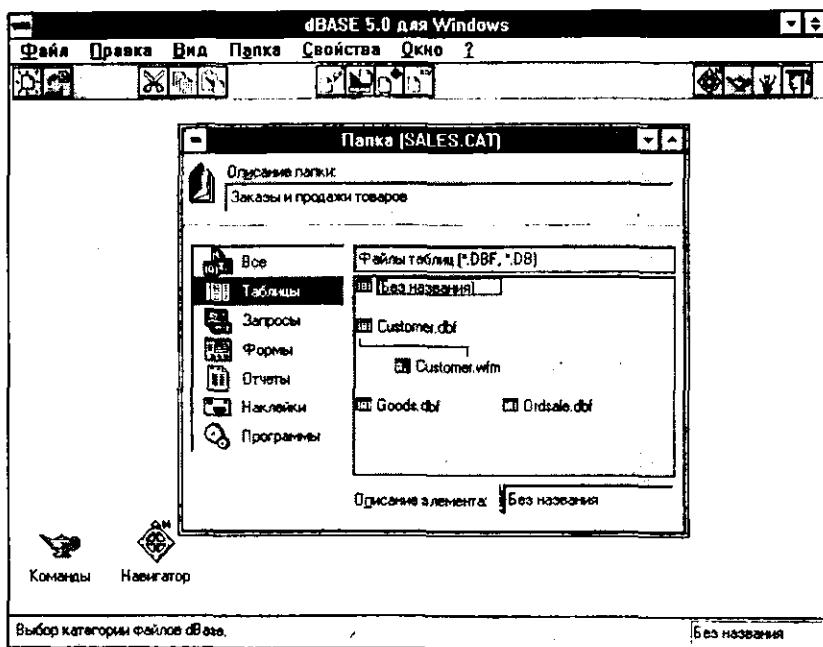


Рис. 1.29. Режим отображения связей между объектами.

Создание папки

Для создания новой папки выберите **Файл | Создать | Папка** из основного меню dBASE или в окне навигатора установите указатель на пиктограмму "Без названия" и выполните одно из следующих действий:

- Дважды нажмите кнопку мыши
- Выберите команду **Навигатор | Новая папка**
- Выберите команду **Новая папка** из контекстного меню
- Нажмите комбинацию клавиш *Shift+F2*

После выполнения любого из этих действий откроется окно диалога "Создание папки", в котором вы задаете имя файла папки. После ввода имени папки нажмите кнопку **ОК** и в следующем окне диалога (рис. 1.30) задайте краткое описание папки и вновь нажмите **ОК**. На экране появится окно созданной папки, которое в данный момент не содержит ни одного объекта.

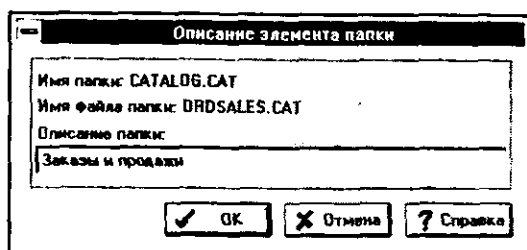


Рис. 1.30. Ввод краткого описания папки.



Краткое описание папки отображается в верхней части окна папки и доступно для редактирования. Поэтому в дальнейшем вы можете его изменить не выходя из окна папки.

Добавление объекта в папку

При добавлении объекта в папку вы не перемещаете его из одного каталога в другой. Вместо этого в окно папки помещается ссылка на объект. Вы можете добавить объект в папку с помощью меню, панели управления, путем переноса. Кроме этого объекты могут добавляться в папку автоматически.

Чтобы добавить файл в папку с помощью меню выполните следующие действия:

1. Выберите окно папки, чтобы сделать его активным.
2. Выполните команду **Папка | Добавить файл** или нажмите на панели управления кнопку 
3. Выберите файл в окне диалога "Добавление файла в папку" (рис. 1.31). По умолчанию используется тип файла, который является текущим в окне папки. Выбранный файл добавляется в активную папку.

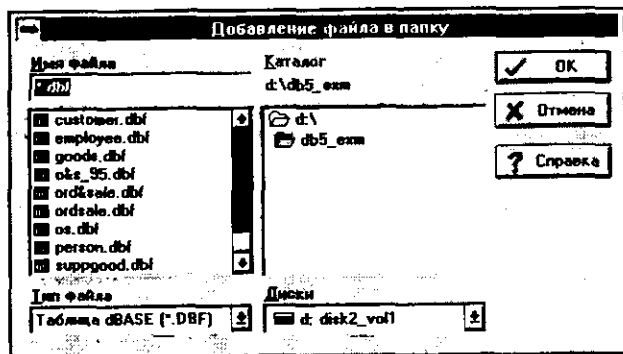
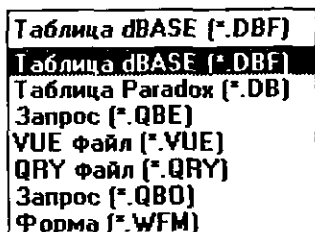


Рис. 1.31. Окно диалога выбора файла.

Аналогичные окна диалога выбора файла появляются на экране при открытии всех существующих объектов dBASE, поэтому остановимся на данном типе окна диалога более подробно.



Список **Имя файла** окна диалога содержит наименование всех объектов, тип которых совпадает с текущим значением из списка **Тип файла**. Если вам нужен объект другого типа, откройте список **Тип файла** и выберите требуемый тип. После чего вы можете выбрать из списка требуемый файл.

Если вы знаете точное название файла, можете ввести его непосредственно в поле ввода имени файла. В этом же поле вы можете задать шаблон поиска файла. Например, если вы введете в строку ввода *.DBF, dBASE поместит в список только наименование таблиц dBASE, а если *.DB, — то только таблицы Paradox. Данное средство очень удобно при выборе всех файлов, связанных в конкретной таблицей. Введите в строку ввода ADDRESS.* и список будет содержать наименования всех файлов, начинающихся с ADDRESS.

Раскрывающийся список **Диски** содержит список всех устройств вашего компьютера, а область **Каталог** позволяет выбрать требуемый каталог выбранного устройства.

После того, как вы выберете в окне выбора файла требуемый файл, нажмите **OK**, в результате чего указанный файл, будет добавлен в окно папки.



Опция **Тип файла** присутствует во многих окнах диалога выбора файла, однако далеко не при всех операциях dBASE позволит вам изменить стандартный тип объекта.

Для добавления файла в папку путем переноса с помощью мыши выполните следующие действия:

1. Откройте окно папки.
2. Перейдите в окно, которое содержит файл, который вы собираетесь добавить. Это может быть окно навигатора, другое окно папки или окно диспетчера файлов Windows.

3. Выберите файл, перенесите его с помощью мыши и поместите в открытое окно папки.

При открытом окне папки вновь создаваемые файлы автоматически будут добавлены в папку. При этом в папку добавляется не только создаваемый объект, но и все связанные с ним объекты. Например, при создании многотабличного запроса в папку добавится файл запроса и все входящие в запрос таблицы.

Удаление объектов

Для удаления объект из папки выберите его и выполните любую из следующих действий:

- Выберите команду **Удалить** контекстного меню
- Нажмите кнопку  на панели инструментов
- Нажмите клавишу *Del*

На экране появится окно диалога “Удаление файла из папки” (рис. 1.32). Нажмите **Да** для подтверждения удаления объекта из папки. На экране появится следующее окно диалога, в котором запрашивается подтверждение на удаление объекта с диска. Если вы решили полностью удалить объект, нажмите кнопку **Да**, в противном случае нажмите кнопку **Нет**.



Удаление объекта из окна папки не удаляет его с диска. В случае необходимости вы вновь можете добавить его в окно папки.



Будьте осторожны при удалении объекта с диска. Даже если этот объект не будет в дальнейшем использоваться в активной папке, он может использоваться в других папках или приложениях.

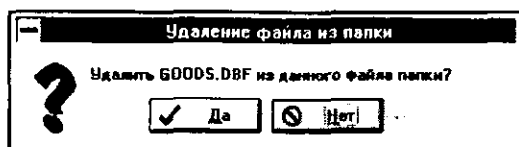


Рис. 1.32. Окно диалога “Удаление файла из папки.”.

Свойства окна папки

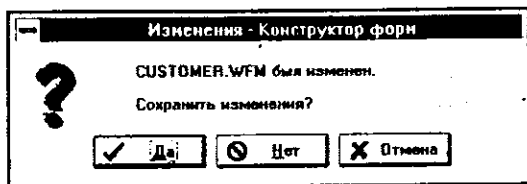
Свойства окна папки аналогичны свойствам навигатора. Для определения свойств папки выполните команду **Свойства | Окно папки** основного меню или выберите **Свойства окна папки** из контекстного меню.

Выход из dBASE

После завершения текущего сеанса работы вам необходимо покинуть dBASE. Для этого вы можете воспользоваться одним из следующих способов:

- Выберите **Файл | Выход**
- Выберите **Закреть** в системном меню главного окна dBASE

- Укажите на кнопку системного меню в главном окне dBASE и дважды нажмите кнопку мыши



Если в каком-либо из открытых окон содержится несохраненная информация, dBASE предложит вам указать как поступить с ней. Для этого на экране появится окно диалога, аналогичное следующему.

Для продолжения операции выберите одну из опций:

- Да** Сохраняет файл и выходит из dBASE. Если файлу еще не присвоено имя, dBASE откроет окно диалога “Сохранение файла”.
- Нет** Осуществляет выход из dBASE без сохранения файла.
- Отмена** Закрывает текущее окно диалога и возвращается в dBASE.



Никогда не выключайте ваш компьютер, предварительно не выйдя из dBASE, поскольку в этом случае существует потенциальная возможность потери данных.

Заключение

- В dBASE все данные хранятся в таблицах. Каждая таблица образована строками и столбцами, которые называются записями и полями, соответственно.
- Отдельные компоненты баз данных, которые используются для хранения и представления информации, называются объектами. К объектам относятся: таблицы, формы, отчеты, наклейки, запросы и программы. Все эти объекты являются основными объектами dBASE.
- Все объекты одного класса характеризуются одинаковым набором параметров, которые называются свойствами.
- Каждый объект хранится в отдельном файле, причем имена файлов, содержащих основные объекты, вы задаете самостоятельно, а наименование файлов, содержащих объекты, связанные с таблицей (индексы и файлы Метод-полей), совпадает с именем таблицы.
- Перед запуском dBASE запустите Windows. Если Windows располагается на устройстве C: в каталоге WINDOWS (эти параметры соответствуют стандартной установке), введите следующие команды:

```
C:  
CD\WINDOWS  
WIN
```

- Для запуска dBASE укажите на пиктограмму “dBASE для Windows” и дважды нажмите кнопку мыши. В качестве альтернативного способа запуска вы можете с

помощью мыши или клавиатуры выбрать пиктограмму "dBASE для Windows", а затем выполнить команду **Файл | Открыть**.

- Взаимодействие с dBASE осуществляется посредством выбора команд из текущего меню, расположенного в строке меню в верхней части экрана. В результате выбора большинства команд меню на экране появляется ниспадающее меню или открывается окно диалога для задания конкретных параметров.
- Вы можете выбрать команду с помощью любой из следующих процедур:
 - поместите курсор мыши на требуемую команду и нажмите кнопку мыши;
 - переместите курсор в строку меню, нажав клавишу *F10*. Затем поместите световой указатель на требуемую команду и нажмите *Enter*;
 - нажмите *Alt*+подчеркнутый символ имени команды. При выборе команд из ниспадающих меню, открытии списков и нажатии кнопок в окнах диалога нажмите подчеркнутый символ имени.
- Кнопки, расположенные на панели инструментов, дублируют выполнение некоторых из команд меню или позволяют ускорить перемещение по экрану, а также обеспечивают возможность работы с окнами папки и навигатора.
- Для ввода требуемой комбинации клавиш, нажмите клавишу, указанную в комбинации первой (*Alt*, *Ctrl* или *Shift*), а затем, не отпуская ее, нажмите вторую клавишу.
- При работе в главном окне, а также в окнах, в которых отображаются основные объекты, вы можете воспользоваться следующими средствами:
 - кнопкой системного меню;
 - вертикальной и горизонтальной полосами прокрутки;
 - кнопками распахивания и минимизации окна.
- Для получения справочной информации выберите команду ? из меню или нажмите клавишу *F1*.
- При работе с таблицами в dBASE вместо полного наименования каталога вы можете использовать краткое имя, называемое псевдонимом.
- Окно папки предназначено для ускорения доступа к часто используемым объектам. Для того чтобы открыть окно папки, выберите **Файл | Открыть** из основного меню dBASE и выберите файл с расширением .CAT или в окне навигатора установите тип объекта "Папки" и выберите файл папки в области объектов.
- Для выхода из dBASE выберите **Файл | Выход** или воспользуйтесь системным меню.

Глава 2. Создание таблиц

В главе 1 вы кратко познакомились с вопросами, которые могут решаться с помощью баз данных. Теперь мы переходим к изучению dBASE и работе с ним.

Итак, вы приступаете к разработке собственной базы данных. Первое, с чего вы должны начать, — это создать таблицу, в которую впоследствии будет вводиться информация. В дальнейшем данные в таблице могут дополняться новыми, редактироваться или исключаться из таблицы. Вы можете просматривать данные в таблицах или упорядочивать их по некоторым признакам. Информация, содержащаяся в таблицах, может быть использована для составления отчетов. Кроме того, вы можете дать графическую интерпретацию информации, содержащейся в базе данных. С решением этих задач вы познакомитесь в последующих главах.

В настоящей главе вы узнаете, каким образом создается таблица в dBASE, познакомитесь со структурой и форматами таблиц. Основное внимание будет уделено вопросам создания и изменения структуры таблиц. Кроме того, вы познакомитесь с доступными типами данных в dBASE.

Структура таблиц в dBASE

Как вы уже знаете, база данных может состоять из одной или нескольких таблиц. Каждая таблица состоит из строк и полей. При этом структура таблицы определяется ее полями. Для описания структуры таблицы необходимо перечислить поля с указанием наименования и типа каждого из них. В dBASE для создания структуры таблиц используется окно диалога “Структура таблицы”, показанное на рис. 2.1. Как вы увидите в дальнейшем, dBASE позволяет создавать таблицы в одном из форматов dBASE или Paradox.

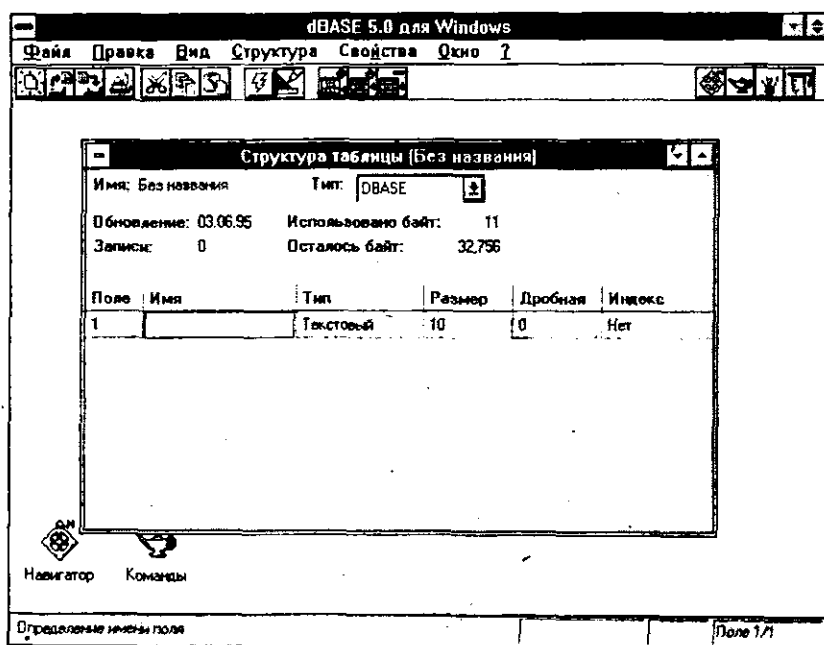


Рис. 2.1. Окно диалога “Структура таблицы”.

Окно диалога “Структура таблицы”, представленное на рис. 2.1, используется при создании таблицы в формате dBASE 5.0 для Windows, о чем свидетельствует значение поля Тип.

Окно диалога содержит таблицу из шести столбцов, которые содержат всю информацию о структуре таблицы.

1. Первый столбец содержит номера полей в таблице. Этот столбец определяет порядковый номер поля в таблице и автоматически заполняется dBASE при добавлении новых полей. Число полей в создаваемой таблице будет равно числу записей в окне диалога.
2. Столбец с наименованием “Имя” содержит наименования полей. Наименование каждого из полей таблицы, как правило, выбирается произвольно, но таким образом, чтобы отразить характер информации, которая будет храниться в данном поле.
3. Столбец с наименованием “Тип” содержит типы полей. Тип поля определяется типом данных, хранящихся в этом поле.
4. Столбец с наименованием “Размер” служит для задания ширины поля. При рассмотрении конкретных типов полей мы укажем для каких типов необходимо задавать ширину поля и приведем допустимые размеры полей.
5. Столбец с наименованием “Дробная” служит для задания размера дробной части числового поля.
6. Столбец “Индекс” предназначен для задания ключевых полей.

Наименования всех столбцов, указаны в их верхней части (рис. 2.1).

Остановимся более подробно на наименованиях полей и допустимых типах данных.

Наименования полей

Наименования полей становятся заголовками столбцов создаваемой таблицы. При задании наименований полей для таблицы dBASE вы должны следовать следующим правилам:

- Длина имени может достигать 10 символов.
- Первый символ должен быть буквой.
- В имени используются буквы, цифры и знак подчеркивания.
- Внутри имени не должно быть пробелов.
- В качестве имени поля вы не можете использовать зарезервированное имя поля `_DBASELOCK`. Это поле хранит сведения о блокировке при работе в многопользовательской среде.

Ограничений на наименования полей для таблицы Paradox меньше. Вы при создании таблицы Paradox должны следовать следующим правилам:

- Наименование поля может содержать до 25 символов.

- Наименование поля может содержать буквы, цифры и специальные символы (включая \$, %, &, !, #, @ и пробел).
- Два поля в одной таблице не могут иметь одинаковых наименований.
- В наименовании поля недопустимо наличие квадратных [], фигурных {} и обычных открывающей и закрывающей скобок (), а также комбинации символов ->.
- Наименование поля может содержать один или несколько знаков #, но не может состоять только из этих знаков.



dBASE не допускает ввод запрещенных символов и выдает звуковой сигнал при попытке их ввода.

Несмотря на то, что вы можете задать наименование поля до 10 символов, мы рекомендуем вам использовать имена, которые не превышают ширины поля. При отображении таблицы на экране dBASE отводит под каждое поле ширину, определяемую наибольшим из двух значений: длина наименования поля и ширина поля. Таким образом, если длина наименования поля значительно превосходит ширину самого поля, такое поле будет занимать на экране недопустимо много места. При этом значительная часть экрана не будет содержать информации, в то время как полезная для вас информация окажется за пределами экрана. В этом случае при работе с таблицей вы можете уменьшить ширину столбца, содержащего такое поле, непосредственно в экранном образе. Однако, часть наименования поля окажется невидимой.



В качестве наименования поля мы рекомендуем использовать аббревиатуры или краткие названия.

Типы данных dBASE и Paradox для Windows

Таблица типа dBASE допускает использование данных восьми типов, приведенных в табл. 2.1.

Табл. 2.1. Типы данных dBASE для Windows

Тип данных	Ширина поля
Текстовый	1 — 254
Числовой	1 — 20
Фиксированный	1 — 20
Дата	8
Текстовый произвольной длины (Мемо)	10
OLE	10
Логический	1
Двоичный	10



Поля логического и двоичного типа, а также поля дат, Мемо, OLE не допускают задания длины. Их размер автоматически устанавливается dBASE. Для данных числового и фиксированного типа указывается число знаков после десятичного разделителя.

Таблица типа Paradox допускает использование данных семнадцати типов, приведенных в табл. 2.2.

Табл. 2.2. Типы данных Paradox для Windows

Тип данных	Символ	Ширина поля
Текстовый	A	1 — 255
Числовой	N	
Целочисленный	S	
Длинное целое	I	
Фиксированный	#	0 — 32
Денежный	\$	
Дата	D	
Время	T	
Дата и время	@	
Текстовый произвольной длины (Мемо)	M	1 — 240
Форматируемый (форматируемое Мемо)	F	0 — 240
Графический	G	0 — 240
Вложенных объектов (OLE)	O	0 — 240
Логический	L	
Счетчик	±	
Двоичный	B	0 — 240
Байтовый	Y	1 — 255

В настоящем разделе мы подробно остановимся на каждом из типов данных таблицы типа dBASE. Подробную информацию о типах данных таблицы типа Paradox вы можете найти в документации или справочной системе.

Текстовые поля



Текстовые поля могут содержать буквы, цифры и специальные символы. Максимальная ширина поля составляет 254 символа.



Для задания текстового поля в столбце "Тип" необходимо выбрать из раскрывающегося списка значение "Текстовый", а в столбце "Размер" задать число, определяющее ширину поля (от 1 до 254).

Например, вы можете создать текстовое поле шириной 25. Максимальное количество символов, которые могут быть размещены в этом поле, равно 25. Это означает, что в такое поле не могут быть введены значения, длина которых превышает 25 символов. Ниже приведены примеры данных, которые вы можете ввести в различные записи такого поля:

IBM PC/AT Computer
ЦКБ "ЦИКЛОН"
123-34-78 (домашний)

При определении текстового поля вы обязательно должны задавать его длину. В противном случае dBASE выдаст сообщение (рис. 2.2). Аналогичная ситуация произойдет при задании длины поля, превышающей 254 символа.

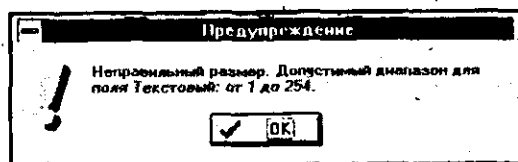


Рис. 2.2. Предупреждение при неправильном вводе данных.

Числовые поля фиксированного формата и числовые поля с плавающей точкой

Познакомившись с текстовыми полями, вы знаете, что в качестве допустимых символов этот тип поля может содержать цифры. В связи с этим возникает вопрос: с какой целью вводится числовой тип полей? В качестве основных аргументов по использованию числовых полей приведем следующие:

- При вводе данных числового типа автоматически производится проверка данных. Если данные содержат текстовые или специальные символы, dBASE выдаст предупреждение, и ошибочные данные не будут введены.
- Только над числовыми полями возможно выполнение математических операций.



Для задания числового поля с плавающей точкой выберите из раскрывающегося списка в столбце "Тип" значение "Числовой", в столбце "Размер" задайте число, определяющее ширину поля и в столбце "Дробная" — размер дробной части поля.



Для задания числового поля фиксированного формата выберите в столбце "Тип" значение "Фиксированный" из раскрывающегося списка и задайте ширину поля и размер дробной части.

Числовые поля фиксированного формата и числовые поля с плавающей точкой могут иметь длину до 20 цифр и до 18 знаков в дробной части. Разделитель целой и дробной части занимает один знак.



В отличие от dBASE IV, система dBASE для Windows одинаково обрабатывает числовые поля фиксированного формата и числовые поля с плавающей точкой. При выполнении математических операций для обоих типов предусмотрена одна и та же точность (19 знаков).

Поля дат

Использование для хранения дат полей, в которых данные представлены в специальном формате, предоставляет вам ряд дополнительных преимуществ:

- dBASE позволяет вводить и отображать даты в нескольких форматах, определяемых программой настройки IDAPI.
- При вводе дат dBASE осуществляет проверку формата данных, корректно обрабатывает високосные годы и столетия, на которые приходится перерасчет календаря, и допускает ввод только правильных дат.

- При работе с данными могут быть использованы операторы поиска и сортировки. Кроме того, вы имеете возможность вычислить количество дней между двумя датами или дату, которая наступит спустя заданное количество дней.



Для обозначения полей, в которые будут заноситься даты, в столбце "Тип" необходимо выбрать значение "Дата".

При задании дат XX века вы можете вводить только две последние цифры года. В противном случае необходимо указывать все четыре цифры. Например, 21.3.91, но 21.3.2091.

Текстовые поля произвольной длины (Мето-поля)

Текстовые поля произвольной длины (Мето-поля) могут содержать те же типы данных, что и простые текстовые поля. Отличие между этими полями заключается в том, что размер поля Мето не ограничен 254 символами, а зависит от объема данных хранящихся в этом поле. Максимальный размер Мето-поля не ограничен. Фактически объем данных, которые вы можете ввести в поле данного типа определяется объемом жесткого диска вашего компьютера.

dBASE автоматически создает файл для хранения данных Мето-поля. Этот файл имеет расширение .DBT, а его имя совпадает с наименованием таблицы.



Для задания Мето-поля выберите в столбце "Тип" значение "Мето".

Поля вложенных объектов

Поля вложенных объектов или OLE-поля (OLE — сокращение от Object Linking and Embedding) позволяют использовать данные из других приложений, работающих под Windows. Например, вы можете включить в свою базу данных информацию, хранящуюся в электронных таблицах. Данное средство очень удобно, поскольку при его использовании осуществляется не экспорт/импорт данных, а непосредственный доступ к данным из dBASE.



Для задания поля вложенных объектов выберите в столбце "Тип" значение "OLE".

Логические поля

Логические поля предназначены для хранения логических значений: Т и F (для "True" (Истина) и "False" (Ложь)) или Д и Н (для "Да" и "Нет"). При вводе данных вы можете пользоваться любыми из символов Т, F, Д и Н (как прописными, так и строчными буквами), Yes/No (Да/Нет) или True/False (Правда/Ложь). Размер логического поля равен 1.



Для задания поля двоичного типа выберите в столбце "Тип" значение "Логический".

Поля двоичного типа

Двоичные поля могут содержать изображение или звуковые сигналы (речь, музыку и т.д.). Размер двоичных полей, так же как и Мемо-полей, зависит от объема данных, хранящихся в этом поле.

В интерактивном режиме вы не можете редактировать двоичные поля. Вам доступен только просмотр сохраненного изображения или проигрывание звуковых сигналов.



Для задания поля двоичного типа выберите в столбце "Тип" значение "Двоичный".



Мемо, OLE и двоичные поля иногда все вместе называют BLOB-полями. BLOB является сокращением от Binary Large Objects и обусловлено тем, что в данных полях хранятся большие двоичные объекты.

Индексы

В настоящем разделе мы остановимся на принципиальных моментах использования и создания индексов таблиц.

Одним из основных требований, предъявляемых к СУБД, является возможность быстрого поиска требуемых записей среди большого объема информации. Индексы представляют собой наиболее эффективное средство, которое позволяет значительно ускорить поиск данных в таблицах по сравнению с таблицами, не содержащими индексов. В зависимости от количества полей, используемых в индексе, различают *простые* и *составные индексы*. В простом индексе для ключевого выражения используется имя одного поля. Составной индекс использует набор из одного или нескольких полей или выражение, включающее функции и операторы dBASE.

Значения индексов хранятся в отдельном файле, называемом *индексным файлом*, в упорядоченном виде. Поиск в этом файле осуществляется оптимальным методом. Для каждого значения индекса в индексном файле содержится уникальная ссылка, указывающая на местонахождение в таблице записи, соответствующей индексу. Поэтому при поиске записи осуществляется не последовательный просмотр всей таблицы, а прямой доступ к записи на основании упорядоченных значений индекса.

Другой важной особенностью индексов является то, что вы можете использовать индексы, содержащие только уникальные значения. Это означает, что для таблицы, содержащей только одно индексное поле, уникальными должны быть значения этого поля. Для составных индексов величины в каждом из индексных полей могут иметь повторяющиеся значения. Однако значения выражения для индекса должны быть уникальными.

Вы можете создать простой индекс непосредственно при определении структуры таблицы с помощью столбца "Индекс". Значение столбца может иметь одно из трех состояний:

Значение	Описание
Нет	Поле не является индексным.
По возрастанию	По данному полю создается простой индекс, в котором значения упорядочены по возрастанию.
По убыванию	По данному полю создается простой индекс, в котором значения упорядочены по убыванию.

Для создания составного индекса откройте таблицу, для которой будет устанавливаться индекс, а затем выполните команду **Таблицы | Операции | Индексы**. На экране появится окно диалога “Индексы” (рис. 2.3). Для создания нового индекса нажмите кнопку **Создать** и на экране появится следующее окно диалога (рис. 2.4) “Создание индекса”.

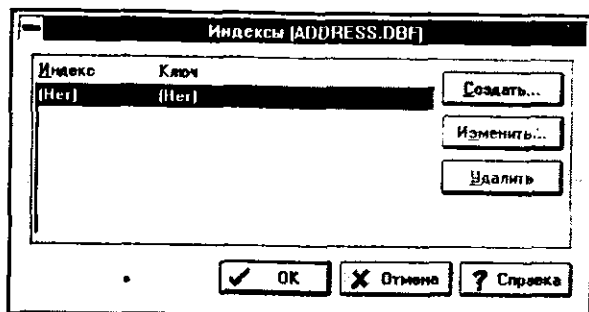


Рис. 2.3. Окно диалога “Индексы”.

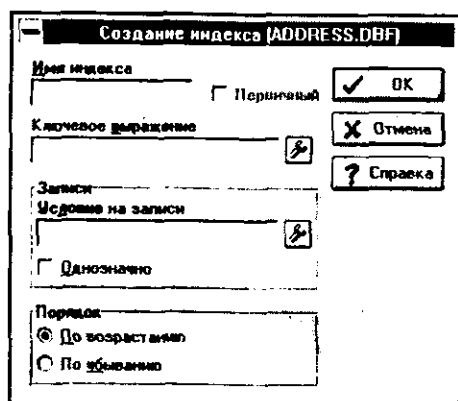


Рис. 2.4. Окно диалога “Создание индекса”.

В dBASE для определения составного индекса используется не список индексных полей, как во многих реляционных базах данных, а индексное выражение, которое включает индексные поля таблицы. При использовании в выражении полей разных типов вы должны проверять допустимость введенного выражения. Каждый индекс имеет имя, на которое вы можете в дальнейшем ссылаться для упорядочения отображения данных в соответствии с данным индексом.

Переключатели **По убыванию** и **По возрастанию** определяют порядок упорядочивания значений индексного выражения. Флажок **Единозначно** используется для создания индексов, содержащих только уникальные значения. Вы можете ввести имя индексного выражения непосредственно в строке ввода или же нажать кнопку **Ключевое выражение** для формирования выражения, используя окно диалога “Построение выражения” (рис. 2.5).

Аналогично индексному выражению вы можете определить для индекса фильтр, который используется для ограничения формируемых индексных значений. Результат выражения, используемого для фильтра, должен иметь логический тип.

В dBASE все индексы одной таблицы хранятся в одном файле (.MDX). Кроме того, система распознает файлы .MDX, созданные в более ранних версиях dBASE.

Система dBASE для Windows обеспечивает совместимость с более ранними версиями dBASE, которые использовали отдельный файл для каждого индекса (.NDX). Однако эти файлы открываются только с помощью команд dBASE в окне команд или в программе.

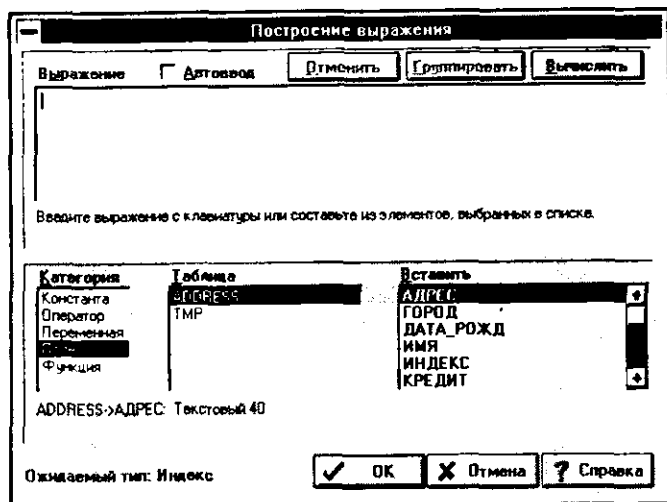


Рис. 2.5. Окно диалога "Построение выражения".

Индексы представляют собой весьма полезное средство. Однако вы всегда должны иметь в виду ту цену, которую придется заплатить за их использование. Так, если вы используете составной индекс, в состав которого входят текстовые поля большой ширины, размер индексного файла может оказаться сравнимым с размером самой таблицы.



В качестве индексных полей не могут быть использованы логические, двоичные и Метод-поля.

Определение структуры таблицы

Вы познакомились с допустимыми типами данных, а также с окном диалога, которое позволяет задать всю информацию о структуре таблицы. Однако, прежде чем приступить к созданию структуры таблицы, вы должны представлять цель, которая должна быть достигнута в результате создания базы данных. Исходя из поставленной цели, вы должны определить, какие таблицы будут использованы для ее достижения, а затем присвоить имена таблицам и определить их структуру. Только после этого вы можете создавать требуемую структуру таблиц.

Если вы создадите однотабличную базу данных, определение структуры таблицы, как правило, не вызывает затруднений. Однако при разработке связанных таблиц с различными типами отношений между ними распределение информации между таблицами, определение связей между таблицами, а следовательно, и определение структуры самих таблиц может быть сопряжено со значительными трудностями. В первой части книги мы ограничимся рассмотрением однотабличных баз данных. Вопросы проектирования многотабличных баз данных будут обсуждаться во второй части книги.

Процесс создания структуры таблицы рассмотрим на следующем примере. Предположим, что нам требуется создать базу данных, в которой будут содержаться имена и фамилии некоторой группы людей. Кроме того, база данных должна содержать номера их телефонов, адреса, наименование предприятий и кредит, а также некоторую дополнительную информацию. Фактически такая база данных представляет собой специальный “Телефонный справочник”. Этот пример мы будем часто использовать в последующих главах.

С помощью “Телефонного справочника” вы сможете быстро:

- найти номер телефона и адрес абонента по его имени и фамилии;
- получить список номера телефонов и адреса всех абонентов, живущих в заданном городе;
- выбрать всех клиентов, кредит которых составляет менее некоторого определенного значения;
- получить список людей, имеющих день рождения на текущей неделе.

Кроме того, “Телефонный справочник” позволит вам:

- распечатать список абонентов в алфавитном порядке;
- направить выбранной группе людей приглашение на встречу;
- напечатать информационные письма и адреса для их наклейки на конверты.

На данном этапе для решения поставленной задачи достаточно одной таблицы, которую мы назовем ADDRESS. Для включения в таблицу всей необходимой информации ее структура должна содержать поля, приведенные в табл. 2.3.

Табл. 2.3. Структура “Телефонного справочника”

	Наименование поля	Тип поля	Размер
1	Фамилия	Текстовый	16
2	Имя	Текстовый	16
3	Отчество	Текстовый	16
4	Телефон	Текстовый	13
5	Индекс	Текстовый	6
6	Страна	Текстовый	20
7	Область	Текстовый	20
8	Город	Текстовый	16
9	Адрес	Текстовый	40
10	Дата рождения	Дата	8
11	Предприятие	Текстовый	20
12	Кредит	Числовой	14(2)
13	Примечание	Мето	

В нашей первой базе данных мы не будем определять в качестве ключевого поля ни одно из полей таблицы ADDRESS. На первый взгляд кажется, что в качестве ключевого выражения можно было бы использовать объединение трех полей: “Фамилия”, “Имя” и “Отчество”. При этом вся информация была бы упорядочена в соответствии с фамилиями, именами и отчествами абонентов, что совершенно логично. Для однозначной идентификации записи этот индекс должен быть уникальным. Однако может возникнуть ситуация, при которой у двух различных людей будут совпадать значения всех трех полей. В этом случае вы просто не сможете ввести относящуюся к ним ин-

формацию в таблицу. Конечно, вы можете определить составной ключ, выбрав в качестве ключевых поля "Фамилия", "Имя" и "Телефон". Но в этом случае данные будут упорядочены по значению поля "Телефон", а не "Отчество", что не всегда удобно.

В качестве альтернативного решения мы можем предложить вам ввести в таблицу новое числовое поле, которое будет содержать уникальные значения идентификатора абонента. Во второй части книги при проектировании многотабличных баз данных мы воспользуемся этим приемом. Однако сейчас мы ограничимся созданием неключевой таблицы.

Под адрес абонента отведено четыре поля, которые названы "Страна", "Область", "Город" и непосредственно "Адрес". Такое представление адреса позволит вам эффективно осуществлять поиск информации по названиям стран, городов и адресам абонентов.

Поле "Примечание" содержит дополнительную информацию о каждом абоненте. Наименования остальных полей ясно указывают на то, какой тип данных содержат эти поля.

После определения структуры таблицы мы можем приступить к ее созданию в dBASE. На этом этапе вам необходимо выполнить следующую последовательность действий:

1. Выберите тип таблицы.
2. Задайте имена и типы полей.
3. Укажите длину полей для соответствующих типов полей.
4. Задайте имя и сохраните созданную таблицу.

Выполнение данных действий мы проиллюстрируем на примере "Телефонного справочника".

Для перемещения в процессе задания структуры таблицы вы можете использовать мышь или клавиши, представленные в табл. 2.4.

Табл. 2.4. Клавиши, используемые для задания структуры таблицы

Клавиша	Результат
<i>Backspace</i>	Перемещает курсор на одну позицию влево, удаляя символ
<i>Tab, Enter</i>	Перемещает курсор в следующий столбец
<i>Shift+Tab</i>	Перемещает курсор в предыдущий столбец
↑ и ↓	Перемещает курсор на одну строку вверх или вниз
<i>PgDn</i> и <i>PgUp</i>	Перемещает курсор на один экран вверх или вниз

Структура	
<u>Д</u> обавить поле	^A
<u>В</u> ставить поле	^N
<u>У</u> далить поле	^U
<u>П</u> ерейти к полю...	^G
<u>И</u> ндексы...	

Меню Структура содержит команды добавления нового поля, удаления существующего и перехода на заданное поле. Опция Индексы этого меню используется для создания индекса таблицы.

Пример 2.1. Создание “Телефонного справочника”

1. Запустите dBASE. Если у вас возникли проблемы, обратитесь к главе 1. После запуска на экране появится главное окно dBASE.
2. Для того чтобы приступить к созданию таблицы, выберите **Файл | Создать | Таблица** и на экране появится окно диалога “Структура таблицы” (рис. 2.1).
3. Прежде всего вам необходимо выбрать тип таблицы. Выбор типа таблицы (Paradox или dBASE) определит какие типы полей вы сможете использовать при создании таблицы. Давайте попытаемся сравнить типы полей Paradox и dBASE чтобы лучше оценить достоинства и недостатки каждого из типов таблиц. Для этого обратимся к табл. 2.5.

Табл. 2.5. Сопоставление типов таблиц Paradox и dBASE

Средство	Paradox	dBASE
Просмотр удаленных записей	-	+
Двоичные поля	+	+
Поля произвольной длины	+	+
Форматируемые поля произвольной длины	+	-
Имя поля (до 25 символов)	+	-
Первичные индексы	+	-
Упорядочение таблиц в соответствии со значениями первичного индекса	+	-

dBASE-формат имеет одно очень важное достоинство, которое не нашло отражения в табл. 2.5. Помимо самого dBASE вы можете использовать dBASE-файлы при работе с FoxPro, Clipper и другими dBASE-совместимыми программами. Оценив все “за” и “против” на данном этапе вам необходимо выбрать один из допустимых типов таблиц.

В качестве типа таблиц, используемого по умолчанию установлен dBASE 5.0 для Windows. Если вы решили создать таблицу в формате Paradox нажмите мышью кнопку открытия списка в поле “Тип”. В результате на экране появится список из двух доступных форматов таблиц: dBASE и Paradox, из которого вам остается выбрать требуемый формат.

В рассматриваемом примере создайте таблицу в формате dBASE 5.0 для Windows.

4. Теперь приступим к описанию полей таблицы.

Первое поле в таблице описания структуры, в котором содержатся номера полей, является недоступным для пользователя. dBASE автоматически вводит номер поля, после чего перемещает курсор в столбец “Имя”, указывая на то, что данное поле является текущим. Фактически вы сразу должны задавать наименование поля.

Для задания первого поля напечатайте в столбце “Имя” фамилия. Отметьте, что dBASE уже ввел в левый столбец номер этого поля 1.

5. После ввода наименования поля вам необходимо задать его тип. Для этого переместите курсор в столбец “Тип”, нажав клавишу → или Tab. Для задания типа поля нажимайте стрелки вверх или вниз и в столбце по очереди будут отобра-

жаться возможные типы полей. Кроме этого вы можете нажать кнопку раскрытия списка, в результате чего на экране появится список допустимых типов полей. Для текстового поля выберите значение "Текстовый".



6. Перейдите в следующий столбец и напечатайте 16.

Как было сказано ранее, в нашем первом примере мы не будем создавать ключевых полей, поэтому столбец "Индекс" должен оставаться пустым.

7. Для завершения задания первого поля нажмите *Enter*. Далее dBASE введет номер следующей записи и переместит курсор в столбец "Имя". В результате ваших действий экран будет иметь вид, показанный на рис. 2.6.
8. Продолжите заполнение раздела "Структура таблицы" до тех пор, пока не введете наименования всех полей за исключением поля "Страна", приведенных в табл. 2.3, не зададите их типы и длину текстовых и Мемо-полей. Поле "Страна" будет добавлено в примере 2.3.



dBASE автоматически пропускает столбцы, заполнение которых не требуется.

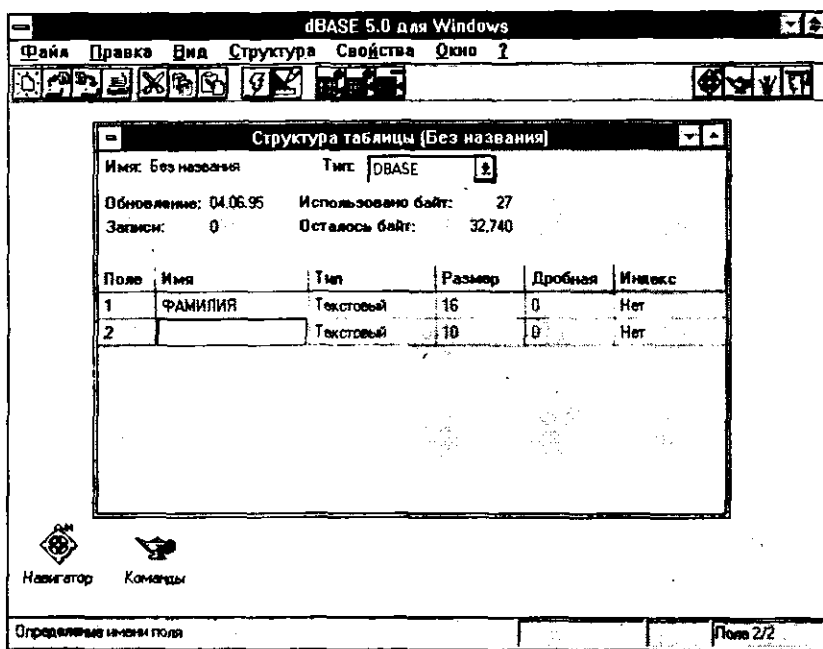


Рис. 2.6. Добавление нового поля.

Сохранение структуры таблицы

После того как вы создали структуру таблицы, ее необходимо сохранить. Для этого выполните команду **Файл | Сохранить как**, и на экране появится окно диалога "Сохранение таблицы" (рис. 2.7).

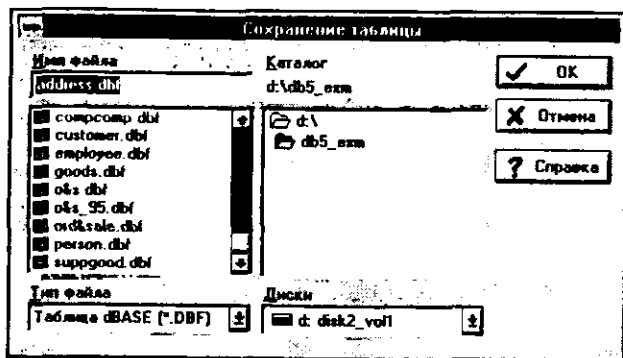


Рис. 2.7. Окно диалога “Сохранение таблицы”.

Задание имени таблицы

Перед тем, как сохранить созданную структуру таблицы, вам необходимо указать имя новой таблицы. Имя присваивается каждой таблице для того, чтобы связать таблицу с содержащейся в ней информацией. Как правило, название таблицы в лаконичной форме отражает основную идею представленной в ней информации.

В dBASE существует несколько основных правил по заданию наименования таблицы:

- Имя таблицы может содержать до восьми символов.
- Имя таблицы может содержать буквы, цифры и специальные символы (\$, %, &, !, #, @, и _), но первым символом должна быть буква или цифра. Рекомендуется избегать использования специальных символов, так как это может привести к некоторым проблемам при работе с таблицами (например, при создании вычисляемых полей, в которых необходимо указать псевдоним таблицы).
- В наименовании таблицы недопустимо наличие пробелов.
- Таблица в каталоге должна иметь уникальное имя.

При создании таблицы в текущем каталоге создается файл с указанным вами наименованием и расширением .DBF (для таблицы dBASE) или .DB (для таблицы Paradox). Если вам требуется создать таблицу в каталоге, отличном от текущего, воспользуйтесь областью выбора каталога.

Пример 2.2.

Завершите создание “Телефонного справочника”, сохранив таблицу под именем ADDRESS.

1. Выполните команду **Файл | Сохранить как**.
2. Выберите требуемый каталог.
3. Укажите на строку ввода “Имя файла”, удалите символ * и введите ADDRESS. В результате строка ввода должна иметь вид: ADDRESS.DBF
4. Выберите OK и dBASE перейдет в главное окно.

Открытие вновь созданной таблицы

После сохранения структуры таблицы вы можете дать указание dBASE открыть вновь созданную таблицу. Для этого выполните команду **Файл | Открыть** и установите тип выбираемого файла (например таблица dBASE). Затем выберите требуемую таблицу и нажмите **OK**. dBASE откроет таблицу в окне таблицы (рис. 2.8).

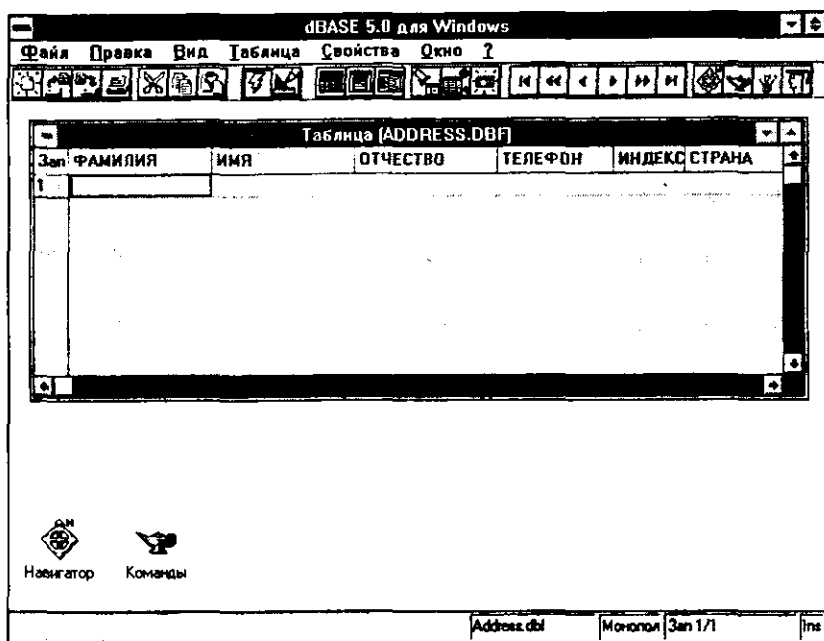
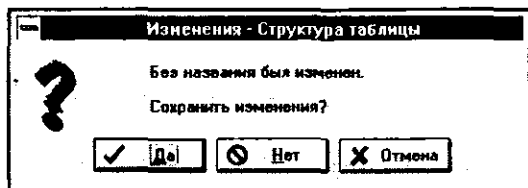


Рис. 2.8. Открытие вновь созданной таблицы.

Отказ от структуры таблицы

Вы можете выйти из окна диалога “Структура таблицы” без создания новой таблицы. Для отказа от результатов создания таблицы вы можете воспользоваться одним из следующих способов:

- Выберите команду **Заккрыть** из системного меню.
- Нажмите **Ctrl+F4**.
- Выполните команду **Файл | Заккрыть**.



Если вы воспользуетесь любым одним из этих способов, dBASE предложит вам подтвердить ваше решение.

Модификация структуры таблицы

Если вы ввели всю информацию в разделе “Структура таблицы”, не сделав при этом ни одной ошибки, и не предполагаете накладывать условия на поля создаваемой таблицы, вам необходимо присвоить созданной таблице имя и сохранить ее структуру. Данная процедура описана в следующем разделе.

Однако вполне вероятно, что при задании структуры вы могли допустить ошибки. Не стоит огорчаться, потому что dBASE предоставляет вам средства для исправления ошибок. К их числу относятся:

- изменение наименования поля и/или его типа;
- вставка пропущенного поля;
- удаление ошибочно введенного поля;
- изменение порядка следования полей в таблице.

Остановимся подробнее на некоторых моментах модификации структуры таблиц.

Изменение наименование поля и/или его типа

Предположим, что вы допустили ошибку при задании имени поля или его типа. При этом, ошибка была обнаружена только после окончания ввода этого поля.

Вначале установите курсор на наименование поля, тип или размер, которые требуется изменить. Для этого вы можете воспользоваться мышью или клавишами, приведенными в табл. 2.4. При переходе в новое поле оно окажется выделенным. В этом случае нажатие любой клавиши приведет к удалению содержимого поля и замене его введенным вами символом. Если вам потребуется не заменить, а изменить содержимое поля, перейдите в режим редактирования поля. Для этого нажмите ← или → или же нажмите кнопку мыши. В результате dBASE снимет выделение с поля и вы сможете перемещаться в пределах поля, вставлять и удалять символы.

Вставка пропущенного поля

Если вам необходимо ввести поле в середине таблицы, воспользуйтесь командой Структура | Вставить поле или Структура | Добавить поле. Первая команда вставляет пустую строку перед текущим полем таблицы, а вторая за текущим полем. После этого вы можете задать имя, тип и размер поля, как было описано ранее.

Вместо приведенных команд вы можете использовать эквивалентные кнопки на панели управления:



Добавить поле



Вставить поле

Пример 2.3.

При создании структуры таблицы было пропущено поле “Страна”, которое должно находиться перед полем “Область”. Исправьте допущенную ошибку.

1. Для вставки поля установите курсор в поле “Область” и нажмите кнопку Вставить поле. В результате на экране будет отображена ситуация, показанная на рис. 2.9.

2. Введите имя поля — Страна, его тип — Текстовый и размер — 20.

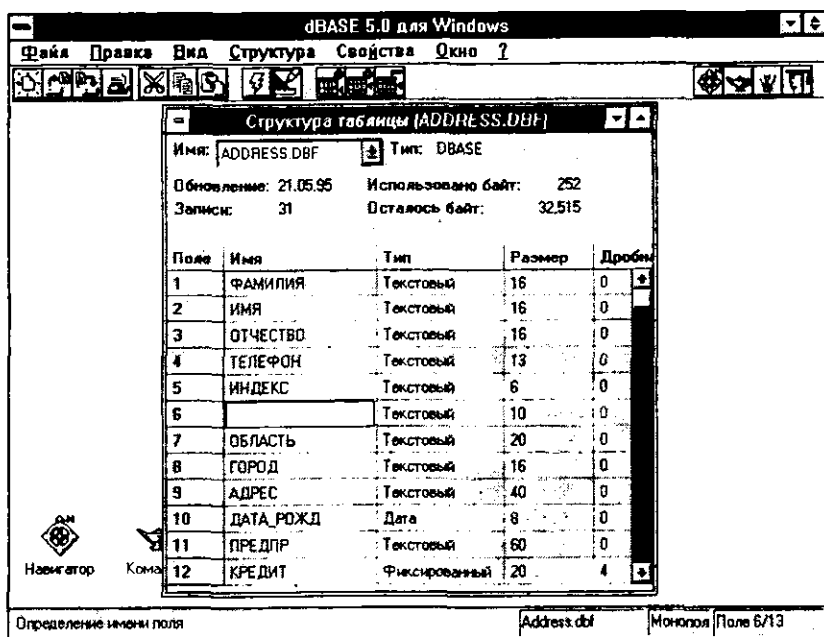


Рис. 2.9. Вставка пустой строки.

Удаление поля

Если после создания структуры таблицы окажется, что ошибочно введено лишнее поле, вы можете удалить его. Для этого установите курсор на удаляемое поле и выполните команду **Структура | Удалить поле** или нажмите кнопку **Удалить поле** на панели управления. В качестве примера на рис. 2.10 показан результат удаления поля "Город".

Изменение порядка следования полей

Для того чтобы переместить поле нажмите кнопкой мыши номер строки, которая содержит его описание. При этом курсор мыши примет форму руки. Удерживая кнопку мыши, переместите ее в нужную позицию, после чего отпустите кнопку мыши.

Заключение

- В dBASE все данные хранятся в таблицах. Каждая таблица образована строками и столбцами, которые называются соответственно записями и полями. Каждое поле имеет наименование и тип, определяющий характер данных, которые хранятся в этом поле.
- В таблицах dBASE допустимы следующие типы данных: текстовый, числовой, фиксированный, дата, Мемо, OLE, логический и двоичный.
- В таблицах Paradox допустимы следующие семнадцать типов данных: текстовый, числовой, целочисленный, длинные целые, фиксированный, денежный, дата,

время, дата/время, Метод, форматизируемое Метод, графический, OLE, логический, счетчик, двоичный и байтовый.

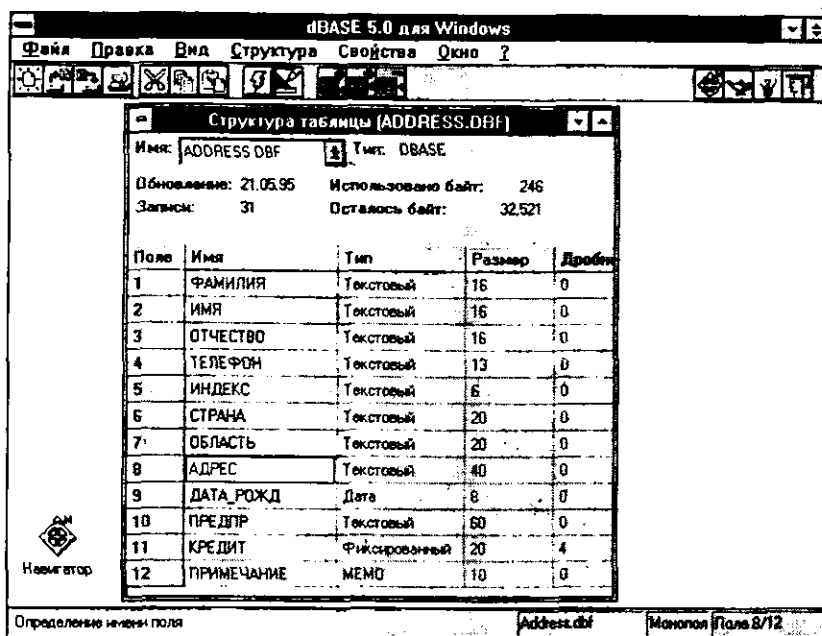


Рис. 2.10. Удаление поля "Город".

- Для создания новой таблицы используйте команду **Файл | Создать | Таблица** из меню главного окна dBASE. Эта команда позволяет вам определить структуру таблицы и задать ее наименование.
- Выбор типа таблицы (Paradox или dBASE) определяет какие типы полей вы сможете использовать при создании таблицы. Сравнительные характеристики типов полей в форматах Paradox и dBASE приведены в табл. 2.5.
- Для вставки нового поля используйте кнопки **Добавить поле** или **Вставить поле**.
- Для удаления поля используйте кнопку **Удалить поле**.
- Для изменения наименований и типов полей используйте режим редактирования поля, переход в который осуществляется нажатием мышью требуемого поля или с помощью клавиш **←** или **→**. Если вы не снимете выделение с поля, существующее значение будет заменено на вводимые вами символы.

Глава 3. Просмотр и редактирование данных

В главе 2 вы познакомились со структурой таблиц в dBASE, допустимыми типами данных, а также со способами задания и редактирования структуры таблиц. В настоящей главе вы узнаете, как вводить данные в таблицу, редактировать содержимое таблицы, сохранить результаты редактирования, а также просмотреть заполненную таблицу на экране.

Вы познакомитесь с вводом значений в поля дат, в поля числового и двоичного типа и в Метод-поля. В заключение мы введем понятие бланка, перечня и формы и покажем, как использовать их при вводе данных в таблицу.

Ввод данных в таблицы



Как вы уже знаете, все данные в базах данных хранятся в таблицах. Поэтому прежде, чем приступить ко вводу данных вам необходимо открыть таблицу. Для этого выполните команду **Файл | Открыть** в меню главного окна dBASE или нажмите кнопку **Открыть** на панели инструментов.

Далее укажите наименование таблицы, воспользовавшись одним из способов, описанных в главе 1. В результате dBASE откроет окно таблицы и поместит в него указанную вами таблицу. При этом dBASE находится в режиме таблицы, который часто называют режимом BROWSE по имени соответствующей этому режиму команды dBASE.

Как вы уже знаете из главы 1, в dBASE существует четыре режима отображения информации на экране: режимы таблицы, бланка, перечня, формы. Для отображения информации в этих режимах используется окно вывода данных. Наименование окна, в котором в настоящий момент представлены данные, отображается в заголовке окна таблицы.

Панель инструментов содержит кнопки, предназначенные для выполнения действий, которые обычно выполняются при выборе команды меню или нажатии комбинаций клавиш, а также для быстрого перемещения по данным. При этом состав кнопок зависит от режима отображения данных. Окно таблицы имеет две дополнительные по отношению к окнам бланка и перечня кнопки, которые используются для перемещения курсора вниз и вверх на размер страницы таблицы. На рис. 3.1 показана панель инструментов в окне таблицы, а в табл. 3.1 приведены состав и назначение ее кнопок.



Рис. 3.1. Панель инструментов в окне таблицы.

Табл. 3.1. Кнопки панели инструментов в окне таблицы

Кнопка	Название	Эквивалентная команда меню	Назначение
	Создать	Файл Создать	Создает новый файл
	Открыть	Файл Открыть	Открывает существующий файл

Кнопка	Название	Эквивалентная команда меню	Назначение
	Сохранить запись	Файл Сохранить запись	Сохраняет текущую запись
	Печать	Файл Печать	Печатает содержимое активного окна
	Удалить в буфер	Правка Вырезать	Удаляет выделенный текст или выделенные объекты и помещает их в буфер
	Копировать в буфер	Правка Копировать	Копирует в буфер выделенный текст или выделенные объекты, не удаляя из таблицы
	Вставить из буфера	Правка Вставить	Копирует содержимое буфера. Конкретно действие этой кнопки зависит от того, в каком из режимов вы работаете
	Изменить структуру	Таблица Изменить структуру	Модифицирует структуру текущей таблицы
	Режим таблицы	Таблица Режим таблицы	Отображает данные таблицы в режиме таблицы
	Режим бланка	Таблица Режим бланка	Отображает данные таблицы в режиме бланка
	Режим перечня	Таблица Режим перечня	Отображает данные таблицы в режиме перечня
	Найти запись	Таблица Найти записи	Осуществляет поиск определенной записи таблицы
	Добавить запись	Таблица Добавить запись	Добавляет в таблицу новую запись
	Создать запрос	Таблица Создать запрос	Создает запрос для открытых в данный момент таблиц
	Первая запись	Запись к первой записи	Выполняет переход к первой записи
	Предыдущая страница	Таблица К предыдущей странице	Выполняет переход к предыдущей странице записей
	Предыдущая запись	Таблица к предыдущей записи	Выполняет переход к предыдущей записи
	Следующая запись	Таблица к следующей записи	Выполняет переход к последующей записи

Кнопка	Название	Эквивалентная команда меню	Назначение
	Следующая страница	Таблица к следующей странице	Выполняет переход к последующей странице
	Последняя запись	Таблица к последней записи	Выполняет переход к последней записи
	Навигатор		Открывает окно навигатора
	Окно команд		Открывает окно команд
	Эксперт	? Эксперты	Осуществляет вызов эксперта, предназначенного для создания специальных форм
	Инструктор	? Инструктор	Открывает окно инструктора, в котором вы можете указать тему и урок для обучения

Пример 3.1. Ввод новых записей

В качестве примера введем несколько записей в таблицу ADDRESS, структуру которой вы создали в главе 2.

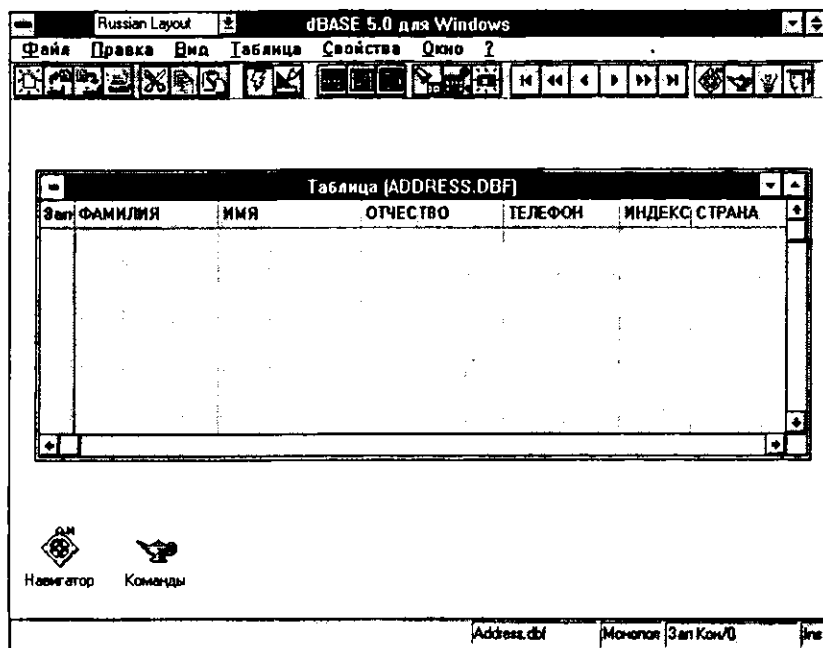


Рис. 3.2. Открытие вновь созданной таблицы.

1. Выполните команду **Файл | Открыть** и укажите наименование ADDRESS в качестве таблицы, которую следует открыть.

Результат выбора этой таблицы показан на рис. 3.2. Отметьте, что в заголовке окна отображено наименование таблицы ADDRESS. Очевидно, что поскольку после создания структуры таблицы ADDRESS вы не вводили в нее данные, при первом вызове таблица будет пустой (рис. 3.2). В строке состояния отображается количество записей в текущей таблице, в данном случае — 0.

В соответствии с созданной структурой таблица имеет 13 полей. Наименования полей, заданные при определении структуры, отображены в верхней части таблицы. Кроме того, dBASE автоматически формирует дополнительное поле, которое является крайним левым полем любой таблицы, которое имеет заголовок “Зап”. Вы не можете ввести данные в это поле. Номера записей автоматически вводятся dBASE по мере заполнения таблицы.

Обратите внимание на то, что ширины экрана вашего компьютера недостаточно для одновременного отображения всех полей таблицы. Как изменить экранный образ таблицы, чтобы на экране одновременно могло помещаться большее число полей? К этому вопросу мы вернемся в главе 4.

2. Добавьте новую пустую запись в таблицу, воспользовавшись командой **Таблица | Добавить запись** или кнопкой **Добавить запись** на панели управления. В результате в строке состояния появится сообщение **Зап 1/1**, свидетельствующее о том, что вы добавили одну запись в таблицу (рис. 3.3).

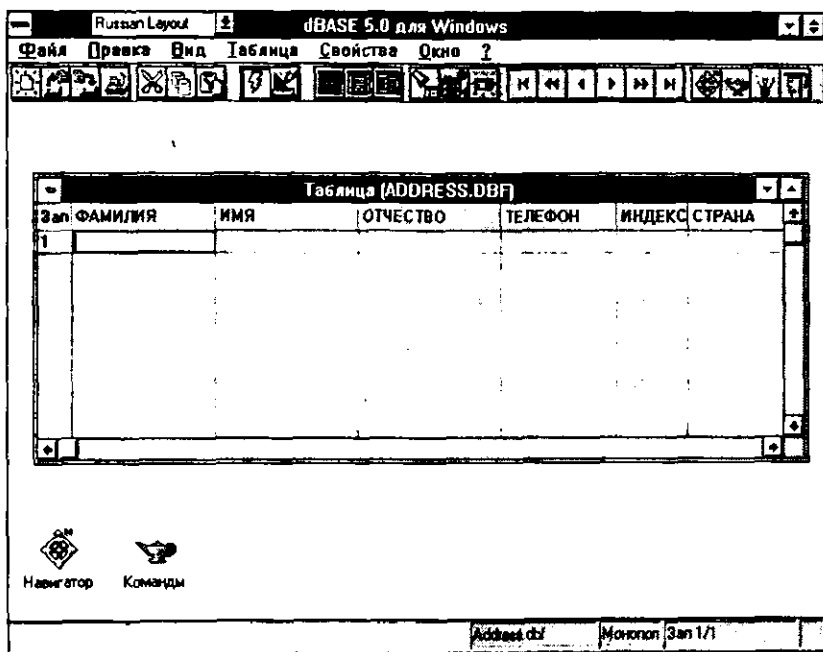


Рис. 3.3. Добавление пустой записи в таблицу.

На рис. 3.3 вы видите, что dBASE автоматически ввел номер 1 для первой записи. Теперь вы можете ввести данные в первую строку таблицы.

3. Введите данные в первое поле (“Фамилия”) и нажмите **Enter**. Курсор переместится в следующее поле (“Имя”), и вы можете вводить данные в это поле. На

рис. 3.4 показан результат ввода данных в первые пять полей, которые вы первоначально видели на экране. При вводе данных в следующие поля dBASE будет автоматически сдвигать столбцы, позволяя тем самым вводить информацию во все поля таблицы.

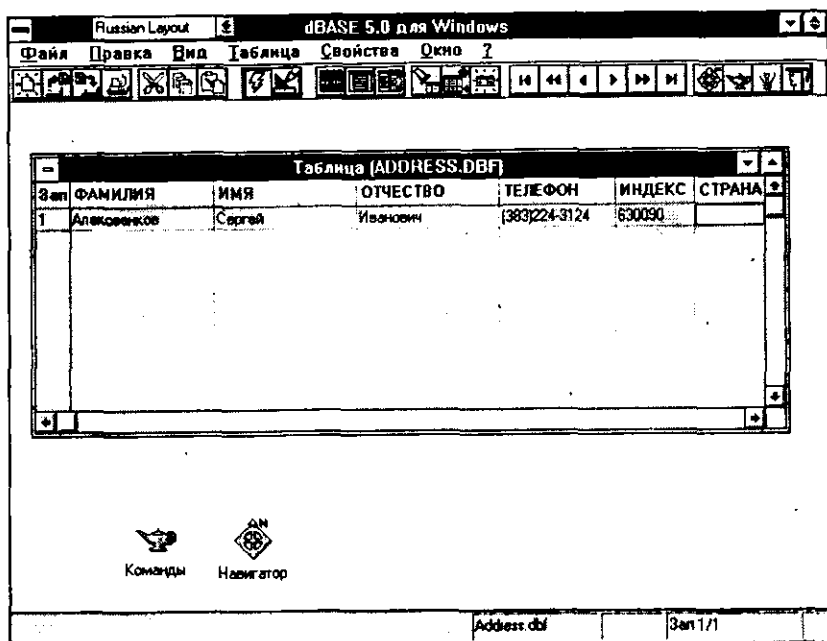


Рис. 3.4. Таблица ADDRESS после ввода данных в первые пять полей.

После заполнения последнего поля текущей записи и нажатия *Enter* dBASE запросит у вас подтверждения добавления новой записи (рис. 3.5) и утвердительном ответе перейдет на новую запись. При этом в крайнем левом поле будет помещен следующий номер. Курсор будет расположен в поле "Фамилия", и вы можете продолжить ввод данных в таблицу.

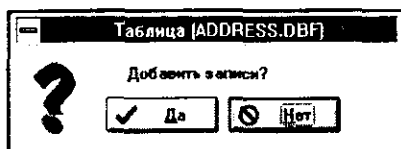


Рис. 3.5. Запрос подтверждения добавление новой записи.

Вы можете не вводить значения во все поля каждой строки. Если вы по какой-либо причине не хотите вводить значение в какое-либо из полей, перейдите в нужное вам поле. Для этого вам достаточно указать на него мышью. Для перемещения курсора мы можете также использовать клавиши. Например, для того, чтобы перейти к следующему полю, следует нажать клавишу *Enter* или *Tab*. Аналогично, если вам требуется переместить курсор на следующую строку, вы можете сделать это, нажав клавишу *↓*.

Наиболее простой способ перемещения курсора при вводе и редактировании данных заключается в том, чтобы нажать мышью требуемое поле. В качестве альтернативного способа вы можете воспользоваться клавишами, которые в некоторых слу-

чаях являются очень полезным средством для перемещения курсора. В рассмотренном примере переход на следующую строку без заполнения всех полей текущей записи осуществляется достаточно просто. При этом dBASE не позволит вам выполнить аналогичные действия с помощью мыши.

В табл. 3.2 приведены список и назначение клавиш, используемых при вводе данных в режиме таблицы.

Табл. 3.2. Комбинации клавиш, используемых при вводе данных в режиме таблицы

Клавиша	Назначение
<i>Enter</i> , <i>Tab</i> и $\rightarrow$	Переходит к следующему полю
<i>Shift+Tab</i> и $\leftarrow$	Переходит к предыдущему полю
$\uparrow$ и $\downarrow$	Переходит к предыдущей или последующей записи
<i>Ctrl+Home</i>	Переходит к первому полю текущей записи таблицы
<i>Ctrl+End</i>	Переходит к последнему полю текущей записи таблицы
<i>PgUp</i> и <i>PgDn</i>	Перемещают курсор к предыдущей или последующей странице записей
<i>Ctrl+PgUp</i>	Переходит к первой записи таблицы
<i>Ctrl+PgDn</i>	Переходит к последней записи таблицы

В процессе ввода данных вы можете удалить часть или все введенные символы в каком-либо из полей. Если вам нужно удалить все содержимое редактируемого поля, выделите его целиком, и нажмите *Del* или выберите команду Правка | Удалить. Еще одним средством является комбинация клавиш *Ctrl+Backspace*, предназначенная для удаления части поля, расположенной слева от курсора. Если же вам необходимо очистить все поля записи используйте команду Таблица | Очистить запись.



При вводе данных в режиме таблицы вы можете распахнуть окно таблицы, что позволит вам видеть одновременно на экране большее число записей. Для этого нажмите мышью кнопку распахивания окна или дважды нажмите мышью строку заголовка окна. Вы можете также открыть системное меню и выбрать из него Развернуть.

Итак, вы познакомились с общими положениями о вводе данных. Однако, ввод данных в различные поля имеет свои особенности. Перед тем, как приступить к рассмотрению специальных приемов ввода данных в поля различных типов, нам необходимо обсудить редактирование поля.

Редактирование поля

При установке курсора на поле, в которое уже введены данные, dBASE выделит это поле. Если теперь вы начнете вводить символы, содержимое поля будет полностью уничтожено и заменено новыми символами. Для редактирования содержимого поля вам необходимо перейти в режим редактирования поля, в котором вы можете перемещаться в пределах поля, удалять один или несколько символов, добавлять новые символы и т.д.

Для перехода в режим редактирования поля воспользуйтесь любым из приведенных ниже способов:

- Нажмите клавиши перемещения (*Home*, *End*, $\leftarrow$ или $\rightarrow$).
- Нажмите мышью требуемое поле.

При переходе в режим редактирования поля выделение с данного поля снимается, а курсор принимает вид вертикальной полоски. После этого вы можете перемещать курсор в пределах текущего поля и редактировать его содержимое, используя клавиши, приведенные в табл. 3.3.

Табл. 3.3. Клавиши для перемещения курсора и редактирования данных в режиме редактирования поля

Клавиша	Назначение
→ и ←	Перемещают курсор на один символ вправо или влево
Ctrl← и Ctrl→	Перемещает курсор на одно слово влево или вправо
Home	Перемещает курсор в начало текущего поля
End	Перемещает курсор в конец текущего поля
Del	Удаляет символ справа от курсора
Backspace	Перемещает курсор на одну позицию влево, удаляя символ
Ctrl+ Backspace	Удаляет слово слева от курсора

Пример 3.2. Редактирование данных в поле

Допустим, что вам необходимо в таблице ADDRESS ввести номер телефона одного из абонентов — 526-4216. При вводе была допущена ошибка и введен номер телефона 256-4216. При этом ошибка была обнаружена после того, как вы покинули поле. Если вы не воспользуетесь режимом редактирования поля, вам придется вводить все цифры заново. Очевидно, что это не лучший метод, поскольку вам требуется заменить только две цифры.

1. Откройте таблицу ADDRESS. Добавьте новую запись.
2. В поле “Телефон” введите значение (831)526-4216 и перейдите в следующее поле.
3. Возвратитесь в поле “Телефон” и обратите внимание, что значение поля выделено. Перейдите в режим редактирования поля, воспользовавшись одним из приведенных выше методов. В данном случае мы рекомендуем воспользоваться мышью.
4. Установите курсор между цифрами 2 и 5 и нажмите кнопку мыши. При таком включении режима редактирования поля курсор сразу оказывается в нужной вам позиции. В противном случае вам пришлось бы после включения режима поля устанавливать курсор с помощью клавиш.
5. Удалите ошибочно введенные символы, используя клавиши *Del* и *Backspace*, после чего введите требуемые символы.

Теперь, после того, как вы познакомились с режимом редактирования поля, приступим к рассмотрению настройки параметров ввода данных, а также ввода в поля различного типа. Вначале мы сделаем несколько замечаний относительно стандартных полей (дат и числовых полей), а затем обсудим ввод данных в двоичные и Метопля.

Настройка параметров ввода

Для настройки параметров ввода данных выполните команду **Свойства | Среда dBASE** и в окне диалога “Свойства среды dBASE” выберите карточку “Ввод данных” окна диалога” (рис. 3.6). Эта карточка содержит области “Клавиатура”, “Звуковой сигнал” и “Разделители”.

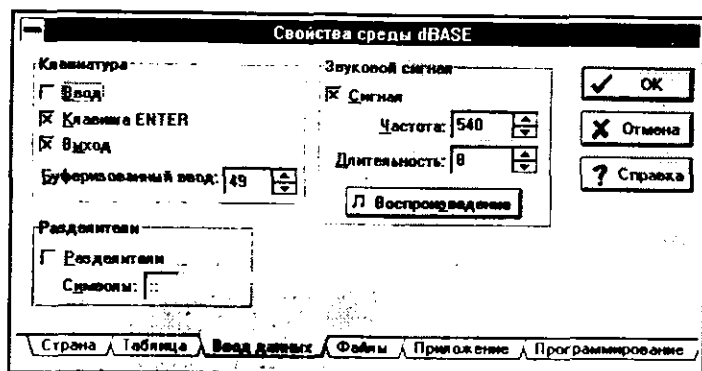


Рис. 3.6. Настройка параметров ввода.

Область “Клавиатура” определяет действия, наступающие при нажатии определенных клавиш.

Флажок **Ввод** определяет режим перехода к следующему полю. Когда флажок установлен, курсор переходит к следующей области ввода только после нажатия клавиши *Enter* или клавиш перемещения по полям, или при выборе другого поля с помощью мыши. Если же флажок снят, курсор автоматически переходит к следующему полю, когда текущее поле оказывается заполненным. По умолчанию флажок снят.

Флажок **Клавиша ENTER** устанавливает режим использования клавиши *Enter* при работе с формой. Если флажок установлен, клавиша *Enter* не выполняет действий в текущей форме. При снятом флажке клавиша *Enter* позволяет переходить в форме от одного объекта к другому. По умолчанию флажок установлен.

Флажок **Выход** определяет, будет ли нажатие клавиши *Esc* прерывать выполнение программы. По умолчанию флажок установлен.

Счетчик **Буферизованный ввод** задает размер буфера клавиатуры. Вы можете задать диапазон от 0 до 1600. По умолчанию значение равно 50 символам.

Область **Разделители** используется для определения разделителей, отмечающих начало и конец данных в полях. При помощи флажка **Разделители** вы определяете использование разделителей, а поле **Символы** задает символы разделители, стоящие в начале и конце данных в полях. По умолчанию значение поля выделяется двоеточиями (:).

Область “Звуковой сигнал” определяет параметры звукового сигнала, который выдается в момент достижения конца поля во время ввода данных или при вводе недопустимых данных. Флажок **Сигнал** включает и выключает звуковой сигнал. Счетчик **Частота** задает частоту звукового сигнала, измеряемую в циклах в секунду. Вы можете использовать диапазон от 19 до 10000. По умолчанию установлено значение 512 циклов в секунду. Счетчик **Длительность** задает продолжительность звукового сигнала в диапазоне от 1 до 19, измеряемую в тиках (1/18 секунды). По умолчанию установлено значение 2. После определения частоты и длительности сигнала вы можете прослушать его звучание с помощью кнопки **Воспроизведение**.

Значения параметров, установленные в окне диалога, действуют в течение сеанса работы.

Ввод данных в поля дат

Как вы уже знаете, dBASE поддерживает несколько форматов представления дат. При вводе даты dBASE автоматически подставляет разделители даты, соответствующие установленному в настройке dBASE формату даты, и вам остается только ввести само значение даты. В качестве примера в табл. 3.4 представлены три способа ввода даты начала Отечественной войны 1812 года (12 июня 1812 г.).

Табл. 3.4. Форматы ввода дат в dBASE

Формат	Пример
ММ/ДД/ГГГГ	06/12/1812
ДД/ММ/ГГГГ	12/06/1812
ДД.ММ.ГГГГ	12.06.1812

В главе 4 мы покажем, как изменить формат отображения дат в полях таблиц, а сейчас рассмотрим установку формата даты в среде dBASE.

Для установки принятого по умолчанию в течение сеанса работы формата дат используйте команду **Свойства | Среда dBASE** и в окне диалога “Свойства среды dBASE” выберите карточку “Страна” (рис. 3.7). Область “Дата” окна диалога предназначена для настройки формата отображения даты. Прежде всего вы должны выбрать формат отображения даты из раскрывающегося списка **Формат**. Кроме формата отображения даты вы можете ввести в поле **Разделитель** символ (обычно “/”, “-” или “.”), используемый в качестве разделителя компонент даты. Наконец, флажок **Год полностью** полностью обеспечивает вывод дат на экран с двумя или четырьмя цифрами года. При установленном флажке отображаются четыре цифры, в противном случае только две цифры.

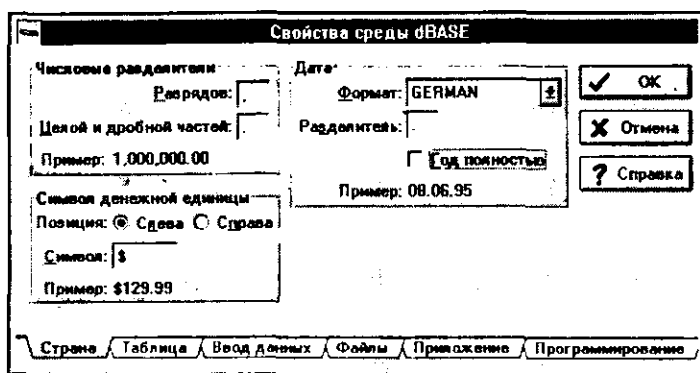


Рис. 3.7. Настройка формата отображения дат.



Отметим, что привычному в России формату отображения даты ДД.ММ.ГГ соответствует формат GERMAN. Кроме того, вы можете выбрать формат DMY и установить в качестве разделителя символ “.”.



Напомним, что dBASE игнорирует неправильно введенные данные. Например, будет проигнорирована дата 31.09.93 (поскольку в сентябре всего 30 дней).

Ввод данных в числовые поля

Аналогично датам, вы можете установить формат отображения числовых полей в таблице. Вы можете установить символы, используемые в качестве числовых разделителей, а также символ денежной единицы, отображаемый на экране при вводе денежных данных.

Также как и для настройки полей дат, выберите карточку “Страна” (рис. 3.7).

Область “Числовые разделители” задает символы, используемые для деления дробной и целой частей и деления разрядов больших чисел. Как правило в качестве разделителя дробной и целой части используется символ “.”, но в некоторых странах таким разделителем является символ “,”. Использование разделителя разрядов повышает читабельность больших чисел за счет того, что каждая группа из трех цифр разделяется указанным символом. Обычно в качестве разделителя используется символ “,”.

Область “Символ денежной единицы” задает текст, который будет выводиться при отображении числовых данных в денежном формате. Вы можете не только задать текст, но и его расположение относительно числа (слева или справа). По умолчанию в качестве символа денежной единицы используется “\$”, но вы можете заменить этот символ на сокращенное наименование национальной валюты, например “руб.”.

Ввод данных в Мемо-поле

Мемо-поля предназначены для хранения в таблице большого объема текстовой информации. В отличие от текстового поля Мемо-поле хранится в специальном файле с расширением .DBT.

Вначале перейдите в нужное Мемо-поле. Для просмотра и редактирования Мемо-поля вы можете воспользоваться любым из следующих способов:

- Выберите команду **Содержимое поля** из меню **Вид**
- Дважды нажмите мышью поле, в которое вы предполагаете вводить данные
- Нажмите **F9**

В результате выполнения этих действий откроется окно, называемое окном просмотра Мемо-поля, в котором отображается все содержимое Мемо-поля. При первом открытии окна просмотра Мемо-поля, оно будет пустым. Далее вы можете ввести в текущее поле всю требуемую информацию. Объем информации, которая может быть введена в Мемо-поле ограничивается только возможностями вашего компьютера. На рис. 3.8 показано Мемо-поле после ввода данных. Если при вводе данных в Мемо-поле вы достигнете нижней границы окна и попытаетесь выйти за его нижнюю границу, изображение в этом окне будет автоматически прокручено вверх.

Как и для большинства окон, с которыми вы работаете в dBASE, вы можете изменить положение окна просмотра Мемо-поля и/или изменить его размер.

По сравнению с текстовым полем dBASE предоставляет вам дополнительные средства для редактирования содержимого Мемо-поля. Например, если вы находитесь в текстовом поле, в результате нажатия **Enter** или **Tab** курсор переместится в следующее поле. При работе с Мемо-полем нажатие **Tab** приведет к вставке табуляции. По умолчанию позиции табуляции располагаются через каждые полдюйма. При нажатии **Enter** dBASE устанавливает код “жесткого перехода”, который не зависит от размеров окна, а при двукратном нажатии **Enter** — вставит пустую строку.

В отличие от кода жесткого перевода строки, который вставляется в Мемо-поле при нажатии **Enter**, в конце каждой строки Мемо-поля устанавливается код “мягкого перехода”. В результате устанавливается режим, при котором текст, не помещающийся-

ся в текущей строке будет автоматически перенесен на следующую строку. Это означает, что при изменении размеров окна слова будут автоматически переходить на следующие строки при уменьшении окна или возвращаться на предыдущие строки при увеличении окна.

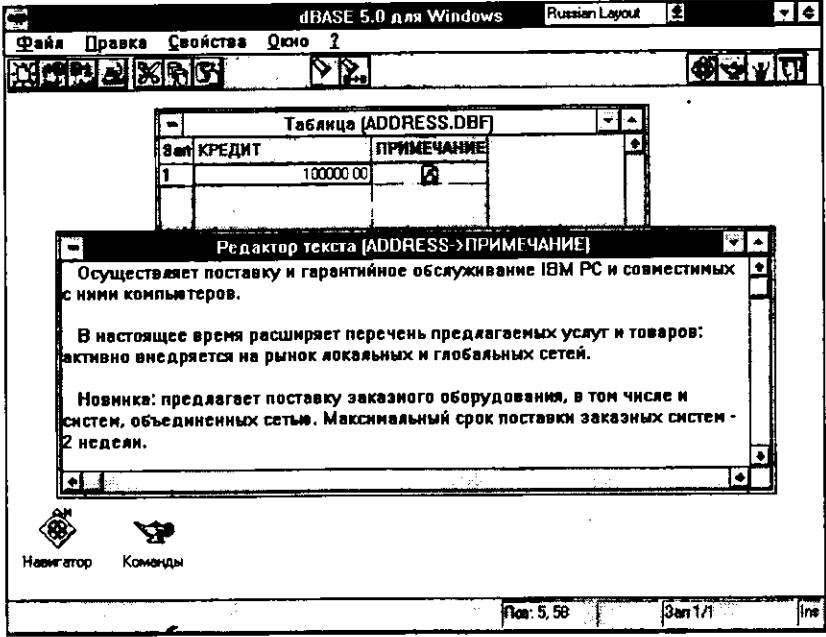


Рис. 3.8. Режим Метод-поля.

В табл. 3.5 приведен список клавиш, которые могут быть использованы при работе в режиме Метод-поля.

Табл. 3.5. Клавиши, используемые при редактировании Метод-полей

Клавиша	Назначение
← и →	На один символ влево или вправо
↑ и ↓	На одну строку вверх или вниз
Ctrl← и Ctrl→	На одно слово влево или вправо
Home	В начало текущей строки
Ctrl+ Home	В начало Метод-поля
End	В конец текущей строки
Ctrl+ End	В конец Метод-поля
Del	Удалить выделенный текст
Backspace	Удалить символ слева от курсора
Shift← и Shift→	Выделить один символ слева или справа
Shift↑ и Shift↓	Продлить выделение на одну строку вверх или вниз
Shift+ Home	Выделить до начала строки
Shift+ End	Выделить до конца строки
Shift+ Ins	Вставить содержимое буфера
Ctrl+ Ins	Копировать в буфер
Shift+ Del	Удалить в буфер

Клавиша	Назначение
<i>Ctrl+Backspace</i>	Удалить слово слева от курсора
<i>Alt+Backspace</i>	Отменить все изменения, внесенные в текущее Мемо-поле
<i>Esc</i>	Выход из Мемо-поля
<i>Tab</i>	Вставить в текст символ табуляции
<i>Enter</i>	Вставить в текст код “жесткого перевода строки”

Кроме того, при редактировании в данном режиме вы можете выполнять операции с блоками текста, о которых мы расскажем при использовании временного буфера Windows в качестве средства редактирования данных в настоящей главе.

При вводе данных в Мемо-поля для копирования текста в пределах одного поля или из одного поля в другое вы можете воспользоваться буфером обмена. Данное средство будет рассмотрено в разделе “Редактирование данных в таблицах”.

Для завершения режима Мемо-поля вы можете воспользоваться любым из стандартных способов, рассмотренных выше. В качестве альтернативного средства для завершения сеанса ввода данных дважды нажмите мышью кнопку системного меню Мемо-поля или нажмите указанную кнопку один раз, после чего выберите из системного меню команду **Заккрыть**. При выходе из Мемо-поля dBASE выполняет следующие действия:

- Сохраняет данные, введенные в поле.
- Закрывает окна просмотра Мемо-поля.

Настройка редактора Мемо-поля

Редактирование Мемо-поля осуществляется редактором текстов, характеристики которого вы можете настроить с помощью команды **Свойства | Редактор текста**. При выполнении этой команды открывается окно диалога “Свойства редактора текста” (рис. 3.9), которое выводит на экран стандартные значения параметров и позволяет задавать новые значения. Изменения, внесенные с помощью этого окна диалога, сохраняются в файле DBASEWIN.INI. Давайте более подробно остановимся на средствах, предоставляемых для настройки параметров редактора.

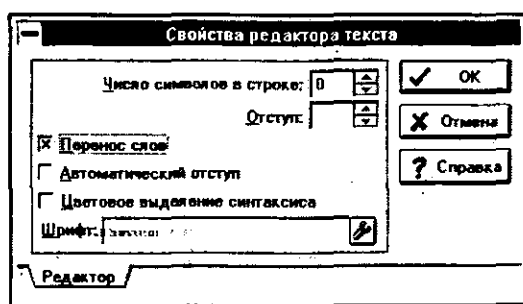


Рис. 3.9. Окно диалога “Свойства редактора текста”.

Флажок **Перенос слов** определяет, будет ли автоматически осуществляться перенос слов, если число символов в строке превышает значение, указанное в поле **Число символов в строке**. Если это флажок установлен, то добавление или удаление текста, а также изменение ширины окна автоматически изменяет структуру абзаца. По умолчанию для Мемо-полей этот флажок установлен.

Стандартное значение счетчика **Число символов в строке** равно 0. Для данного значения перенос строк будет осуществляться при достижении размера окна. Ненулевое значение (от 0 до 999) задает длину строки в символах средней ширины. Правая граница обозначается вертикальной чертой.

Если строки текста заканчиваются символами жесткого возврата каретки, их можно преобразовать в символы мягкого возврата каретки командой **Правка | Объединить строки**.

Флажок **Автоматический отступ** создает для новой строки отступ на уровне предыдущей строки. Если вы установите этот флажок, то при нажатии клавиши **Enter** новая строка получает такое же количество начальных пробелов, как и предыдущая. Кроме того, если этот флажок установлен, при нажатии клавиши **Tab** редактор dBASE для Windows вставляет количество пробелов, равное числу, заданному в счетчике **Отступ**. При каждом нажатии клавиши **Tab** добавляется следующая группа пробелов. При снятом флажке **Автоматический отступ** нажатие клавиши **Tab** приводит к вставке символа табуляции, который равен восьми символам средней ширины.

Действие клавиши **Tab** зависит от того, установлен или снят режим вставки. Если режим вставки установлен и клавиша **Tab** нажата после первого символа, не являющегося пробелом, то в текст вставляется символ табуляции. Если же режим вставки снят и клавиша **TAB** нажата после первого символа, не являющегося пробелом, то курсор перемещается к следующей позиции табуляции.

Флажок **Цветовое выделение синтаксиса** используется при редактировании программ, поэтому мы не будем останавливаться на нем.

Поле **Шрифт** задает шрифт, используемый в окне редактора текстов. Этим шрифтом отображается весь текст в окне. Вы не можете выделять слова и часть текста отдельным шрифтом. Для открытия окна диалога "Шрифт" (рис. 3.10) нажмите кнопку, расположенную в правой части поля, после чего выберите шрифт, его стиль, размер и цвет.

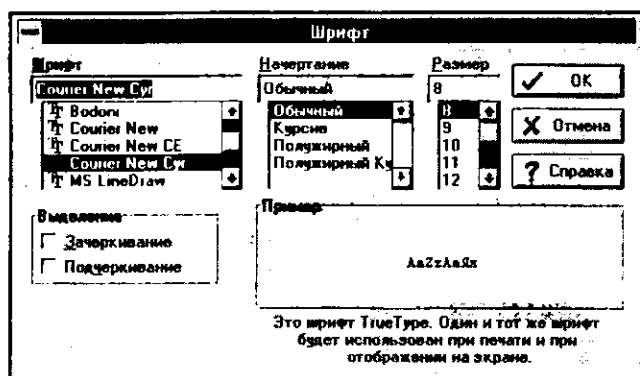


Рис. 3.10. Окно диалога "Шрифт".

Ввод данных в двоичные поля

Двоичные поля предназначены для хранения графических изображений и звуковых образов. Для создания графических изображений могут быть использованы графические редакторы, сканер или другие средства, формирующие изображения в одном из допустимых форматов. dBASE поддерживает форматы графики BMP и PCX. Если ваша программа подготовки графических изображений использует другой формат, прежде, чем импортировать изображение в базу данных dBASE вам необходимо преобразовать его к одному из допустимых форматов. Для этой цели можно использо-

вать большинство графических редакторов или других программ работы с графикой, поскольку практически все графические пакеты программ имеют систему преобразования изображений по крайней мере в один из указанных форматов. Для создания звуковых файлов могут использоваться программы обработки звуков, обычно входящие в состав мультимедийных комплексов.

Вы можете сохранить данные в двоичном поле. Для этого перейдите в него и выполните любое из следующих действий:

- Выберите команду Содержимое поля из меню Вид
- Дважды нажмите мышью поле, в которое вы предполагаете вводить данные
- Нажмите F9

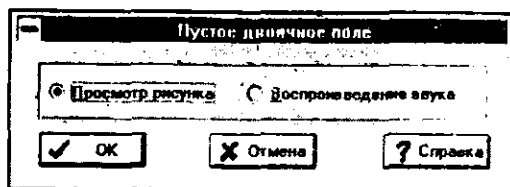


Рис. 3.11. Выбор типа данных, помещаемых в поле.

Далее вы можете вставить в поле содержимое графического или звукового файла. Если двоичное поле пустое, dBASE запрашивает вас тип помещаемых в него данных (рис. 3.11). После этого для графических данных открывается пустое окно просмотра графических данных (рис. 3.12), а для звуковых данных окно диалога воспроизведения звука (рис. 3.13). При этом главное меню dBASE изменяется, позволяя вам выполнять только действия по вставке в двоичное поле данных из файла с графической или звуковой информацией.

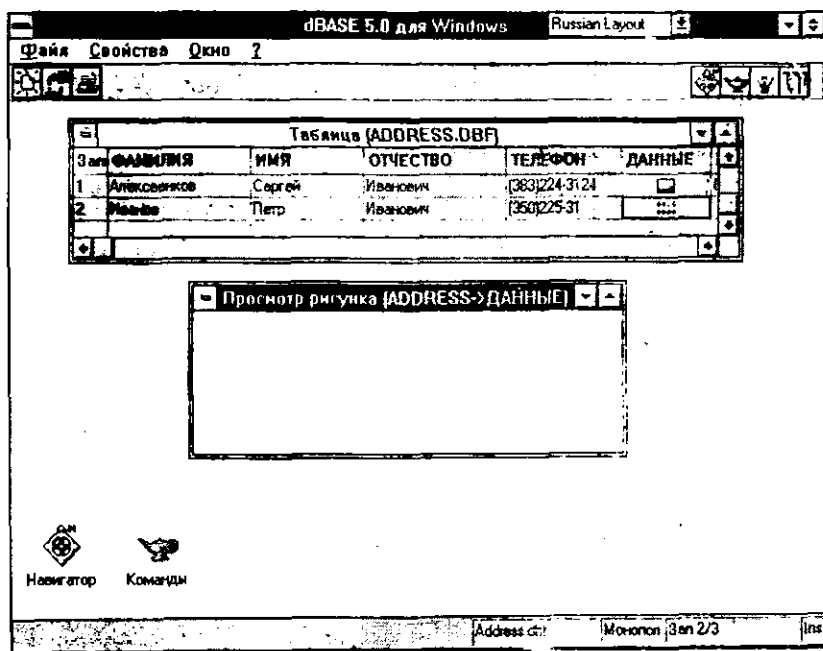


Рис. 3.12. Окно просмотра графической информации.

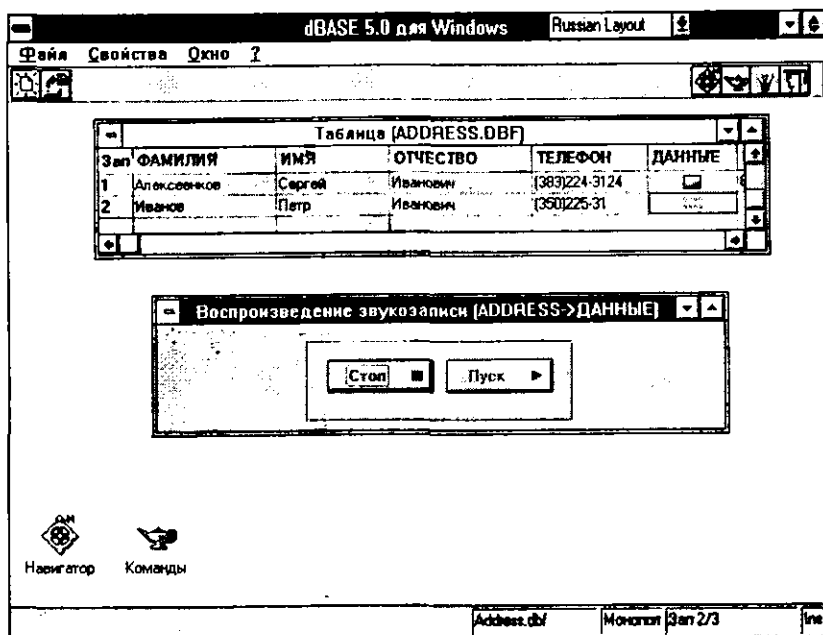


Рис. 3.13. Окно диалога воспроизведения звука.

Пример 3.3. Вставка изображения в двоичное поле

Прежде чем поместить изображение в двоичное поле, вы должны с помощью какого-либо средства создать требуемое изображение и поместить его в файл. После этого вам предстоит скопировать графическую информацию в таблицу dBASE. Для этого выполните следующие действия:

1. Добавьте в таблицу ADDRESS двоичное поле с наименованием "Фотография", в котором будут храниться изображения абонентов. Откройте окно редактирования для этой таблицы.
2. С помощью мыши или клавиши *Tab* перейдите в двоичное поле "Фотография".
3. Нажмите клавишу *F9* или два раза нажмите на кнопку мыши и укажите, что вы собираетесь помещать в данное поле изображение.
4. Выберите команду Вставить из файла из меню Файл. На экране появится окно диалога "Выбор рисунка" (рис. 3.14).
5. Выберите нужный графический файл.
6. Для завершения выбора файла нажмите ОК. В результате графическое изображение будет вставлено в указанное поле. При этом содержимое файла-источника останется без изменения.

На рис. 3.15 показана таблица ADDRESS, запись которой содержит графическое изображение.

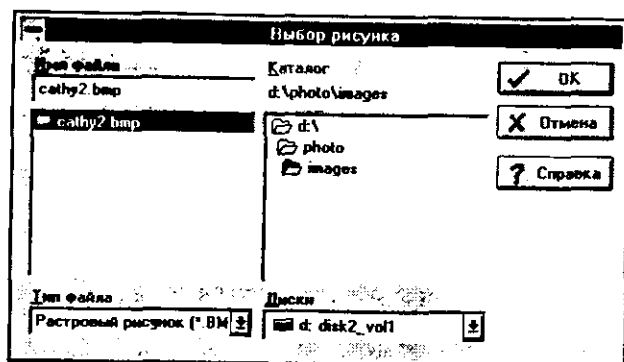


Рис. 3.14. Окно диалога "Выбор рисунка".

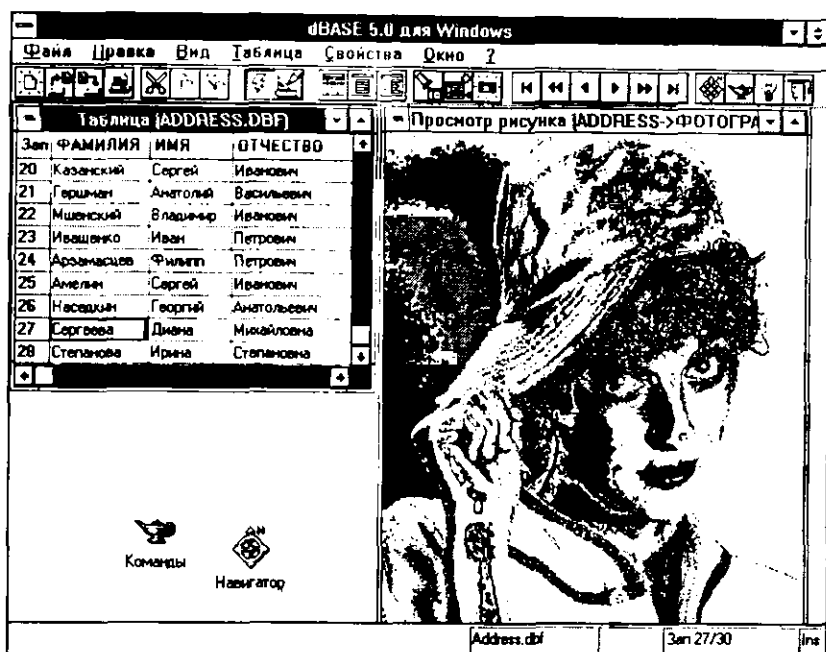


Рис. 3.15. Таблица ADDRESS, в которую вставлено графическое изображение.

Пример 3.4. Вставка звукового файла в двоичное поле

Двоичное поле может содержать как изображения, так и звуковой образ. При этом допускается хранение в разных записях одного и того же поля как изображения, так и звуковые образы. В качестве звукового образа может служить любой звуковой файл с расширением .WAV, например любимая мелодия, звуковое поздравление, краткая звуковая характеристика.

1. Добавьте в таблицу ADDRESS двоичное поле с наименованием "Звук". Откройте окно редактирования для этой таблицы.
2. Добавьте новую запись и введите фамилию, имя и отчество.

3. С помощью мыши или клавиши *Tab* перейдите в двоичное поле “Звук”.
4. Нажмите клавишу *F9* или два раза нажмите на кнопку мыши и укажите, что вы собираетесь помещать в данное поле звуковой образ.
5. Выберите команду **Вставить из файла** из меню **Файл**. На экране появится окно диалога “Выбор звукозаписи” (рис. 3.16).
6. Выберите требуемый звуковой файл.
7. Для завершения выбора файла нажмите **ОК**. В результате звуковое изображение будет вставлено в указанное поле.

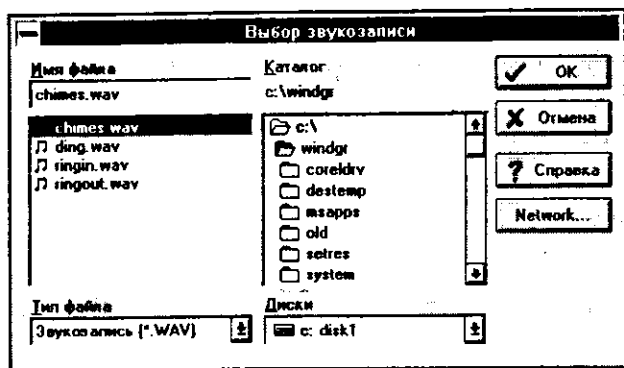


Рис. 3.16. Окно диалога “Выбор звукозаписи”.

Вы не можете редактировать сохраненный в поле звуковой образ, но можете прослушать его используя встроенный в компьютер динамик или установленный в вашем компьютере Sound Blaster.

Ввод данных в OLE-поле

OLE (Object Linking and Embedding) — это механизм связи и внедрения, который используется в среде Windows. OLE-поле позволяет установить связь данных в таблице dBASE с данными, хранящимися в других приложениях, работающих под Windows. Обмен данными с приложениями Windows будет рассмотрен в главе 16, в которой мы подробно обсудим размещение и модификацию OLE-объектов.

Сохранение данных и отказ от сделанных изменений

При работе в режиме редактирования dBASE блокирует текущую запись. Запись остается заблокированной до тех пор, пока вы не покинете ее пределы. Это сделано для того, чтобы несколько пользователей не могли одновременно редактировать содержимое одной и той же записи. Режим блокировки записей действует как в многопользовательском, так и в однопользовательском режиме.

Вы можете отказаться от сделанных изменений в текущей записи до того момента, как выйдете из нее. Это можно сделать двумя способами.

1. Если вам требуется отказаться от изменений, сделанных в текущем поле записи, выполните любое из приведенных ниже действий.

- Выполните команду **Правка | Отменить**
- Выполните команд **Файл | Отменить изменения**
- Нажмите клавишу **Alt+Backspace**

2. Для того чтобы отказаться от изменений, сделанных во всех полях текущей записи, выполните команду **Файл | Отменить изменения** и **закройте** или нажмите клавиши **Esc** или **Ctrl+Q**.

При выходе из текущей записи dBASE сохраняет все сделанные изменения и разблокирует запись. Поэтому вы можете воспользоваться приведенными средствами отказа от результатов редактирования только до того, как вы покинете эту запись.

Поскольку dBASE автоматически сохраняется данные по мере их ввода (или редактирования) запись за записью, вам не нужно предпринимать специальных мер для сохранения данных при завершении сеанса ввода или редактирования данных.

Поиск требуемой записи

В процессе работы с информацией, содержащейся в таблицах, вам приходится осуществлять поиск требуемой записи. Например, прежде чем приступить к редактированию таблицы, вам необходимо установить курсор в требуемую запись. Для этой цели вы можете использовать мышь или клавиши, приведенные в табл. 3.2.

Реакция dBASE на нажатие клавишей, предназначенных для перемещения курсора, приведена в табл. 3.2. Если вы используете мышь, вы можете просто переместить курсор в требуемое поле, после чего нажать кнопку мыши. С помощью мыши вы можете также осуществить прокрутку изображения для размещения курсора.



Для горизонтальной прокрутки изображения на один столбец поместите курсор мыши на правую или левую кнопку (стрелку) горизонтальной полосы прокрутки, расположенную в нижней части окна, и нажмите кнопку мыши.



Для вертикальной прокрутки изображения на одну строку поместите курсор мыши на верхнюю или нижнюю кнопку (стрелку) вертикальной полосы прокрутки, расположенную в правой части окна, и нажмите кнопку мыши.



Для ускоренной вертикальной или горизонтальной прокрутки изображения установите курсор мыши на движок соответствующей полосы прокрутки. Далее, нажмите кнопку мыши и, не отпуская ее, переместите курсор в требуемое поле.

Ускоренный способ поиска требуемой величины

При работе с большими таблицами вам часто требуется не просто переместить курсор на несколько полей или строк, а найти определенную запись или значение. В этом случае поиск требуемой записи с помощью мыши или клавишей может оказаться крайне неэффективным.

Для поиска требуемой величины в указанном поле вы можете воспользоваться командой **Найти записи** из меню **Таблица** (рис. 3.17). Кроме того, имеется команда **Перейти к записи** из этого же меню, которая позволяет вам быстро переместить курсор

в требуемое поле и найти строку с указанным номером записи. Команда Таблица | Заменить записи позволяет также осуществить операцию поиска указанной величины и замену ее новым значением, которая будет рассмотрена в разделе “Редактирование данных в таблицах”.

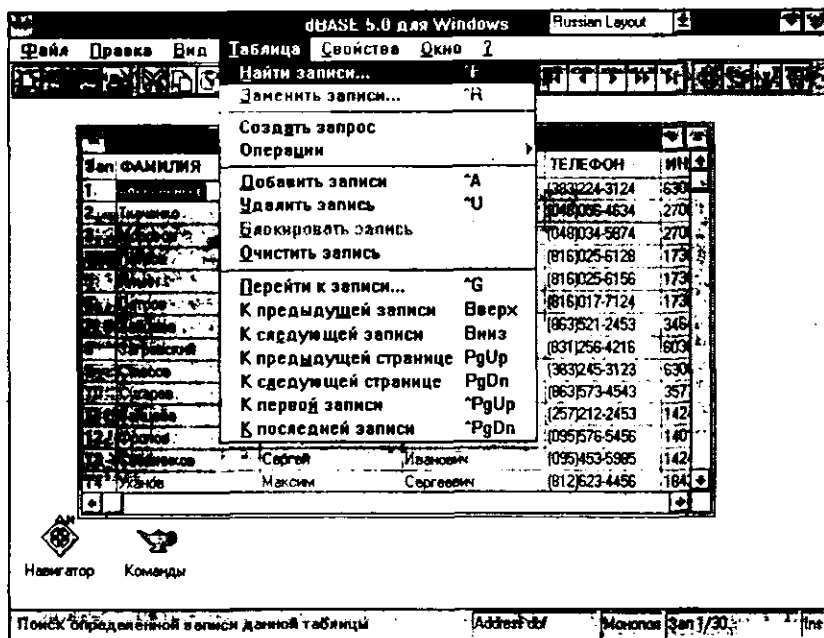


Рис. 3.17. Команды меню Таблица.

Поиск записи с определенным номером

dBASE автоматически присваивает номера каждой записи. Номера записей отображаются в левом столбце. Для того, чтобы отыскать запись с нужным номером, вызовите меню Таблица, из которого выберите команду Перейти к записи. В результате на экране появится окно диалога, в котором отображается номер записи в таблице (рис. 3.18). Введите в строку ввода номер требуемой записи и нажмите ОК. По окончании поиска вы увидите на экране фрагмент таблицы, содержащий запись с указанным номером. При этом найденная запись станет текущей.

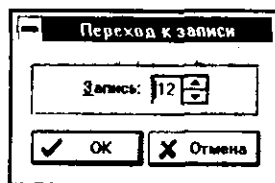


Рис. 3.18. Окно диалога “Переход к записи”.

Команду Перейти к записи целесообразно использовать в тех случаях, когда вы знаете хотя бы приблизительное расположение записи в большой таблице. Используя эту опцию, вы можете быстро переместить курсор в определенное место, после чего найти требуемую запись с помощью клавиш *PgUp* и *PgDn*.

Поиск записи, содержащей указанное значение

Наиболее эффективным средством является поиск определенного значения в требуемом поле. Для этого вы можете воспользоваться одним из следующих способов:

- Выполните команду Таблица | Найти записи
- Нажмите на панели инструментов кнопку Найти записи
- Нажмите на клавиатуре *Ctrl+F*

Независимо от выбранного вами способа на экране появится окно диалога “Поиск записей”, которое содержит список всех полей таблицы (рис. 3.19). Если в строке ввода Поля для поиска вы не увидите имени нужного вам поля, найдите требуемое поле с помощью полосы прокрутки. После выбора поля напечатайте в строке ввода значение, по которому будет осуществляться поиск. В заключение выберите Найти или нажмите *Enter*.

Независимо от текущего места расположения курсора в таблице dBASE переместит его в первую запись таблицы, которая содержит указанную величину.

Возможно, что поле, в котором вы осуществляете поиск, содержит несколько указанных величин. При этом первая строка, найденная dBASE, не содержит требуемой вам записи. В этом случае вы можете осуществить повторный поиск установив для этого переключатель поиска в состояние “Текущая запись”.

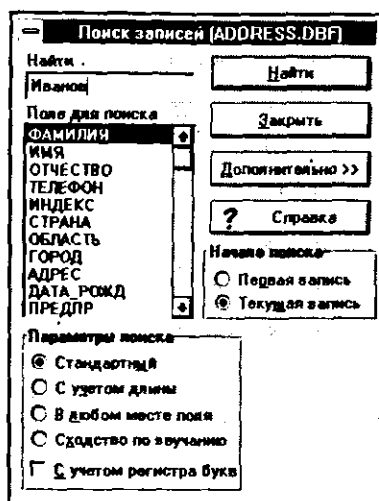
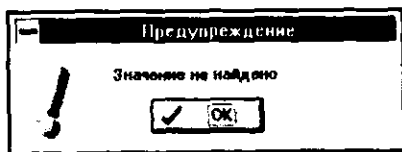


Рис. 3.19. Окно диалога “Поиск записей”.

Эти средства позволяют вам быстро просмотреть все записи в таблице, содержащие требуемую величину.



Если при поиске вы указали величину, которая отсутствует в данном поле, dBASE выдаст следующее предупреждение и курсор останется в текущей позиции.

Аналогичное сообщение появится на экране, если вы воспользовались средствами поиска повторного поиска, но в таблице больше не содержится записей с указанной величиной в поле поиска.

Давайте еще раз обратимся к окну диалога “Поиск записей”, представленному на рис. 3.19. Это окно содержит следующие опции:

Опция	Назначение
Стандартный	Осуществляет поиск значения, совпадающее или начинающееся со значения, указанного в поле Найти .
С учетом длины	Ищет записи, поле которых содержит точное значение поля Найти с учетом длины поля.
В любом месте поля	Ищет записи, поле которых содержит внутри себя в любом месте значение поля Найти .
Сходство по звучанию	Осуществляет поиск по фонетическому сходству, т.е. ищет записи, поле которых содержит значение, похожее по звучанию на значение в поле Найти .
С учетом регистра букв	При поиске учитывает регистр написания букв.

Если вы предполагаете искать значения поля, которые точно совпадают с заданным значением, включите опцию **С учетом длины**. Такой поиск нельзя назвать достаточно гибким. Предположим, что ваша таблица содержит несколько сот (или даже тысяч) записей. В этом случае вы можете просто не помнить точно, какое же значение требуется найти. Например, поле “Предприятие” содержит наименование компаний Bel-ABM и Bel-IBM, и вы не помните в какой из этих компаний работает нужный вам абонент. В такой ситуации вы можете вначале просмотреть записи о всех абонентах, работающих в Bel-ABM, а затем в Bel-IBM. Однако, такой подход представляется неэффективным. dBASE предоставляет в ваше распоряжение средство, с помощью которого можно осуществить поиск не по значению всего поля, а лишь по его части. Если в рассматриваемом примере вы будете выполнять поиск в поле “Предприятие”, задав в качестве значения для поиска Bel, то dBASE найдет записи со значениями Bel, Bel-ABM и Bel-IBM (если таковые имеются в указанном поле). В этом случае вы должны включить опцию **Стандартный**. Если же вы знаете только часть текстового значения, которое к тому же может находиться в любом месте поля, используйте опцию **В любом месте поля**.

По умолчанию dBASE при поиске величин не делает различия между прописными и строчными буквами. Вы можете изменить эту установку и тем самым сделать поиск чувствительным к регистру, который был использован при задании значения для поиска. Для этого выберите опцию **С учетом регистра букв**. После этого dBASE будет воспринимать “Тихомиров” и “тихомиров” как разные значения. Поэтому, если при поиске вместо прописной буквы вы введете строчную (или наоборот), требуемое значение не будет найдено.

При поиске записей вы можете использовать условие совпадения значения выбранного поля со значением другого поля, имя которого в круглых скобках указывается в поле **Найти**. Например, в таблице ORDSALE, которая содержит список заказов и продаж, вы можете найти записи, для которых дата продажи равна дате заказа. Для этого выберите поле для поиска “Дата_прод”, а в строке ввода напечатайте Дата_заказ. В поле **Найти** вы можете указывать не только имя поля, но и любое выражение, допустимое в dBASE. Например, для поиска в таблице ADDRESS записей, для которых фамилия и имя начинаются с одной и той же буквы, выберите поле “Фамилия”, а в строке ввода напечатайте (substr(ИМЯ,1,1)).

Расширенные возможности поиска

В окне диалога “Поиск записей” вы можете задать дополнительные условия выбора записей. Выберите для этого кнопку **Дополнительно>>** и в окне диалога появится новая область “Отбор записей” (рис. 3.20). Эта область содержит следующие опции:

Опция	Назначение
Все	Осуществляет поиск по всем записям таблицы.
До конца таблицы	Осуществляет поиск от текущей записи (включая ее) до конца файла.
До записи	Осуществляет поиск от текущей записи (включая ее). В поиске участвует указанное в счетчике количество записей. Если вы укажете больше записей, чем содержит таблица, действие будет выполнено со всеми записями от текущей записи до конца файла. Этот параметр недоступен для таблиц в формате Paradox и таблиц, хранящихся на SQL-сервере.
Запись	Поиск осуществляется в единственной записи с заданным номером. При поиске записей использование этой опции не имеет большого смысла.
Условие на запись	Определяет условие, позволяющее применить действие только к некоторым записям, которые удовлетворяют этому условию.
Условие на диапазон	Определяет условие поиска, при котором поиск осуществляется до тех пор, пока это условие истинно.

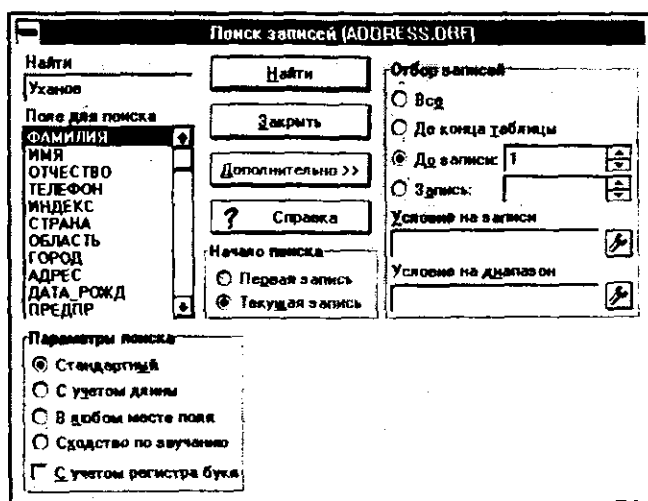


Рис. 3.20. Окно диалога “Поиск записей” с дополнительными опциями.

Пример 3.5. Расширенные возможности поиска

В качестве примера рассмотрим поиск клиентов с фамилией “Морозов”, проживающих в России.

1. Откройте окно редактирования таблицы ADDRESS.
2. Выполните команду **Таблица | Поиск записей** и нажмите кнопку **Дополнительно>>**.
3. Выберите для поиска поле "Фамилия" и строке ввода напечатайте Морозов.
4. Перейдите в поле **Условие** на записи и введите в этом поле условие выбора **СТРАНА='Россия'**. Для ввода условия выбора можете воспользоваться построителем выражений, который вызывается при нажатии на кнопку **Инструмент**.
5. Для поиска записей нажмите **Найти** и dBASE найдет первую запись, удовлетворяющую условию поиска (рис. 3.21). Для перехода к следующей записи, удовлетворяющей условию поиска нажмите еще раз кнопку **Найти**.

Зап.	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ТЕЛЕФОН	ИНДЕКС	СТРАНА
1	Алексенков	Сергей	Иванович	3832243124	630090	Россия
2	Ткаченко	Марина	Михайловна	0480564634	270005	Украина
3	Морозов	Юрий	Анатолевич	898	270029	Украина
4	Петров	Михаил	Николаевич	8160256128	173024	Россия
5	Ильин	Филипп	Петрович	8160256156	173003	Россия
6	Петров	Михаил	Николаевич	8160177124	173003	Россия
7	Зайцев	Людмила	Николаевна	8635212453	346415	Россия
8	Загребский	Андрей	Анатолевич	8312564216	603006	Россия
9	Смисов	Филипп	Петрович	3832453123	630056	Россия
10	Сидоров	Андрей	Анатолевич	8635734543	357736	Россия
11	Зайцев	Дарья	Ивановна	2572122453	142490	Россия
12	Фролов	Виктор	Михайлович	0955765456	140160	Россия
13	Савиных	Сергей	Иванович	0954535986	142432	Россия
14	Уханов	Максим	Сергеевич	8126234456	184230	Россия
15	Морозов	Андрей	Михайлович	3432324542	620151	Россия
16	Райский	Максим	Сергеевич	8462534542	443115	Россия
17	Антонов	Максим	Юрьевич	0955321532	125167	Россия

Рис. 3.21. Запись, удовлетворяющая условию поиска.

Теперь, после того как вы научились перемещать курсор в пределах таблицы и осуществлять поиск записей, перейдем непосредственно к редактированию содержимого таблиц.

Редактирование данных в таблицах

В процессе работы с данными вам приходится вводить в таблицу новые данные, корректировать уже существующие и удалять ненужные записи, т.е. редактировать содержимое таблицы. В табл. 3.8 представлены клавиши, используемые при редактировании в dBASE.

При редактировании Метод-поля используются специальные клавиши, приведенные в табл. 3.5. Далее мы рассмотрим общие средства для редактирования данных в таблицах dBASE, после чего остановимся на некоторых специальных средствах, применимых только для редактирования полей конкретных типов.

Табл. 3.8. Клавиши, используемые при редактировании

Клавиша	Назначение
<i>Ctrl+A</i>	Добавляет в таблицу новую запись
<i>Del</i>	Удаляет выделенный текст или символ, на который установлен курсор
<i>Ins</i>	Переключает режим вставки и замены
<i>Ctrl+U</i>	Удаляет из таблицы существующую запись
<i>Ctrl+L</i>	Блокирует текущую запись
<i>Backspace</i>	Удаляет символ слева от курсора или выделенный текст
<i>Alt+Backspace</i> или <i>Ctrl+Z</i>	Осуществляет отказ от всех изменений, сделанных в текущем поле
<i>Ctrl+Backspace</i>	Удаляет слово слева от курсора или выделенный текст
<i>Esc</i>	Осуществляет отказ от всех изменений, сделанных в текущей записи
<i>Ctrl+Ins</i> или <i>Ctrl+C</i>	Копирует выделенный текст в буфер
<i>Shift+Del</i> или <i>Ctrl+X</i>	Удаляет выделенный текст в буфер
<i>Shift+Ins</i> или <i>Ctrl+V</i>	Вставляет содержимое буфера
<i>Tab</i> и <i>Enter</i>	Завершает ввод значения и переходит к следующему полю
<i>Shift+Tab</i>	Завершает ввод значения и переходит к предыдущему полю

Удаление записи

Действия по удалению записей из таблицы зависят от типа таблицы. Для таблиц в формате Paradox для удаления записей перейдите на любое поле удаляемой записи и нажмите *Ctrl+U* или выберите Таблица | Удалить запись.

Для таблиц в формате dBASE удаление записей состоит из двух шагов. Вы сначала помечаете запись для удаления, а затем физически удаляете помеченные записи из таблицы. Процесс физического удаления записей называется упаковкой таблицы. Для установления метки удаления записи нажмите *Ctrl+U* или выберите Таблица | Удалить запись. Для упаковки таблицы, т.е. физического удаления всех помеченных для удаления записей, выполните команду Таблица | Операции | Удалить помеченные.

Для таблиц в формате dBASE имеется два режима отображения данных. В первом режиме помеченные записи не отображаются на экране. В этом случае после выполнения команды Таблица | Удалить запись, помеченная для удаления запись исчезнет из окна таблицы. Во втором режиме в окне таблицы появляется отдельный столбец, расположенный рядом со столбцом "Зап" и имеющий заголовок "Уд." (рис. 3.22). Значок "X" в этом столбце означает, что запись помечена для удаления. В этом режиме для установки метки удаления вы можете нажать мышью флажок удаления записи.

Для установки режима отображения данных используется карточка "Таблица" (рис. 3.23) окна диалога "Свойства среды dBASE". Для открытия этой карточки выполните команду Свойства | Среда dBASE и в окне диалога нажмите на корешок "Таблица". Флажок Скрыть удаленные записи в этой карточке задает режим отображения удаленных записей. Если флажок установлен, то записи, помеченные для удаления, не будут отображаться в окне таблицы. В противном случае в таблице будут отображаться все записи, включая и удаленные.

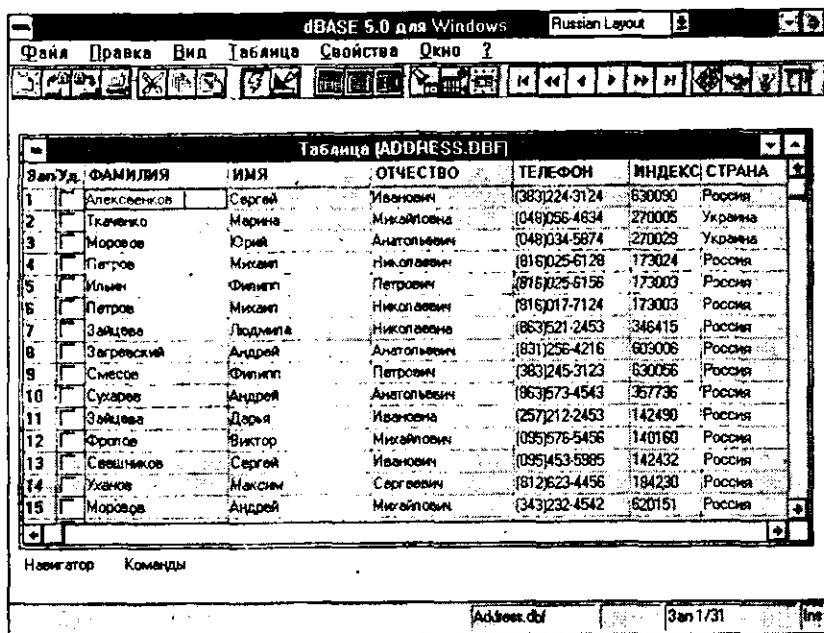


Рис. 3.22. Отображение помеченных для удаления записей.

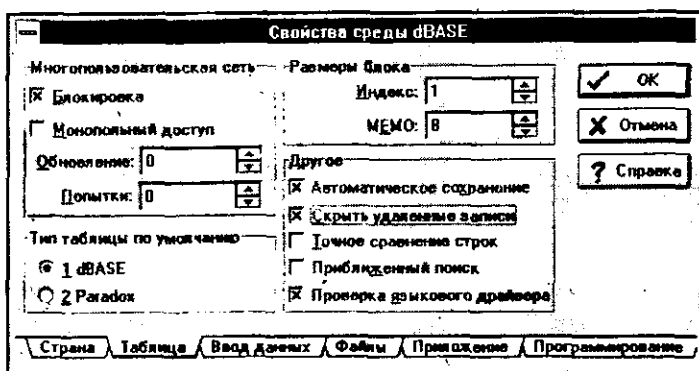


Рис. 3.23. Карточка "Таблица" окна диалога "Свойства среды dBASE".

До упаковки таблицы все помеченные для удаления записи могут быть восстановлены. Для восстановления помеченной записи вы можете выполнить любое из следующих действий:

- Выполните команду Таблица | Снять пометку
- Нажмите *Ctrl+U*
- Нажмите мышью флажок удаления записи

Удаление помеченных записей окончательно удаляет их из таблицы.

В качестве примера установим курсор на поле "Фамилия" третьей записи. Если после этого вы нажмете *Ctrl+U*, в третьей записи появится флажок удаления записи (рис. 3.24).

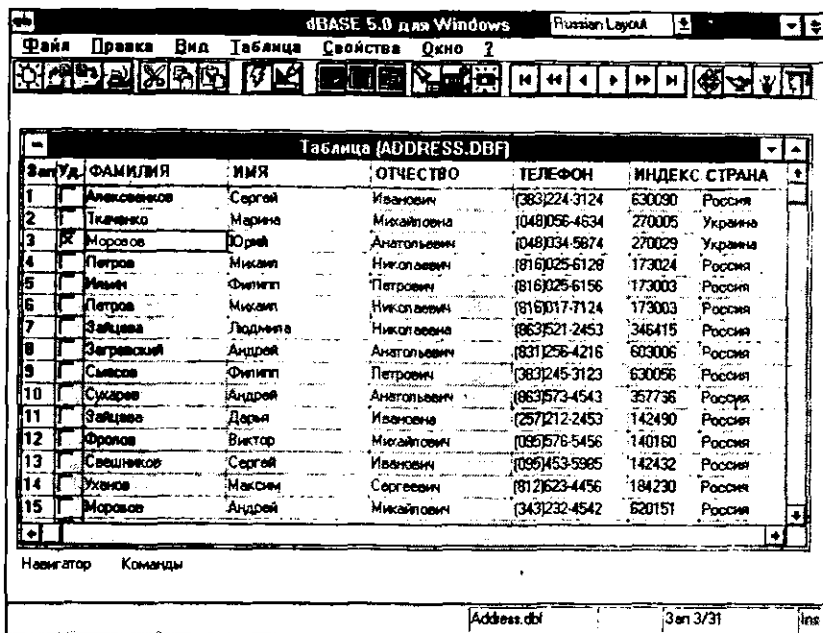


Рис. 3.24. Установка метки удаления записи.

Удаление и восстановление группы записей

В dBASE имеются средства удаления и восстановления не только текущей записи, но и целой группы записей, удовлетворяющих заданным условиям. Для удаления группы записей выполните команду Таблица | Операции | Пометить для удаления. На экране появится окно диалога “Удаление записей”, которое содержит поля для определения условия удаления записей. Назначение этих полей совпадает с полями области “Отбор записей” окна диалога “Поиск записей”, которое мы уже рассмотрели в этой главе. В качестве примера введем условие удаления всех клиентов с фамилией “Морозов” (рис. 3.25).

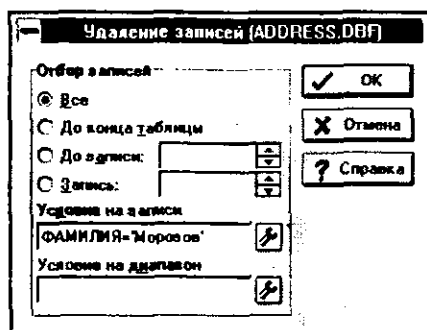


Рис. 3.25. Определение условия удаления группы записей.

Аналогично осуществляется и восстановление помеченных для удаления записей командой Таблица | Операции | Снять пометки.

Кроме этих двух команд в dBASE имеется команда Таблица | Операции | Удалить все записи, которая удаляет все записи таблицы. Данная команда осуществляет пол-

ную очистку таблицы и физическое удаление всех ее записей, в том числе и записей, не помеченных для удаления. Восстановление записей после выполнения данной команды невозможно.

Очистка записи

Еще одной полезной командой является команда **Таблица | Очистить запись**. Эта команда очищает все поля текущей записи и они становятся пустыми. Очистка записи наиболее полезна при замены пустыми значениями логических и числовых полей. Разница между пустым и нулевым значением в числовых полях становится очень важной при вычислении статистических значений, таких как среднее значение по столбцу.

Добавление новой записи

Для добавления новых записей в таблицу воспользуйтесь командой **Таблица | Добавить записи** или клавишей **Ctrl+A**. С помощью этих средств в таблицу добавляется пустая строка, в которую впоследствии могут быть введены данные.

Процедура добавления записей в индексированную таблицу аналогично приведенной выше за исключением того, что вы можете не заботиться о точном местонахождении новой записи. После ввода данных dBASE упорядочит данные в соответствии со значением ключевых значений. В результате упорядочивания записей может оказаться, что введенная вами запись исчезнет с экрана. Для того чтобы убедиться в том, что запись введена в таблицу, обратите внимание на изменившееся число записей, отображающееся в правом углу строки состояния.

Добавление записей из одной таблицы в другую будет рассмотрено в главе 10.

Редактирование полей

Для редактирования данных выделите требуемое поле. По умолчанию dBASE работает в режиме замены, поэтому при вводе новых данных в выделенное поле, старое значение будет удалено. Если вы хотите не удалить, а изменить старое значение перейдите в режим редактирования поля.

Редактирование Метод-полей

Если вы находитесь в Метод-поле, для перехода в режим редактирования Метод-поля нажмите **F9**. При переходе в этот режим открывается окно редактирования Метод-поля и вы можете редактировать содержимое этого поля.

При редактировании Метод-полей вы можете использовать клавиши, список которых приведен в табл. 3.5.

Пример 3.6. Использование буфера обмена Windows для копирования данных

В некоторых случаях вам может потребоваться скопировать содержимое какого-либо из полей. При работе с dBASE вы можете использовать для копирования буфер обмена Windows. Это средство позволяет скопировать значения из любых записей и может быть использовано при работе с полями любых типов. Однако, наиболее полезным это средство оказывается при работе с Метод-полями, в которых может быть использовано для копирования всего содержимого поля или выделенного блока.

Для копирования данных с помощью буфера Windows выполните следующие действия.

1. В качестве примера откройте окно для таблицы ADDRESS.
2. Установите курсор в поле, содержимое которого вы предполагаете скопировать. Переместите курсор в поле "Примечание" первой записи.
3. Введите режим Метод-поля, для чего выполните команду Вид | Содержимое поля или нажмите клавишу F9.
4. Выделите текст, который вы предполагаете скопировать. Вы можете выделить все содержимое поля или его часть. Наиболее просто это сделать с помощью мыши. Установите курсор на начало текста и нажмите кнопку мыши. Удерживая кнопку мыши, переместите курсор в конец текста, после чего отпустите кнопку мыши. В результате выбранная часть поля будет выделена более интенсивной окраской (рис. 3.26).

Если вам необходимо выделить текст, расположенный в нескольких строках, вы можете воспользоваться клавишами, приведенными в табл. 3.9.

Табл. 3.9. Клавиши выделения

Клавиша	Назначение
<i>Shift+ Home</i>	Выделить до начала строки
<i>Shift+ End</i>	Выделить до конца строки
<i>Shift ← и Shift →</i>	Выделить одно символ слева или справа
<i>Ctrl+Shift+→</i>	Продлить выделение на одно слово вправо
<i>Ctrl+Shift+←</i>	Продлить выделение на одно слово влево
<i>Shift ↑ и Shift ↓</i>	Продлить выделение на одну строку вверх или вниз

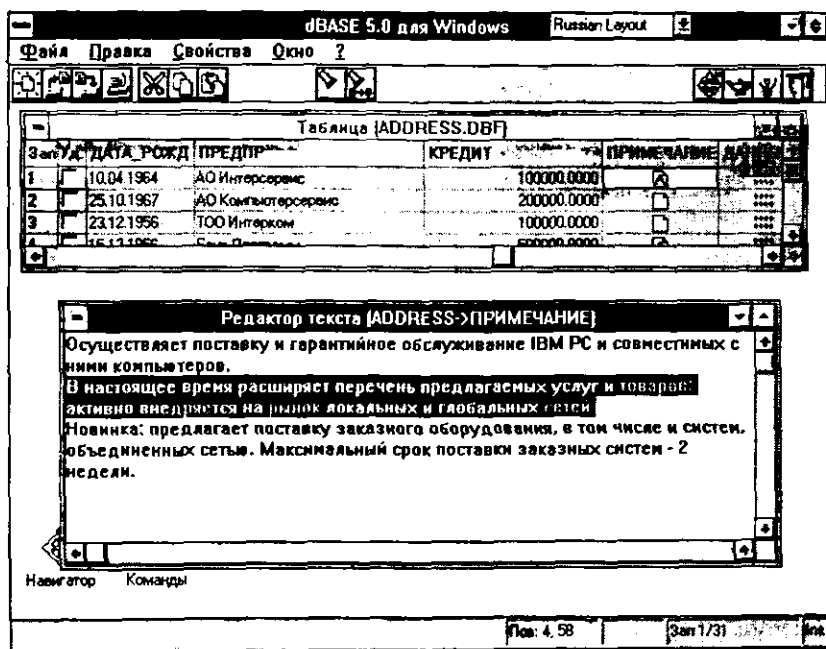


Рис. 3.26. Выделение фрагмента текста для копирования.

5. Скопируйте выделенный фрагмент текста в буфер. При выполнении операций с блоками текста вы можете использовать команды меню **Правка**, кнопки панели инструментов или оперативные клавиши:

Кнопка	Команда	Клавиши	Назначение
	Правка Вырезать	<i>Ctrl+Ins</i>	Удаляет выделенный текст или выделенные объекты и помещает их в буфер
	Правка Копировать	<i>Shift+Del</i>	Копирует в буфер выделенный текст или выделенные объекты, не удаляя из таблицы
	Правка Вставить	<i>Shift+Ins</i>	Копирует содержимое буфера

В текущем примере выберите **Правка | Копировать**, что позволит вам поместить содержимое выделенной области в буфер без удаления.

6. Установите курсор в ту запись, поле и место, куда нужно скопировать содержимое буфера. Далее выберите **Правка | Вставить**. Результат копирования содержимого буфера представлен на рис. 3.27.

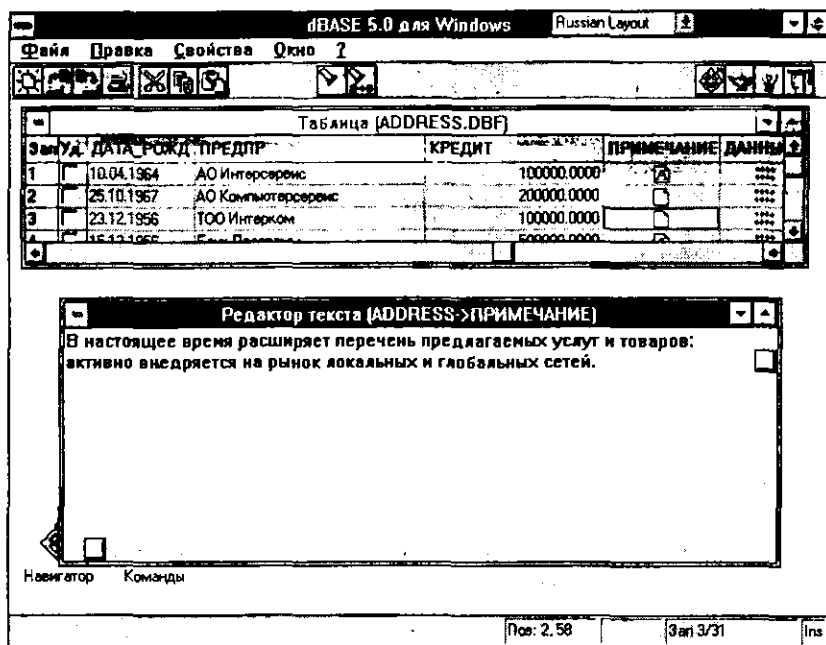


Рис. 3.27. Копирование фрагмента текста с помощью буфера обмена Windows.



В поле может быть вставлено только значение допустимого типа. Например, dBASE не позволит вам скопировать текст, содержащийся в буфере в поле дат.



Данные, введенные в буфер, сохраняются там до тех пор, пока вы не совершите одно из следующих действий:

- очистите буфер;
- поместите в буфер новые данные;
- выйдите из Windows.



Если при операциях с буфером вы допустили ошибку, для отмены ошибочного действия воспользуйтесь командой **Правка | Отменить**.

Кроме копирования значений полей вы можете скопировать в поле содержимое файла или наоборот скопировать в файл содержимое поля или его часть.

Пример 3.7. Копирование данных во внешний файл

Команда **Правка | Копировать в файл** позволяет вам скопировать во внешний файл содержимое Мемо-полей. Последовательность действий при копировании значения поля во внешний файл, рассмотрим на следующем примере.

1. Откройте таблицу ADDRESS и перейдите в поле “Примечание”.
2. Откройте окно редактирования Мемо-поля и выделите фрагмент текста, который вы предполагаете копировать.
3. Выберите **Правка | Копировать в файл**, в результате чего на экране появится окно диалога “Копирование в файл” (рис. 3.28).

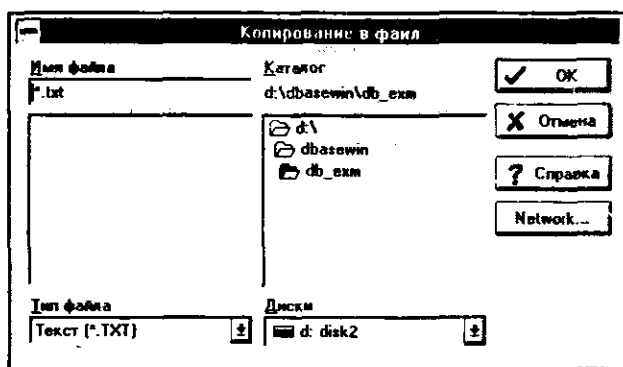


Рис. 3.28. Окно диалога “Копирование в файл”.

3. Введите в строку ввода **Имя файла** имя нового файла. При этом в строке ввода приведены расширения файлов, используемые dBASE по умолчанию (.PRG, .TXT, .WFM и .H). Не забудьте указать каталог, в котором должен располагаться создаваемый вами файл.
4. В заключение выберите **OK**, после чего dBASE создаст новый файл с указанным именем и поместит в него содержимое текущего поля.

Пример 3.8. Копирование из внешнего файла

Команда **Правка | Вставить из файла** предназначена для вставки значения из внешнего файла в текущее поле. С помощью этого средства вы можете скопировать данные в Мемо-поле. При этом должно быть открыто окно редактирования Мемо-поля.

Для того чтобы вставить значение из внешнего файла в поле dBASE, выполните приведенную ниже последовательность действий.

1. Откройте окно для таблицы ADDRESS и перейдите в поле, в которое требуется вставить значение (“Примечание”).
2. Откройте окно редактирования Мемо-поля.
3. Выберите **Правка | Вставить из файла**. dBASE откроет окно диалога “Вставка из файла” (рис. 3.29).

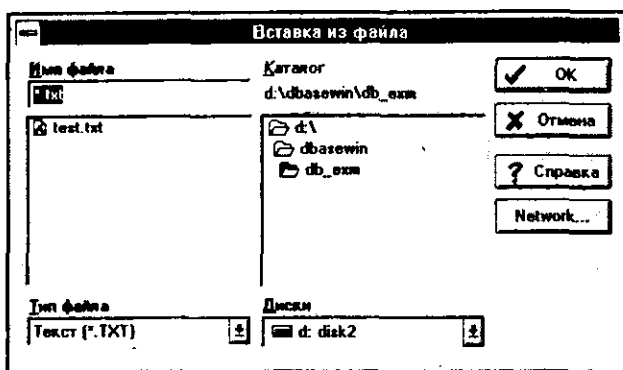


Рис. 3.29. Окно диалога “Вставка из файла”.

4. Список в левой части окна содержит наименования файлов. Выберите требуемый каталог и имя файла из списка.
5. Выберите OK. dBASE поместит содержимое указанного файла в текущее поле.

Замена данных

Возможна ситуация, когда вам потребуется заменить несколько одинаковых значений поля новым значением. Предположим, что ввод данных в таблицу осуществляется несколькими операторами. При этом один из операторов в поле “Страна” вводил значение Россия, а другой — сокращенное наименование Рос. В таком случае вам придется изменить все значения, введенные одним из операторов.

Для решения этой и аналогичных задач вы можете воспользоваться командой **Таблица | Заменить запись**. Эта команда осуществляет поиск всех вхождений указанной величины и заменяет их новым значением.

Пример 3.9.

В качестве примера заменим сокращенные значения Рос, введенные вторым оператором в таблицу ADDRESS, полным наименованием — Россия.

1. Откройте окно для таблицы ADDRESS.
2. Выберите **Заменить записи** из меню **Таблица** или воспользуйтесь комбинацией клавиш **Ctrl+R**.
3. На экране появится окно диалога “Замена записей” (рис. 3.30).

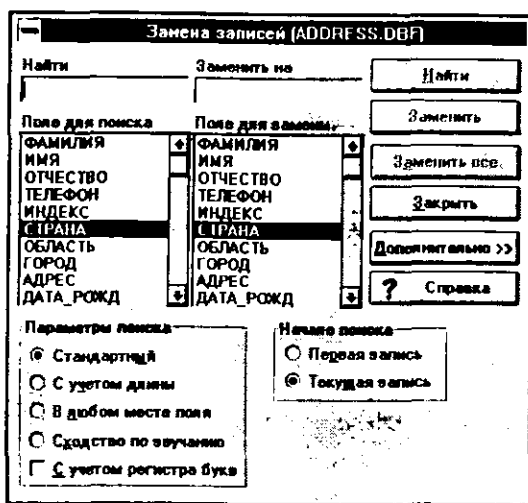


Рис. 3.30. Окно диалога “Замена записей”.

4. Это окно аналогично окну диалога “Поиск записей”, за исключением того, что окно “Замена записей” содержит дополнительную область для ввода значения, предназначенного для замены. Введите искомое значение **Рос** и значение, на которое его следует заменить **Россия**.

Поскольку оператор не уверен сколько букв он вводил в сокращенном наименовании и всюду ли использовал точку (**Рос.** — **Рос** — а может быть **Росс.**), при задании искомого значения был использован параметр поиска **Стандартный**.

В зависимости от конкретной задачи вы можете использовать любые параметры поиска, включая и расширенные возможности задания условия поиска.

5. Из списка полей поиска выберите поле, в котором вы предполагаете осуществлять поиск. В текущем примере выберите поле “**Страна**”.
6. Из списка полей замены выберите поле, в котором вы предполагаете осуществлять замену. В текущем примере выберите поле “**Страна**”. В строку ввода **Заменить на** введите значение для замены **Россия**.
7. Выберите **Найти**. Обратите внимание, что до нахождения первой записи, удовлетворяющей условию поиска, кнопка **Заменить** недоступна.
8. dBASE найдет первую запись, содержащую вхождение искомого значения в указанном поле и предложит выбрать один из трех вариантов:
 - заменить текущее значение;
 - заменить все значения;
 - найти следующее вхождение без замены значения.

Для замены текущего значения выберите **Заменить**, а для пропуска значения — **Найти**. В зависимости от сделанного вами выбора dBASE заменит или пропустит текущее значение и перейдет к поиску следующего вхождения искомого значения. После завершения процесса поиска и замены курсор останется в последней из записей, в которых была произведена замена.

Описанное средство для замены величин является весьма полезным средством. Однако, при его использовании нужно соблюдать осторожность, поскольку неточно заданный критерий для поиска может привести к непредсказуемым результатам. Например, вам потребовалось заменить в поле "Предприятие" значение "Bel-IBM" на "Bel&Rus-IBM". Если в качестве искомой величины вы укажете "Bel" и выберете стандартный режим замены, то в результате окажется измененным наименование требуемой компании. Найдя в таблице запись со значением "Bel-ABM", dBASE также предложит вам подтвердить ее изменение. Поэтому будьте внимательны при подтверждении замены, поскольку в dBASE отсутствуют средства для отказа от ошибочно сделанных изменений.

Описанное средство позволяет осуществлять поиск и замену значений, хранящихся в таблице. Однако, при замене содержимого Метод-поля необходимо быть внимательным, так как dBASE заменит в найденной записи все содержимое Метод-поля, а не его фрагмент, совпадающий со строкой поиска.

Параметры отображения записей

При работе с dBASE вы должны понимать, что информация, введенная в таблицу, хранится на диске. Записи, которые вы видите на экране, являются изображением таблицы, которое называют экранном образом. Хотя экранный образ и выглядит, как настоящая таблица, на самом деле это далеко не одно и то же. При просмотре и редактировании таблицы вы работаете с изображением таблицы. Если вы изменили значения одного или нескольких полей текущей таблицы, данные изменения присутствуют только в экранном образе. Сделанные вами изменения заносятся на диск только после того, как вы покинете текущую запись.

После выбора таблицы появляется в окне таблицы в режиме, установленном вами для данной таблицы. Параметры отображения таблицы сохраняются в свойствах таблицы, которые вы можете задать в окне диалога "Свойства таблицы" (рис. 3.31). В данном разделе мы рассмотрим свойства таблицы, содержащиеся в карточке "Записи", которая содержит параметры отображения записей в окне таблицы.

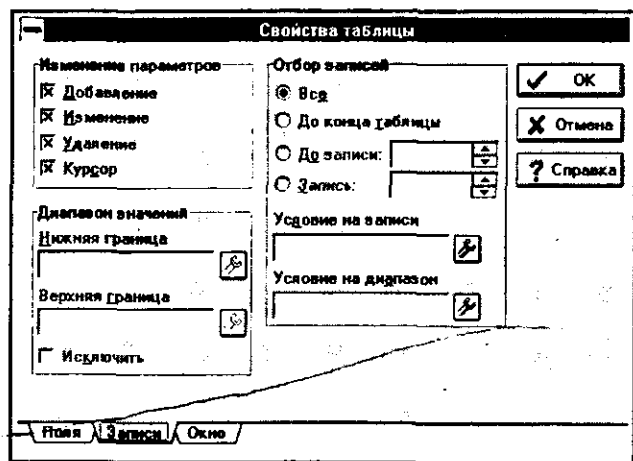


Рис. 3.28. Окно диалога "Свойства таблицы".

Область “Изменение параметров” содержит флажки, которые задают следующие режимы управления записями:

Флажок	Назначение
Добавление	Позволяет добавлять записи в данную таблицу.
Изменение	Позволяет изменять записи данной таблицы.
Удаление	Позволяет удалять записи данной таблицы.
Курсор	Позволяет перемещаться на новое место записи в индексе при изменении значения индексированного поля текущего индекса.

Область “Диапазон значений” используется для таблиц, имеющих индексы, и определяет условие отображения записей по значению текущего основного индекса. Поля **Нижняя граница** и **Верхняя граница** должны содержать выражения того же типа, что и текущий основной индекс. Если вы установите диапазон отображаемых записей, на экране появятся только те записи ключевые значения которых не меньше значения, заданного в поле **Нижняя граница** и не больше значения, заданного в поле **Верхняя граница**. Флажок **Исключить** определяет, будут ли отображаться записи, значения которых точно совпадают со значениями в полях **Нижняя граница** и **Верхняя граница**. Для создания выражений вы можете использовать построитель выражений.

Область “Отбор записей” содержит поля, которые определяют условия отображения записей.

Используя эти параметры вы можете защитить таблицу от случайных изменений или удаления данных, а также настроить ее для конкретного пользователя.

Редактирование данных в режиме бланка, перечня и формы

До настоящего момента вы видели отображение данных на экране в виде таблицы, т.е. ее строк и столбцов. Такой формат представления информации так и называется табличным. Однако существуют и другие форматы отображения данных — в виде бланка, перечня и формы. Очевидно, что все эти форматы обеспечивают просмотр данных одних и тех же таблиц. Давайте посмотрим, в чем же заключаются особенности представления данных в каждом из форматов.

При табличном представлении данных вы можете отображать в окне таблицы набор записей, число которых определяется размерами окна и параметрами шрифта. При этом на экране одновременно могут разместиться, как правило, не более 4 — 8 полей. Количество полей, одновременно отображаемых на экране, определяется их размером. При просмотре в режимах бланка и перечня на экране одновременно отображается меньшее количество записей, но вы можете разместить на экране значительно больше полей.

В режиме бланка на экран выводится только одна запись в виде горизонтальной строки или нескольких последовательно расположенных строк полей. Для отображения таблицы в режиме бланка выполните команду Вид | Режим бланка. Пример отображения таблицы ADDRESS в режиме бланка приведен на рис. 3.32.

В режиме перечня на экран выводится только одна запись в виде вертикального столбца полей. В левой части перечня выводятся имена полей, а в правой части — их значения. Для отображения таблицы в режиме перечня выполните команду Вид | Режим перечня. Пример отображения таблицы ADDRESS в режиме перечня приведен на рис. 3.33.

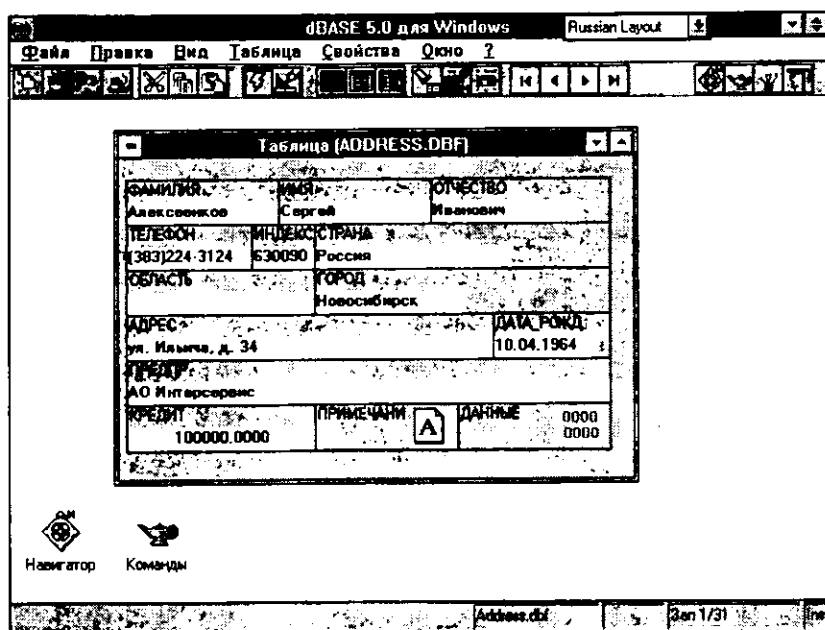


Рис. 3.32. Отображения таблицы ADDRESS в режиме бланка.

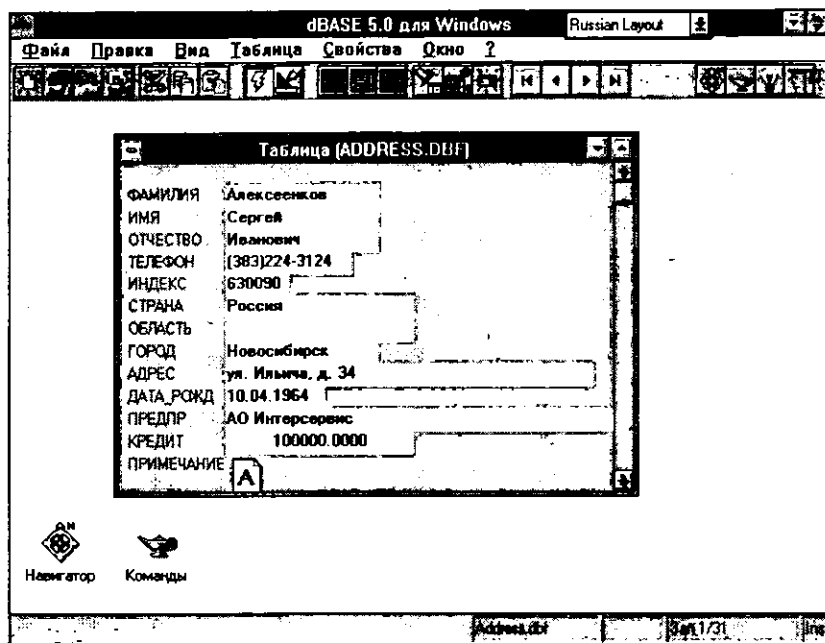


Рис. 3.33. Отображения таблицы ADDRESS в режиме перечня.

Дополнительно для переключения режимов отображения вы можете использовать клавишу **F2**.

Формы являются наиболее удобным средством для просмотра и редактирования таблицы. На рис. 3.34 показана форма для таблицы ADDRESS. Вы видите, что с по-

мощью этой формы на экране отображены все поля одной записи. Формы создаются с помощью конструктора форм и позволяют вам одновременно отображать одну или несколько записей из одной таблицы или же связанных между собой таблиц. С конструктором форм мы познакомим вас в главе 8.

The screenshot shows the dBASE 5.0 for Windows interface. The main window is titled 'Форма' (Form) and contains a sub-window titled 'Список клиентов' (Client List). The form has several fields: 'Фамилия' (Last Name) with value 'Алексеев', 'Имя' (First Name) with value 'Сергей', 'Отчество' (Patronymic) with value 'Иванович', 'Предприятие' (Company) with value 'АО Интерсервис', 'Постоянный' (Permanent) with a checked box, 'Связь с клиентом' (Contact with client) with a checked box, 'Адрес' (Address) with value 'г. Новосибирск, ул. Ильича, д. 34', 'Кредит' (Credit) with value '200000', and a text area for 'Осуществляет поставку и гарантийное обслуживание IBM PC и совместимых с ними компьютеров' (Provides supply and warranty service for IBM PC and compatible computers). The status bar at the bottom shows 'Address.dbf' and 'Зап 1/30'.

Рис. 3.34. Форма для таблицы ADDRESS.

Для того, чтобы открыть созданную ранее форму воспользуйтесь одним из следующих способов.

- Выберите **Открыть** с типом файла .WFM .
- Выберите тип **Формы** в навигаторе и затем откройте требуемую форму из списка существующих форм.

Внешний вид панели инструментов при отображении данных в режимах бланк, перечня и формы содержит те же кнопки управления, что и в режиме таблица, за исключением кнопок перемещения по группам записей.

В табл. 3.11 приведены клавиши, используемые для перемещения курсора и редактирования данных.

Табл. 3.11. Клавиши, используемые при работе в режимах бланк, перечень и форма

Клавиша	Назначение
← и →	Перемещает курсор на один символ влево или вправо
↑ и ↓	Перемещает курсор в предыдущее или следующее поле
PgUp	Перемещает курсор на предыдущую запись
PgDn	Перемещает курсор на следующую запись
Home	Перемещает курсор в начало текущего поля
End	Перемещает курсор в конец текущего поля

Клавиша	Назначение
<i>Ctrl+U</i>	Удаляет запись
<i>Backspace</i>	Удаляет символ слева или выделенный текст
<i>Del</i>	Удаляет символ справа или выделенный текст
<i>Alt+Spacebar</i>	Отменяет все изменения, сделанные в текущей записи

Заключение

- Панель инструментов в окне таблицы показана на рис. 3.1, назначение соответствующих кнопок приведено в табл. 3.1.
- Изменения, сделанные при редактировании, по умолчанию автоматически сохраняются запись за записью. Вы можете изменить этот режим используя окно диалога “Свойства среды dBASE”.
- Для отказа от изменений, сделанных в текущем поле, выберите **Правка | Отменить** или воспользуйтесь клавишным эквивалентом данной команды — *Alt+Backspace*.
- Для того, чтобы отказаться от изменений, сделанных в текущей записи, нажмите *Esc*.
- Режим редактирования поля предоставляет вам возможность редактировать содержимое поля, вместо того, чтобы заменять его. Для перехода в режим редактирования нажмите любую из клавиш перемещения в поле (*←*, *→*, *Home*, *End*) или кнопку мыши.
- Ввод данных в Мемо-поле осуществляется в режиме Мемо-поля. Для перехода в режим Мемо-поля нажмите *F9*.
- Команды **Поиск записей**, **Перейти к записи**, **Заменить записи** из меню **Таблица** позволяют осуществить поиск и замену значений в поле.
- Для копирования всего значения Мемо-поля или выделенного фрагмента может быть использован буфер обмена Windows. Кроме того, команды **Копировать в файл** и **Вставить из файла** из меню **Правка** позволяют вам копировать значение Мемо-поля во внешний файл и из внешнего файла, соответственно.
- Для редактирования и просмотра информации в таблицах в dBASE существуют несколько режимов отображения данных — режим таблицы, режим перечня, режим бланка и режим формы.
- Для перехода между режимами таблицы, перечня и бланка нажмите клавишу *F2*.
- Назначение кнопок управления в режимах перечня, бланка и формы приведено в табл. 3.11.

Глава 4. Дополнительные средства для просмотра и проверки данных

dBASE предоставляет в ваше распоряжение значительное количество средств, предназначенных для упрощения создания, редактирования и отображения информации на экране. В главе 3 вы познакомились с некоторыми из средств для ввода и редактирования данных. В настоящей главе мы рассмотрим средства управления отображением информации на экране. Кроме того, мы обсудим, как обеспечить достоверность вводимых данных.

Изменение параметров экранного образа таблицы

В главе 3 вы научились просматривать содержимое таблицы. При просмотре таблицы вы работаете с изображением или экранным образом таблицы, который может отличаться от содержимого таблицы. Используя средства dBASE, вы можете по своему усмотрению настроить образ, не изменяя хранящееся на диске содержимое таблицы. Ниже приведен список параметров экранного образа таблицы, которые вы можете изменить по своему усмотрению:

- количество записей, отображаемых на экране;
- ширина столбцов;
- порядок следования столбцов;
- формат представления данных.

Зан/Уд.	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ТЕЛЕФОН	ИНДЕКС	СТРАНА
1	Александров	Сергей	Иванович	(383)224-3124	630090	Россия
2	Ткаченко	Марина	Михайловна	(048)056-4634	270005	Украина
3	Морозов	Юрий	Анатолиевич	(048)034-5874	270029	Украина
4	Петров	Михаил	Николаевич	(816)025-6128	173024	Россия
5	Ильин	Филипп	Петрович	(816)025-6158	173003	Россия
6	Петров	Михаил	Николаевич	(816)017-7124	173003	Россия
7	Зайцев	Людмила	Николаевна	(863)521-2453	346415	Россия
8	Загребский	Андрей	Анатолиевич	(831)256-4216	603006	Россия
9	Снесов	Филипп	Петрович	(863)245-3123	630056	Россия
10	Сухарев	Андрей	Анатолиевич	(863)573-4543	357736	Россия
11	Зайцев	Дарья	Ивановна	(257)212-2453	142490	Рос.
12	Фролов	Виктор	Михайлович	(095)576-5456	140160	Россия
13	Савосников	Сергей	Иванович	(095)453-5965	142432	Россия
14	Ухолов	Максим	Сергеевич	(812)623-4456	184230	Россия
15	Морозов	Андрей	Михайлович	(343)232-4542	520151	Россия
16	Райский	Максим	Сергеевич	(845)253-4542	443115	Россия
17	Антонов	Максим	Юрьевич	(095)532-1532	125167	Россия
18	Морозов	Андрей	Петрович	(095)131-7714	117334	Россия

Рис. 4.1. Таблица ADDRESS в окне таблицы.

В предыдущей главе вы уже столкнулись с необходимостью редактирования экранного образа, когда ширина таблицы превышала ширину экрана. Например,

ширина полей таблицы ADDRESS значительно превосходит размер распахнутого окна, в результате чего часть полей оказывается скрытой за правой границей экрана (рис. 4.1).

В некоторых случаях вам может потребоваться изменить порядок следования столбцов для того, чтобы разместить на экране большее количество полей. Например, в той же таблице ADDRESS вы хотите одновременно увидеть на экране поля "Фамилия", "Страна", "Город", "Адрес", "Телефон". В этом случае вам придется поменять столбцы местами и уменьшить их ширину. Кроме того, dBASE предоставляет вам широчайшие возможности по изменению отдельных элементов экранного образа.

В зависимости от типа элемента экранного образа для изменения его визуального представления вы можете использовать мышь, команды меню **Свойства** или соответствующие им оперативные клавиши. Для того чтобы приступить к работе с экранным образом таблицы, откройте ее в окне таблицы, воспользовавшись командой **Файл | Открыть** из меню главного окна dBASE. Далее укажите наименование таблицы, изображение которой вы хотите поместить в рабочую область.

Кроме того, для открытия таблицы вы можете воспользоваться окном навигатора или папки. Выберите категорию файла "Таблица", укажите на наименование таблицы и дважды нажмите кнопку мыши или клавишу **Enter**.

В результате в окне таблицы будет открыта таблица со стандартными значениями параметров изображения. Ваши дальнейшие действия зависят от конкретных изменений экранного образа таблицы, которые вы предполагаете осуществить.

Изменение расположения и размера окна

Изменение вертикального и горизонтального размеров окна относится к стандартным операциям оконного интерфейса и выполняется с помощью мыши. Для изменения вертикального размера окна поместите курсор мыши на нижнюю границу окна, в результате чего, его форма изменится на вертикальную двустороннюю стрелку. Переместите мышью нижнюю границу окна в позицию, соответствующую требуемому размеру, после чего отпустите кнопку мыши. Аналогичным образом вы можете изменить и горизонтальный размер окна. В этом случае вы должны переместить мышью правую границу окна.

Если вам требуется изменить сразу вертикальный и горизонтальный размеры окна, установите курсор мыши в угол окна. В этом случае курсор будет отображаться двусторонней стрелкой, направленной под углом. Вы можете перемещать курсор как в горизонтальном, так и вертикальном направлениях, изменяя соответствующие размеры окна.

После изменения размеров вам может потребоваться изменить место положения окна в рабочей области. Для этого установите курсор мыши на заголовок окна и переместите его в требуемую позицию.

Перемещение столбцов

В некоторых случаях возникает необходимость просмотреть таблицу, в которой порядок следования столбцов отличается от порядка, установленного в структуре таблицы.

Пример 4.1.

Просмотрите список абонентов, номера телефонов и наименования предприятий, на которых они работают. Требуемая информация содержится в таблице ADDRESS.

1. Откройте указанную таблицу.
2. Для размещения столбца "Предпр" следом за столбцом "Телефон" установите курсор на заголовок столбца "Предпр". В результате курсор примет форму руки.
3. Нажмите кнопку мыши и удерживая ее в нажатом состоянии переместите курсор в поле, следующее за полем "Телефон" (в рассматриваемом примере, это поле "Индекс") и отпустите кнопку мыши.

Результат перемещения поля "Предпр" показан на рис. 4.2.

ЗанУд	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ТЕЛЕФОН	ПРЕДПР
1	Александров	Сергей	Иванович	[383]224-3124	АО Интерсервис
2	Ткаченко	Марина	Михайловна	[048]056-4634	АО Компьютерсервис
3	Морозов	Юрий	Анатолеевич	[048]034-5874	ТОО Интерком
4	Петров	Михаил	Николаевич	[816]025-6128	Банк Программ
5	Ильин	Филипп	Петрович	[816]025-6156	ИЧП Элизис
6	Петров	Михаил	Николаевич	[816]017-7124	Вел-АВМ
7	Зайцева	Людмила	Николаевна	[063]521-2453	СП КомпСофт
8	Загребский	Андрей	Анатолеевич	[831]256-4216	
9	Смирнов	Филипп	Петрович	[383]245-3123	СП АВМ-Сервис
10	Сухарев	Андрей	Анатолеевич	[063]573-4543	Стенгер
11	Зайцева	Дарья	Ивановна	[257]212-2453	КомпГраф
12	Фролов	Виктор	Михайлович	[095]576-5456	
13	Савиных	Сергей	Иванович	[095]453-5985	PC Sales Service
14	Уханов	Максим	Сергеевич	[812]523-4456	Оффис-Сервис
15	Морозов	Андрей	Михайлович	[343]232-4542	АО РокКом
16	Райский	Максим	Сергеевич	[846]253-4542	АО Фиган

Рис. 4.2. Перемещение поля.

Для того чтобы облегчить позиционирование курсора, dBASE выделяет пунктирным прямоугольником перемещаемый столбец. В результате перемещения dBASE автоматически сдвинет все остальные поля таблицы. Данная процедура не оказывает влияния на расположение полей в исходной таблице.

Изменение ширины столбцов

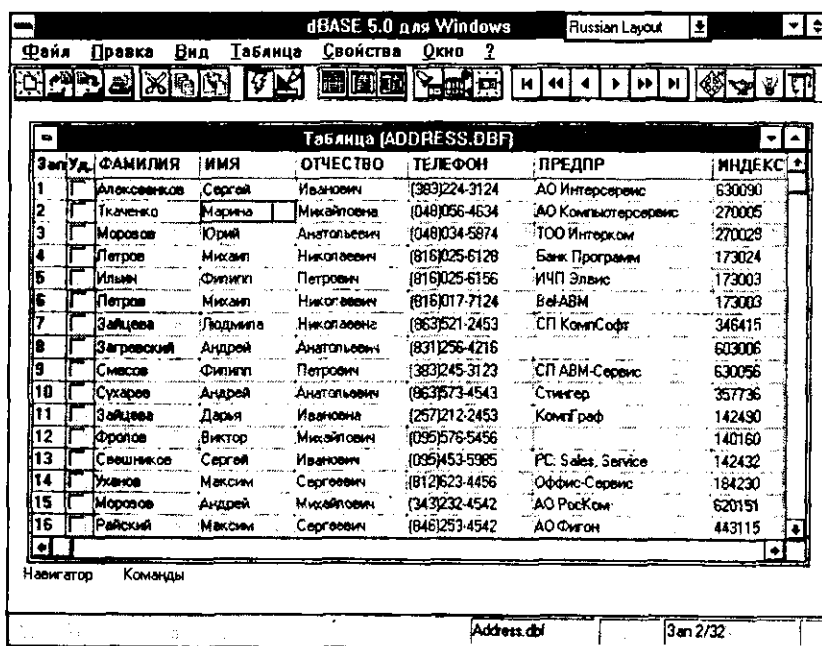
dBASE автоматически устанавливает ширину каждого столбца в соответствии с типом поля, используемого для отображения данных этого поля. Достаточно часто возникает ситуация, при которой ширины экрана оказывается недостаточно для отображения всех требуемых полей. Например, на рис. 4.2 вы видите, что содержимое перемещенного вами поля "Предпр", полностью не помещается на экране. Вместе с тем столбец, в котором содержатся номера записей, а также столбцы "Фамилия", "Имя" и "Отчество" имеют ширину, превосходящую ширину данных в соответствующих полях. Для того, чтобы полностью просмотреть данные, содержащиеся в поле "Предпр", вам необходимо уменьшить ширину одного или нескольких столбцов в изображении таблицы.

Пример 4.2.

В качестве примера уменьшите ширину поля “Имя”.

1. Установите курсор на правую линию, ограничивающую столбец, ширину которого вы предполагаете изменить. Курсор может располагаться на любой записи столбца. В этом случае форма курсора изменится на двустороннюю горизонтальную стрелку.
2. Нажмите кнопку мыши и, удерживая ее, переместите границу столбца вправо или влево в зависимости от того хотите вы увеличить или уменьшить ширину столбца. В текущем примере переместите правую границу столбца “Имя” влево.
3. Для завершения установки новой ширины столбца отпустите кнопку мыши.
4. Если вы проделаете указанные действия, все содержимое столбца “Предпр” будет отображаться на экране.

Результат изменения ширины столбцов “Фамилия”, “Имя”, “Отчество” и “Предпр” показан на рис. 4.3. Данная процедура не влияет на ширину полей в исходной таблице.



дBASE 5.0 для Windows Russian Layout

Файл Правка Вид Таблица Свойства Окно ?

Таблица (ADDRESS.DBF)

Зап. №	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ТЕЛЕФОН	ПРЕДПР	ИНДЕКС
1	Алексеев	Сергей	Иванович	(383)224-3124	АО Интерсервис	630090
2	Ткаченко	Марина	Михайловна	(048)056-4634	АО Компьютерсервис	270005
3	Морозов	Юрий	Анатолиевич	(048)034-5874	ТОО Интерком	270029
4	Петров	Михаил	Николаевич	(816)025-6128	Банк Программ	173024
5	Ильин	Филипп	Петрович	(816)025-6156	ИЧП Элизис	173003
6	Петров	Михаил	Николаевич	(816)017-7124	ВетАВМ	173003
7	Зайцева	Людмила	Николаевна	(863)521-2453	СП КоннСофт	346415
8	Загребский	Андрей	Анатолиевич	(831)256-4216		603006
9	Синцов	Филипп	Петрович	(383)245-3123	СП АВМ-Сервис	630056
10	Сухарев	Андрей	Анатолиевич	(863)573-4543	Стеттер	357736
11	Зайцева	Дарья	Ивановна	(257)212-2453	КомпГрад	142430
12	Аронов	Виктор	Михайлович	(095)576-5456		140160
13	Савенков	Сергей	Иванович	(095)453-5985	PC Sales Service	142432
14	Уланов	Максим	Сергеевич	(812)623-4456	Оффис-Сервис	184230
15	Морозов	Андрей	Михайлович	(343)232-4542	АО РосКом	620151
16	Райский	Максим	Сергеевич	(846)253-4542	АО Фигон	443115

Навигатор Команды

Address.dbf 3an 2/32

Рис. 4.3. Изменение ширины столбцов.



Если ширина поля будет установлена меньше длины данных, размещаемых в этом поле, часть данных может выйти за правую границу столбца. В этом случае вы будете видеть только начальные символы, расположенные в таком поле. В этом случае для просмотра требуемого поля выполните прокрутку изображения через окно.



dBASE устанавливает ширину столбца на основании типа поля и длины заголовка, расположенного в нем. При этом, ширина столбца не может быть короче заголовка поля, даже в тех случаях, когда данные поля короче заголовка. Поэтому, если заведомо известно, что поле будет содержать данные, длина которых не превышает нескольких позиций, не стоит давать ему слишком длинное наименование. Например, в зависимости от величины предприятия табельный номер сотрудника может занимать не более трех или четырех позиций. Поэтому для поля, содержащего табельные номера более подойдет наименование "Таб_N", чем "Таб_номер". Изменение структуры таблиц, в том числе и редактирование наименований полей, будет рассмотрено в главе 11.

Изменение свойств таблицы

Свойства

Среда dBASE...

Окно таблицы...

Основным средством для изменения свойств таблицы является меню **Свойства**.

Опция	Назначение
Среда dBASE	Изменяет свойства главного окна dBASE.
Окно таблицы	Задаёт параметры таблицы.

Для того чтобы приступить к работе со свойствами таблицы, вы можете выполнить команду **Свойства | Окно таблицы** или нажать правой кнопкой мыши любое место внутри окна таблицы и выбрать из меню объекта опцию **Свойства окна таблицы**.

В результате ваших действий на экране появится окно диалога "Свойства таблицы", которое содержит карточки "Поля", "Записи", "Окно". Выберите карточку "Окно" (рис. 4.4), которое содержит параметры настройки окна и отображения записей в окне таблицы.

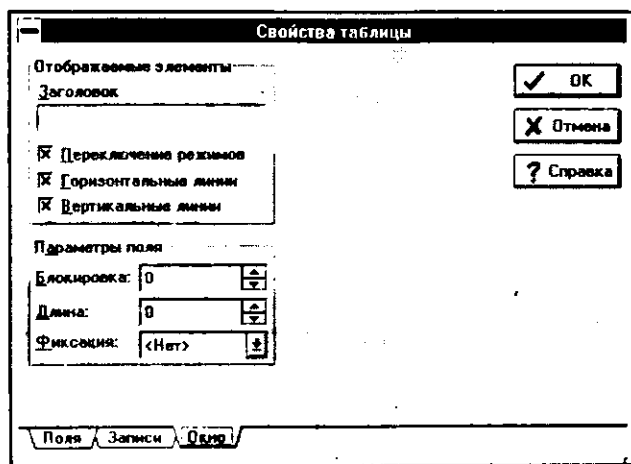


Рис. 4.4. Карточка "Окно" окна диалога "Свойства таблицы".

По умолчанию заголовок окна таблицы содержит имя таблицы. Это не очень удобно, так как имя не может содержать более 8 символов и вы не можете использо-

вать в имени таблицы кириллицу. Используя поле “Заголовок” вы можете задать собственный заголовок для окна таблицы. В качестве примера введите для окна таблицы ADDRESS заголовок “Телефонно-адресная книга”.

Флажки Горизонтальные линии и Вертикальные линии используются для отображения линий сетки в режиме таблицы.

По умолчанию вы можете переключаться между режимами таблицы, бланка и перечня. Если вы решили, что данная таблица должна отображаться только в одном из этих режимов, перейдите в требуемый режим и снимите установку флажка Переключение режимов. После этого все попытки перейти в другой режим отображения таблицы закончатся неудачей.

Только в режиме таблицы вы можете использовать параметры Блокировка, Длина и Фиксация. Параметр Блокировка блокирует указанное количество полей в левой части окна. Это означает, что при перемещении по полям таблицы левые поля, количество которых задано в поле Блокировка, не изменят свое положение в таблице. Этот режим особенно удобен при просмотре таблиц, не все поля которых помещаются в окне таблицы. Например, для таблицы ADDRESS вы можете разместить поля “Фамилия”, “Имя” и “Отчество” в левой части окна таблицы и установить количество блокируемых полей равное трем. В этом случае при вводе данных, а также при их просмотре вы всегда будете видеть на экране фамилию, имя и отчество клиента.

Если вы решили ограничить доступ к таблице только к одним полем, воспользуйтесь полем Фиксация. Для определения поля для фиксации нажмите кнопку раскрывающегося списка и выберите из этого списка требуемое поле. Чтобы снять фиксацию, выберите значение <Нет>.

Дополнительно для отображения текстовых данных вы можете установить режим вывода на экран всех текстовых данных в полях одинаковой длины. Для этого перейдите в поле Длина и введите подходящее вам значение.

На рис. 4.5 представлено окно таблицы ADDRESS, для которого задан заголовок окна и установлен режим блокировки трех левых полей.

ЗанУд	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ГОРОД	АДРЕС
1	Алексеев	Сергей	Иванович	Новосибирск	ул. Ильича, д. 34
2	Ткаченко	Марина	Михайловна	Одесса	ул. Индустриальная, д. 23
3	Морозов	Юрий	Анатолиевич	Одесса	ул. Манежная, д. 42
4	Петров	Михаил	Николаевич	Новгород	ул. Свободы, д. 18
5	Яковлев	Филипп	Петрович	Новгород	ул. С.Петербургская, д. 4
6	Петров	Михаил	Николаевич	Новгород	ул. С.Петербургская, д. 3
7	Зайцева	Людмила	Николаевна	Новочеркасск	ул. Энергетиков 12/6
8	Загребская	Андрей	Анатолиевич	Н.Новгород	ул. Горького, д. 160
9	Смирнов	Филипп	Петрович	Новосибирск	ул. Ветлужская, д. 22 а
10	Сухарев	Андрей	Анатолиевич	Кисловодск	ул. Севастопольская, д. 6
11	Зайцева	Дарья	Ивановна	Электросталь	ул. Заповедная, д. 1

Рис. 4.5. Задание заголовка и установка режима блокировки левых полей.

Выбор полей, отображаемых в окне таблицы

Карточка "Поля" окна диалога "Свойства таблицы" (рис. 4.6) определяет состав выводимых в окне таблицы полей, а также свойства отдельных полей.

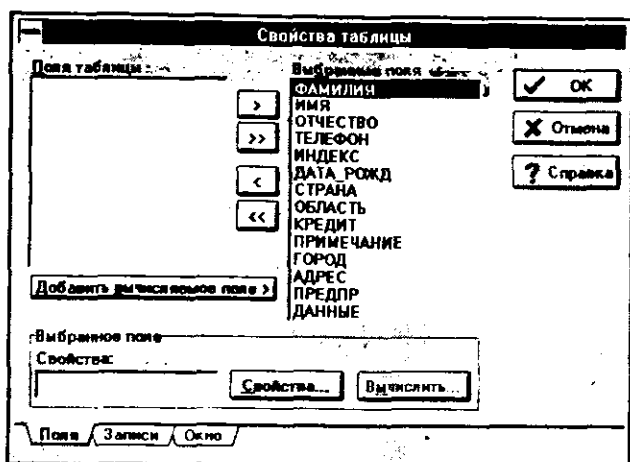


Рис. 4.6. Карточка "Поля" окна диалога "Свойства таблицы".

По умолчанию в окне таблицы отображаются все поля таблицы. В некоторых случаях вас может интересовать только часть полей таблицы. В этом случае излишняя информация в окне таблицы затруднит просмотр данных. Вы можете настроить список отображаемых в окне таблицы полей. Список **Выбранные поля** содержит список полей, которые отображаются в окне таблицы, а список **Поля таблицы** — список не отображаемых полей таблицы. Для перемещения полей между этими списками используются следующие кнопки:

Кнопка	Назначение
Добавить (>)	Добавляет выделенное поле в список выбранных полей.
Удалить (<)	Удаляет выделенное поле из списка выбранных полей.
Добавить все (>>)	Добавляет все поля списка Поля таблицы в список Выбранные поля .
Удалить все (<<)	Удаляет все поля из списка Выбранные поля и помещает их в список Поля таблицы .

Вы можете не только размещать в окне таблицы имеющиеся поля таблицы, но и добавить в него новые вычисляемые поля. Например, при отображении таблицы ORDSALE вы можете добавить поле, которое содержит стоимость партии заказанных товаров. Для добавления вычисляемого поля используется кнопка **Добавить вычисляемое поле>**, которая открывает окно диалога "Вычисляемое поле" (рис. 4.7).

Окно диалога содержит поля **Имя** и **Выражение**. Имя поля может содержать до 32 символов и отображается в заголовке поля. В поле **Выражение** вы можете ввести любое допустимое в dBASE выражение. Обычно для создания выражения используется построитель выражений, который вызывается при нажатии на кнопку **Инструмент**.

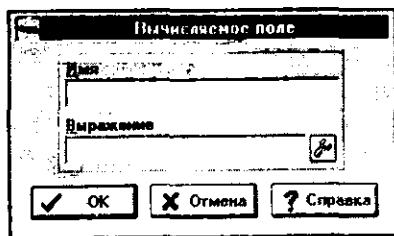


Рис. 4.7. Окно диалога “Вычисляемое поле”.

Рассмотрим более подробно создание выражения с помощью построителя выражения.

Работа с построителем выражений

Построитель выражений предназначен для создания выражений, допустимых в системе dBASE, и вставки их в текущее поле. Однако, прежде чем приступить к созданию выражений для формирования вычисляемых полей, остановимся на общих положениях о выражениях dBASE.

Выражение в dBASE состоит из одного или нескольких элементов dBASE, результатом вычисления которых является одно значение. Ниже приведены примеры выражений:

```
TRIM(ФАМИЛИЯ) + ", " + ИМЯ
ЦЕНА*ЗАКАЗАНО
DATE()+100
```

В выражениях могут использоваться следующие элементы:

- Имена полей
- Константы
- Функции
- Операторы
- Переменные
- Элементы массива

Результат вычисления выражения может принадлежать к любому из следующих типов данных: текстовому, дате, логическому или числовому. Вы можете строить выражение на основе данных разного типа, но при применении к ним арифметических операторов, предварительно должны привести элементы к одному типу данных с помощью функций преобразования данных. При работе с операторами сравнения слева и справа от оператора должны стоять данные одного типа. Оператор сравнения всегда преобразует выражение в логическое значение.



Выражение может состоять и из одного элемента.



Для вызова построителя выражений используется кнопка **Инструмент**, при нажатии на которую открывается окно диалога построителя выражений (рис. 4.8).

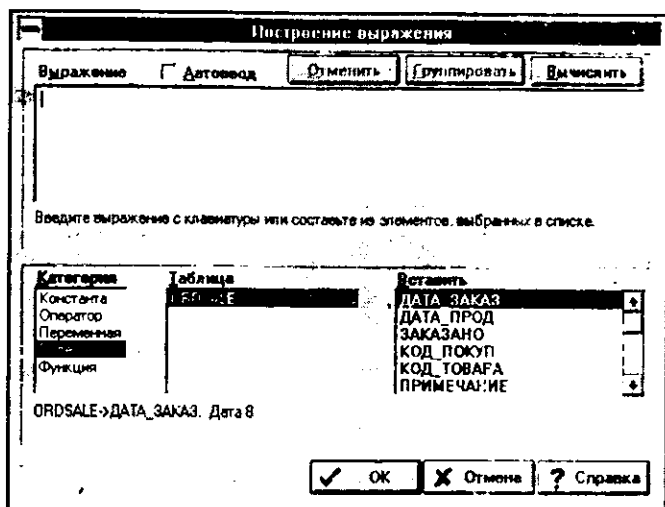


Рис. 4.8. Окно диалога “Построение выражения”.

Вы можете непосредственно вводить выражение в поле **Выражение**. Но этот режим используется достаточно редко. Наиболее часто строят выражение добавляя в него готовые элементы из списка допустимых значений. При выборе используются три уровня группировки данных, которые позволяют вам ускорить поиск требуемого элемента.

Прежде всего вы должны выбрать категорию добавляемого в выражение элемента. Для этого перейдите в список **Категория** и выберите один из допустимых разделов: “Константа”, “Оператор”, “Переменная”, “Поле”, “Функция”. Для каждой категории существует второй уровень выбора, который для категории “Таблица” содержит список открытых в настоящий момент таблиц, а для остальных категорий тип констант, операторов, переменных и функций. Для выбранной таблицы список **Вставить** содержит перечень полей выбранной таблицы. Для функций этот список содержит перечень всех функций выбранного типа, а для операторов - допустимые операторы данного типа. Для констант список **Вставить** не может содержать все возможные константы. Поэтому он содержит примеры некоторых наиболее часто используемых констант, которые вы можете использовать в качестве шаблона при самостоятельном вводе констант.

Пример 4.3. Создание вычисляемого поля

В качестве примера обратимся к таблице **ORDSALE**. Данная таблица содержит информацию о количестве товаров каждого типа, заказанных в каждой партии (поле “Заказано”), и цене изделия (поле “Цена”). Данные об общей стоимости каждого заказа в исходной таблице отсутствуют. Если вам потребуются эти данные, вы можете получить их путем перемножения значений полей “Заказано” и “Цена” по каждой партии.

В настоящем примере вам предстоит добавить в таблицу **ORDSALE** вычисляемое поле, которое содержит стоимость партии заказанных товаров.

1. Откройте таблицу ORDSALE.
2. Откройте окно диалога “Свойства таблицы” и в карточке “Поле” нажмите кнопку **Добавить** **вычисляемое поле**. На экране появится окно диалога “Вычисляемое поле”.
3. В окне диалога “Вычисляемое поле” введите имя создаваемого поля “Стоимость”.
4. Для создания выражения с использованием построителя выражений нажмите кнопку **Инструмент**. На экране появится окно диалога “Построение выражения”.
5. Выберите категорию “Поле”, а затем поле “Заказано” таблицы ORDSALE. Выбранное вами поле появится в поле ввода выражения.
6. Выберите категорию “Оператор”, тип оператора числовой, а затем из списка **Вставить** — оператор умножения. Выбранный оператор будет добавлен в поле ввода выражения. Конечно, вы можете ввести оператор умножения и непосредственно в поле ввода выражения.
7. В заключение выберите категорию “Поле”, а затем поле “Цена” таблицы ORDSALE. Выбранное вами поле добавится в поле ввода выражения (рис. 4.9).

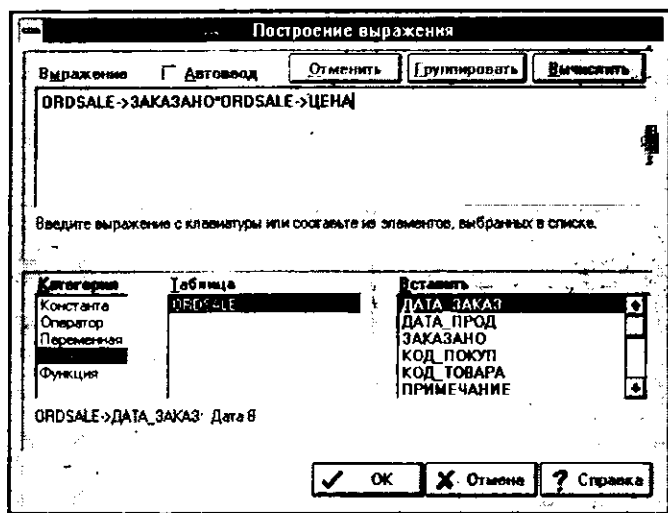


Рис. 4.9. Создание выражения.

8. В поле **Выражение** окна диалога “Вычисляемое поле” появится созданное вами выражение.
9. Для завершения создания вычисляемого поля нажмите **ОК** и вы увидите, что таблица в окне таблицы содержит теперь новое поле, отображающее стоимость заказа партии товаров (рис. 4.10).

В построителе выражений существует возможность полного контроля над созданием выражения. Для этого вы должны перейти в режим автоввода, установив флажок **Автовод** в окне построителя выражения. При установке данного флажка вы не

можете с помощью клавиатуры вводить данные в поле **Выражение**. Шаблоны функций, элементы выражения обозначаются разными цветами, облегчая создание выражения. Справочная панель под полем **Выражение** содержит указание по вводу очередного элемента выражения. Если для изменения порядка выполнения операций вам необходимо заключить некоторые элементы выражения в скобки, выделите эти элементы и нажмите кнопку **Группировать**. В результате выделенные элементы будут заключены в скобки. В процессе ввода выражения вы можете переключаться из режима автоввода в обычный режим создания выражения.

Зан	Уд	КОД_ТОВАРА	КОД_ПОКУП	ДАТА_ЗАКАЗ	ЦЕНА	ЗАКАЗАНО	СТОИМОСТЬ
1			2	1 02.04.1995	610.00	5	3050.00
2		1		2 02.01.1995	2600.00	2	5200.00
3		2		2 03.01.1995	625.00	15	9375.00
4		2		2 01.02.1995	630.00	7	4410.00
5		2		2 05.02.1995	645.00	10	6450.00
6		2		2 13.02.1995	605.00	500	302500.00
7		3		2 01.01.1995	1025.00	10	10250.00
8		5		2 12.04.1995	270.00	10	2700.00
9		8		2 05.03.1995	27.00	100	2700.00
10		2		6 05.01.1995	620.00	20	12400.00
11		4		7 05.03.1995	210.00	20	4200.00
12		8		7 05.01.1995	27.00	50	1350.00
13		1		10 25.02.1995	2670.00	3	8010.00
14		4		10 26.02.1995	220.00	20	4400.00
15		5		10 10.02.1995	270.00	20	5400.00
16		1		11 10.02.1995	2600.00	1	2600.00
17		2		11 10.01.1995	635.00	1	635.00

Рис. 4.10. Таблица ORDSALE, содержащая вычисляемое поле.

Форматирование данных

До тех пор, пока вы не измените формат отображения чисел и дат, dBASE представляет их в стандартном формате, установленном с помощью Панели Управления Windows. Вы можете по своему усмотрению изменить форматы отображения полей текстового и числового типа, а также полей дат. При этом информация в таблице, хранящаяся на диске, остается без изменений.

Для изменения формата отображения поля выполните следующие действия:

1. В окне диалога "Свойства таблицы" откройте карточку "Поле".
2. Выберите требуемое поле и нажмите кнопку **Свойства** данного поля.
3. В окне диалога "Свойства поля" перейдите на поле шаблон и нажмите на кнопку **Инструмент** и на экране появится окно диалога "Выбор шаблона" (рис. 4.11).
4. Выберите тип данных и для данного типа укажите требуемый вам формат данных.

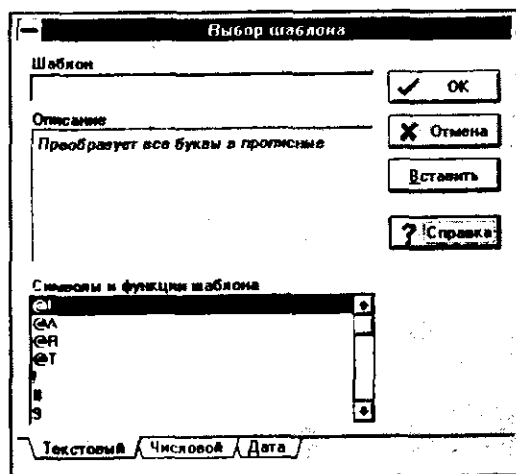


Рис. 4.11. Окно диалога "Выбор шаблона".

Давайте более подробно остановимся на форматировании текстовых и числовых полей, а также полей дат.

Форматирование текстовых полей

Для выбора символов форматирования текстовых данных укажите на карточку "Текстовый". В списке Символы и функции шаблона появятся все допустимые символы шаблона форматирования текстовых данных:

Символ	Описание
@!	Преобразует все буквы в прописные.
@A	Допускает только ввод букв.
@R	Не позволяет сохранять символы, которые не являются кодами шаблона.
@T	Удаляет при отображении начальные и конечные пробелы.
!	Преобразует вводимую букву в прописную.
#	Допускает ввод только цифры, пробела, знака "+" или "-".
9	Допускает ввод только цифры.
A	Допускает ввод только одной буквы.
L	Допускает ввод только букв T, t, F, f, Д, д или Н, н.
Y	Допускает ввод только Д, д, Н и н. Отображает .Т., Т, t и д как Д; .F., F, f и н как Н.
X	Оставляет символ неизменным. Используйте его вместе с другими шаблонами.

Пример 4.4. Выбор текстового формата

В данном примере вам предстоит создать формат для отображения полей "Фамилия", "Имя" и "Отчество" таблицы ADDRESS, которые могут содержать только буквы, а первая из них должна быть заглавной. Кроме этого создадим формат для отображения номеров телефонов в привычном для нас виде, например (095) 101-45-74.

1. Откройте окно таблицы ADDRESS.
2. Выделите поле “Фамилия” в окне диалога “Свойства таблицы”, выберите требуемое поле и нажмите кнопку Свойства для данного поля. Далее в окне диалога “Свойства поля” перейдите на поле Шаблон, нажмите на кнопку Инструмент и вызовите карточку “Текстовый” окна диалога “Выбор шаблона”.
3. Вначале создадим шаблон для ввода данных в поле “Фамилия”.

Наиболее просто разработать формат, используя список Символы и функции шаблона. При выборе любого из допустимых шаблонов в поле Описание появляется краткое описание выбранного шаблона. Для преобразования отображения первой буквы в прописную выберите символ “!”, а для разрешения ввода в поле только букв добавьте в шаблон символы “А”. Количество этих символов должно быть на единицу меньше ширины поля. В результате строка шаблона будет выглядеть следующим образом: “!AAAAAAAAAAAAAAAA”. Выберите кнопку ОК сначала в окне диалога “Выбор шаблона”, а затем в окне диалога “Свойства поля”.

4. Аналогично создайте шаблоны для полей “Имя” и “Отчество”.

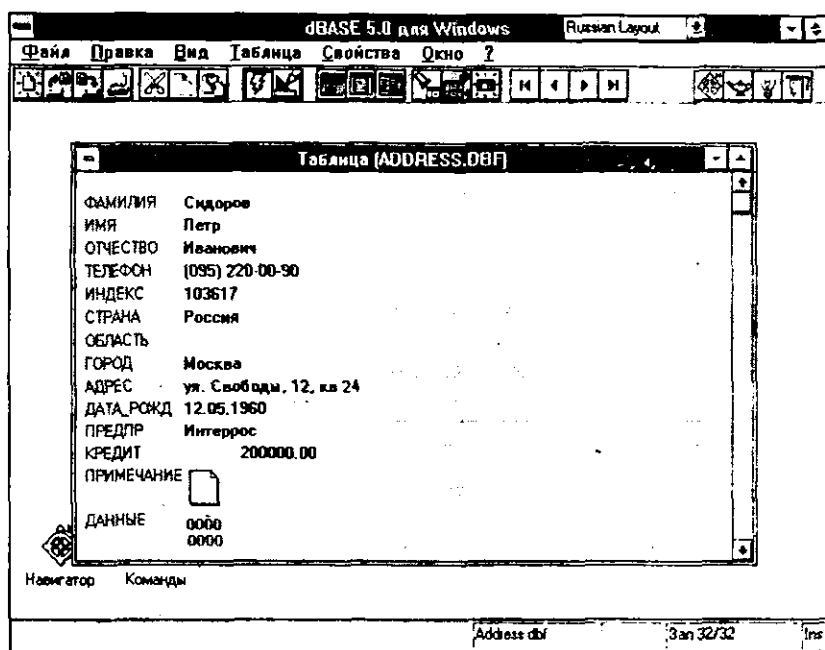


Рис. 4.12. Форматирование текстовых данных.

5. Теперь создадим шаблон для ввода данных в поле “Телефон”.

Данные этого поля хранятся в символьном виде и содержат цифры номеров телефонов. Вы могли бы включить в номера телефонов круглые скобки и дефис для приведения номеров к принятому виду отображения. Но в этом случае эти символы будут храниться в поле и потребуют дополнительный объем жесткого диска. Более рациональным является хранение в поле только цифр и использование шаблона поля для отображения данных этого поля на экране. Для этих целей воспользуйтесь символом шаблона “@R”, который отображает на экране указанные в шаблоне символы, не

являющимися символами шаблона, но не записывает их в таблицу. Вслед за этим символом шаблона введем формат отображения данных на экране "(999) 999-99-99". Сохраните созданный вами шаблон для ввода значений в поле "Телефон" и вернитесь в окно диалога "Свойства таблицы".

6. Нажмите кнопку ОК и вы увидите, что данные на экране отображаются в указанном вами формате (рис. 4.12). При вводе новых значений в поля "Фамилия", "Имя" и "Отчество", первая буква независимо от ее ввода по верхнему или нижнему регистру, будет отображаться в виде прописной буквы. Поле "Телефон" теперь отображается в более читабельном виде.

Форматирование числовых полей

Форматирование числовых полей осуществляется аналогично форматированию текстовых полей. В данном случае для выбора символов форматирования выберите карточку "Числовой". В списке Символы и функция шаблона появятся все допустимые символы шаблона форматирования числовых данных:

Символ	Описание
@\$	Отображает символ денежной единицы до или после значения.
@Z	При отображении заменяет начальные нули пробелами.
@^	Отображает значение в экспоненциальном формате. Использование экспоненциального формата очень удобно для отображения очень больших или малых чисел, а также полей, в которых диапазон изменения значений велик.
\$	Отображает символ денежной единицы.
*	При отображении заполняет пустые старшие разряды числа символом *.
,	Вставляет разделитель групп разрядов числа.
.	Вставляет разделитель целой и дробной части числа.
9	Допускает только ввод цифры, знака "+" или "-" для числовых значений.

Денежный символ и его положение (слева или справа) задаются в области "Символ денежной единицы" в полях Позиция и Символ карточки "Страна" окна диалога "Свойства среды dBASE".

Пример 4.5. Выбор числового формата

В данном примере вам предстоит создать формат для отображения значений денежных полей, для которых цены выражены в тысячах рублей.

1. Откройте окно таблицы ADDRESS.
2. Выделите поле "Кредит" в окне диалога "Свойства таблицы", выберите требуемое поле и нажмите кнопку Свойства данного поля. Далее в окне диалога "Свойства поля" перейдите на поле шаблон, нажмите на кнопку Инструмент и вызовите карточку "Числовой" окна диалога "Выбор шаблона".
3. Создадим шаблон для ввода, который содержит символ разделения разрядов числа ",", символ разделения целой части от дробной и единицу измерения значений 'тыс. руб.'. В завершённом виде шаблон должен выглядеть следующим образом: '999,999.99 тыс. руб.'.

4. Увеличьте ширину поля до 20 позиций для размещения в нем значений поля вместе с символами шаблона. Для сохранения шаблона выберите кнопку ОК сначала в окне диалога “Выбор шаблона”, а затем в окне диалога “Свойства поля”.
5. Вернитесь в окно диалога “Свойства таблицы”, нажмите кнопку ОК и вы увидите, что данные на экране отображаются в указанном вами формате (рис. 4.13).

Скриншот окна dBASE 5.0 для Windows. В центре экрана отображается таблица [ADDRESS.DBF] с шестью столбцами: Зан.Уд., ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, КРЕДИТ, ТЕЛЕФОН. Данные в столбцах КРЕДИТ и ТЕЛЕФОН отформатированы с использованием шаблонов, включающих скобки и запятые. Например, в столбце КРЕДИТ значения имеют вид 200,000.00 тыс. руб. (863) 521-24-53. В столбце ТЕЛЕФОН значения имеют вид (863) 521-24-53. В нижней части окна видны панели "Навигатор" и "Команды". В статусной строке в правом нижнем углу отображены: Address.dbf, Монофон, Зан 8/30, lms.

Зан.Уд.	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	КРЕДИТ	ТЕЛЕФОН
7	Защова	Людмила	Николаевна	200,000.00 тыс. руб.	(863) 521-24-53
8	Защовский	Андрей	Анатолевич	80,000.00 тыс. руб.	(831) 256-42-16
9	Смеев	Филипп	Петрович	360,000.00 тыс. руб.	(363) 245-31-23
10	Сухарев	Андрей	Анатолевич	400,000.00 тыс. руб.	(863) 573-45-43
11	Защова	Дарья	Ивановна	200,000.00 тыс. руб.	(257) 212-24-53
12	Фролов	Виктор	Михайлович	50,000.00 тыс. руб.	(095) 576-54-56
13	Савельчиков	Сергей	Ильич	200,000.00 тыс. руб.	(095) 453-53-65
14	Уханов	Максим	Сергеевич	100,000.00 тыс. руб.	(812) 623-44-56
15	Морозов	Андрей	Михайлович	200,000.00 тыс. руб.	(343) 232-45-42
16	Рейский	Максим	Сергеевич	800,000.00 тыс. руб.	(846) 253-45-42
17	Антонов	Максим	Юрьевич	25,000.00 тыс. руб.	(095) 532-15-32
18	Морозов	Андрей	Петрович	300,000.00 тыс. руб.	(095) 131-77-14
19	Александров	Александр	Иванович	250,000.00 тыс. руб.	(013) 225-65-41
20	Каванский	Сергей	Иванович	140,000.00 тыс. руб.	(044) 453-67-64
21	Гершман	Анатолий	Васильевич	300,000.00 тыс. руб.	(846) 223-48-72

Рис. 4.13. Форматирование числовых данных.

Форматирование полей дат

Мы уже познакомились с определением форматов дат в карточке “Страна” окна диалога “Свойства среды dBASE”. Установленный в этой карточке формат действует при отображении полей дат любых таблиц. Кроме того, для полей дат отдельных таблиц вы можете установить режим отображения даты не в текущем формате даты, а в европейском формате — ДД.ММ.ГГ. Для выбора символов форматирования дат выберите карточку “Дата” и из списка Символы и функции шаблона — допустимые символы шаблона форматирования:

Символ	Описание
@D	Отображает значения в текущем формате даты.
@E	Отображает дату в европейском формате (ДД.ММ.ГГ).

Изменение заголовка поля

По умолчанию в заголовке полей отображаются наименования полей таблицы. При необходимости вы можете изменить заголовки полей таблицы. Например, заголовок “Дата заказа” будет выглядеть более привлекательным чем имя поля “Дата_зак”. Для изменения заголовка поля откройте окно диалога “Свойства таб-

лицы", выберите требуемое поле и нажмите кнопку **Свойства** данного поля. Далее в окне диалога "Свойства поля" перейдите на поле "Заголовок" и введите новый заголовок поля.

Запрет изменения значения поля

При выполнении некоторых операций с таблицами вы редактируете только часть полей. При этом вам необходимо просматривать остальные поля, но запрещается их редактировать. Например, при изменении телефона или адреса клиента вы должны видеть его фамилию, имя и отчество, но ни в коем случае не должны их изменять. В таких случаях используйте флажок **Только чтение** окна диалога "Свойства поля". Поля с установленным флажком будут отображаться в окне таблицы, но они будут недоступны для редактирования данных.

Флажок **Только чтение** устанавливает полный запрет редактирования поля. В некоторых случаях вам необходимо установить запрет редактирования поля в зависимости от значений других полей таблицы. В этих случаях используйте поле **Когда** окна диалога "Свойства поля". Это поле может содержать любое выражение, результат которого должен быть логическим.

Пример 4.5. Определение условия редактирования поля

В данном примере зададим условие редактирования поля "Продано" таблицы ORDSALE. Будем предполагать, что пользователь не должен вводить значение в это поле, если не определено значение поля "Заказано".

1. Откройте окно таблицы ORDSALE.
2. Выделите поле "Продано" в окне диалога "Свойства таблицы" и нажмите кнопку **Свойства** данного поля. Далее в окне диалога "Свойства поля" перейдите на поле **Когда** и нажмите на кнопку **Инструмент**.
3. В окне диалога построителя выражения создайте выражение `ORDSALE->ЗАКАЗАНО>0` (рис. 4.14) и нажмите кнопку **ОК**. Созданное вами выражение будет помещено в поле **Когда**.
5. Вернитесь в окно диалога "Свойства таблицы" и нажмите кнопку **ОК**. Откроется окно таблицы ORDSALE.
6. Перейдите в поле "Продано" записи, которая содержит пустое значение заказанного товара. При попытке редактирования поля "Продано" будет выдаваться звуковое предупреждение и вы не сможете изменить содержимое поля.

Проверка достоверности данных

Вероятно, вы уже успели отметить, что dBASE автоматически осуществляет проверку достоверности данных, вводимых в поля определенного типа. Например, вы не сможете ввести букву в числовое поле или ввести несуществующую дату.

Проверка достоверности данных предоставляет вам дополнительные средства для контроля данных, вводимых в поля определенного типа. С помощью этого средства вы можете установить правила и значения, которые dBASE будет использовать при проверке достоверности данных.

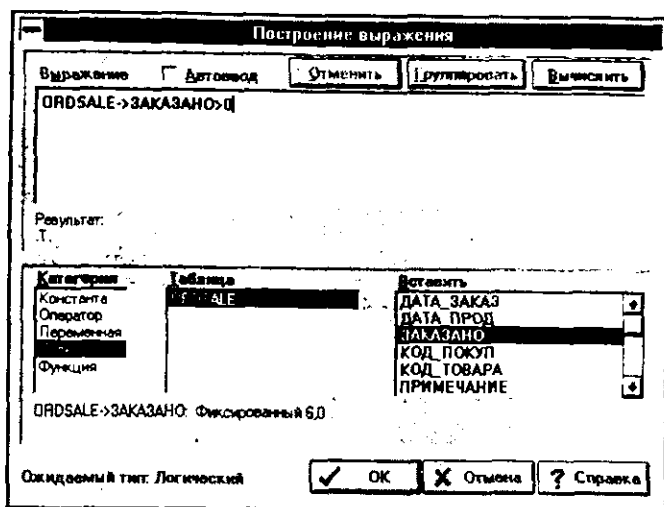


Рис. 4.14. Условие редактирования поля.

В предыдущих разделах мы использовали окно диалога “Свойства поля” для создания шаблона и заголовка поля. В этом же окне диалога вы можете определить проверку достоверности ввода данных и сообщение, выводимое на экран при возникновении ошибки. Для этих целей используются поля **Выражение**, **Сообщение об ошибке** и **Проверка** из области “Условие”.

Условие, определяющее достоверность вводимых данных, задается в поле **Выражение**. Результат выражения должен быть логическим, а непосредственная проверка условия осуществляется при выходе из этого поля. Для ввода выражения вы можете воспользоваться построителем выражений.

В поле **Сообщение об ошибке** вы можете ввести сообщение, которое выводится в окне диалога, если условие проверки достоверности вводимых данных не выполняется. Это сообщение может быть просто текстом, или же содержать выражение, результат которого является символьным. Флажок **Проверка** запрещает выход из поля, пока не введены допустимые данные.

Пример 4.6. Проверка достоверности данных

В данном примере зададим проверку достоверности ввода данных и сообщение об ошибке для поля “Продано” таблицы ORDSALE. Будем предполагать, что количество проданного товара не может превысить заказанное количество.

1. Откройте окно таблицы ORDSALE.
2. Выделите поле “Продано” в окне диалога “Свойства таблицы” и нажмите кнопку **Свойства данного поля**. Далее в окне диалога “Свойства поля” перейдите на поле **Выражение** и нажмите на кнопку **Инструмент**.
3. В окне диалога построителя выражения создайте выражение `ORDSALE->ПРОДАНО<=ORDSALE->ЗАКАЗАНО` и нажмите кнопку **ОК**. Созданное вами выражение будет помещено в поле **Выражение**.
4. В окне диалога “Свойства поля” перейдите на поле **Сообщение об ошибке** и введите текст сообщения об ошибке: **Вы пытаетесь отпустить товар больше заказанного.**

- Вернитесь в окно диалога “Свойства таблицы” и нажмите кнопку ОК. Откроется окно таблицы ORDSALE.
- Перейдите в поле “Продано” и введите количество проданного товара больше заказанного. При попытке покинуть поле открывается окно диалога “Предупреждение”, которое содержит текст заданного вами сообщения об ошибке (рис. 4.15).

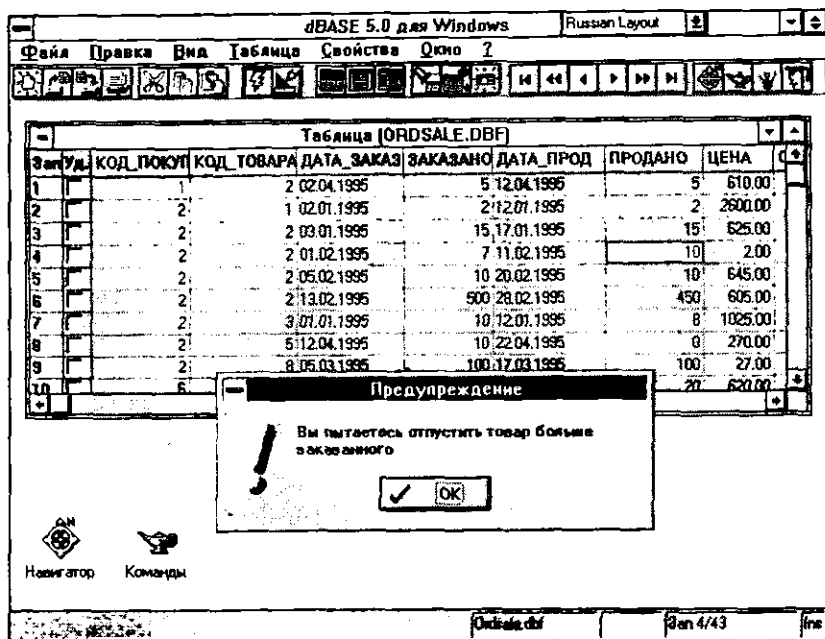


Рис. 4.15. Проверка достоверности ввода данных.

Установка минимального и максимального значений

Дополнительно к проверке достоверности ввода данных вы можете определить диапазон допустимых значений поля. Поле **Нижняя граница** предназначено для установки минимального допустимого значения данных в выбранном поле. Установив минимальное значение для поля, вы не сможете ввести в это поле величину, которая имеет значение меньше установленного вами. Аналогичный контроль данных осуществляет и поле **Верхняя граница**. Вы можете также задать двусторонний диапазон допустимых значений, указав для поля минимальное и максимальное значения. Для ввода как минимального, так и максимального значения вы можете использовать построитель выражений. Флажок **Проверка области “Диапазон”** используется для запрета выхода из поля, если в него не введены допустимые данные.

Пример 4.7. Установка диапазона допустимых значений

В данном примере установим диапазон допустимых значений для поля “Дата_прод” таблицы ORDSALE. Очевидно, что дата продажи не может предшествовать дате заказа. Кроме того, предположим, что покупатель должен получить заказ в течение пяти дней.

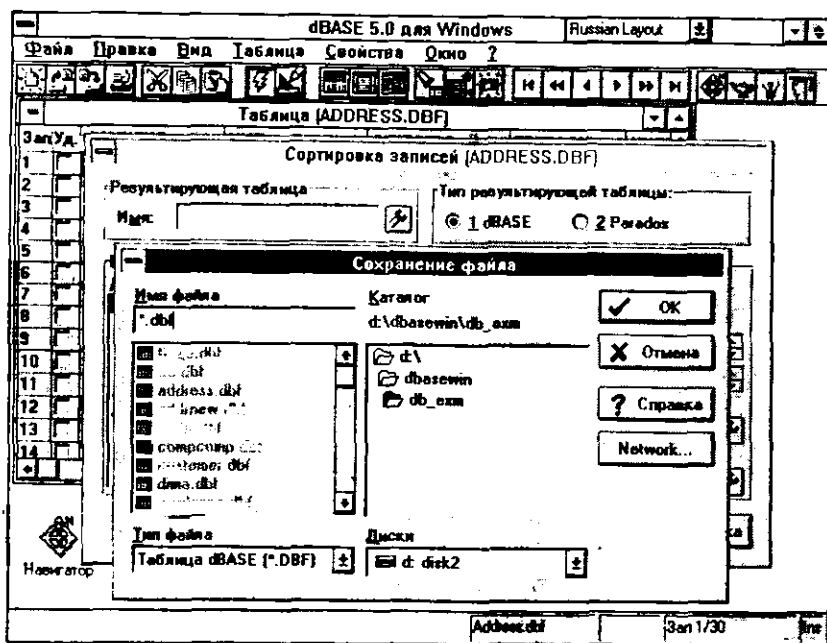


Рис. 5.3. Окно диалога “Сохранение файла”.

1. Для того чтобы включить поле в список **Ключевые поля**, выделите требуемое поле в списке доступных полей таблицы. Далее нажмите кнопку **Добавить (>)** и поле будет перенесено в список **Ключевые поля**. Если вы перенесли поле по ошибке, используйте кнопку **Удалить (<)**. При этом выбранное поле удаляется из списка **Ключевые поля** и помещается в конец списка **Доступные поля**. Для удаления всех полей из списка **Ключевые поля** используйте кнопку **Удалить все (<<)**.



Сортировка записей dBASE не может осуществляться по логическим, двоичным и Мемо-полям. Поля этих типов в списке допустимых полей таблицы не отображаются.

2. Завершив формирование списка полей, определите порядок сортировки. При сортировке dBASE сначала сортирует записи в соответствии со значениями в первом поле в списке сортируемых полей, затем в соответствии со значениями во втором поле и т.д. Последовательность полей в списке выбранных для сортировки определяется последовательностью их переноса из списка доступных полей. Если вам необходимо изменить установленную последовательность полей в выбранном списке, выделите поле, которое требуется переместить, и нажатием кнопки **Удалить (<)** переместите его в список доступных полей. Затем выделите в списке **Ключевые поля** поле, перед которым вы собираетесь разместить перемещаемое поле. Для завершения операции переноса перейдите в список **Доступные поля**, выделите перемещаемое поле и нажмите кнопку **Добавить (>)**.
3. На следующем этапе вам необходимо определить порядок сортировки значений в выбранных полях. По умолчанию поля сортируются по возрастанию значений в каждом из полей. Об этом свидетельствует индикатор направления сортировки, расположенный перед каждым полем в списке **Ключевые поля** (рис. 5.4).

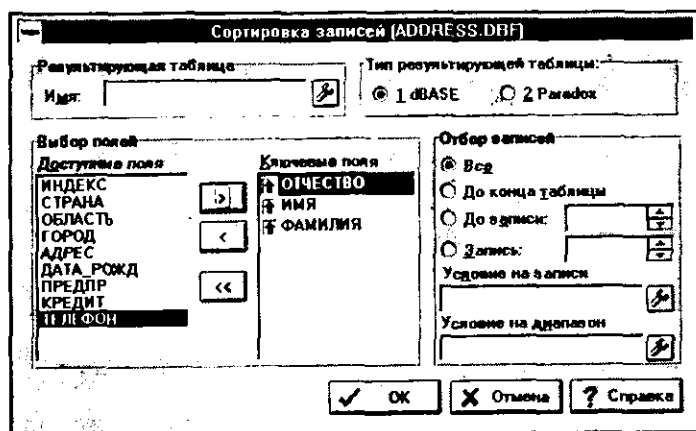


Рис. 5.4. Сортировка по возрастанию значений.

Данные в числовых полях упорядочиваются по возрастанию значений от минимального к максимальному.

Текстовые поля упорядочиваются в алфавитном порядке, начиная с буквы А и заканчивая Я. Если текстовые данные содержат буквы, цифры или другие символы, dBASE упорядочивает их по возрастанию значений OEM-кодов величин (от минимального значения к максимальному).

Поля дат упорядочиваются от наиболее ранней даты к поздней.

☒	По возрастанию
☒	Без учета регистра
☒	По убыванию
☒	Без учета регистра

Для того чтобы изменить направление сортировки, нажмите кнопку индикатора сортировки. В результате раскроется список допустимых вариантов сортировки.

Вы можете выбрать сортировку записей по любому из четырех возможных вариантов упорядочивания:

Порядок сортировки	Описание
По возрастанию	Сортирует значения в возрастающем порядке.
По возрастанию без учета регистра	При сортировке значений по возрастанию не учитывается регистр букв.
По убыванию	Сортирует значения в убывающем порядке.
По убыванию без учета регистра	При сортировке значений по убыванию не учитывается регистр букв.

Независимо от количества полей, помещенных в список **Ключевые поля**, в результирующую таблицу будут включены все поля исходной таблицы в том порядке, в каком они располагаются в исходной таблице.

Использование индексов для изменения упорядочивания

Если вам требуется просмотреть упорядоченную таблицу, для этого вовсе не обязательно сортировать исходную таблицу; вам достаточно упорядочить записи в экран-

ном образе таблицы. Индексы позволяют вам упорядочить образ таблицы по различным критериям. Для одной и той же таблицы dBASE позволяет создать индексы по каждому из полей исходной таблицы. Кроме того, вы можете создать составные индексы сразу по нескольким полям исходной таблицы.

При работе с индексированными таблицами вы можете упорядочить их в порядке, определяемом выбранным индексом. Например, на рис. 5.5 показаны два экранных образа индексированной таблицы CUSTOMER, упорядоченной по кодам и по фамилиям клиентов. Отметим, что в данной таблице ключевым полем является "Код покупателя", а поле "Фамилия" — дополнительным индексом.

Кроме определения альтернативных способов упорядочения записей индексы имеют еще два важных назначения:

- связывание таблиц в многотабличных базах данных;
- ускорение выполнения запросов и поиска значений.

Список клиентов, упорядоченных по кодам						
Занят	КОД ПОКУП	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ТЕЛЕФОН	ИНДЕКС
1	1	Алексеев	Сергей	Иванович	(383)224-3124	630090
2	2	Ткаченко	Марина	Михайловна	(048)056-4634	270005
3	3	Морозов	Юрий	Анатолийевич	(048)034-5874	270029
4	4	Петров	Михаил	Николаевич	(816)025-6128	173024
5	5	Ильин	Филипп	Петрович	(816)025-6156	173003
6	6	Петров	Михаил	Николаевич	(816)017-7124	173003

Список клиентов, упорядоченных по фамилиям						
Занят	КОД ПОКУП	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ТЕЛЕФОН	ИНДЕКС
15	15	Александров	Александр	Иванович	(013)225-6541	228010
1	1	Алексеев	Сергей	Иванович	(383)224-3124	630090
26	26	Амелин	Сергей	Иванович	(017)245-7982	220131
24	24	Ананиев	Юрий	Степанович	(032)279-7659	290052
17	17	Антонов	Максим	Юрьевич	(095)532-1532	125167
25	25	Арзамасцев	Филипп	Петрович	(846)223-4564	443080
21	21	Гершман	Антоний	Васильевич	(846)223-4872	443115

Рис. 5.5. Отображение различных экранных образов таблицы CUSTOMER.

Пример 5.1. Создание индекса для упорядочивания

В данном примере вам предстоит создать составной индекс по полям "Фамилия", "Имя" и "Отчество" таблицы CUSTOMER.

1. Для определения дополнительного индекса для уже существующей таблицы, выполните команду **Таблица | Операции | Индексы**, в результате чего будет открыто окно диалога "Индексы" (рис. 5.6).
2. Выберите **Создать**. dBASE откроет окно диалога "Создание индекса" (рис. 5.7).
3. Напечатайте имя индекса ФИО.
4. Задайте выражение для индекса **ФАМИЛИЯ+ИМЯ+ОТЧЕСТВО**.

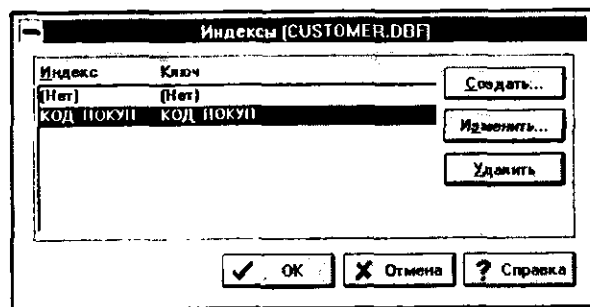


Рис. 5.6. Окно диалога "Индексы".

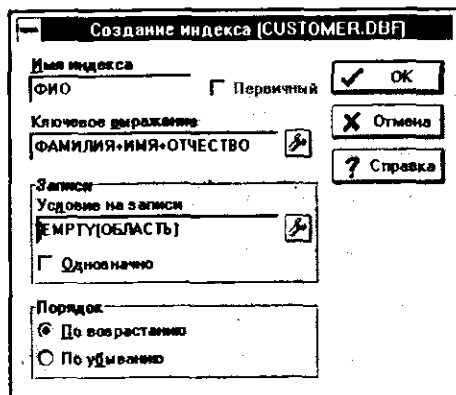


Рис. 5.7. Окно диалога "Создание индекса".

- Вы можете не только задать критерий упорядочивания таблицы, но и ограничить список отображаемых записей при выборе создаваемого индекса. Для этой цели воспользуйтесь полем Условие на записи. Для отображения клиентов, проживающих в областных центрах введите условие: ЕМРТУ (ОБЛАСТЬ).
- Выберите ОК. dBASE сохранит составной индекс, присвоив ему указанное имя.

В окне таблицы записи будут отсортированы по заданному критерию. Обратите внимание на то, что в окне таблицы отображаются только клиенты, проживающие в областных центрах (рис. 5.8).

Аналогичным образом вы можете создать следующий индекс. Индексы удобны для просмотра данных по разным критериям сортировки, но каждый индекс занимает определенное место на жестком диске. Кроме того, при добавлении записи, а также при изменении значения индексных полей, индексный файл обновляется. Это может привести к некоторому снижению скорости отображения данных.

Управление индексами

Информация о индексах

Для получения информации о индексах откройте таблицу и выберите команду Таблица | Операции | Индексы. При наличии индексов в окне диалога "Индексы" будет отображен их список (рис. 5.9).

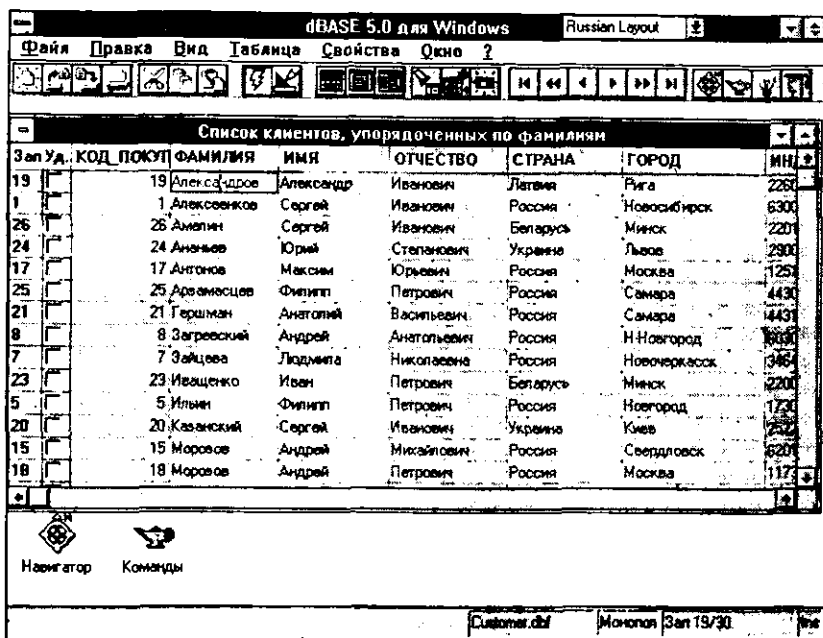


Рис. 5.8. Сортировка таблицы по дополнительному индексу.

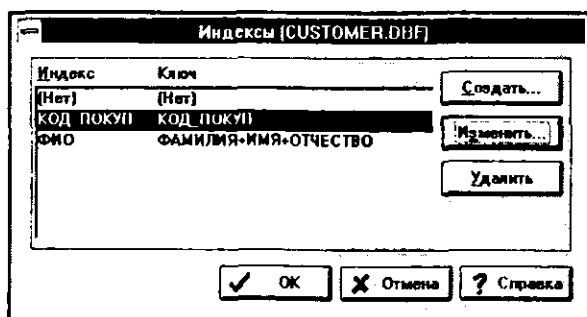


Рис. 5.9. Список индексов таблицы CUSTOMER.

Удаление и изменение индексов

Для удаления индекса выделите его в списке индексов в окне диалога "Индексы" и выберите **Удалить**.

Если вам потребуется внести изменения в определение индекса, выделите его наименование в списке индексов в окне диалога "Индексы" и выберите **Изменить**. dBASE откроет окно диалога "Изменение индекса", в которое поместит описание выделенного индекса. Внесите требуемые изменения и выберите **ОК**.

Упорядочение таблицы при просмотре данных

Для просмотра записей таблицы в порядке, определяемом индексом, выделите его наименование в списке индексов в окне диалога "Индексы" и нажмите кнопку **ОК**.

Заключение

- Основным средством сортировки таблиц является окно диалога “Сортировка записей”. Для того чтобы открыть это окно, выберите **Таблица | Операции | Сортировка записей** в главном окне dBASE.
- Для определения порядка, в котором должны быть отсортированы записи таблицы, необходимо выбрать нужные поля из списка допустимых полей таблицы и добавить их к списку **Ключевые поля** в порядке сортировки. В случае необходимости, вы можете изменить принятый по умолчанию порядок сортировки.
- Для того чтобы изменить направление сортировки, укажите на индикатор направления сортировки требуемого поля и выберите один из допустимых порядков сортировки данных.
- Вы должны указать имя результирующей таблицы и ее тип (dBASE или Paradox).
- Индексы позволяют вам просматривать записи индексированной таблицы в порядке, определяемом выбранным индексом.
- Индекс можно определить в окне диалога “Индексы”. Для того чтобы открыть это окно диалога выполните команду **Таблица | Операции | Индексы**.
- Выражение для индекса задается в поле **Ключевое выражение**.
- Вы можете не только задать критерий упорядочивания таблицы, но и ограничить список выводимых записей при выборе создаваемого списка. Используйте для этого поле **Условие на записи**.
- Для получения информации о индексах выберите **Таблица | Операции | Индексы**. Для изменения выделенного индекса выберите кнопку **Изменить**, а для удаления индекса кнопку **Удалить**.
- Для того чтобы просмотреть содержимое таблицы в порядке, определяемом требуемым индексом, выберите **Таблица | Операции | Индексы**, затем нажмите кнопку **ОК**.

Глава 6. Выбор данных из таблиц с помощью запросов

В предыдущей главе вы познакомились с командой Таблица | Операции | Сортировка записей, которая позволяет упорядочить данные в таблице. Однако на практике часто требуется получить не просто отсортированную таблицу, а выбрать из исходной таблицы часть записей, удовлетворяющих определенным критериям, и упорядочить выборку. Критерии могут определяться сочетанием ряда условий. Например, вам необходимо выбрать из таблицы записи о жителях Минска, чей кредит превышает 1 000 000 рублей, и упорядочить их в алфавитном порядке по полю “Фамилия”. Для решения таких задач предназначено средство, называемое запросом по образцу.

Запрос по образцу является мощным и гибким средством dBASE, с помощью которого вы сможете:

- Формировать сложные критерии для выбора записей из одной или нескольких таблиц
- Указать поля, которые должны быть отображены для выбранных записей
- Редактировать группы записей, удовлетворяющие определенным критериям
- Выполнять вычисления с использованием выбранных данных

Запросы в dBASE являются итеративным средством. Если запрос не дает нужного результата, вы можете внести в него изменения и выполнить еще раз. Кроме того, dBASE позволяет составлять запросы по результатам предыдущих запросов, тем самым просматривать многоступенчатые ситуации типа “А что если?”.

В пределах одной главы невозможно в полном объеме рассмотреть все средства по созданию запросов, предоставляемых в ваше распоряжение dBASE. В настоящей главе мы начнем знакомить вас с использованием запросов в dBASE для обработки информации, содержащейся в одной таблице. Вы познакомитесь с бланком запроса и языком запросов. Вы узнаете, как сформировать выражение запроса для того, чтобы выбрать из таблицы записи, удовлетворяющие определенным условиям, и отобразить в них все или часть полей исходной таблицы. Кроме того, мы обсудим некоторые специфические виды условий выбора записей.

В главе 7 мы продолжим изучение возможностей запросов в dBASE для обработки информации, содержащейся в одной таблице. Кроме того, в ряде глав мы будем возвращаться к рассмотрению дополнительных возможностей запросов при их использовании совместно с другими средствами dBASE.

Что такое “Запрос по образцу”

В главе 1 нами были рассмотрены общие положения, относящиеся к базам данных. В ней мы отмечали, что одним из основных назначений законченного приложения является быстрый поиск информации в базе данных и получение ответов на разнообразные вопросы. Вопросы, формулируемые по отношению к базе данных, называются *запросами*. В dBASE для формирования запросов используется “Запрос по образцу” (Query by Example или QBE). Хотя идея использования такого средства в СУБД не нова, в dBASE она реализована достаточно полно.

Что же такое “Запрос по образцу”? Запрос по образцу — это таблично-ориентированный язык для выбора данных из одной или нескольких таблиц. При формиро-

вании запроса вам необходимо указать критерии поиска записей в исходной таблице и определить вид результирующей таблицы. При этом вместо того чтобы печатать предложения на специальном языке, вы должны просто заполнить соответствующие поля в окне запроса. Метод формирования запроса путем заполнения бланка запроса прост для изучения и понимания. Он способствует эффективному использованию возможностей dBASE пользователями, имеющими даже минимальный навык работы с системой или не имеющими его вовсе.

Бланк запроса

Для того чтобы создать новый запрос, выполните следующие действия:

1. Выберите **Файл | Создать | Запрос**. В качестве альтернативного способа вы можете использовать навигатор. Выделите раздел "Запросы" и выберите из списка запросов <Без названия>.
2. На экране появится окно "Выбор таблицы". Выберите таблицу, для которой вы предполагаете создать запрос, из списка файлов или же из списка таблиц определенной вами ранее базы данных. В качестве примера укажите наименование результирующей таблицы. Выбор таблицы завершите нажатием ОК.
3. dBASE откроет пустое окно запроса (без названия), в которое поместит бланк запроса для указанной таблицы (рис. 6.1). Далее этот бланк используется для составления запроса.

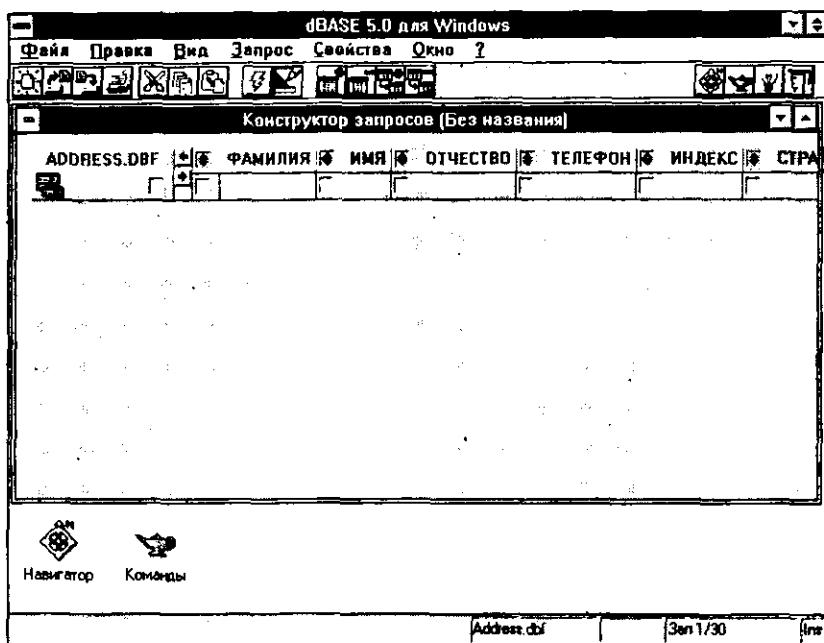


Рис. 6.1. Окно запроса для таблицы ADDRESS.



При составлении запроса для связанных таблиц окно запроса может содержать несколько бланков запроса.

Остановимся подробнее на средствах, которые вы можете использовать при работе в окне запроса. Для этого обратимся к рис. 6.1.



Бланк запроса содержит те же поля, что и исходная таблица. Отметим, что все поля не помещаются в окне запроса. Для перемещения по полям бланка запроса вы можете использовать кнопки, расположенные в бланке запроса сразу же за именем таблицы, а также клавиши *Enter*, *Tab*, *Shift+Tab*, *Ctrl+Home*, *Ctrl+End*.

Каждое из полей бланка запроса содержит флажок, который имеет вид пустой рамки. Назначение этих флажков будет рассмотрено в разделе "Выбор полей для включения в результат запроса".

Данные в бланке запроса отсутствуют. При заполнении бланка запроса в его строках указываются критерии для поиска, выбора и редактирования данных в исходной таблице.

В качестве данных вы можете ввести в бланк запроса операторы запроса, функции и константы, которые составляют язык запросов. Кроме этого, в бланк запроса могут вводиться данные, предназначенные для задания условий выбора записей. Вы можете вводить и редактировать запросы в бланке так же, как делали это при работе с обычными таблицами.



При вводе операторов и функций языка запроса вы можете использовать как прописные, так и строчные буквы.

Меню и панель инструментов при создании запроса

В окне запроса меню dBASE содержит команды **Файл**, **Правка**, **Вид**, **Запрос**, **Свойства**, **Окно** и **?**. Меню **Файл**, **Правка**, **Вид** и **?** содержат стандартный состав команд. Назначение и состав команд меню **Свойства** будет рассмотрено далее в настоящей главе.

Вызов меню **Запрос** осуществляется с помощью мыши или нажатием *Alt+3*.

Добавить таблицу...	^A
Удалить выбранную таблицу	
Копировать результаты в новую таблицу	
И<u>н</u>дексы...	
У<u>с</u>тановить связь...	
И<u>з</u>менить связь...	
У<u>д</u>алить связь	
С<u>о</u>здать вычисляемое поле	
У<u>д</u>алить вычисляемое поле	
Д<u>о</u>бавить условия	
У<u>д</u>алить условия	

Назначение команд меню **Запрос** приведено в табл. 6.1.

Табл. 6.1. Команды меню Запрос

Команда	Назначение	Эквивалентная кнопка панели инструментов
Добавить таблицу	Добавляет в окно запроса бланк для новой таблицы	
Удалить выбранную таблицу	Удаляет бланки запроса одной или нескольких таблиц из окна запроса	
Копировать результаты в новую таблицу	Копирует результаты в новую таблицу	
Индексы	Определяет индекс, который должен стать основным	
Установить связь	Устанавливает связь между двумя таблицами	
Изменить связь	Изменяет ранее установленную связь между таблицами	
Удалить связь	Удаляет связь между таблицами	
Создать вычисляемое поле	Добавляет в бланк запроса новое вычисляемое поле	
Удалить вычисляемое поле	Удаляет из бланка запроса вычисляемое поле	
Добавить условия	Добавляет условие фильтрации, которое применяется ко всему запросу	
Удалить условия	Удаляет условие фильтрации по текущему полю из области условий	

Кроме перечисленных кнопок панель инструментов содержит кнопки **Удалить в буфер**, **Копировать в буфер**, **Вставить из буфера**, **Навигатор**, **Эксперт**, **Инструктор**, **Окно команд**, которые выполняют свои обычные функции. Кроме того, на панели инструментов присутствуют новые кнопки:



Выполнить запрос

Выполняет текущий запрос



Модифицировать запрос

Переходит в режим конструктора запросов



Печать

Печатает активное окно

Результирующая таблица

После задания в бланке запроса условий для выбора записей и указания полей, которые вы хотите увидеть в результате выполнения запроса, нажмите кнопку **Выполнить запрос** на панели инструментов или **F2** на клавиатуре. dBASE создаст временную таблицу, в которую поместит выбранные записи, и отобразит ее на экране. Эта таблица содержит поля, указанные вами в бланке запроса, и записи, удовлетворяющие заданным условиям. Результирующая таблица называется *представлением данных*, поскольку она не содержит реальные данные, а позволяет увидеть данные, хранящиеся в таблицах, на основе которых выполнен запрос. Вы можете просматривать и редактировать содержимое этой таблицы в табличном виде или в виде формы.

На практике процесс выбора данных из таблицы по сложному критерию часто носит итерационный характер. В этом случае вам необходимо сохранять результат текущей итерации в новой таблице и формировать следующую выборку на основе сохраненной таблицы.

При выходе из dBASE результирующая выборка будет уничтожена. Поэтому, если вы хотите сохранить результаты выборки, выполните команду **Запрос | Копировать результаты в новую таблицу**.

Сохранение запроса

Заполнив бланк запроса, вы можете сохранить его, что позволит вам сэкономить время при последующих выполнениях данного запроса. Для сохранения запроса выберите **Файл | Сохранить**, **Файл | Сохранить как** или **Файл | Сохранить и закрыть**. Если вы попытаетесь закрыть окно запроса, не сохранив его содержимое, вам будет предложено сохранить текущий запрос.



При задании имени запроса действуют те же правила, что и при назначении имен другим объектам dBASE.

После сохранения запрос становится стандартным объектом dBASE, с которым вы можете выполнять обычные действия: открыть запрос, копировать его, удалить и т.д. В последующих главах вы узнаете как разрабатывать формы и отчеты, используя для этого непосредственно сохраненные запросы, а не результирующие таблицы, получаемые в результате их выполнения.

Выполнение запроса

Заполнив бланк запроса, вы тем самым сформировали текущее выражение запроса. Для выполнения запроса воспользуйтесь одним из следующих способов:

- Выберите **Вид | Результаты запроса**
- Нажмите кнопку **Выполнить запрос** на панели инструментов
- Нажмите **F2**

Если запрос составлен корректно, dBASE выполнит его и откроет в новом окне результирующую таблицу, в которую поместит выбранные записи.

Открытие и выполнение существующего запроса

В случае необходимости, вы можете внести изменения в существующий запрос. Для того чтобы открыть запрос, выберите **Файл | Открыть** в главном окне. Далее в

окне диалога “Открытие файла” укажите в качестве типа выбираемых файлов “Запрос” и выберите из списка файлов запрос, который вы предполагаете изменить. Вы можете открыть запрос в режиме выполнения или конструктора. Установите для этого соответствующий переключатель в правом нижнем углу окна диалога “Открытие файла”.

В качестве альтернативного способа вы можете использовать навигатор. Выделите для этого раздел “Запросы” и выберите из списка требующийся вам запрос. Для выполнения запроса нажмите клавишу *F2* или выберите **Открыть** из меню объекта. Для открытия запроса в окне конструктора нажмите клавишу *Shift+F2* или выберите **Конструктор запросов** из меню объекта.

Печать результирующей таблицы



Перед печатью результата запроса проверьте, чтобы ваш принтер был готов к работе. После чего нажмите на панели инструментов кнопку **Печать**.

Более подробно печать документов будет обсуждаться в главах 8 и 9. Кроме того, в главе 9 вы узнаете как создать отчет непосредственно на основании запроса.

Выбор полей для включения в результат запроса

Отбор полей

Для отбора полей, которые должны присутствовать в результирующей таблице, вам необходимо пометить их в бланке запроса. Для маркировки полей в зависимости от требований, предъявляемых к выборке, могут быть использованы различные специальные символы.

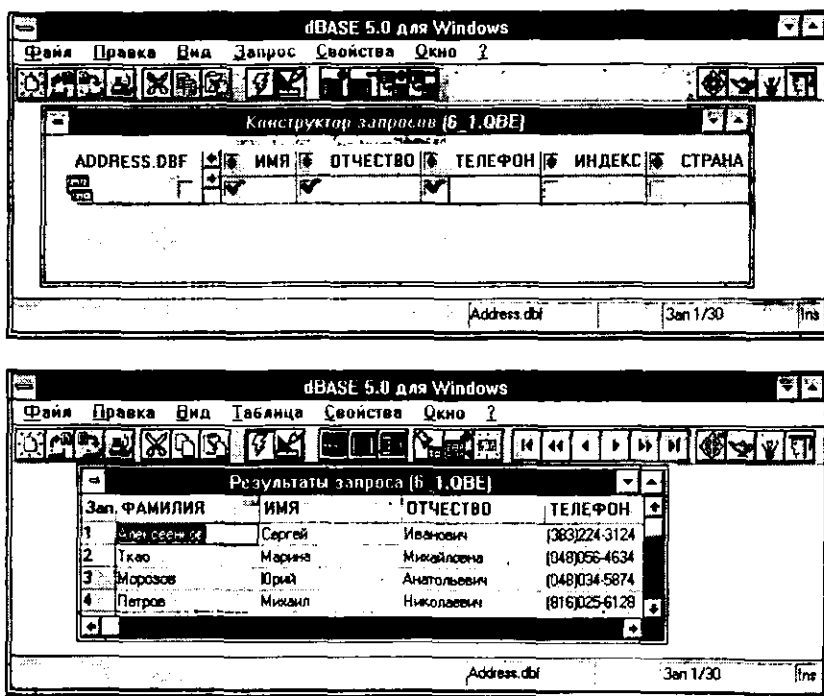


Рис. 6.2. Выбор полей.

На рис. 6.2 представлен заполненный бланк запроса. Вы видите, что поля “Фамилия”, “Имя”, “Отчество” и “Телефон” содержат метки. Метка является специальным символом языка запросов и указывает на то, что отмеченные им поля должны присутствовать в результирующей таблице. На том же рисунке вы видите результирующую таблицу, которая содержит только отмеченные поля. Поля исходной таблицы, не содержащие метки, в результирующей таблице отсутствуют.

Рассмотрим правила установки меток. Укажите на флажок в поле бланка запроса, который вы хотите отметить, и нажмите кнопку мыши. В результате в рамке этого поля появится метка (✓). Если вы измените свое мнение и решите, что отмеченное поле должно отсутствовать в результирующей таблице, повторно нажмите мышью флажок. Метка исчезнет из бланка запроса.

Вы можете также воспользоваться клавиатурой для размещения и удаления метки. Для того чтобы установить метку, установите курсор в требуемое поле с помощью клавиш *Tab* и *Shift+Tab*, после чего нажмите *F5*. Клавиша *F5* является переключателем. Повторно нажатие *F5* приведет к удалению метки.

Если вам требуется отметить сразу все поля в бланке запроса, нажмите *Ctrl+F5* или укажите на пустой флажок, расположенный в крайнем левом поле бланка, и нажмите кнопку мыши. Метки появятся во всех полях бланка. Повторное нажатие мышью флажка или *Ctrl+F5* приведет к удалению всех меток.

Изменение порядка сортировки записей

При выполнении запроса порядок расположения полей в результирующей таблице определяется последовательностью полей в бланке запроса, которые содержат символы маркировки.

Для изменения порядка следования полей в бланке запроса:

1. Поместите курсор мыши на имя поля в представлении бланк запроса. При этом курсор принимает форму 
2. Нажмите кнопку мыши и, удерживая ее, перенесите поле с помощью мыши на новое место.
3. Отпустите кнопку мыши.

По умолчанию при выполнении запроса dBASE не сортирует записи результирующей таблицы. Порядок следования записей определяется порядком их физического расположения в таблице, т. е. порядком их добавления в таблицу.

Однако вы можете выполнить сортировку одного или нескольких полей в прямом или обратном порядке, используя для этого кнопки сортировки, которые расположены левее наименования поля. В этом случае порядок сортировки определяется последовательностью следования полей слева направо и типом сортировки полей. Для определения порядка сортировки по конкретному полю нажмите на кнопку сортировки поля и выберите одну из возможных стрелок:

Стрелка	Тип сортировки
	Порядок ввода
	По возрастанию, с учетом регистра букв



По убыванию, с учетом регистра букв



По возрастанию, без учета регистра букв (только для текстовых полей)



По убыванию, без учета регистра букв (только для текстовых полей)

При сортировке знаки пунктуации следуют перед цифрами, цифры следуют перед прописными буквами, а прописные буквы следуют перед строчными буквами (сначала А...Я, затем а...я).

Для определения типа сортировки поля вы также можете использовать клавиатуру. Для этого переместите курсор в требуемое поле и нажмите несколько раз **F6** до тех пор, пока в кнопке сортировки поля не появится требуемая стрелка.

В одном бланке запроса вы можете установить разные типы сортировки отдельных полей. При выполнении запроса записи в отмеченных полях будут упорядочены в соответствии с заданным типом сортировки.

Например, на рис. 6.3 показаны заполненный бланк запроса для таблицы ADDRESS и результирующая таблица. Вы видите, что поле "Имя" отмечено стрелкой , а поле "Фамилия" — . В результате в результирующей таблице записи будут вначале упорядочены по возрастанию значений величин в поле "Фамилия", а затем — по убыванию значений величин в поле "Имя".

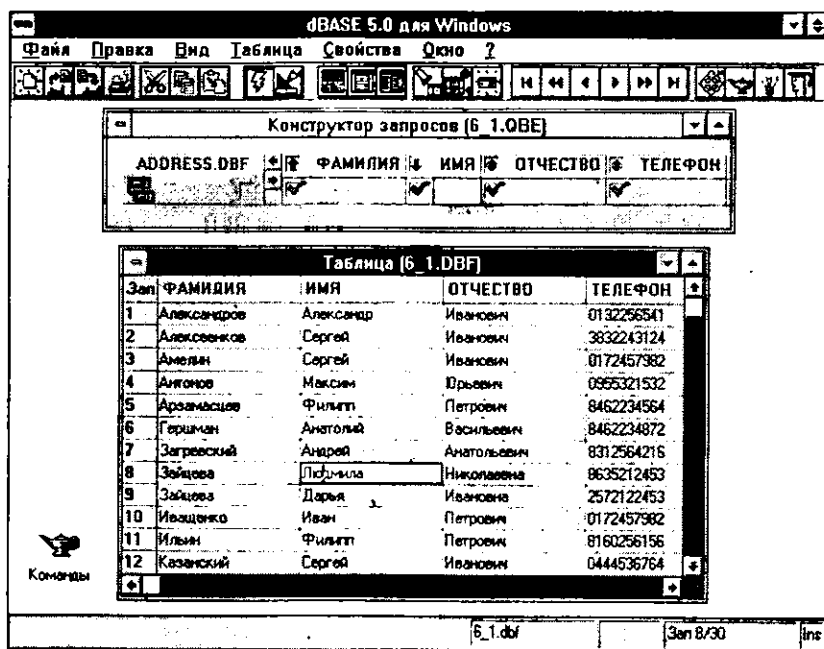


Рис. 6.3. Изменение порядка сортировки записей.

Пример 6.1. Изменение порядка сортировки полей

В предыдущем примере мы получили записи, которые упорядочены по значениям полей "Фамилия" и "Имя". Теперь нам требуется получить аналогичный список, упорядоченный по наименованиям городов. Напомним, что порядок сортировки полей в результирующей таблице определяется последовательностью их расположения в бланке запроса. После выполнения запроса поля таблицы будут отсортированы в указанном вами порядке.

1. Откройте окно запроса для результирующей таблицы и пометьте поля "Фамилия", "Имя", "Отчество", "Телефон" и "Город".
2. Перенесите поле "Город" в начало бланка запроса и установите метку выбора поля.
3. Установите тип сортировки "По возрастанию".
4. Выберите ОК и выполните запрос. Результат выполнения запроса показан на рис. 6.4.

Результаты запроса [6 1.0BE]

Зап	ГОРОД	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ТЕЛЕФОН
1	Ална-Ата	Наседкин	Георгий	Анатолевич	(327)256-7562
2	Алшты	Мшенский	Владимир	Иванович	(044)264-5627
3	Жумовский	Фролов	Виктор	Михайлович	(095)576-5456
4	Киев	Сергеев	Дмитрий	Михайлович	(044)453-6785
5	Киев	Казанский	Сергей	Иванович	(044)453-6764
6	Кировск	Уланов	Максим	Сергеевич	(812)623-4456
7	Кисловодск	Сухарев	Андрей	Анатолевич	(863)573-4543
8	Клин	Яковина	Анатолий	Андреевич	(224)234-7985
9	Львов	Ананий	Юрий	Степанович	(032)278-7659
10	Минск	Ивашенко	Иван	Петрович	(017)245-7982
11	Минск	Амелин	Сергей	Иванович	(017)245-7982
12	Москва	Морозов	Андрей	Петрович	(095)131-7714
13	Москва	Антонов	Максим	Юрьевич	(095)532-1532
14	Н-Новгород	Загребский	Андрей	Анатолевич	(831)256-4216
15	Новгород	Петров	Михаил	Николаевич	(816)017-7124
16	Новгород	Петров	Михаил	Николаевич	(816)025-6128

Рис. 6.4. Изменение порядка сортировки полей.

Изменение результирующей таблицы

Кроме определения порядка сортировки вы можете установить дополнительные свойства результирующей таблицы:

- Изменить имя результирующей таблицы
- Указать каталог, в котором должна быть сохранена результирующая таблица
- Выбрать формат для сохранения результирующей таблицы (dBASE или Paradox)

Для этой цели предназначена команда **Запрос | Копировать результаты в новую таблицу**, при выполнении которой открывается окно диалога “Сохранение таблицы”, показанное на рис. 6.5.

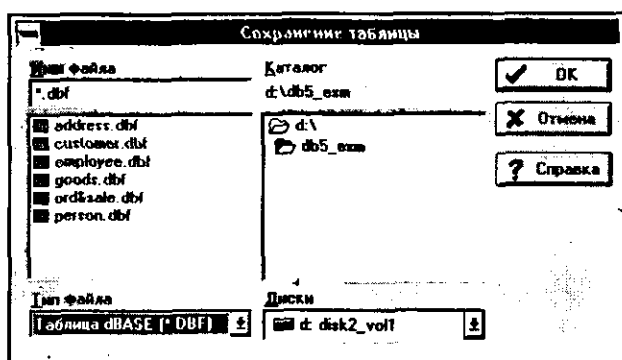


Рис. 6.5. Окно диалога “Сохранение таблицы”.

Определение имени результирующей таблицы

Результирующая таблица является временной таблицей и будет уничтожена при выходе из dBASE. Если вы хотите сохранить результат выполнения запроса, вам необходимо переименовать результирующую таблицу. Для этого введите новое имя таблицы в строку ввода **Имя файла**.

По умолчанию результирующая таблица помещается в каталог, в котором находилась исходная таблица. Если вы предполагаете сохранить результат выполнения запроса в каком-либо другом каталоге, выберите его в разделе “Каталог”.



Если вы укажете каталог, который уже содержит таблицу с таким же именем, dBASE предварительно запросит у вас подтверждения замены существующей таблицы на новую.

Изменение типа таблицы

Независимо от типа исходной таблицы, по умолчанию для результирующей таблицы устанавливается тип dBASE. В случае необходимости вы можете поместить результат выполнения запроса в таблицу Paradox. Для этого в разделе “Тип файла” измените начальную установку dBASE на Paradox.

Пример 6.2. Сохранение результата выполнения запроса

Давайте в качестве примера с помощью окна диалога “Сохранение таблицы” сохраним результат предыдущего запроса в таблице ADDRNEW.

1. Возвратитесь к бланку предыдущего запроса.
2. Выполните команду **Запрос | Копировать результаты в новую таблицу** и в окне диалога “Сохранение таблицы” укажите имя новой таблицы ADDRNEW и тип таблицы dBASE. Каталог оставьте без изменения.
3. Выберите **ОК**, после чего запрос будет выполнен и результат запроса появится на экране.

Результат выполнения запроса показан на рис. 6.6. Теперь вы можете использовать эту таблицу для создания новых запросов, печати отчетов, просмотра в следующих сеансах работы.

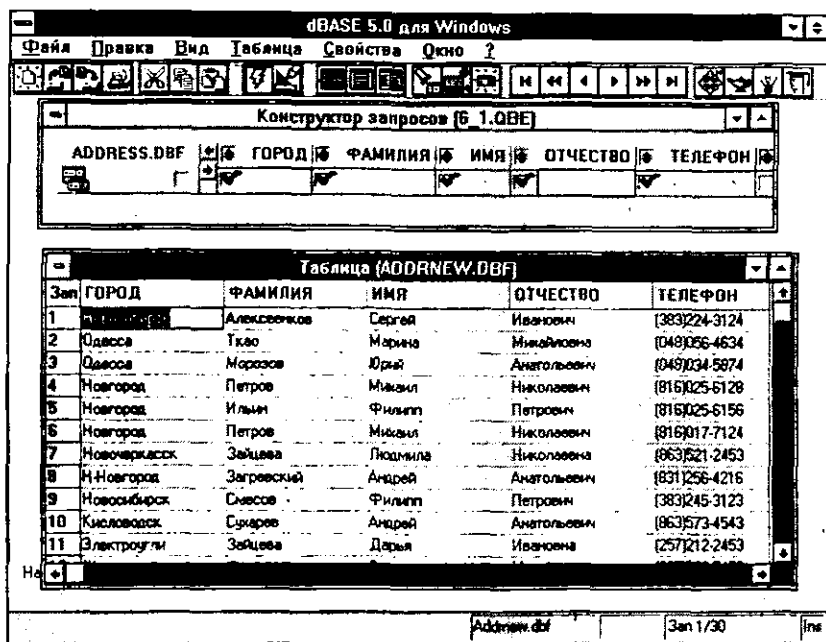


Рис. 6.6. Сохранение результирующей таблицы.

Построение простых условий для выбора записей

Метки, являющиеся одним из элементов языка запросов, с которыми вы познакомились в настоящей главе, позволяют отобразить в результирующей таблице все или часть полей исходной таблицы, упорядоченные соответствующим образом. На практике в большинстве случаев требуется получить не все записи исходной таблицы, а лишь ту часть, которая удовлетворяет определенным условиям. Простейший критерий для выбора записей предполагает точное совпадение значений поля.

Точное совпадение значений одного поля

Точное совпадение значений одного из полей позволяет выбрать из таблицы одну запись или группу записей, одно из полей которых содержит указанные величины. Например, вам требуется просмотреть список всех абонентов, живущих в Минске, или найти адрес и телефон человека по фамилии Морозов.

Для задания условия выбора напечатайте требуемое значение в соответствующем поле бланка запроса. Тем самым вы создадите шаблон, с которым dBASE при выполнении запроса будет сравнивать все записи исходной таблицы. В результирующую таблицу будут помещены только те данные, значения которых совпали с шаблоном.

В обсуждаемых ниже примерах мы будем использовать простые условия для текстовых полей. В бланке запроса текстовые значения заключаются в двойные кавычки (" "), одинарные кавычки (') или квадратные скобки ([]).

Пример 6.3. Точное совпадение значений текстового поля

1. Решение первой задачи начните с того, что получите на экране бланк запроса для результирующей таблицы.
2. При решении данной задачи ограничимся четырьмя полями исходной таблицы, которые содержат фамилии, имена и отчества абонентов, а также номера их телефонов. Для того чтобы эти поля были отображены в результирующей таблице, поместите в каждое из них метку. Поскольку известно, что результат выполнения запроса будет содержать только записи, в поле "Город" которых содержится значение "Минск", отображать это поле в результирующей таблице не имеет смысла.
3. Для задания условия выбора записей переместите курсор в поле "Город" и введите: 'Минск'.
4. Для выполнения запроса нажмите кнопку **Выполнить запрос** на панели инструментов или клавишу **F2**. На экране появится результирующая таблица, которая содержит записи об абонентах, проживающих в Минске. Запрос и результирующая таблица показаны на рис. 6.7.

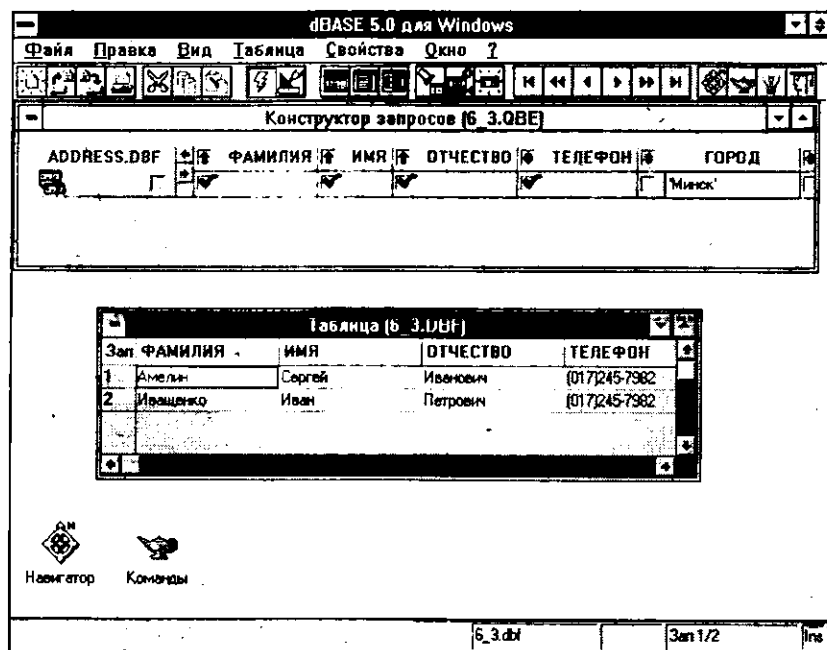


Рис. 6.7. Точное совпадение значения одного поля.

Пример 6.4. Редактирование запроса

Теперь найдем запись о человеке по фамилии Морозов.

1. Перейдите в окно запроса.

2. Удалите старое условие поиска. Для этого перейдите в поле "Город" и удалите введенное в нем условие.
3. Если вас интересует название города, в котором живет Морозов, отметьте это поле.
4. Переместите курсор в поле "Фамилия" и после метки введите: 'Морозов'. Задание выражения запроса завершено — для его выполнения нажмите кнопку **Выполнить** запрос на панели инструментов или клавишу **F2**.

Результирующая таблица представлена на рис. 6.8. dBASE нашел в исходной таблице три записи о людях по фамилии Морозов, и вам остается только выбрать необходимую.

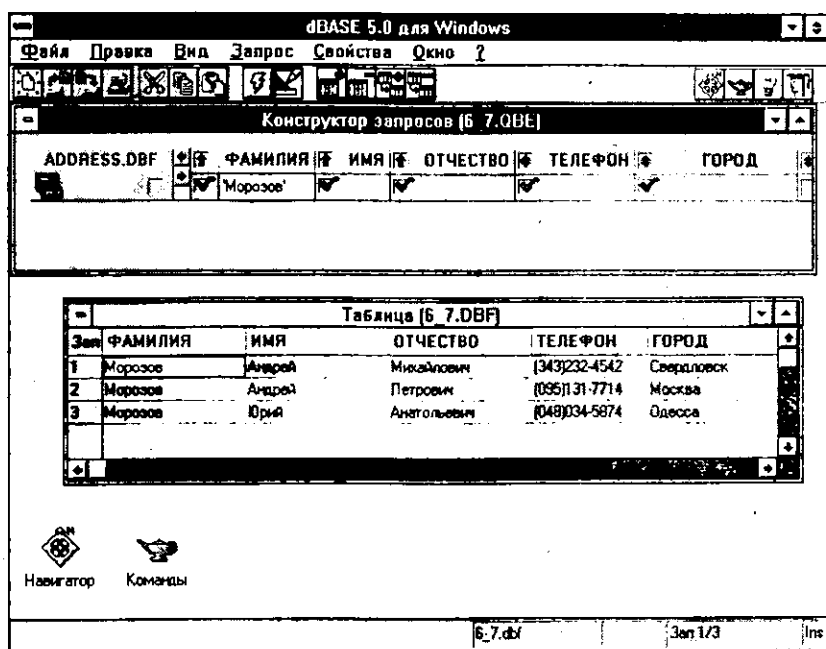


Рис. 6.8. Второй пример точного совпадения значений одного поля.



Не допускайте ошибок при вводе значений в бланк запроса! dBASE считает ошибкой не только неправильно введенные символы, но также и ошибочное использование верхнего и нижнего регистров. Например, если вы введете условие: "МОРОЗОВ", нужные вам записи не появятся в результирующей таблице.

Точное совпадение величин накладывает жесткие ограничения на выбор записи из таблицы. Оператор \$ позволяет вам обойти эти ограничения и более гибко задавать условия совпадения значений поля с заданной величиной. Этот оператор языка запросов будет рассмотрен далее в настоящей главе.

Во всех рассмотренных примерах в качестве условия выбора задавалась текстовая величина. dBASE позволяет также производить выбор записей по совпадению чисел и дат. Условия выбора по значениям чисел или дат задаются аналогично условиям, использующим текстовые величины.

Точное совпадение значений в полях типа дат

dBASE поддерживает следующие форматы даты:

Тип	Формат
AMERICAN	MM/DD/YY
ANSI	YY.MM.DD
BRITISH	DD/MM/YY
FRENCH	DD/MM/YY
GERMAN	DD.MM.YY
ITALIAN	DD-MM-YY
JAPAN	YY/MM/DD
USA	MM-DD-YY
MDY	MM/DD/YY
DMY	DD/MM/YY
YMD	YY/MM/DD

По умолчанию используется формат, установленный в Панели Управления Windows. Для изменения формата используется окно диалога "Свойства Среды dBASE".

При задании условия точного совпадения даты вы можете использовать любой из допустимых разделителей, но порядок следования дня, месяца и года должен соответствовать установленному формату.

Дата в условии вводится в фигурных скобках. В качестве примера на рис. 6.9 показаны сформированный запрос и результат его выполнения. dBASE нашел в таблице ADDRESS две записи о человеке, родившемся 04.10.64. Вы получите аналогичный результат, введя дату в виде 04/10/64 или 4-10-64.

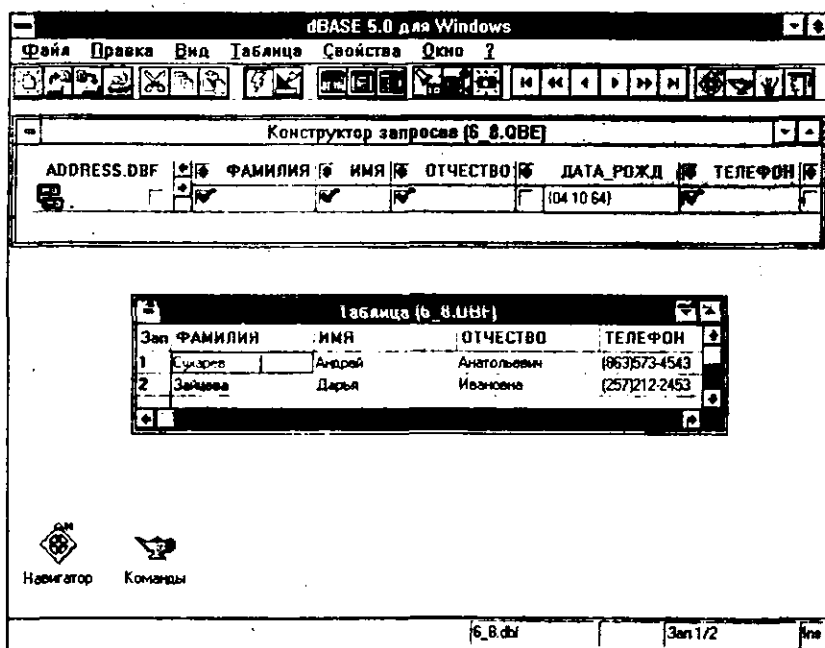


Рис. 6.9. Точное совпадение значений в поле дат.

Точное совпадение значений в цифровых полях

При вводе чисел в бланк запроса существует одно ограничение: разделители в целой части числа должны отсутствовать.

Например, из результирующей таблицы требуется выбрать записи о клиентах, чей кредит точно равен 25 000. Откройте окно запроса для результирующей таблицы и отметьте в нем требуемые поля. Напечатайте в поле "Кредит" число 25000. Требуемые записи вместе с бланком запроса показаны на рис. 6.10.

Ниже приведены правила, которыми следует руководствоваться при вводе чисел в числовые поля в запросах.

- Не вводите специальные символы, используемые при отображении денежных значений (\$, руб. и т.д.).
- Для обозначения отрицательных чисел используйте знак минус перед числом, но не круглые скобки.
- Наличие разделителей целой и дробной частей числа (точки в американской нотации и запятые в международной нотации) обязательно.

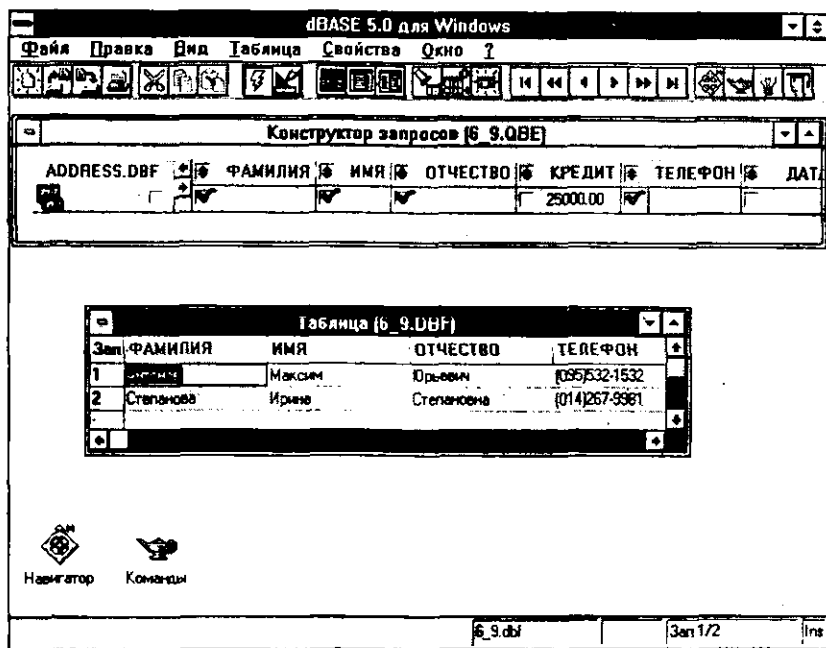


Рис. 6.10. Точное совпадение значений в числовом поле.

До сих пор для задания условия точного совпадения значений одного из полей с заданной величиной вы просто вводили эту величину в соответствующем поле бланка запроса. Теперь мы рассмотрим выбор пустых значений, сравнения с текущей датой и оператор <>, которые также предназначены для задания условий точного совпадения.

Выбор пустых значений

На практике достаточно часто встречаются таблицы, некоторые из полей которых не содержат данных. Такие поля называют пустыми. Исходные таблицы могут содержать пустые поля по различным причинам. Например, таблица ADDRESS содержит

поле "Область", которое будет пустым для всех записей о жителях областных центров. В этом случае поле "Область" должно быть пустым по смыслу. В других случаях причина может быть в несвоевременном получении данных. Тогда по мере поступления информации вам потребуется отыскивать и заполнять пустые поля.

В dBASE пустые поля определены для текстовых полей и полей типа дата. Для выбора пустых значений в текстовых полях вы можете использовать пустую строку "", а для выбора пустых значений дат пустую дату {}. Кроме этого для выбора записей, содержащих пустые поля предназначены функции ISBLANK и EMPTY (Пустое значение). Функция EMPTY, в отличие от функции ISBLANK, в числовых полях не различает пустое значение поля от значения 0. Так как эти функции требуют в качестве параметра имя поля, вы не можете использовать их при задании условия поиска, которое может содержать любое допустимое логическое выражения dBASE. В этой области, в частности, можно использовать и функцию EMPTY.

Пример 6.5. Поиск записей, имеющих пустое поле

В качестве примера предположим, что вам требуется выбрать из таблицы ADDRESS записи о жителях областных центров.

1. Откройте окно запроса для таблицы ADDRESS.
2. Установите следующий порядок полей: "Город" — "Фамилия" — "Имя" — "Отчество" — "Телефон". Поместите метки во все перечисленные поля.
3. Установите сортировку по возрастанию для полей "Город", "Фамилия", "Имя" и "Отчество".
4. В поле "Область" напечатайте "".

Зап	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ТЕЛЕФОН	ГОРОД
1	Иванов	Александр	Иванович	(013)225-6541	Рига
2	Александров	Сергей	Иванович	(383)224-3124	Новосибирск
3	Амелин	Сергей	Иванович	(017)245-7982	Минск
4	Ананиев	Юрий	Степанович	(032)278-7659	Львов
5	Антонов	Максим	Юрьевич	(095)532-1532	Москва
6	Арзамасцев	Филипп	Петрович	(846)223-4564	Самара
7	Гаршман	Антоний	Васильевич	(846)223-4872	Самара
8	Загребский	Андрей	Анатолеевич	(831)256-4216	Н.Новгород
9	Зайцев	Людмила	Николаевна	(863)521-2453	Новочеркасск
10	Ивашченко	Иван	Петрович	(017)245-7982	Минск
11	Ильин	Филипп	Петрович	(816)025-6156	Новгород
12	Казанский	Сергей	Иванович	(044)453-6764	Киев
13	Морозов	Андрей	Михайлович	(343)232-4542	Свердловск

Рис. 6.11. Выборка записей с пустыми значениями одного из полей.

На рис. 6.11 вы видите результирующую таблицу, которая появится на экране после нажатия клавиши *F2* или кнопки **Выполнить запрос**.

Использование области условий

Для задания сложных условий выборки используется область условий окна конструктора запросов. Область условий целесообразно применять в следующих случаях:

- Для выборки по выражению, а не по значению поля
- Для объединения нескольких условий в одно выражение
- Для выборки данных, основанных на сравнении значений полей
- Для выборки данных по полям двух таблиц

Более подробно использование области условий мы рассмотрим в следующей главе. В настоящем разделе мы определим в этой области простое условие для выборки пустых значений.

Для открытия области “Условия” выполните команду **Запрос | Добавить условия**. Затем вы можете ввести в этой области любое допустимое в dBASE логическое выражение.

Пример 6.6. Использование функции EMPTY

В качестве примера выполним предыдущий запрос с использованием функции **EMPTY**.

1. Откройте бланк предыдущего запроса.
2. Удалите условие выборки из поля “Область”.
3. Выполните команду **Запрос | Добавить условия**, в результате чего в бланке запроса откроется область “Условия”.
4. В области “Условия” напечатайте **EMPTY(Область)** для выборки записей с пустыми значениями в поле “Область”.

Бланк запроса и результирующая таблица представлены на рис. 6.12.

Поиск записей за текущий день.

Функция DATE (Текущая дата)

В полях типа дат функция **DATE** определяет сегодняшнюю дату. Для того чтобы функция **DATE** работала правильно, необходимо, чтобы таймер вашего компьютера, был установлен верно.

При решении практических задач функция **DATE** наиболее часто используется в сочетании с арифметическими операторами. К обсуждению такого применения функции **DATE** мы вернемся в главе 7.

Если вам потребуется выбрать записи, введенные в таблицу за сегодняшний день, воспользуйтесь функцией **DATE**. В этом случае вы должны в соответствующем поле бланка запроса напечатать: **DATE()**.

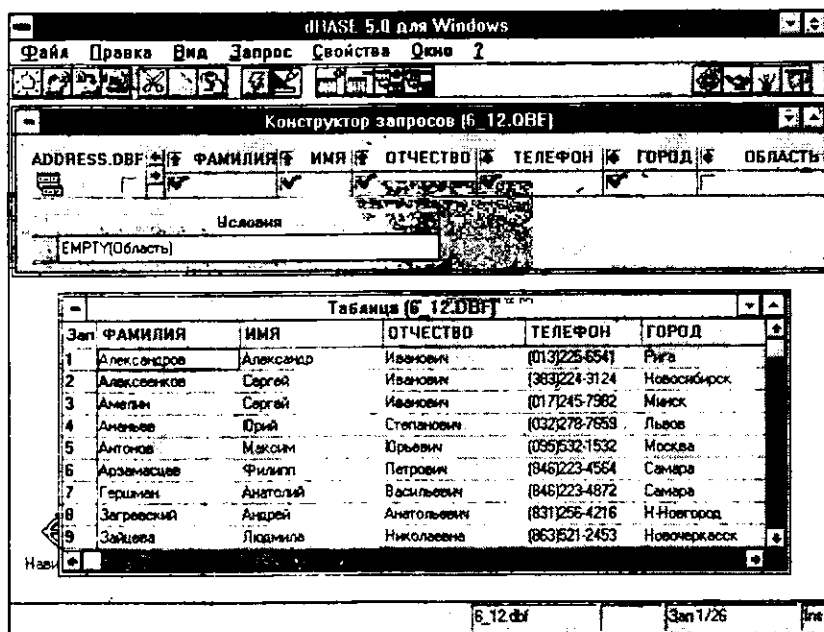


Рис. 6.12. Использование области условия.

Точное несовпадение значений одного из полей.

Оператор <> (Не совпадает с)

До настоящего момента мы осуществляли выбор записей по точному совпадению чисел, дат или последовательности символов в одном из полей. Возможно, вам потребуется найти в таблице записи, значения которых не удовлетворяют определенному условию. Например, вам нужен список людей, проживающих в областях (таблица ADDRESS).

Оператор <> позволяет выбрать из таблицы записи, не удовлетворяющие определенному условию. Для задания условия выбора вы должны в бланке запроса напечатать: <>, далее само значение. Например, напечатав в поле "Страна": <> 'Россия', вы сможете выбрать все записи, за исключением записей со значением "Россия" в поле "Страна".

Оператор <> является оператором логического отрицания. При построении условий отрицания оператор <> можно использовать совместно с функциями EMPTY, DATE, а также с выражениями языка dBASE.

Пример 6.7. Поиск записей, имеющих непустое поле

Воспользуемся сочетанием операторов <> " для получения списка предприятий, на которых работают абоненты (таблица ADDRESS). Поскольку некоторые из записей не содержат значений в поле "Предпр", такие записи не представляют для нас интереса и должны быть исключены из результата запроса. Для этой цели воспользуемся сочетанием <> ", которое является логическим отрицанием условия ".

1. Откройте окно запроса для таблицы ADDRESS.

2. Установите в бланке запроса следующий порядок полей: "Предприятие" — "Фамилия" — "Имя" — "Отчество" — "Телефон".
3. Поместите метки во все перечисленные поля и установите для всех них, за исключением поля "Телефон", сортировку по возрастанию.
4. Напечатайте в поле "Предприятие": <> " ".

Сформированный запрос и результирующая таблица представлены на рис. 6.13.

Зап	ПРЕДПР	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ТЕЛЕФОН
1	ВетАВМ	Ивашенко	Иван	Петрович	(017)245-7982
2	ВетАВМ	Петров	Михаил	Николаевич	(816)017-7124
3	ВетАВМ	Анатолий	Сергей	Иванович	(017)245-7982
4	PC Sales, Service	Савиных	Сергей	Иванович	(095)453-5385
5	AD Интерсервис	Александров	Сергей	Иванович	(383)224-3124
6	AD Компьютерсервис	Ткаченко	Марина	Михайловна	(048)056-4634
7	AD РосКом	Морозов	Андрей	Михайлович	(343)232-4542
8	AD Фигон	Райский	Максим	Сергеевич	(846)253-4542
9	ААриг	Сергеева	Дiana	Михайловна	(044)453-6786

Рис. 6.13. Запрос с условием точного несовпадения значений одного из полей.

Запросы по значениям Мемо-полей

Традиционно запросы формируются по значениям текстовых и числовых полей, а также полей дат. Однако вы можете осуществить запрос по значениям Мемо-полей. С помощью уже известного вам условия <> EMPTY() можно определить наличие значений в выбранном поле. Для графических, двоичных и OLE-полей условия выбора EMPTY() и <> EMPTY() (вы можете также использовать функцию ISBLANK) являются единственными условиями.



Если вам потребуется включить в результирующую таблицу графические, двоичные или Мемо-поля, предварительно убедитесь в правильности составления запроса и только после этого включите требуемые поля. Такой подход обусловлен тем, что включение полей указанных типов требует значительного дискового пространства и продолжительного времени.

Для Мемо-полей вы можете осуществлять выборку по фрагменту значения Мемо-полей, используя для этого оператор \$ и шаблоны, с которыми вы познакомитесь далее в настоящей главе.



Помните, что содержимое Мемо-полей хранится в отдельном файле. При поиске значений Мемо-полей dBASE осуществляет поиск в дополнительных файлах, что замедляет выполнение запроса. Поэтому при создании таблиц старайтесь включать в Мемо-поля только ту информацию, по которой вам вряд ли потребуется выполнять запросы.

Условие включения. Оператор \$

Требование точного задания чисел или последовательности символов в качестве критерия поиска является жестким ограничением. Время от времени возникает неприятная ситуация, при которой вы не помните точного написания условия поиска. Кроме того, вы можете просто не знать какой регистр (верхний, нижний или их сочетание) был использован при вводе каждой записи. При работе с большими таблицами такая ситуация превращается в серьезную проблему.

В подобных случаях вы можете осуществить выбор записей по условию включения заданного значения, используя оператор \$. Данный оператор позволяет найти требуемые записи, указанные поля которых содержат заданное значение. Оператор \$ имеет разный смысл в условии для поля и в области условий. В условии поля оператор \$ означает "содержит". В области условия (и в языке dBASE) оператор \$ означает "содержится в". Например, если ввести \$ "АО" в условии поля "Предпр", результирующая таблица для данного запроса будет включать все записи, которые в поле "Предпр" содержат слово "АО". Условие Город \$ "Новгород" будет содержать все записи, значение полей которых содержится в слове "Новгород". Например, такими значениями могут быть "Новгород", "Новг", "город".

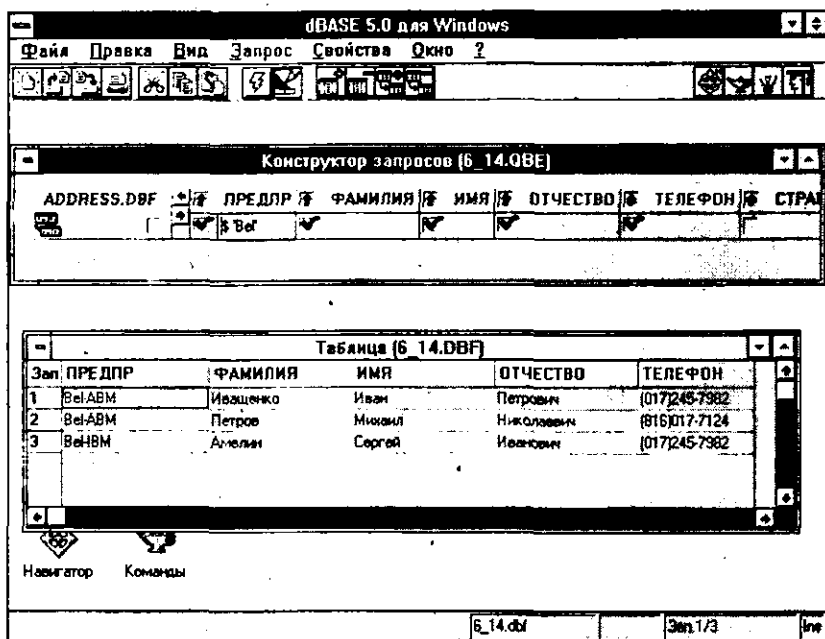


Рис. 6.14. Использование оператора \$.

В качестве примера предположим, что в таблице ADDRESS требуется найти запись о представителе фирмы Bel-ABM. Однако вы не уверены в правильности написания второй части наименования. Попытаемся найти требуемую запись, используя

оператор \$. Для этого в поле "Предпр" бланка запроса для таблицы ADDRESS после метки напечатайте: \$ 'Bel'. На рис. 6.14 показаны бланк запроса и результат поиска. dBASE отобразил в результирующей таблице не только две записи о представителях фирмы Bel-ABM, но и запись об Амелине С. И., сотруднике Bel-IBM. Если вам требовалась именно эта запись, без помощи оператора \$ найти ее было бы непросто.



В тех случаях, когда запрос с точным совпадением значений не дал требуемого результата, воспользуйтесь оператором \$. Возможно, проблема заключается в неправильном написании условия поиска.

Более широкие возможности поиска записей при неточном задании условия вам предоставляют операторы шаблонов.



Оператор \$ позволяет вам выбрать записи с требуемым значением в Метод-поле.

Условия совпадения с шаблонами

Оператор LIKE (Похож на) с символами шаблонов обеспечивает поиск текстов с помощью шаблонов. Поскольку операторы шаблона заменяют единичный символ или целую группу символов, возможности выбора записей по условию неточного совпадения значений существенно расширяются по сравнению с оператором \$.

К символам шаблонов относятся:

Символ	Заменяет
*	Любое количество символов, включая отсутствие символов
?	Любой один символ

Символы шаблона могут располагаться в любой части значения для поиска, в том числе в его начале и конце. Кроме того, условие может содержать несколько символов шаблона.

Символ шаблона *

Символ шаблона * обозначает последовательность любых символов, включая пробелы и отсутствие символов. Поэтому символ шаблона * называют *групповым шаблоном*.

Например, вам требуется найти в таблице ADDRESS запись о человеке, фамилия которого начинается на "А" и заканчивается на "ов". Найти нужную запись вам поможет шаблон А*ов. Введите его в бланк запроса в поле "Фамилия" и отметьте поля, которые вас интересуют. В результате выполнения запроса вы получите все записи, значение которых совпадает с шаблоном (рис. 6.15).



При использовании шаблона * dBASE различает заглавные и строчные буквы.

Как правило, вам редко требуется полная выборка записей, для которых значения поля "Фамилия" совпадают с шаблоном А*ов. Для больших таблиц объем выборки может быть значителен и потребует некоторого времени для отыскания единствен-

ной нужной записи в результирующей таблице. Поэтому, если вы вспомнили еще хотя бы одну букву, используйте более точный шаблон (например, A*т*ов). При выполнении запроса dBASE обнаружил в таблице ADDRESS единственную запись, удовлетворяющую условию поиска, Антонов (рис. 6.16).

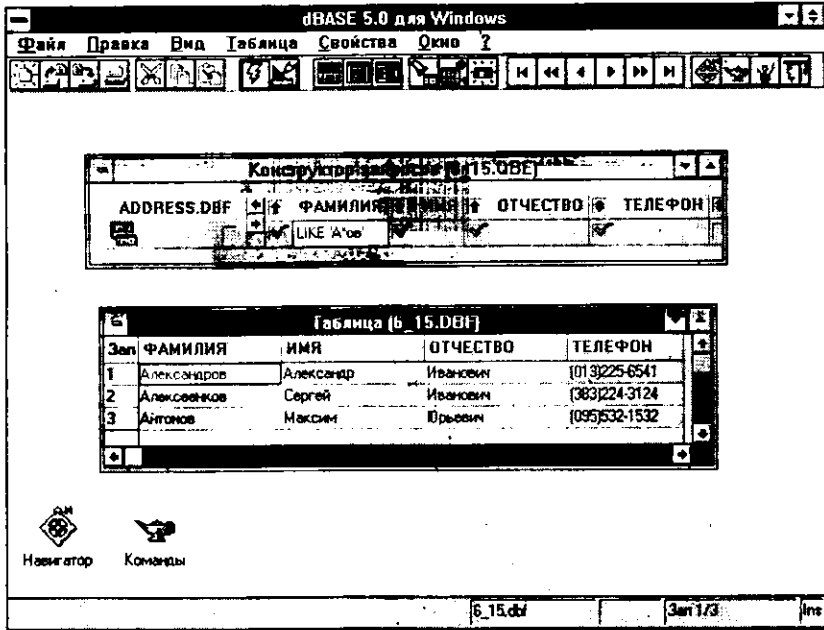


Рис. 6.15. Использование шаблона A*ов.

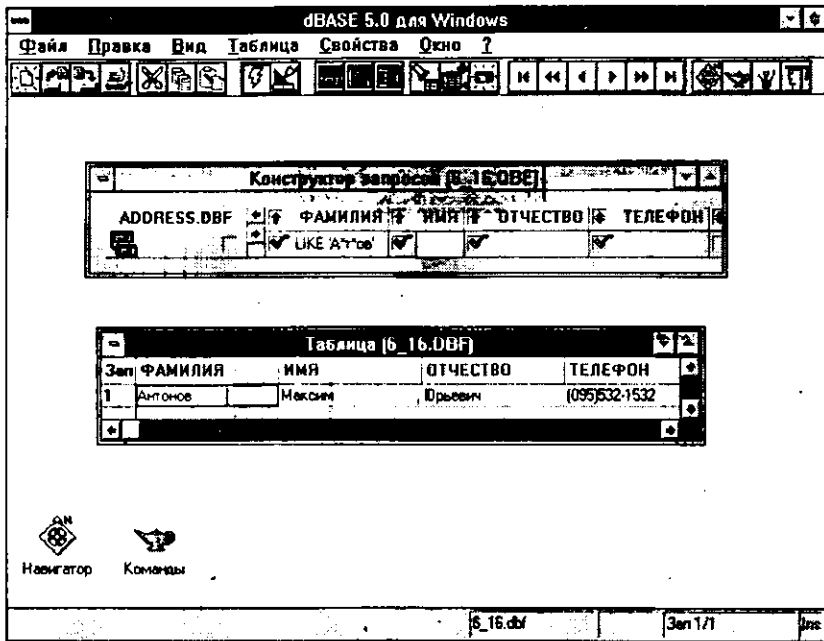


Рис. 6.16. Использование шаблона A*т*ов.

Шаблоны, так же как и оператор \$, могут использоваться для выбора записей с требуемым значением в Мето-поле. В качестве примера на рис. 6.17 представлен результат поиска в таблице ADDRESS по значению Мето-поля с использованием шаблона.

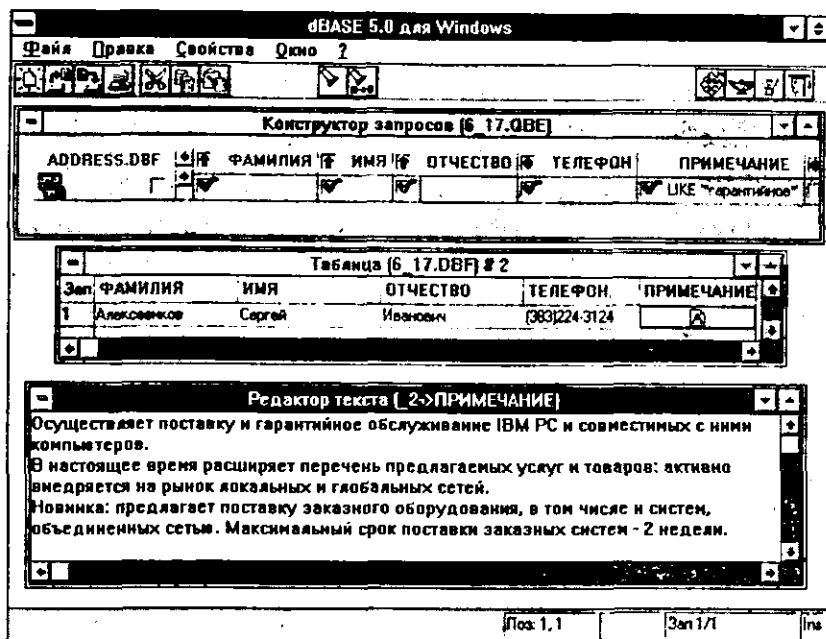


Рис. 6.17. Использование шаблона с Мето-полями.

Оператор шаблона ?

Символ шаблона ? может использоваться для обозначения любого единичного символа, поэтому символ шаблона ? называют *единичным символом шаблона*.

Символ шаблона ? удобно использовать для задания шаблона фиксированной длины. Например, в соответствии с шаблоном И???н (всего пять букв) в таблице ADDRESS будет найдена только одна запись с фамилией Ильин (рис. 6.18). Если вы не уверены в количестве символов, воспользуйтесь символом шаблона *.



При использовании символа шаблона ? dBASE различает заглавные и строчные буквы. Поэтому, результат поиска для двух шаблонов И???н и и???н будет различен.



Если при задании условия выбора для Мето-поля вам потребуется включить символ шаблона ?, он должен быть использован только в сочетании с символом шаблона *.

В рассмотренных примерах шаблоны использовались для поиска текстовых величин. Далее мы покажем, как использовать символ шаблона * при работе с полями дат.

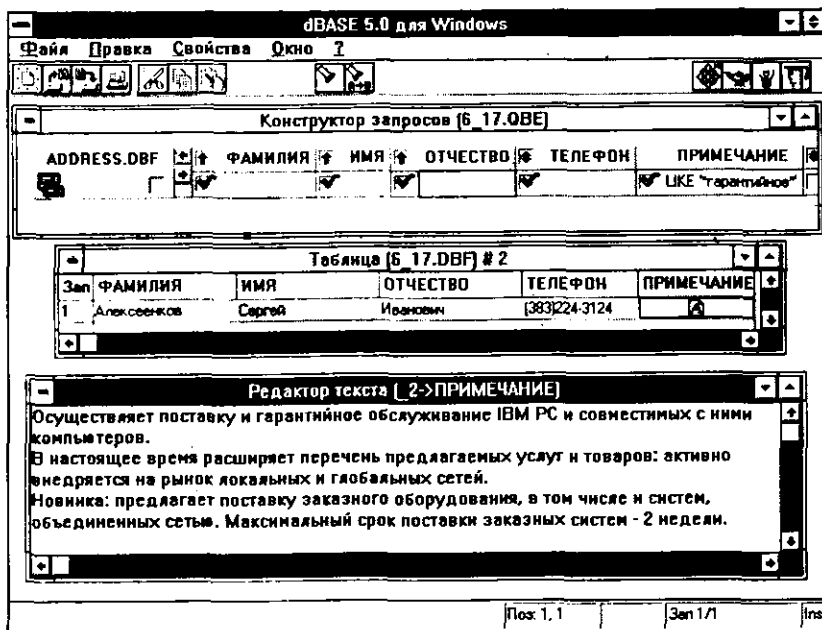


Рис. 6.18. Использование единичного оператора шаблона.

Использование шаблона в полях дат

Оператор LIKE с символами шаблонов применим только к текстовым полям и Мето-полям. Тем не менее вы можете использовать шаблоны для полей типа дат. Для этого вместо оператора LIKE необходимо использовать функцию LIKE в области условий. Функция LIKE содержит два параметра. Первый из них задает шаблон для поиска, а второй — выражение для сравнения. Значение даты для использования в функции LIKE преобразуйте в символьный вид с помощью функции DTOS.



При точном задании значения в качестве условия выбора записи вы могли использовать для ввода даты в поле бланка запроса любой формат. При формировании шаблона для поиска даты необходимо, чтобы ее формат совпадал с установленным форматом отображения дат, который содержится в файле IDAPI.CFG. Если вы воспользуетесь любым другим форматом, результат выборки может быть некорректным.

Для того чтобы определить формат даты, хранящейся в файле IDAPI.CFG, перейдите в Диспетчер Программ Windows, воспользовавшись *Ctrl+Esc* или *Alt+Tab* и запустите программу настройки IDAPI из группы dBASE для Windows. Выберите Даты в меню главного окна IDAPI, в результате чего на экране появится окно диалога, показанное на рис. 6.19. Опции SEPARATOR и MODE содержат информацию об установленном разделителе и последовательности компонентов даты.

После того, как вы получите требуемую информацию, закройте окно диалога, выйдите из программы настройки IDAPI и возвратитесь в dBASE. Подробная информация о настройке системы приведена в приложении II.

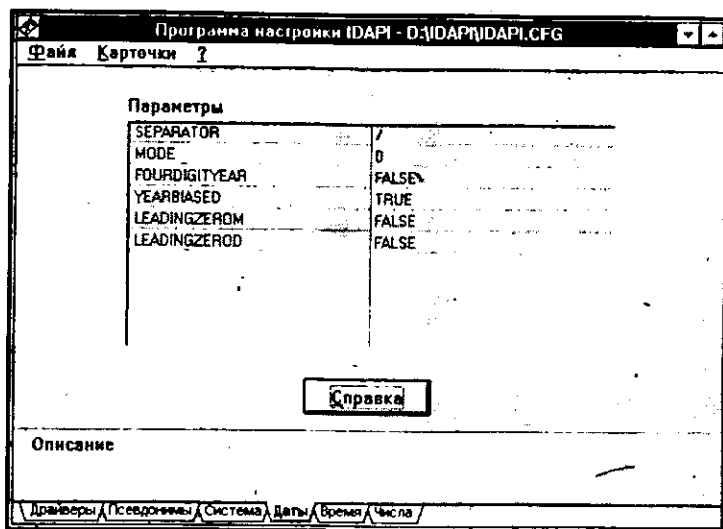


Рис. 6.19. Окно диалога "Настройки".

Пример 6.8. Выбор записей о продажах за определенный месяц

Давайте в качестве примера выберем из таблицы ORDSALE записи о продажах за январь 1995 г.

1. Определите формат представления дат, хранящийся в файле IDAPI.CFG. Предположим, установлено, что текущий формат представления дат дд.мм.гг.
2. Откройте окно запроса для таблицы ORDSALE.
3. Поместите метки в поля, которые должны быть представлены в результирующей выборке.
4. Откройте область условия и напечатайте `LIKE ('??'.01.95', DTOS (DATA_ПРОД))`.

В результате выполнения запроса в результирующей таблице появятся требуемые записи (рис. 6.20).

Итак, мы рассмотрели простейшие случаи запросов по образцу, для которых условия поиска задавались по одному из полей таблицы. Запись из исходной таблицы выбиралась в случае точного или приближенного совпадения ее значения с указанным числом или последовательностью символов. Перейдем к построению более сложных условий выбора. Начнем с рассмотрения критерия выбора записи, основанном на попадании значения одного из полей в указанный диапазон.

Выбор записей по диапазону значений

Предположим, что вам потребовался список клиентов, чей кредит превышает 300 000. В другом случае вам нужно получить сведения о товарах, проданных за первые три месяца 1995 г. В обоих случаях необходимо выбрать из таблицы записи, данные которых попадают в диапазон значений. При этом границы диапазона заданы точно.

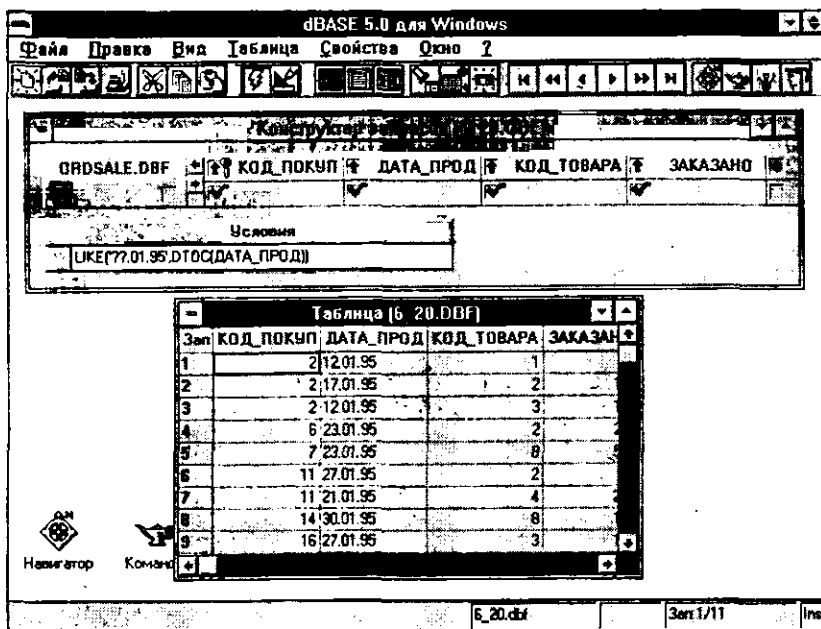


Рис. 6.20. Использование шаблона с полями дат.

Для задания диапазона значений в бланке запроса используются операторы сравнения, которые приведены в табл. 6.2. Поскольку эти операторы предназначены для задания диапазона значений, их называют также *операторами диапазона*.

Табл. 6.2. Операторы сравнения

Оператор	Значение
=	равно
<	больше, чем
>	меньше, чем
>= (или =>)	больше или равно
<= (или =<)	меньше или равно

Операторы сравнения могут использоваться с текстовыми и цифровыми полями, а также полями дат.



При вводе в бланк запроса операторов >= и <= наличие пробела между символами недопустимо. Наличие пробела приведет к тому, что при выполнении запроса dBASE выдаст сообщение об ошибке. Запрос выполнен не будет.

Воспользуемся операторами диапазона для решения задач, условия которых приведены в начале текущего раздела.

Пример 6.9. Выбор записей по диапазону числовых значений

1. Для решения первой задачи вам необходима таблица ADDRESS. Получите на экране окно запроса для этой таблицы и отметьте необходимые вам поля.

2. Далее напечатайте условие поиска >300000 в поле "Кредит".
3. Нажмите на панели инструментов кнопку Выполнить запрос, и вы увидите на экране результирующую таблицу, содержащую требуемые записи (рис. 6.21).

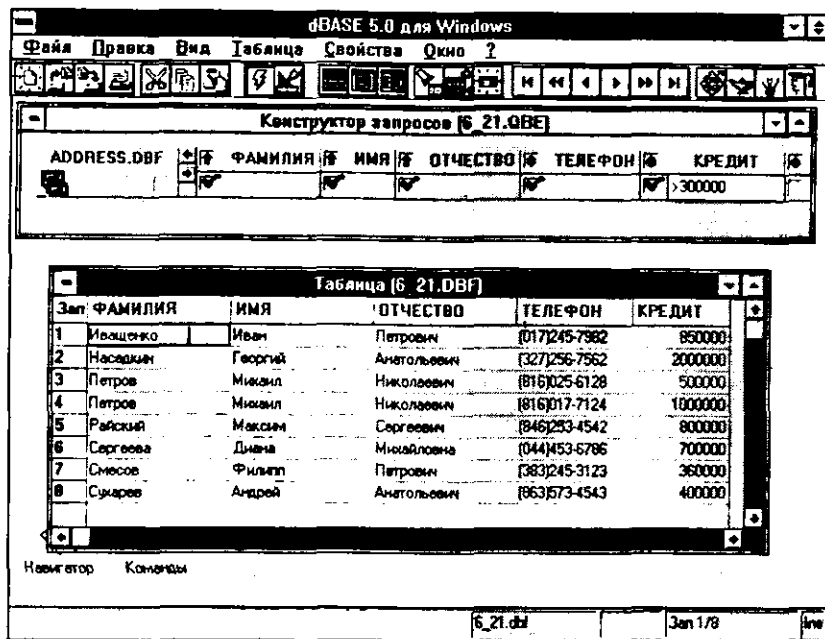


Рис. 6.21. Выбор записей по диапазону значений числового поля.

Если вы хотите, чтобы dBASE также включил в выборку записи о клиентах, чей кредит точно равен 300 000, вы должны в бланке запроса вместо оператора > ввести оператор >=.

В рассмотренной задаче мы задавали только одну нижнюю границу диапазона. Очевидно, что для получения сведений о всех продажах за первые три месяца 1995 г., вам придется задать обе границы (начальную и конечную даты). Оба граничных условия должны быть помещены на одной строке и разделены запятой.



Запятая в числовом поле, за которой следует пробел или любой другой нечисловой символ, кроме точки (например, любой оператор сравнения), считается оператором "И".

Пример 6.10. Выбор записей по диапазону значений дат

1. Информация, необходимая для решения второй задачи, содержится в таблице ORDSALE. Откройте окно запроса для этой таблицы и поместите метки в поля, которые должны присутствовать в результирующей выборке.
2. Для задания интервала дат с 1 января 1995 г. по 31 марта 1995 г. введите условие запроса: >= {01.01.95}, <= {31.03.95} в поле "Дата_прод".
3. Выполните запрос, и вы увидите на экране результирующую таблицу, содержащую информацию о всех продажах за интересующий вас промежуток дат (рис. 6.22).

4. Сохраните созданный запрос.

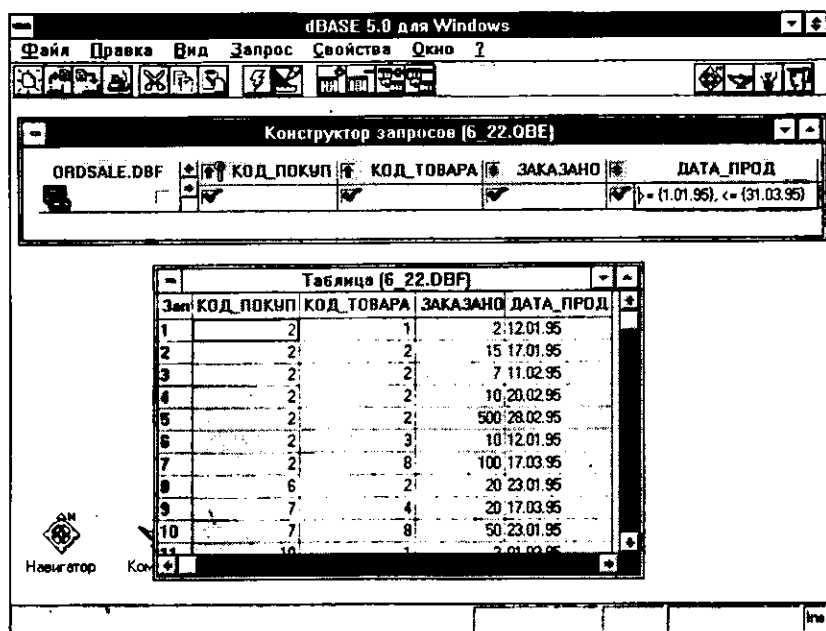


Рис. 6.22. Выбор записей по диапазону значений поля дат.



Обратите внимание на формат даты. В данном случае dBASE позволяет использовать разделитель, отличный от хранящегося в файле IDAPI.CFG, однако требует соблюдения последовательности компонент даты.

Вы можете задать диапазон для значений текстовых полей. В этом случае при выполнении запроса dBASE сравнивает OEM-коды текстовых величин с OEM-кодами границ диапазона. Например, условие выбора $\geq$ П, $<$ Т, помещенное в поле "Фамилия", позволит вам выбрать из таблицы ADDRESS список абонентов, фамилии которых начинаются на П, Р, С.



Использование шаблонов и оператора LIKE при задании диапазонов недопустимо.

Более сложные И/ИЛИ-выражения

При выборе записей за первые три месяца 1995 г. мы использовали условие запроса: $\geq$ {01.01.95}, $\leq$ {31.03.95}. Обратите внимание, что при выполнении запроса выбраны только те записи, для которых одновременно выполнены оба условия. Если запись выбирается только в случае выполнения всех условий, то условие такого выбора называется логическим И, а запрос — И-запросом. Если запись выбирается при выполнении хотя бы одного из всех условий, то условие такого поиска называется логическим ИЛИ, а запрос — ИЛИ-запросом.

Пример 6.11. И-запрос

Для задания И-выражения вы должны просто указать условие выбора для каждого из полей, образующих критерий. При этом все условия должны располагаться в одной строке бланка запроса.

В качестве примера предположим, что из всех записей о продажах за первые три месяца 1995 г. требуется выбрать те, которые содержат данные о товарах типа 2.

1. Воспользуйтесь решением предыдущей задачи. Вызовите запрос, созданный в примере 6.10.
2. Дополните выражение запроса условием 2, поместив его в поле "Код_товара".
3. Выполните запрос, и вы увидите записи, содержащие сведения о товарах типа 2, проданных за указанный период дат (рис. 6.23).

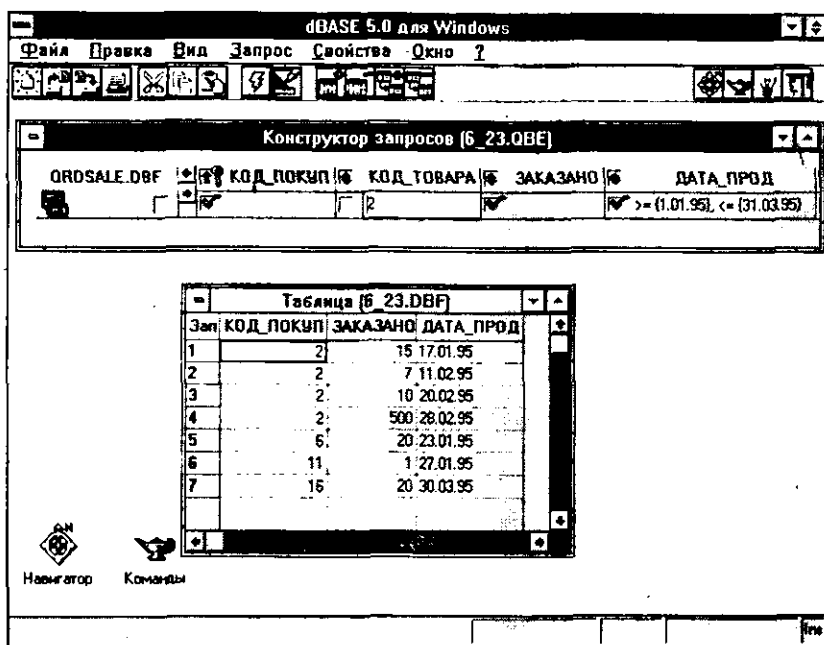


Рис. 6.23. Условие логического И.

dBASE формально относится к заданному вами И-выражению. Если вы попытаетесь выбрать из таблицы ADDRESS записи об абонентах из Минска и Киева, воспользовавшись для этого бланком запроса, представленным на рис. 6.24, то dBASE не найдет ни одной записи. Ни одной записи об абонентах из Минска и Киева! Очевидно, что абонент не может проживать в двух городах одновременно, поэтому данный запрос лишен смысла.

Пример 6.12. ИЛИ-запрос

При задании ИЛИ-выражения каждое из условий выбора, образующих критерий, должно располагаться на отдельной строке бланка запроса. Оказывается, что для выбора записей о абонентах из Минска и Киева условия нужно просто расположить на разных строках!

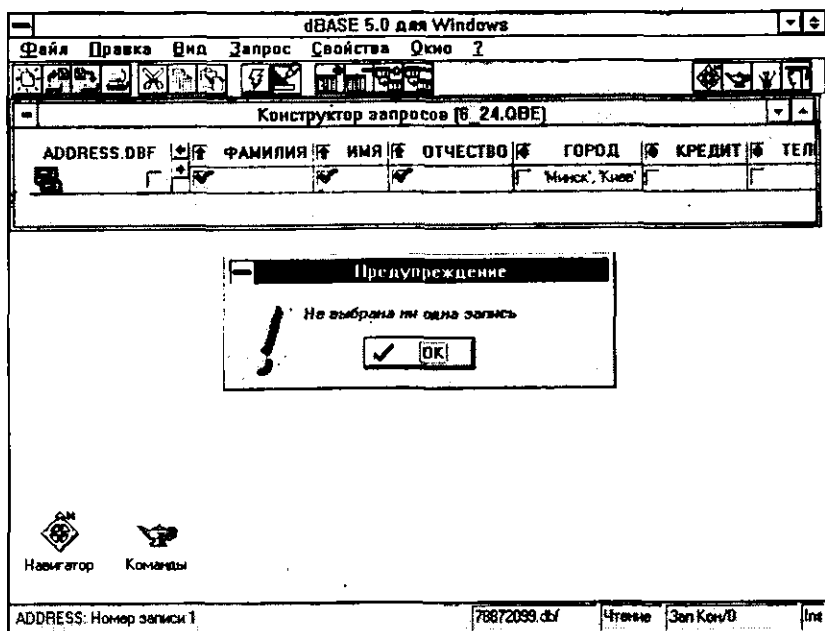


Рис. 6.24. Некорректное использование логического И.

1. Откройте окно запроса для таблицы ADDRESS.
2. Пометьте поля, которые должны присутствовать в результате запроса.
3. Введите в поле “Город” критерий поиска: 'Минск'.
4. Нажмите клавишу ↓, и курсор переместится во вторую строку бланка запроса.
5. Введите во второй строке в поле “Город” условие: 'Киев' и выполните запрос, нажав F2 или кнопку **Выполнить** запрос.

Сформированный запрос и результирующая таблица показаны на рис. 6.25. Как видите, требуемые записи найдены.



При задании ИЛИ-выражения вы должны в каждой строке бланка запроса пометить одни и те же поля. При этом условия поиска могут быть расположены в различных полях.

Пример 6.13. Формирование сложного ИЛИ-выражения.

При формировании сложного ИЛИ-выражения вы можете расположить условия выбора для различных полей в разных строках бланка запроса. Например, вам требуется выбрать записи о абонентах, проживающих на Украине или имеющих кредит свыше 300 000.

1. Откройте окно запроса для таблицы ADDRESS.
2. Пометьте поля, которые должны быть отображены в результирующей таблице.

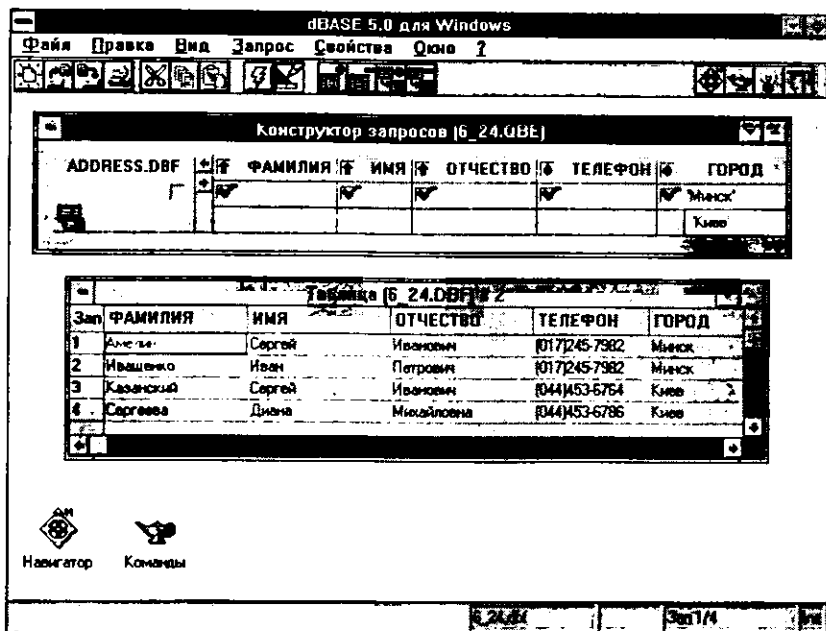


Рис. 6.25. Условие логического ИЛИ.

3. Сформируйте ИЛИ-выражение, для чего в поле "Страна" первой строки введите 'Украина' и в поле "Кредит" второй строки введите > 300000.

На рис. 6.26 показаны запрос и результирующая таблица.

Комбинация И/ИЛИ-запросов

Обратимся к примерам, приведенным на рис. 6.23 и 6.25. На их основе мы можем сделать два очень важных вывода:



Условия, образующие И-выражение, располагаются в одной строке бланка запроса и собираются по условию логического И.



Условия, образующие ИЛИ-выражение, расположены в различных строках бланка запроса и собираются по условию логического ИЛИ.

Сбор условий в пределах одной строки и между строками позволяет вам формировать сложные критерии выбора записей из таблицы. В качестве иллюстрации использования И/ИЛИ-выражений для выбора записей рассмотрим два более сложных примера.

Пример 6.14. Первая стадия формирования И/ИЛИ-выражения

Предположим, что ваше предприятие переходит к выпуску нового продукта, для изготовления которого необходимы компоненты типов 1 и 2. В вашем распоряжении находится таблица СОМРСОМР, содержащая информацию о предприятиях, производящих эти компоненты.

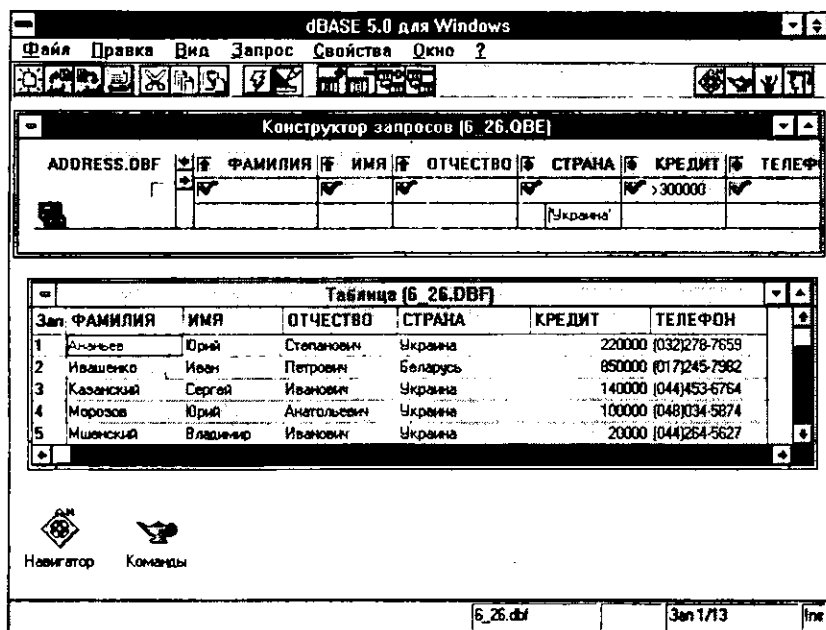


Рис. 6.26. Второй пример использования логического ИЛИ.

Для начала вам нужно выяснить, какие из предприятий производят компонент типа 1. Кроме того, потребности в компоненте этого типа для установочной партии составляют 100—120 штук. Желательно, чтобы цена единицы продукции не превышала 1000. Таким образом, вам требуется найти предприятие, которое одновременно удовлетворяет трем следующим условиям:

1. Предприятие должно производить компонент типа 1.
2. Минимальный размер партии должен быть не более 120 штук.
3. Цена компонента типа 1 должна быть не более 1000 за штуку.

В этой задаче критерий выбора записей из таблицы СОМРСОМР образуется тремя условиями для различных полей, которые собираются по условию логического И.

1. Откройте окно запроса для таблицы СОМРСОМР.
2. Поместите метки в поля "Код_предпр", "Код_товара", "Цена", "Мин_партия".
3. Введите условие выбора:
 <= 1000 ("Цена");
 <= 120 ("Мин_партия");
 1 ("Код_товара").
4. Сохраните созданный запрос.

Формирование И-выражения завершено. Выполните запрос, и вы увидите, что вашим требованиям отвечают два предприятия (рис. 6.27).

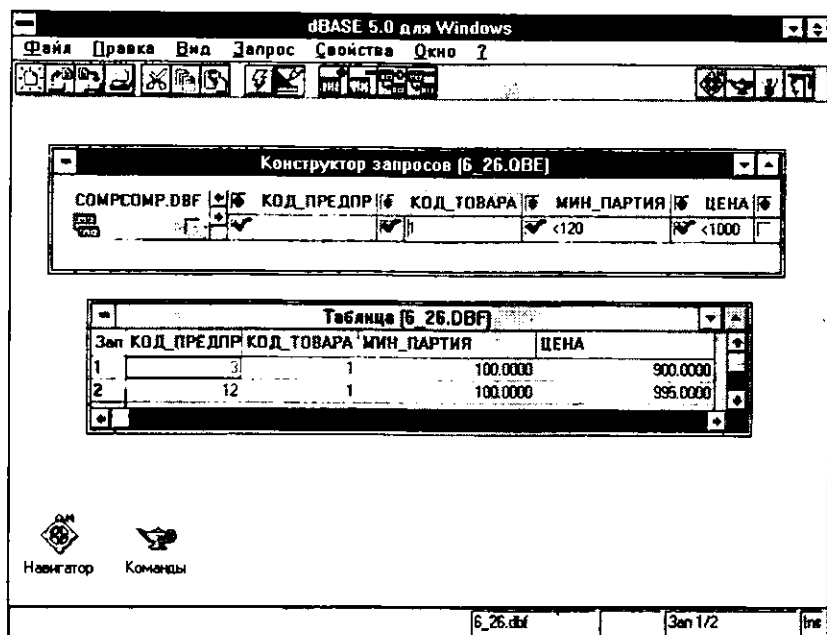


Рис. 6.27. Критерий логического И.

Пример 6.15. Вторая стадия формирования И/ИЛИ-выражения.

Перейдем к выбору предприятия, которое может поставить компонент типа 2. Для производства необходимо 80 штук компонентов этого типа. Кроме того, вам известно, что вместо компонента типа 2 в качестве альтернативного варианта можно использовать компонент типа 3. Однако компонентов типа 3 требуется в два раза больше по сравнению с компонентом типа 2. В зависимости от количества компонентов каждого типа вы определяете цену, которую можете заплатить за компонент каждого типа. Цена компонента типа 2 должна быть не более 200 за штуку, а компонента типа 3 — не более 100. При решении этой задачи для поиска данных вам потребуется сформировать И/ИЛИ-выражение.

Поскольку вас может удовлетворить как компонент типа 2, так и типа 3, выбор между двумя компонентами осуществляется по условию логического ИЛИ. Выражение ИЛИ должно располагаться на двух различных строках бланка запроса. Условие выбора для компонента каждого типа собирается по условию логического И, которое должно располагаться в одной строке бланка запроса.

Теперь вам нужно задать условия по двум компонентам.

1. Откройте запрос, сформированный в примере 6.14.
2. Отредактируйте первую строку. Введите в поля следующие значения:
 <= 200 ("Цена");
 <= 80 ("Мин_партия");
 2 ("Код_товара").
3. Нажмите клавишу ↓ и введите значения:
 <= 100 ("Цена");
 <= 160 ("Мин_партия");
 3 ("Код_товара").

Выполните запрос, нажав кнопку **Выполнить запрос** на панели инструментов или клавишу **F2**, и вы увидите, что в таблице **COMPCOMP** обнаружены три записи, удовлетворяющие вашим требованиям (рис. 6.28).

Будьте внимательны в своих логических построениях, и вы сможете получить информацию в самых запутанных ситуациях.

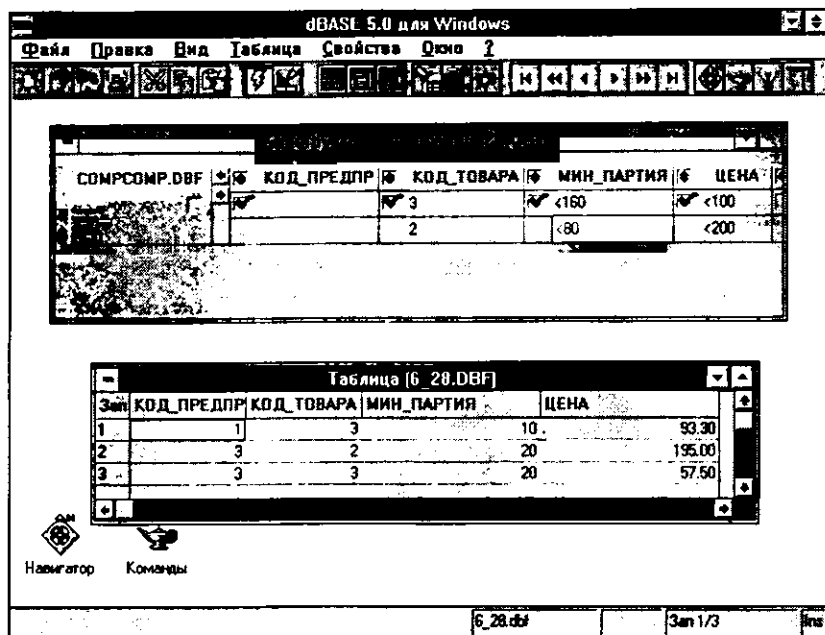


Рис. 6.28. Комбинация И/ИЛИ-запросов.

Операторы .OR. и .AND.

В области условия вы можете использовать операторы **.OR.** и **.AND.**, которые позволяют вам формировать в одной строке сложное условие выборки. Например, при поиске записей о абонентах из Минска и Киева вы можете поместить ИЛИ-выражение в одной строке (рис. 6.29). Сравнив результирующие таблицы, представленные на рис. 6.25 и 6.29, отметьте, что dBASE выбрал из таблицы **ADDRESS** одни и те же записи.



В области условия вы можете формировать любое допустимое в dBASE логическое условие, которое может содержать функции, операторы сравнения, **.OR.**, **.AND.**, **.NOT.** и скобки для изменения порядка выполнения выражения.

Заключение

- Для выбора данных из одной или нескольких таблиц используется запрос по образцу. При формировании запроса вам необходимо указать критерии поиска записей в исходной таблице и определить вид результирующей таблицы. При этом вместо того, чтобы печатать предложения на специальном языке, вы должны просто заполнить соответствующие поля в окне запроса.

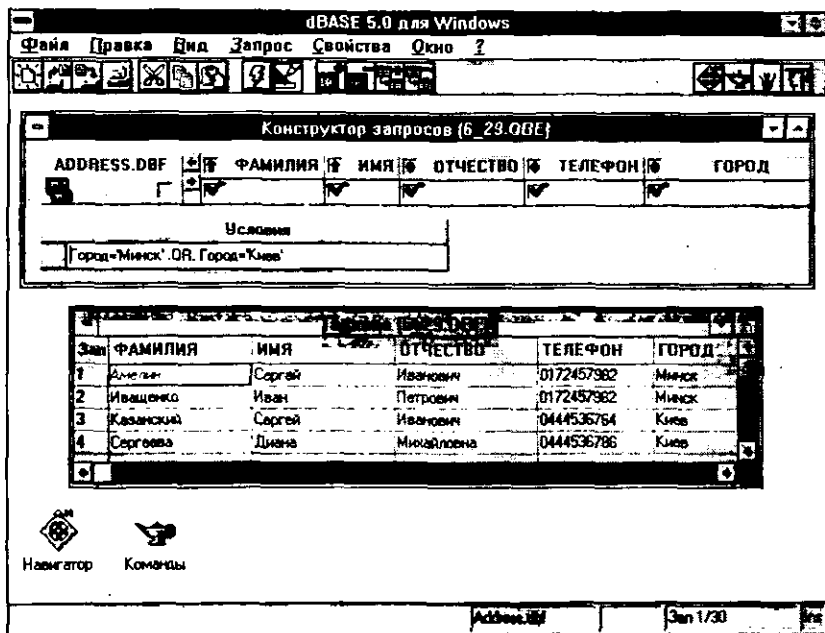


Рис. 6.29. Использование оператора .OR.

- Для отображения поля в результирующей таблице поместите в него метку (✓). Данные в этом поле будут упорядочены в порядке возрастания значений.
- Для определения типа сортировки в поле выберите соответствующую стрелку для кнопки упорядочивания.
- Для задания условия точного совпадения значений напечатайте его в соответствующем поле бланка запроса.
- Для задания условия точного несовпадения значений используйте оператор <> (Не совпадает с).
- Для задания условия, основанного на сегодняшней дате, воспользуйтесь функцией DATE (Текущая дата).
- Для выбора данных из пустых полей используйте функцию EMPTY или ISBLANK (Пустое значение).
- Для задания условия неточного совпадения значений используйте операторы \$ и LIKE (Похож на).
- Для задания диапазона значений в условии поиска воспользуйтесь операторами диапазона, приведенными в табл. 6.2.
- При формировании выражения И все условия располагаются в одной строке. Если поле содержит два условия, то их значения разделяются запятой.
- При формировании выражения ИЛИ все условия располагаются на разных строках.

Глава 7. Дополнительные возможности запросов по образцам

В настоящей главе мы продолжим изучение возможностей, предоставляемых в ваше распоряжение “запросами по образцу” для обработки информации, содержащейся в одной таблице. Мы покажем, как выполнять вычисления над полями различных типов. Кроме того, вы познакомитесь с использованием запросов для редактирования данных в исходной таблице. Обработка информации из связанных таблиц будет рассмотрена во второй части книги.

Использование вычисляемых полей

dBASE позволяет производить вычисления над полями каждой записи исходной таблицы. При необходимости вы можете задавать любые допустимые в dBASE выражения, которые могут включать имена полей и переменных, функции, арифметические и логические операторы и операторы сравнения. Например, таблица ORDSALE содержит данные о количестве проданного товара каждого типа (поле “Продано”) и цене (поле “Цена”). На основании этой информации вы можете вычислить стоимость каждой партии проданных товаров и отобразить результат в результирующей таблице.

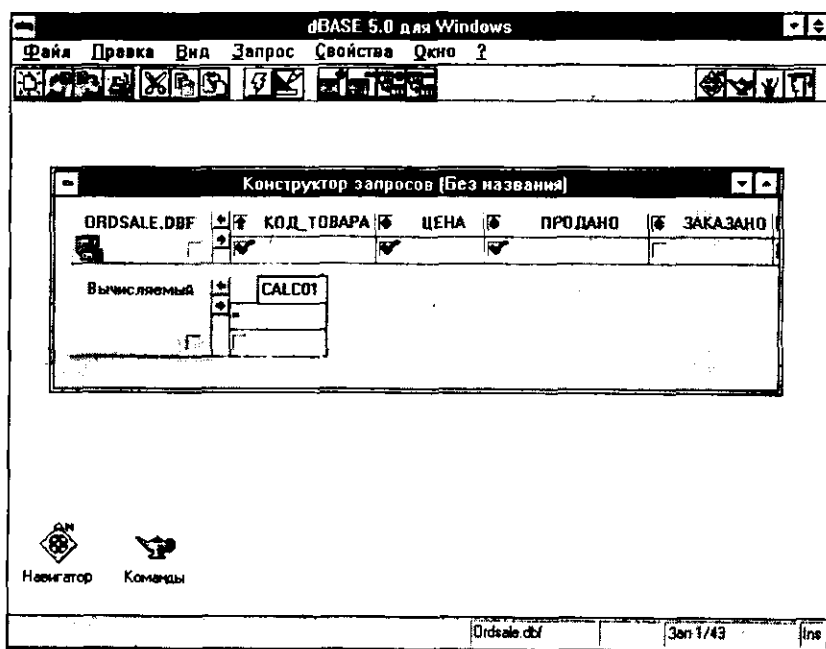


Рис. 7.1. Создание вычисляемого поля.

Для создания или добавления вычисляемого поля выполните команду Запрос | Создать вычисляемое поле. Если вы создаете первое вычисляемое поле в данном запросе, то появляется бланк пустого вычисляемого поля (рис. 7.1). Если же вычисляемые поля уже присутствуют в запросе, бланк нового поля будет добавлен правее существующих. Этот бланк содержит три текстовых поля, в которых вы можете ввести заголовок поля, выражение для вычисления значения поля и условие выборки по

значению данного поля. Введенный вами заголовок поля будет появляться при отображении результатов выборки. Для отображения вычисляемого поля в результирующей таблице установите метку в этом поле. Если вычисляемое поле используется только для ввода условия выборки, вы можете снять метку с данного поля и этим запретить вывод его в результирующую таблицу.

Выражение для вычисления

Выражение для вычисления содержит элементы формулы, которые связаны арифметическими и логическими операторами и операторами сравнения. В качестве элементов формулы могут использоваться:

- Имена полей
- Константы
- Функции
- Переменные
- Элементы массива

Результат вычисления выражения должен относиться к одному типу данных (текстовому, дате, логическому или числовому). Выражения можно строить на основе данных разного типа, но все они должны преобразовываться к одному типу данных с помощью функций преобразования данных. При работе с операторами сравнения слева и справа от оператора должны находиться данные одного типа, при этом оператор сравнения всегда преобразует выражение в логический тип.



Выражение может состоять из одного элемента.

В выражение могут входить арифметические и логические операторы, а также операторы сравнения. Ранее вы уже познакомились с логическими операторами и операторами сравнения. Ниже приведены арифметические операторы с указанием допустимых типов полей:

Оператор	Функция	Допустимый тип поля
+	Сложение	C, N, D
-	Вычитание	N, D
*	Умножение	N
/	Деление	N

где, C - символьный тип, N - числовой тип, D - тип дат.

Для изменения порядка вычислений и группировки данных в выражениях используются круглые скобки. В выражениях без круглых скобок умножение и деление выполняются раньше, чем сложение и вычитание. Операции с равным приоритетом выполняются слева направо.

Чтобы определить вычисление, выполняемое в вычисляемом поле, выберите поле ввода выражения, которое находится правее знака равенства и введите выражение в это поле.

Условие выборки для вычисляемого поля

Для каждого вычисляемого поля вы можете задать условие выборки, которое выполняет те же функции, что и условие выборки для обычных полей, т.е. определяет, какие записи должны помещаться в результирующую таблицу при выполнении запроса.

Условие выборки вводится в поле условия, которое находится сразу под вычисляемым полем. Для формирования выражения И вы можете использовать запятые, а для выражения ИЛИ — добавить новую строку.

Заголовок вычисляемого поля

По умолчанию dBASE для каждого вновь создаваемого вычисляемого поля автоматически задает заголовок поля (например, 'CALC01'). Но вы можете изменить этот заголовок и ввести текст, более соответствующий содержанию вычисляемого поля (например, 'ИТОГО').

Для изменения заголовка перейдите на текстовое поле, содержащее заголовок вычисляемого поля, которое находится над полем ввода вычисляемого выражения, и отредактируйте текущий заголовок.

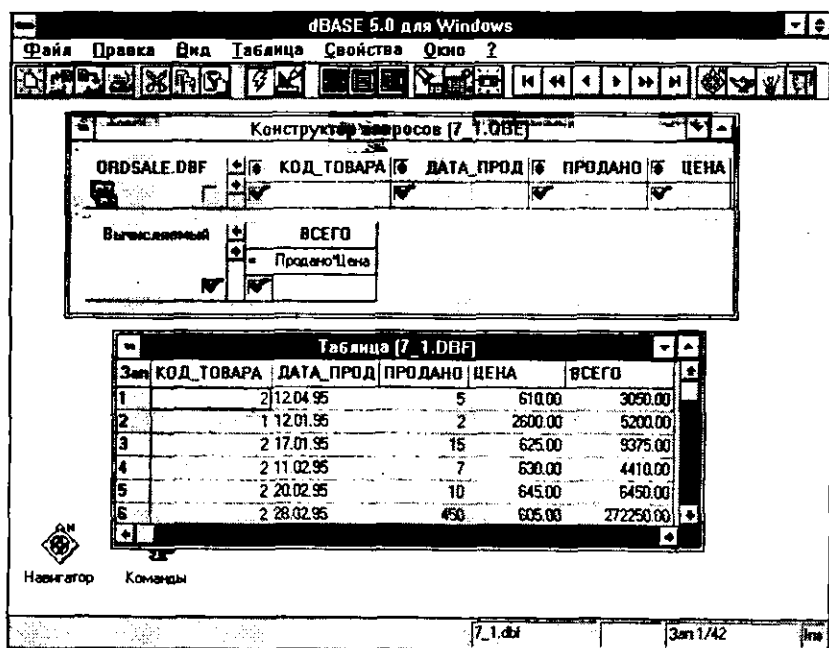


Рис. 7.2. Вычисление над числовыми полями.

Пример 7.1. Вычисления над числовыми полями

Для создания вычисляемого поля обратимся к таблице ORDSALE. Данная таблица содержит информацию о количестве товаров каждого типа, проданных в одной партии (поле "Продано"), и цене единицы изделия (поле "Цена"). Данные об общей стоимости товаров, проданных в каждой партии, в исходной таблице отсутствуют. Если вам потребуются эти данные, вы можете получить их путем перемножения значений полей "Продано" и "Цена" по каждой записи.

1. Откройте окно запроса для таблицы ORDSALE.
2. Поместите метки в поля “Код товара”, “Дата продажи”, “Продано” и “Цена”.
3. Далее выполните команду Запрос | Создать вычисляемое поле и в бланке запроса появится бланк вычисляемого поля. Введите в этом бланке вычисляемое выражение ПРОДАНО*ЦЕНА.
4. По умолчанию заголовок вычисляемого поля равен ‘CALC01’. Удалите этот заголовок и замените его на более смысловой— ‘ВСЕГО’.
5. Установите метку в вычисляемом поле для его отображения в результирующей таблице.

На рис. 7.2 показаны заполненный бланк запроса и результирующая таблица, в которой появилось поле с заголовком “Всего”, содержащее результаты вычислений.

Пример 7.2. Выбор записей из таблицы по результату вычислений

Вы можете использовать вычисляемые поля при формировании предложения запроса для выбора записей из таблицы по результату вычислений над полями исходной таблицы. Допустим, вы хотите получить информацию о продажах тех партий товаров, стоимость которых превышает 10 000.

1. Откройте созданный в предыдущем примере запрос.
2. Перейдите в поле ввода условия бланка вычисляемого поля и напечатайте условие выбора: >10000.
3. Выполните запрос и вы увидите на экране результат, показанный на рис. 7.3.

Конструктор запросов [7_1.DBF]

ORDSALE.DBF	КОД ТОВАРА	ДАТА ПРОД	ПРОДАНО	ЦЕНА
Вычисляемый				ВСЕГО
				Продано*Цена
				>10000

Таблица [7_3.DBF]

Зам	КОД ТОВАРА	ДАТА ПРОД	ПРОДАНО	ЦЕНА	ВСЕГО
1	2	28.02.95	450	605.00	272250.00
2	2	23.01.95	20	620.00	12400.00
3	1	31.03.95	5	2675.00	13375.00
4	3	27.01.95	15	1250.00	18750.00
5	7	28.02.95	10	1025.00	10250.00
6	1	31.01.95	10	2400.00	24000.00
7	1	01.03.95	10	2380.00	23800.00
8	7	27.02.95	20	950.00	19000.00

Рис. 7.3. Выбор записей по результатам вычислений.

Пример 7.3. Вычисления над текстовыми полями.

Вы можете использовать вычисления для объединения двух или более текстовых полей. Например, в таблице ADDRESS данные об адресах клиентов, проживающих в областных центрах, представлены в трех отдельных полях таблицы: "Страна", "Город", "Адрес", что затрудняет их просмотр. Объединим данные из этих трех полей и поместим их во вновь созданное поле. Кроме адреса клиента поместим в это же поле данные из поля "Индекс".

1. Откройте окно запроса для таблицы ADDRESS.
2. Введите в поле "Адрес" формулу для вычисления:

`RTRIM(Индекс)+' ', '+RTRIM(Страна) +', '+RTRIM(Город) +', '+Адрес`

В соответствии со введенной формулой dBASE создаст новое поле в результирующей таблице, в которое поместит данные из всех перечисленных полей, разделив их запятыми и пробелами. Функция RTRIM используется для удаления всех правых пробелов указанных полей. При отсутствии в формуле членов, заключенных в кавычки, разделители в результирующем поле будут также отсутствовать. Такой вид ухудшит читабельность результата.

3. Аналогичным образом объедините в одном столбце данные из полей "Фамилия", "Имя" и "Отчество".

Заполненный бланк запроса и результирующая таблица показаны на рис. 7.4.

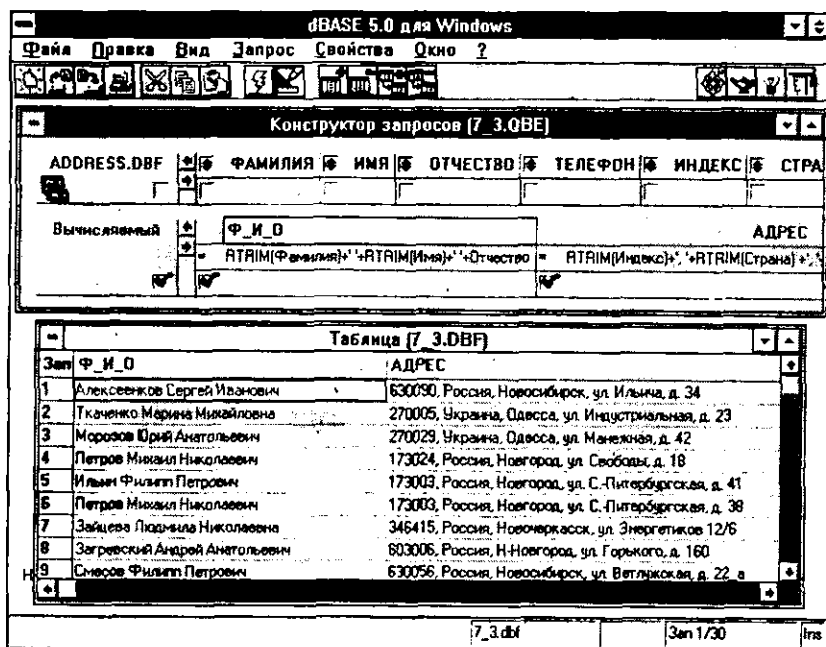


Рис. 7.4. Объединение текстовых полей.

Пример 7.4. Вычисления над полями дат

Формула для вычисления над полями дат может содержать константы (даты или числа), функции даты, а также операторы.

Предположим, что нам требуются сведения о продажах за последние 300 дней товара типа 2.

1. Откройте окно запроса для таблицы ORDSALE.
2. Поместьте поля, которые должны присутствовать в результирующей таблице.
3. Введите условия выбора
2 (поле "Код товара");
>=DATE()-300 (поле "Дата продажи")

Сформированный запрос и результат его выполнения представлены на рис. 7.5.

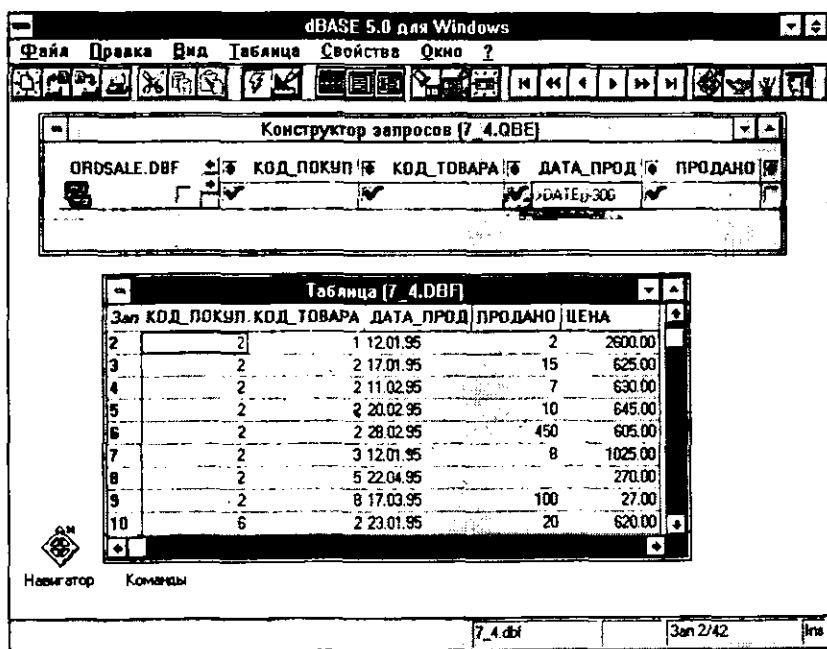


Рис. 7.5. Использование функции DATE (Текущая дата) для определения условий выбора.

Результат вычисления над пустыми полями

Если одно из полей, участвующих в вычислении, окажется пустым, результат будет зависеть от типа поля.

dBASE воспринимает пустые числовые поля как нули. Поэтому результат умножения пустого числового значения на любое число будет равен нулю. В некоторых случаях это может привести к неверным результатам. Например, если в столбце "Цена" не введена цена товара, то стоимость опущенного товара будет считаться равным нулю, что совершенно неправильно.

При вычислениях над полями дат, одно из которых является пустым, результат всегда будет пустым полем. При сравнении пустой даты с любой другой датой считается, что пустая дата более поздняя.

При вычислениях над текстовыми полями в результирующем поле пустым будет только его часть, соответствующая исходному пустому полю.

Для анализа пустых значений вы можете воспользоваться функцией ISBLANK, которая использует в качестве параметра значение поля или переменной и возвращает (.T.) если значение пустое.

Пример 7.5. Использование условий выбора при вычислениях

В некоторых случаях формула, по которой выполняются вычисления над полями исходной таблицы, может зависеть от значения данных в этих полях. Например, при продаже партии товаров менее 10 штук ее стоимость определяется выражением:

Продано * Цена

А стоимость партии товаров более 10 штук определяется выражением:

$10 * \text{Цена} + 0.9 * (\text{"Продано"} - 10) * \text{Цена}$

Если вам требуется определить стоимость каждой проданной партии товаров, вы должны использовать одну из приведенных формул в зависимости от размера партии. Очевидно, что выбор формулы для вычисления должен производиться на основании условия выбора.

Для выполнения таких вычислений удобно использовать функцию IIF, которая анализирует заданное условие и в зависимости от результата возвращает одно из двух возможных значений.

Синтаксис функции:

IIF(<логическоеВыражение>, <Выражение1>, <Выражение2>)

Выражение1 возвращается при выполнении условия, заданного аргументом логическоеВыражение. В противном случае возвращается **Выражение2**.

Необходимая информация для определения стоимости проданных партий товаров содержится в таблице ORDSALE. Количество товаров в партии может быть больше, меньше или равно 10 штукам. При определении стоимости партии товаров в 10 единиц вы можете воспользоваться любой из формул.

1. Откройте окно запроса для таблицы ORDSALE.
2. Создайте вычисляемое поле и напечатайте выражения для вычисления значения поля:

IIF(Продано<=10, Продано*Цена, 10 * Цена + 0.9 * (Продано - 10) * Цена)

Заполненный бланк запроса и результирующая таблица показаны на рис. 7.6.

Редактирование записей с помощью результирующей таблицы

В процессе выполнения запроса dBASE выделяет из исходной таблицы подмножество записей, удовлетворяющее заданным условиям. Если вы внесете изменения в результирующую таблицу, то все изменения автоматически переносятся в исходную таблицу. Таким образом, вы можете выделить требуемые записи в результирующую

таблицу, после чего отредактировать их. Это весьма удобное средство для редактирования таблиц.

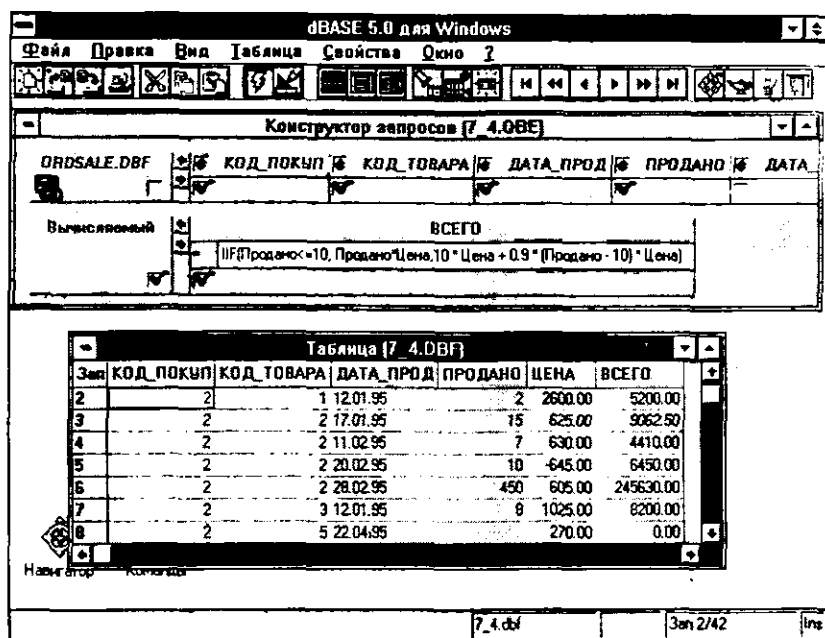


Рис. 7.6. Стоимость товаров, определяемая от размеров проданной партии.

Пример 7.6.

Допустим, вам требуется обновить некоторые из записей таблицы GOODS, относящиеся к категории 2 (вычислительная техника).

1. Откройте окно запроса для таблицы GOODS.
2. Пометьте все поля.
3. Введите в поле "Категория" условие отбора записей: 2.
4. Выполните запрос. На экране появится результат выполнения запроса.
5. Измените значение в одной из записей. При выходе из нее dBASE обновит значения в исходной таблице (GOODS).

Результат ваших действий представлен на рис. 7.7.

Заключение

- При выполнении вычислений над различными полями таблицы создайте вычисляемое поле с помощью команды **Запрос | Создать вычисляемое поле**. Для вычисляемого поля вы можете задать выражение для вычисления, определить заголовок поля и условие выборки.

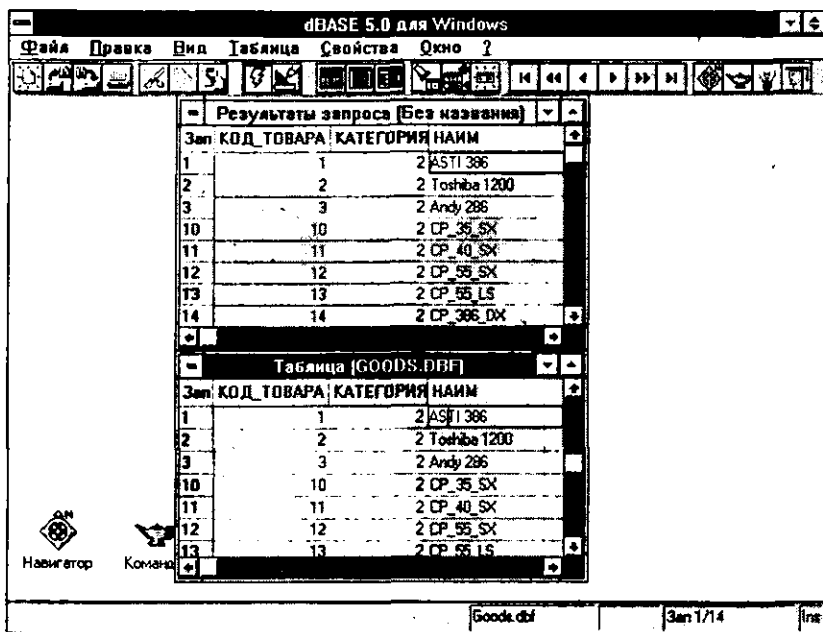


Рис. 7.7. Использование результирующей таблицы для редактирования записей.

- Выражение для вычисления вводится в поле бланка вычисляемого поля, которое следует сразу за знаком равенства.
- Для изменения заголовка вычисляемого поля перейдите на поле заголовка бланка вычисляемого поля и отредактируйте имя заголовка, задаваемое по умолчанию.
- dBASE позволяет создавать условия для вычисляемого поля. Для ввода условия перейдите на поле, расположенное ниже поля ввода выражения, и введите требуемое условие выборки на основе результатов вычисления.
- Для редактирования исходной таблицы вы можете использовать результирующую таблицу. Внесенные вами изменения автоматически будут перенесены в исходную таблицу.

Глава 8. Создание форм

В настоящей главе мы обсудим формы. Как вы уже знаете, в dBASE существует несколько режимов отображения содержимого таблицы — режим таблицы, бланка, перечня и в виде формы. В главе 3 вы познакомились с первыми тремя видами отображения содержимого таблицы, которые автоматически создаются dBASE для любой таблицы. Кроме этих видов вы можете создать и использовать формы. Формы являются мощным и гибким средством представления информации. Например, если вы разрабатываете законченное приложение по бухгалтерии, вы можете создать ряд форм, которые будут выглядеть на экране точно так же, как стандартные бланки бухгалтерских документов.

На рис. 8.1 показано использование формы для просмотра содержимого таблицы ADDRESS. Данная форма позволяет объединить поля в группы по определенным признакам, что облегчает восприятие информации.

Конструктор форм позволяет вам разрабатывать собственные формы для представления, ввода и редактирования данных. С помощью таких форм вы сможете отобразить через форму:

- несколько записей из одной таблицы;
- информацию из нескольких таблиц;
- значения полей, которые получены в результате вычислений над полями исходной таблицы и т.д.

The screenshot shows the dBASE 5.0 for Windows interface with a form titled "Форма" (Form) for the "ADDRESS" table. The form is titled "СПИСОК КЛИЕНТОВ" (Client List) and displays data for a client named Alexeev, Sergey Ivanovich, born 08.04.1964, with a phone number 3832243124 and a credit of 100001. Below this, the "АДРЕС" (Address) section shows the client's location in Novosibirsk. At the bottom, the "ПРИМЕЧАНИЕ" (Note) section contains text about IBM PC service and equipment supply. The form is organized into sections with labels like "Фамилия", "Имя", "Отчество", "Предпр", "АО Интерсервис", "Телефон", "Дата_рожд", "Кредит", "АДРЕС", and "ПРИМЕЧАНИЕ". The status bar at the bottom indicates "Address.dbf" and "3 of 1/30" records.

СПИСОК КЛИЕНТОВ			
Фамилия	Предпр	АО Интерсервис	3832243124
Имя	Алексеев	Телефон	08.04.1964
Отчество	Сергей	Дата_рожд	100001
	Иванович	Кредит	
АДРЕС			
Индекс	630090		
Страна	Россия	Область	
Город	Новосибирск		
Адрес	ул. Ильича, д. 34		
ПРИМЕЧАНИЕ			
Осуществляет поставку и гарантийное обслуживание IBM PC и со			
В настоящее время расширяет перечень предлагаемых услуг и то			
Новинка: предлагает поставку заказного оборудования, в том чис			

Рис. 8.1. Форма для просмотра данных.

В текущей главе мы обсудим основные средства создания форм для одной таблицы. В главе 14 мы продолжим рассмотрение вопросов, связанных с созданием и использованием форм. Там вы узнаете как создавать многотабличные и многостраничные формы.

Создание форм с помощью экспертов

Простейший способ создать форму заключается в использовании экспертов. В настоящей главе мы остановимся на использовании эксперта для создания однотабличной формы. Однако, в полной мере, вы сможете по достоинству оценить это средство при разработке форм для связанных таблиц.

Выполните команду **Файл | Создать | Форма** и на экране откроется окно диалога “Эксперт по разработке форм” (рис. 8.1). Если вы решили прибегнуть к помощи эксперта, в этом окне диалога установите переключатель **Помощь эксперта** и нажмите кнопку **Далее**. Кроме этого вы можете выбрать кнопку **Эксперт** на панели инструментов, в результате чего dBASE предложит вам список доступных экспертов. В версии dBASE 5.0 вам будет предложен только один эксперт, но в версии 5.5 уже имеется множество разнообразных экспертов для создания форм и отчетов.

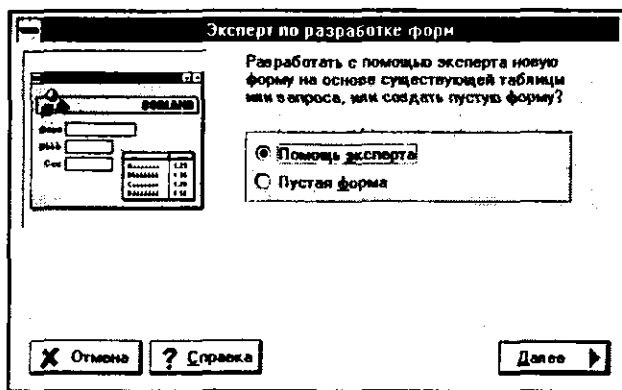


Рис. 8.2. Окно диалога “Эксперт по разработке форм”.

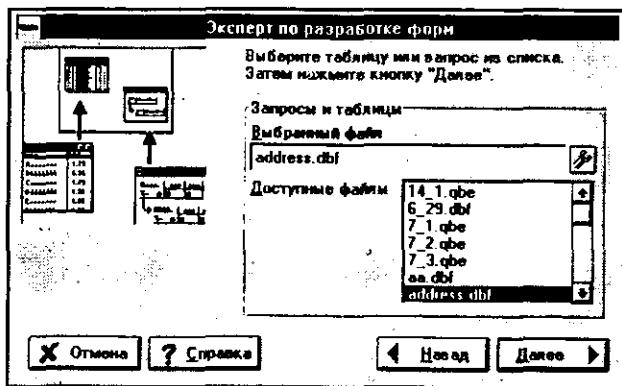


Рис. 8.3. Выбор таблицы для создания формы.

Пример 8.1.

Давайте воспользуемся услугами эксперта для создания собственной формы.

1. Выполните команду **Файл | Создать | Форма**.

- В окне диалога “Эксперт по разработке форм” установите переключатель Помощь эксперта и нажмите кнопку Далее. В результате будет вызван эксперт по разработке форм (рис. 8.3).
- Укажите таблицу, для которой предполагается создать форму (ADDRESS). Если вы не помните имя таблицы или если она расположена в каталоге, отличном от текущего, нажмите на кнопку Инструмент и выберите требуемую таблицу в окне диалога “Поиск” (рис. 8.4). Для перехода к следующему шагу нажмите кнопку Далее.

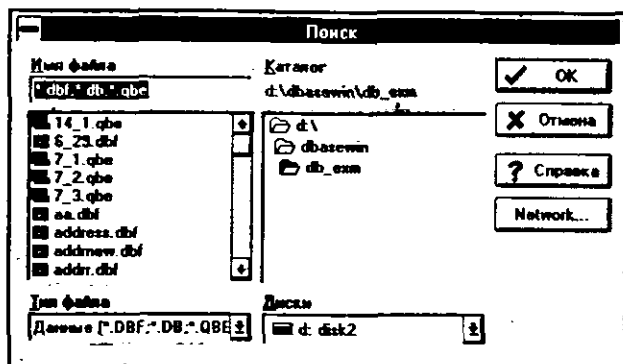


Рис. 8.4. Выбор таблицы с помощью окна диалога “Поиск”.

- Выберите поля таблицы, которые должны присутствовать в форме (рис. 8.5).

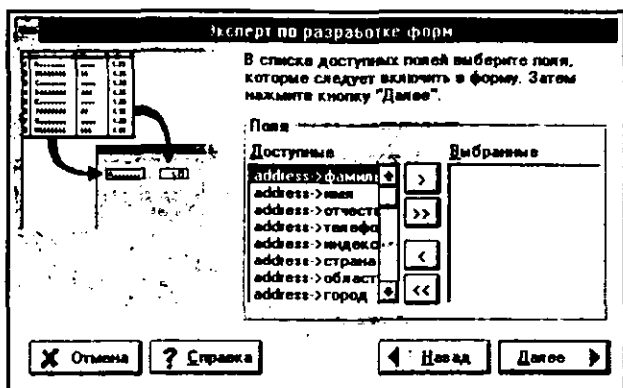


Рис. 8.5. Выбор полей таблицы.

По умолчанию все поля указанной таблицы размещаются в списке **Доступные**. Если вы хотите включить какие-либо поля в форму, укажите на поле, после чего нажмите кнопку **Добавить (>)**. В результате указанное поле будет помещено в список **Выбранные**.

Если вы измените свое мнение и решите не размещать в форме выбранное поле, укажите на него в списке **Выбранные** и нажмите кнопку **Удалить (<)**. В результате чего поле возвратится в список **Доступные**.

Для добавления всех полей списка **Доступные** в список **Выбранные** нажмите кнопку **Добавить все (>>)**. Для удаления всех выбранных для отображения в форме полей нажмите кнопку **Удалить все (<<)**.

В данном примере переместите все поля из списка доступных полей за исключением полей “Область”, “Дата рождения”, “Примечание”, “Фотография” и “Звук”.

Для перехода к следующему шагу нажмите кнопку Далее.

5. Выберите тип макета для формы. В данном примере воспользуйтесь типом **Перечень**, установленным по умолчанию.

Если вы хотите выбрать другой тип макета, нажмите соответствующий ему переключатель в правой части окна диалога. Для перехода к следующему шагу нажмите кнопку Далее.

6. Задайте внешний вид формы (рис. 8.6). Вы можете определить значения атрибутов шрифта, цвета и рамок различных элементов формы. Выбранные вами значения будут отображаться в примере формы, расположенном в верхнем левом углу.

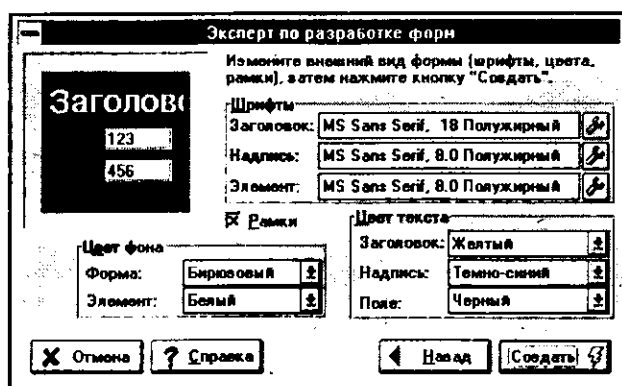


Рис. 8.6. Определение внешнего вида формы.

В заключение нажмите кнопку **Создать**, и после непродолжительной паузы на экране появится форма в режиме конструктора форм.



7. Вы можете продолжить работу над формой с помощью конструктора форм или же запустить ее на выполнение. Создание форм с помощью конструктора будет рассмотрено в следующем разделе, а сейчас сохраните форму выполнив команду **Файл | Сохранить как** и откройте ее для просмотра.

На рис. 8.7 показана форма, созданная в результате ваших действий.

Следует отметить, что созданная вами форма пока далека от совершенства. Однако эта форма может быть с успехом использована в качестве черновика при создании рабочей формы.

Создание форм с помощью конструктора

При создании форм dBASE предоставляет в ваше распоряжение широчайший спектр средств для управления отображением информации, представленной в форме. Объекты интерфейса позволяют придать форме вид, в наилучшей степени раскрывающий ее содержание. В число объектов интерфейса входят поля, кнопки, графические объекты, OLE-объекты и т.д.

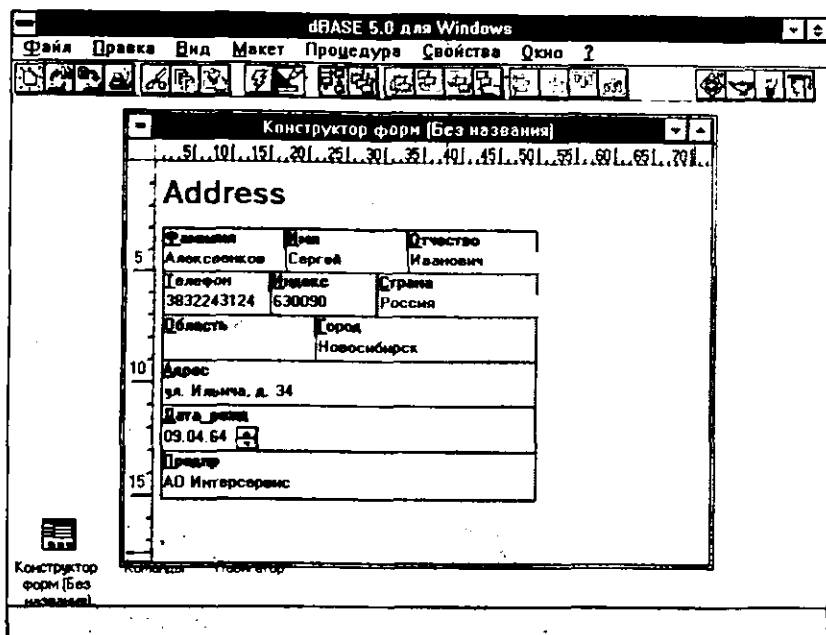


Рис. 8.7. Форма, созданная при помощи эксперта.

Вы можете поместить в любом месте формы заголовки; тексты, предназначенные для раскрытия содержания полей формы; описание функциональных клавиш, используемых при работе, и т.д. Поля формы могут непосредственно соответствовать полям исходной таблицы или являться результатом вычислений над ними. Вы можете установить цвет и управлять параметрами отображения любых элементов и областей формы. Например, для наиболее значимых из полей может быть установлен специальный цвет и стиль. Кроме того, для улучшения восприятия информации вы можете заключить в рамки ряд полей вместе с сопутствующим текстом. Со всеми этими и другими средствами вы познакомитесь далее в настоящей главе.

Для создания новой формы выберите **Файл | Создать | Форма**. Для создания новой формы из окна навигатора или папки выберите элемент списка "Без названия".

Независимо от того, каким из средств вы воспользуетесь будет открыто окно диалога "Эксперт по разработке форм" (рис. 8.2). С помощью этого окна вы должны выбрать, воспользуетесь ли вы при создании формы экспертом или предпочитаете создавать форму самостоятельно.

Сейчас мы хотим самостоятельно создать форму для таблицы ADDRESS. Поэтому установите флажок **Пустая форма** и на экране появится окно конструктора форм, которое не содержит никаких объектов (рис. 8.8).

Вы размещаете в форме объекты интерфейса такие, как поля ввода, списки, флажки и т.д. Эти объекты являются стандартными для dBASE и позволяют вам вводить и отображать любые данные таблиц приложения. Дополнительно вы можете использовать специальные объекты интерфейса, поставляемые вместе с dBASE или созданные независимыми разработчиками. Для размещения объектов интерфейса используется панель элементов. Каждый из объектов экранной формы имеет ряд параметров, называемых *свойствами*. Некоторые из свойств (например, цвет) присущи практически всем объектам. При размещении объекта в форме для него устанавливаются стандартные свойства, принятые по умолчанию. В дальнейшем вы можете вызвать окно свойств объекта и выбрать из него требуемые свойства. В последующих

разделах мы рассмотрим каждый из типов объектов и их свойства. Кроме свойства объекта интерфейса вы можете определить обрабатываемые этим объектом *события*, а также *методы*, связанные с ним. Для определения свойств объекта, обрабатываемых событий, а также связанных с объектом методов используется окно свойств объекта, которое содержит отдельные карточки для определения свойств, событий и методов.

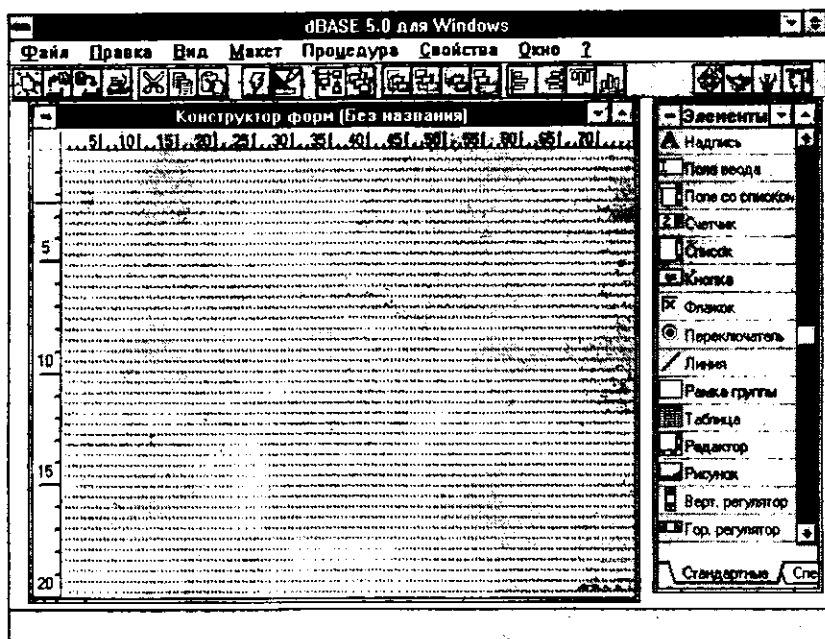


Рис. 8.8. Пустое окно конструктора форм.



Форма также является объектом интерфейса.

Вы можете использовать формы для ввода и просмотра данных таблиц, для ввода значений переменных, используемых при формировании отчетов, запросов, а также для вывода разнообразной информации.

Работа в окне конструктора форм

Окно конструктора форм является основным средством для создания форм. В этом окне выполняется большинство операций по размещению, перемещению, удалению и модификации всех экранных объектов. При работе в окне конструктора возможны два режима работы: макет и упорядочивание.

Режим макета является основным режимом работы в окне конструктора форм и предназначен для выполнения следующих действий:

- размещения элементов управления в окне формы
- определения свойств объектов
- определения действий для событий объектов

В режиме упорядочивания вы можете определить порядок расположения объектов по слоям.

Знакомство со средствами, доступными при создании форм, мы начнем с панели инструментов в режиме макета и упорядочивания. Затем рассмотрим меню конструктора формы, а также контекстное меню в окне формы, панель элементов, окно свойств объектов интерфейса.

Панель инструментов

Внешний вид панели инструментов в режиме макета показан на рис. 8.9, а наименования кнопок приведены в табл. 8.1.



Рис. 8.9. Панель инструментов в окне конструктора форм в режиме макета.

Табл. 8.1. Кнопки панели инструментов

Кнопка	Наименование	Назначение
	Создать	Создает новый файл
	Открыть	Открывает существующий файл
	Сохранить	Сохраняет текущий файл
	Печать	Позволяет напечатать текущий документ, используя для задания параметров окно диалога "Печать файла"
	Удалить в буфер	Удаляет выделенный текст или объект и помещает их в буфер
	Копировать в буфер	Копирует в буфер выделенный текст или объект, не удаляя их из документа
	Вставить из буфера	Копирует содержимое буфера. Результат нажатия этой кнопки зависит от содержимого буфера
	Запуск	Закрывает окно конструктора форм и предоставляет возможность просмотра или редактирования данных с помощью текущей формы
	Режим макета	Переключает конструктор форм в режим макета
	Режим упорядочивания	Переключает конструктор форм в режим упорядочивания
	Установить первым	Устанавливает выбранный объект первым в порядке обхода объектов формы

Кнопка	Наименование	Назначение
	Установить за следующим	Устанавливает выбранный объект за следующим объектом в порядке обхода объектов
	Установить перед предыдущим	Устанавливает выбранный объект перед предыдущим объектом в порядке обхода объектов
	Выровнять по левому краю	Выравнивает выбранные объекты по левому краю самого левого из выбранных объектов
	Выровнять по правому краю	Выравнивает выбранные объекты по правому краю самого правого из выбранных объектов
	Выровнять по верхнему краю	Выравнивает выбранные объекты по верхнему краю самого верхнего из выбранных объектов
	Выровнять по нижнему краю	Выравнивает выбранные объекты по нижнему краю самого нижнего из выбранных объектов
	Навигатор	Открывает окно навигатора для текущего каталога
	Эксперт	Осуществляет выбор эксперта
	Инструктор	Запускает инструктора dBASE

В режиме упорядочивания в панели управления исчезают кнопки выравнивания объектов и определения порядка их обхода. Вместо них появляется счетчик слоев расположения объектов, с помощью которых задается номер слоя выбранного объекта. В dBASE каждый объект формы располагается в отдельной плоскости, которая находится впереди или позади плоскости другого объекта. Порядок расположения объектов по слоям называется Z-порядком. Более подробно изменение порядка расположения объектов по слоям мы рассмотрим далее в настоящей главе.



Рис. 8.10. Панель инструментов в окне конструктора форм в режиме упорядочивания.

Меню окна конструктора форм

Окно конструктора форм содержит следующие меню:

Наименование меню	Назначение
Файл	Позволяет создавать новые файлы, открывать и запускать существующие файлы, выводить файлы на печать и выходить из конструктора форм.
Правка	Выполняет операции выбора, копирования и удаления одного или группы объектов формы.

Наименование меню	Назначение
Вид	Выполняет переход между режимами выполнения или разработки формы, а также между режимами макета и упорядочивания, открывает панель элементов, окно свойств и окно процедур.
Макет	Осуществляет выравнивание объектов интерфейса в форме, а также управляет расположением слоев объектов. Кроме того, вы можете использовать это меню для вызова конструктора запросов и конструктора меню из окна конструктора форм.
Процедура	Команды данного меню предназначены для работы с процедурами в редакторе процедур. Некоторые из этих команд доступны только при открытом окне редактора процедур. Так как данное меню связано с программированием в среде dBASE мы не будем рассматривать его в рамках данной книги. Необходимую информацию вы можете найти в "Руководстве программиста" и справочной системе dBASE.
Свойства	Настраивает характеристики системы dBASE для Windows, конструктора форм и редактора процедур.
Окно	Размещает, открывает и активизирует окна в рабочей области dBASE. Это меню содержит также список файлов, которые были открыты ранее и позволяет вам быстро открыть их снова простым выбором пункта меню с именем файла.
?	Выполняет свои обычные функции по выводу справочной информации и вызова Экспертов и Инструкторов.

Теперь мы кратко остановимся на меню **Вид** и **Макет**, поскольку в отличие от других меню данные меню выполняют специфические для конструктора форм действия.



В dBASE одно и то же действие может быть выполнено с помощью панели инструментов, меню окна конструктора форм или контекстного меню.

Вид	
Форма	F2
✓ Конструктор форм	Shift+F2
✓ Режим макета	
Режим упорядочивания	
✓ Панель элементов	
Окно свойств	
Процедуры	

Первые четыре пункта меню дублируют соответствующие кнопки панели инструментов. Пункты **Панель элементов**, **Окно свойств** и **Процедуры** из меню **Вид** управляют отображением одноименных окон в окне конструктора форм. Для вывода окна на экран установите метку в пункте меню, а для удаления окна с экрана снимите эту метку.

Команды выравнивания объектов, а также управления расположением слоев из меню **Макет** дублируют кнопки панели управления.

Макет
Выровнять по левому краю
Выровнять по правому краю
Выровнять по верхнему краю
Выровнять по нижнему краю
Выравнивание по центру ▶
На передний план
На задний план
На один слой вверх
На один слой вниз
Операции ▶

Абсолютное горизонтальное
Относительное горизонтальное
Абсолютное вертикальное
Относительное вертикальное

Конструктор меню
Конструктор запросов

Команда **Выравнивание по центру** позволяет вам установить четыре разных варианта выравнивания.

Команда **Операции** предназначена для создания запроса, который может использоваться в качестве источника данных, а также для меню, связанного с данной формой.

При разработке формы удобно пользоваться контекстным меню окна конструктора форм, которое вызывается на экран нажатием правой кнопки мыши в любом месте рабочей поверхности окна конструктора форм. Контекстное меню содержит пункты основного меню, наиболее часто используемые при создании формы:

Свойства конструктора форм...	
Вырезать	^X
Копировать	^C
Вставить	^V
✓ Панель элементов	
✓ Окно свойств	
Процедуры	

Панель элементов

Панель элементов содержит список объектов интерфейса, которые вы размещаете в форме. Панель элементов состоит из двух карточек: "Стандартные" и "Специальные" (рис. 8.11). Для отображения на экране панели инструментов необходимо установить метку в пункте **Панель элементов** меню **Вид**. Альтернативно вы можете установить метку в одноименном пункте контекстного меню формы.

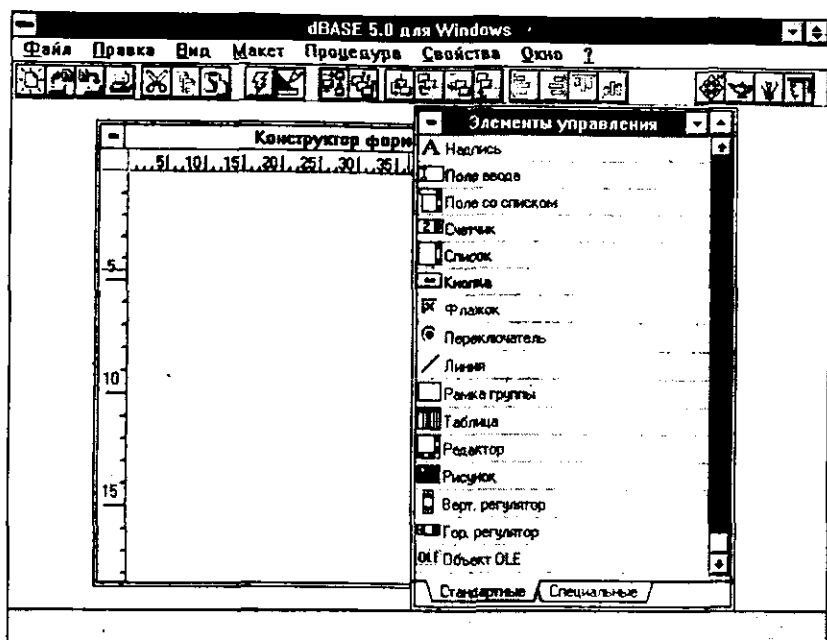


Рис. 8.11. Панель элементов.

Стандартные объекты интерфейса являются частью системы dBASE для Windows. Специальные объекты интерфейса могут быть специальными элементами управления dBASE (они имеют расширение имени файла .CC) или специальными элементами управления, созданными в стандарте Visual Basic с расширением имени файла .VBX. Стандартные объекты интерфейса всегда присутствуют в панели элементов. Список стандартных объектов интерфейса приведен ниже:

Объект	Наименование	Назначение
	Указатель	Выбирает объект интерфейса в форме
	Надпись	Служит для ввода текста
	Поле ввода	Вводит одно значение данных
	Поле со списком	Вводит или выбирает одно значение из нескольких имеющихся вариантов
	Счетчик	Вводит числовое значение или дату непосредственно в поле или с помощью кнопок увеличения и уменьшения значения
	Список	Выбирает одно значение из нескольких имеющихся вариантов

Объект	Наименование	Назначение
	Кнопка	Позволяет выполнить назначенное кнопке действие при ее нажатии
	Флажок	Выбирает одно из двух логических значений
	Переключатель	Выбирает одно из нескольких значений
	Линия	Позволяет улучшить оформление формы
	Рамка группы	Используется для выделения группы однородных объектов
	Таблица	Вводит данные в табличном виде
	Редактор	Позволяет просматривать и редактировать Метополь или длинные текстовые поля
	Рисунок	Позволяет улучшить оформление формы
	Вертикальный регулятор	Выбирает числовое значение или дату
	Горизонтальный регулятор	Выбирает числовое значение или дату
	Объект OLE	Позволяет работать с документом из другого приложения с помощью механизма OLE

Использование специальных объектов интерфейса будет рассмотрено в разделе “Специальные объекты интерфейса” настоящей главы.

Для добавления объекта интерфейса в форму, выберите его на панели элементов. После выбора объекта при перемещении мыши по рабочей поверхности формы его курсор принимает форму перекрестия. Нажмите левую кнопку мыши и, удерживая ее, перемещайте указатель по рабочей поверхности, пока не сформируете область такого размера, какой должен иметь объект интерфейса. После этого отпустите кнопку мыши, в результате чего в форме будет размещен выбранный вами объект заданного размера. Если же вы желаете добавить в форму объект интерфейса стандартного размера, просто выберите место на рабочей поверхности и отпустите кнопку мыши.

Выбор объектов

Прежде чем выполнить в окне конструктора над объектом какую-либо операцию, вам необходимо выбрать его. Вы можете выбрать требуемый объект с помощью кнопки **Выбор объекта** на панели элементов или клавиатуры. При нажатии кнопки **Выбор объекта** курсор мыши приобретет форму стрелки. При входе в окно конструктора форм курсор имеет именно такую форму, поскольку указанная кнопка выбирается по умолчанию.

После того как вы выберете некоторый объект, вокруг него будут отображаться маркеры, с помощью которых вы можете перемещать выбранный объект, а также изменять его размеры и форму. В качестве примера на рис. 8.12 показан выбор текстового поля с заголовком формы. Отметьте наличие маркеров вокруг этого поля.

Если вы укажете мышью на один из маркеров, ее курсор изменит свою форму, показывая в каких направлениях можно перемещать данный маркер. Например, для того чтобы изменить длину объекта, укажите на любой из маркеров, расположенных в середине вертикального отрезка. Курсор мыши примет форму горизонтальной двунаправленной стрелки. После чего переместите маркер до установки требуемого размера. Если вы захотите изменить сразу оба размера, воспользуйтесь одним из маркеров, расположенных по углам выбранного объекта.

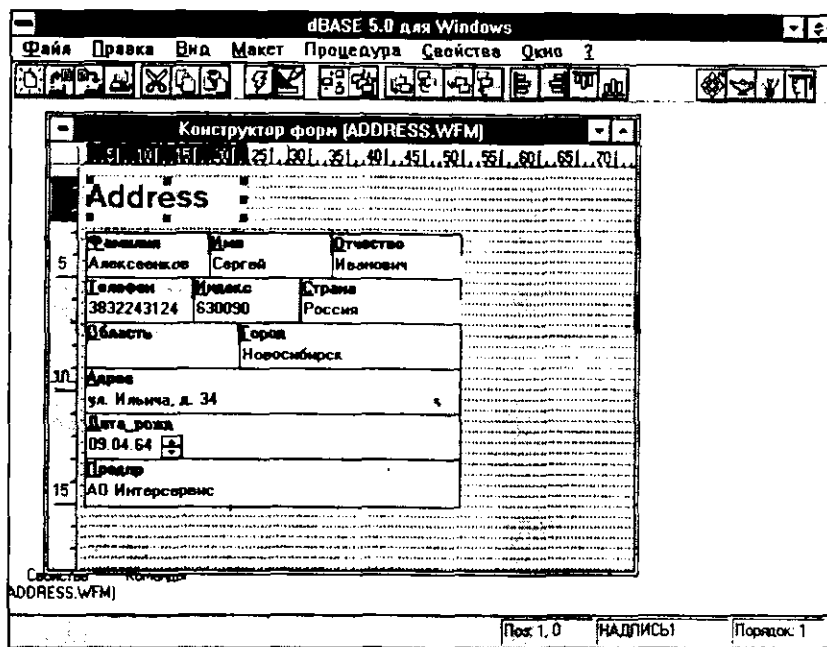


Рис. 8.12. Выбор текстового поля.

Для перемещения выбранного объекта укажите на область внутри объекта (но не на маркер) и переместите объект в нужное место.

Для того чтобы выбрать объект с помощью клавиатуры, нажмите *Tab*, что позволит вам перейти от текущего объекта в форме к последующему. При этом происходит перемещение маркеров, указывающих на выделенный объект. Если вам потребуется переместиться в обратном направлении, воспользуйтесь *Shift+Tab*.

В некоторых случаях (например, при перемещении) может оказаться, что целесообразно управлять не одним, а сразу несколькими объектами. В этом случае вам потребуется выделить сразу нескольких объектов. После чего вы можете перемещать их как единое целое или изменить свойства сразу всех выделенных объектов. Существует несколько способов выделения нескольких объектов. Выбор конкретного способа зависит от расположения объектов и тех действий, которые вы предполагаете выполнить над группой выделенных объектов.

Для выделения нескольких изолированных объектов, нажмите клавишу *Shift* и укажите на первый из объектов. Далее, не отпуская клавиши *Shift*, укажите на следующий из объектов и так далее до тех пор, пока не будут выбраны все необходимые вам объекты.

В том случае, когда необходимо выделить несколько объектов, которые мысленно можно расположить в пределах одной области, используется другой способ. Нажмите клавишу *Shift* и, удерживая ее, укажите на один из углов воображаемой области. Далее нажмите кнопку мыши и, не отпуская ее, переместите курсор мыши в противоположный угол области. В процессе перемещения курсора мыши пунктирная рамка указывает на положение области. После того как все нужные объекты будут находиться внутри пунктирной рамки, отпустите кнопку мыши и клавишу *Shift*. В результате dBASE выделит все объекты, находящиеся внутри рамки.

Еще одним средством для выделения объектов является команда **Правка | Выделить все**. Если вы выберете эту команду в тот момент, когда в окне конструктора форм не выделено ни одного объекта, dBASE выделит все объекты формы.

Если вам потребуется снять выделение с какого-либо из объектов, нажмите клавишу *Shift* и укажите на него. В результате маркеры объекта исчезнут. Если клавиша *Shift* не будет нажата, выделение будет снято со всех выделенных объектов, за исключением того, на который вы укажете.

После выделения одного или более объектов вы можете выполнять с ними различные действия. Например, для того чтобы переместить выделенные объекты, достаточно поместить курсор мыши внутрь одного из этих объектов, нажать кнопку мыши и, не отпуская ее, переместить группу объектов в требуемое место. Вы также можете воспользоваться буфером для копирования группы выделенных объектов. Для удаления, копирования в буфер и копирования из буфера вы можете воспользоваться соответствующими кнопками панели инструментов, командами из меню **Правка** или оперативными клавишами. Ниже приведены команды меню **Правка** и эквивалентные им оперативные клавиши:

Вырезать	<i>Shift+ Del</i>
Копировать	<i>Ctrl+ Ins</i>
Вставить	<i>Shift+ Ins</i>
Удалить	<i>Del</i>

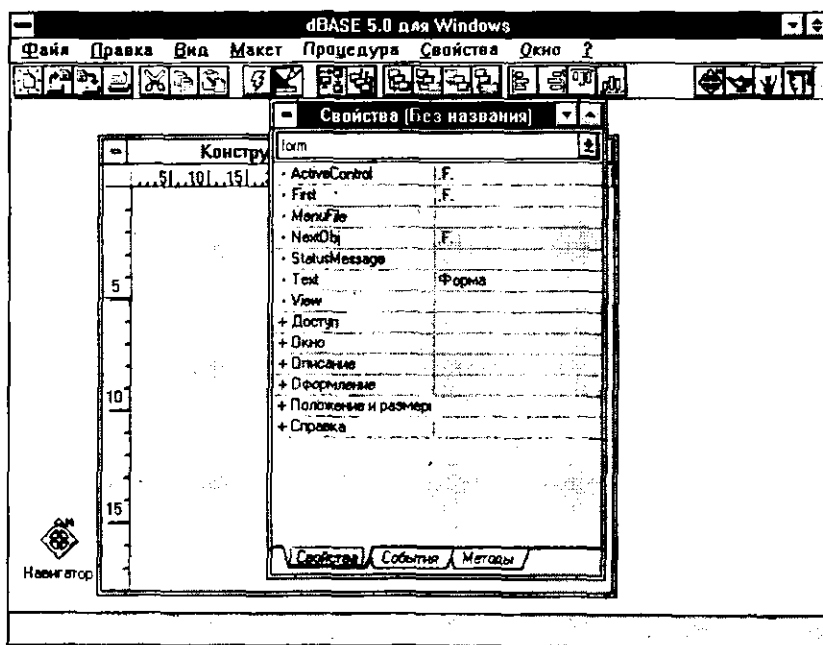


Рис. 8.13. Окно свойств объекта интерфейса.

Изменение свойств объектов

Все объекты dBASE характеризуются свойствами, которые вы можете настроить в соответствии со своими требованиями. Если вы еще не изменяли свойств объектов, то они будут иметь стандартные значения. Для изменения свойств объекта, а также определения тех свойств, которые по умолчанию не определяются, используется окно свойств объекта (рис. 8.13). В этом же окне вы можете также определить обрабатываемые этим объектом события и связанные с ним методы. Для вывода окна свойств установите метку для пункта **Окно свойств** меню Вид или контекстного меню окна конструктора форм. Окно свойств содержит карточки "Свойства", "События" и "Методы". Вначале обсудим определение свойств объекта с помощью карточки "Свойства". В верхней части этой карточки находится раскрывающийся список, в котором представлены все имеющиеся в форме объекты интерфейса. Остальная часть карточки содержит два столбца, в левом из которых перечислены наименования свойств, а в правом — их текущие значения. Каждому типу объекта интерфейса соответствует свой список свойств. Поэтому при выборе различных типов объектов меняется и содержимое окна свойств.

Свойства выводятся на экран в списке, упорядоченном по алфавиту или по категориям. По умолчанию список упорядочен по категориям, так как именно это упорядочение наиболее удобно для выбора и изменения свойств. Если же вам более подходит упорядочивание по алфавиту, выполните следующие действия:

1. Выберите команду **Свойства | Среда dBASE**, и на экране откроется окно диалога "Свойства среды dBASE" (рис. 8.14).
2. Выберите корешок "Приложение".
3. Снимите флажок **Группировать свойства**, чтобы расположить свойства в алфавитном порядке.

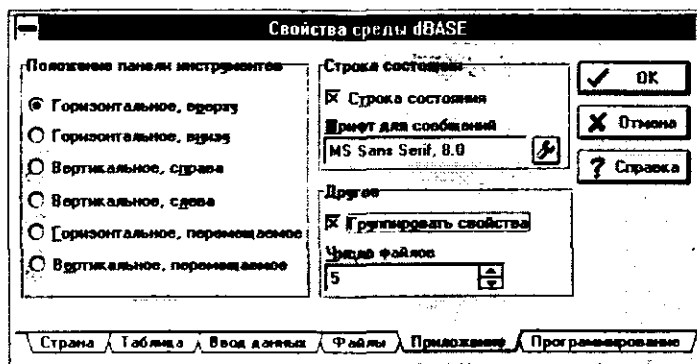


Рис. 8.14. Настройка упорядочивания свойств объекта интерфейса.

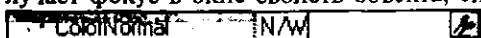
При упорядочении по категориям ряд свойств объекта объединены в категории, наименования которых отображаются в окне свойств. Перед именем категории всегда стоит знак плюса (+) или минуса (-). Имена свойств, не объединенных в категории, перед именем свойства содержат маленький кружок. Знак плюса (+) перед именем категории указывает, что список закрыт, т.е. вместо списка свойств категории в окне свойств отображается только строка с именем категории. Знак минуса (-) указывает, что список раскрыт. В этом случае ниже имени категории приводится список всех

Оформление	
• ColorNormal	N/W
• MousePointer	0 - По умолчанию
• ScrollBar	0 - Не отображать

свойств данной категории, наименования которых содержат отступ от левого края столбца.

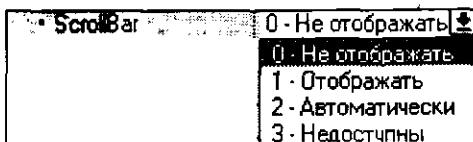
Для того чтобы открыть или закрыть список, нужно дважды нажать имя категории или нажать клавиши (+) или (-) на цифровой клавиатуре. Чтобы открыть все списки категорий нажмите клавишу *Ctrl* с клавишей (+) на цифровой клавиатуре, а для закрытия всех списков — *Ctrl* с клавишей (-).

Вы можете ввести значение свойства непосредственно в поле ввода данного свойства, но для установки значений многих из них вы можете использовать специальные инструменты, предоставляемые вам для облегчения определения значения свойства. Если для свойства существует инструмент, то в момент, когда свойство получает фокус в окне свойств объекта, справа от значения свойства появляется кнопка



Инструмент.

Например, для установки цвета объекта вы можете выбрать соответствующую кнопку. В результате на экране появляется окно диалога выбора цвета, в котором вы в интерактивном режиме задаете цвет объекта. При этом вам наглядно демонстрируется результат вашего выбора.



Некоторые из свойств объекта могут иметь одно из нескольких возможных значений. В этих случаях вы можете выбрать значение с помощью раскрывающегося списка.



В dBASE для определения свойств объектов широко используются инструменты и раскрывающиеся списки, которые не только облегчают выбор значения свойства, но и уменьшают возможность ошибки при вводе.

Настройка свойств конструктора форм

Команда **Свойства | Конструктор форм** позволяет вам выполнить индивидуальную настройку конструктора форм в соответствии с вашими требованиями. При выполнении этой команды на экране появляется окно диалога “Свойства конструктора форм” (рис. 8.15).

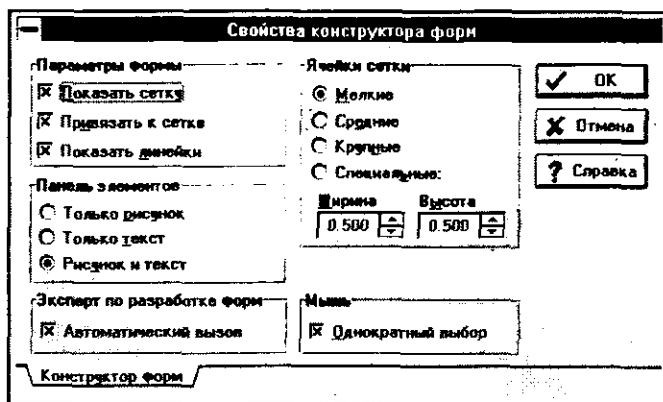


Рис. 8.15. Окно диалога “Свойства конструктора форм”.

Знакомство с окном диалога “Свойства конструктора форм” начнем с определения параметров сетки в окне конструктора формы. Сетка представляет собой набор точек, которые используются для размещения и выравнивания объектов. Флажок **Показать сетку** управляет отображением сетки на экране. Сетка отображается на экране только при установленном флажке. Флажок **Привязать к сетке** устанавливает режим автоматического выравнивания объектов по точкам сетки при добавлении, перемещении объектов или изменении их размеров. При установленном флажке углы объекта всегда привязываются к ближайшему узлу сетки. При этом каждый угол объекта выравнивается по узлу сетки таким образом, чтобы стороны объекта находились на один пиксел над точками сетки и на один пиксел левее точек сетки. При снятом флажке **Привязать к сетке** объект можно добавлять или передвигать в любое место рабочей поверхности, а также произвольно изменять его размеры. Если объект был создан при выключенном флажке **Привязать к сетке**, то при последующей установке этого флажка размер и положение объекта не меняются до тех пор, пока он не будет передвинут или изменен его размер. Как только вы совершили какое-либо из этих действий, размер и положение объекта будут привязаны к узлам сетки.

Вы можете управлять размером ячейки. Значения размера сетки вычисляются относительно текущего размера шрифта. Для определения размера ячейки в области “Ячейки сетки” выберите один из возможных вариантов:

Размер	Описание
Мелкие	Размер, приблизительно равный половине величины текущего шрифта формы.
Средние	Размер, приблизительно равный трем четвертям величины текущего шрифта формы.
Крупные	Размер, равный величине текущего шрифта формы.
Специальные	Размер, определенный пользователем, относительно шрифта формы. Размер задается в счетчиках Ширина и Высота . Счетчик Ширина задает ширину шрифта, а счетчик Высота — высоту шрифта.

В левой и верхней частях рабочей поверхности находятся горизонтальная и вертикальная линейки, которые облегчают позиционирование курсора мыши и объекта при ручном размещении. Курсор мыши отображается на линейках маркерами, в результате чего вы можете определить точное положение курсора. Границы выделенного объекта или группы объектов также отображаются на линейках аналогичным образом.

По умолчанию линейки присутствуют в окне формы, однако вы можете удалить их с экрана сняв флажок **Показать линейки**. Это позволит вам увеличить размер рабочей области окна конструктора.

Шаг линейки равен одному знаку текущего шрифта формы. Это значит, что шаг горизонтальной линейки равен средней ширине символа шрифта. Шаг вертикальной линейки равен высоте шрифта.

Группа переключателей “Панель элементов” определяет отображение стандартных элементов управления. Возможны три варианта:

Только рисунок	Обеспечивает отображение только рисунков объектов интерфейса (без их имен).
Только текст	Обеспечивает отображение только имен объектов интерфейса (без рисунков).

Рисунок и текст Обеспечивает отображение и рисунков и имен объектов интерфейса.

По умолчанию при создании новой формы всегда открывается эксперт форм. Если в создаваемых вами формах использование эксперта нецелесообразно, снимите флажок **Автоматический вызов** в области "Эксперт по разработке форм".

Флажок **Однократный выбор** области "Мышь" определяет поведение указателя при выборе объектов из панели элементов. По умолчанию этот флажок установлен и в этом режиме после выбора объекта и размещения его в форме, с ним можно выполнять разные действия. Вы можете изменить его размер, переместить его на новое место или изменить его свойства. Данный режим удобен в том случае, когда вы размещаете каждый объект по отдельности, сразу же определяя его размер, положение и свойства.

dBASE позволяет вам использовать режим многократного размещения объектов интерфейса. Для этого вы должны снять флажок **Однократный выбор**. В этом режиме вы выбираете объект из панели элементов и размещаете его в форме. Для размещения еще одного такого же объекта в форме вам нет необходимости выбирать этот объект еще раз из панели элементов. Достаточно нажать мышью в требуемом месте рабочей поверхности. Этот режим удобен для размещения на рабочей поверхности нескольких одинаковых объектов интерфейса, не выбирая каждый раз снова объект интерфейса на панели элементов.

Размещение объектов интерфейса

Размещение линий

Объект **Линия** действует точно также, как аналогичные средства в программах создания графических изображений (например, Microsoft Windows Paintbrush). Для того чтобы разместить в форме линию, выполните следующие действия:

1. Выберите на панели элементов объект **Линия**. В результате курсор мыши примет форму перекрестья.
2. Установите курсор на начальную точку линии.
3. Нажмите кнопку мыши, тем самым вы зафиксируете начальную точку линии. Далее, удерживая кнопку мыши, переместите ее в конечную точку линии. Для того чтобы зафиксировать объект, отпустите кнопку мыши.
4. Если в дальнейшем вам потребуется изменить размеры линии, выделите ее, после чего с помощью соответствующего маркера установите требуемый размер.

Линия имеет собственный набор свойств, которые позволяют вам изменить параметры ее отображения. Ниже приведены эти свойства:

Свойство	Описание
ColorNormal	Устанавливает цвет линии, используя для этого цветовую палитру. Для данного свойства определено средство выбора цвета, поэтому вы можете нажать кнопку Инструмент , расположенную в правой части поля ввода значения.

Свойство	Описание
Pen	Позволяет вам использовать сплошной и четыре типа пунктирных линий. При выборе этого свойства появляется список допустимых типов линий, с помощью которого вы можете выбрать требуемый тип.
Width	Изменяет толщину линии. По умолчанию используется тонкая линия. Если вам потребуется изменить этот параметр, выберите свойство Width и с помощью счетчика, который появляется в этом поле, установите требуемую толщину линии.



Если вам потребуется получить дополнительную информацию о свойствах объекта, выделите нужную строку в окне свойств и нажмите *F1*.

Размещение текста

Рассмотрим теперь объект **Надпись**, с помощью которого вы можете разместить в форме текстовые объекты. Вы можете использовать текстовые объекты в качестве заголовка формы или отобразить с их помощью некоторую вспомогательную информацию о работе с формой. Тексты могут носить произвольный характер, однако мы рекомендуем помещать в форме только информативные заголовки и сообщения.

Пример 8.2.

Для создания текстового объекта выполните следующую последовательность действий:

1. Нажмите объект **Надпись** на панели элементов, в результате чего курсор мыши примет форму перекрестья.
2. Установите курсор мыши в одном из углов области, в которой должен отображаться текст. Нажмите кнопку мыши и, удерживая ее, переместите курсор в противоположный угол области, тем самым устанавливая требуемые размеры области. Не заботьтесь о точном расположении и размерах области, вы всегда сможете изменить их в дальнейшем с помощью маркеров. После того как вы отпустите кнопку мыши, вокруг области появится рамка, а внутри объекта будет содержаться текст, определенный по умолчанию при создании объекта.
3. Откройте окно свойств текстового объекта и в поле ввода значения свойства объекта **Text** напечатайте текст.

На рис. 8.16. приведен пример создания текстового объекта.

Текстовый объект содержит два отдельных свойства и шесть категорий свойств. Для того чтобы получить доступ к свойствам текстового объекта, поместите курсор мыши внутрь объекта и нажмите правую кнопку. В результате на экране появится меню свойств (рис. 8.17).

Некоторые из свойств текстового объекта аналогичны свойствам объектов других типов, а ряд свойств являются специфическими для объектов данного типа. Остановимся подробнее на свойствах текстовых объектов.

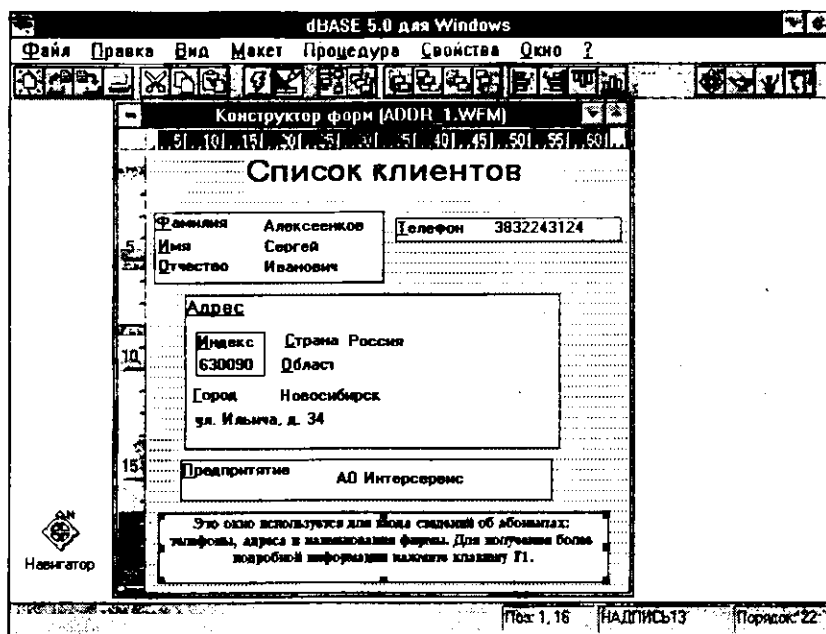


Рис. 8.16. Текстовый объект.

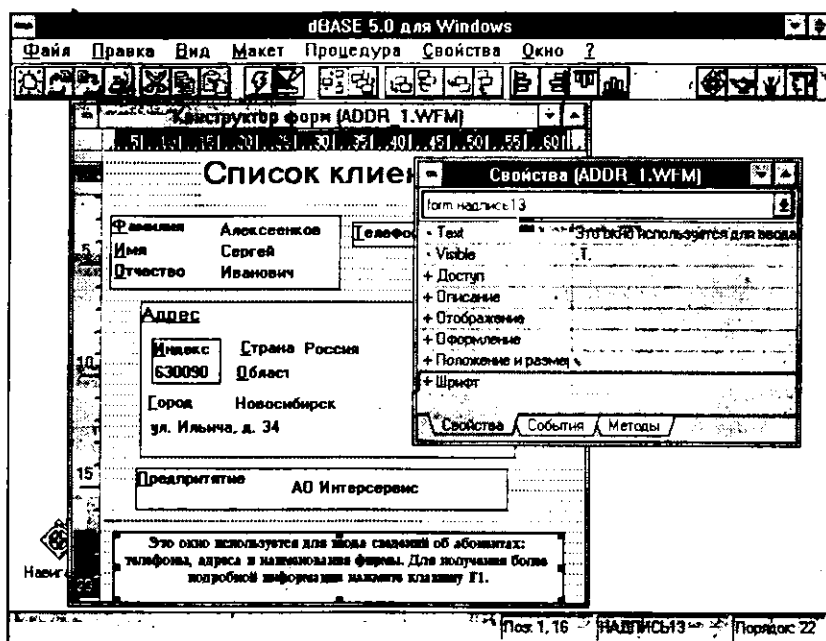


Рис. 8.17. Свойства текстового объекта.

Категория Оформление содержит приведенные ниже четыре свойства.

Наименование	Начальное значение	Описание
Border	.T.	Размещает рамку вокруг текста.
ColorNormal	N/W	Устанавливает цвет текста.
MousePointer	0	Определяет вид указателя мыши, когда он находится над текстовым объектом.
OldStyle	.F.	Определяет стиль отображения текстового объекта (.T. — стиль Windows, .F. — стиль Borland).

Свойство **Border** предназначено для размещения рамки вокруг текстового объекта, что позволяет вам выделить текст, расположенный на форме.

Для определения цвета текста и цвета фона вы можете воспользоваться свойством **ColorNormal**. При определении этого свойства используется специальное окно диалога выбора цвета, которое вызывается при нажатии кнопки **Инструмент** в правой части поля ввода значения поля. Подробно выбор цвета будет рассмотрен в следующем разделе.

Свойство **MousePointer** определяет вид указателя мыши, когда он находится над объектом. Для выбора используйте раскрывающийся список, который содержит 11 различных видов курсора мыши:

Вид курсора	Значение свойства	Вид курсора	Значение свойства
	Стрелка		Размер ВЕРТ
	Крест		Размер ЛВПН
	Текстовый		Размер ГОР
	Значок		Стрелка вверх
	Размер		Песочные часы
	Размер ЛНВП		

Категория **Положение и размеры** содержит свойства, определяющие координаты левого верхнего угла объекта, высоту и ширину объекта, а также способ выравнивания текста в поле. Для выбора способа выравнивания перейдите в поле ввода значения свойства **Alignment**, нажмите на кнопку раскрывающегося списка и выберите подходящее значение из 12 возможных способов выравнивания:

N	Наименование	N	Наименование
0	Вверху слева (по умолчанию)	6	В центре слева
1	Вверху в центре	7	В центре
2	Вверху справа	8	В центре справа
3	Вверху слева	9	Слева с переносом
4	Вверху в центре	10	По центру с переносом
5	Вверху справа	11	Справа с переносом

При выравнивании по левому краю каждая строка текста начинается в первой колонке объекта, при этом правый край может быть не выровнен. При выравнивании по правому краю каждая строка текста заканчивается в последней колонке объекта, при этом левый край может быть неровным. При выравнивании по центру выполняются центровка каждой строки текста, поэтому неровными могут оказаться обе границы. Примеры действия различных опций по выравниванию текста приведены на рис. 8.18.

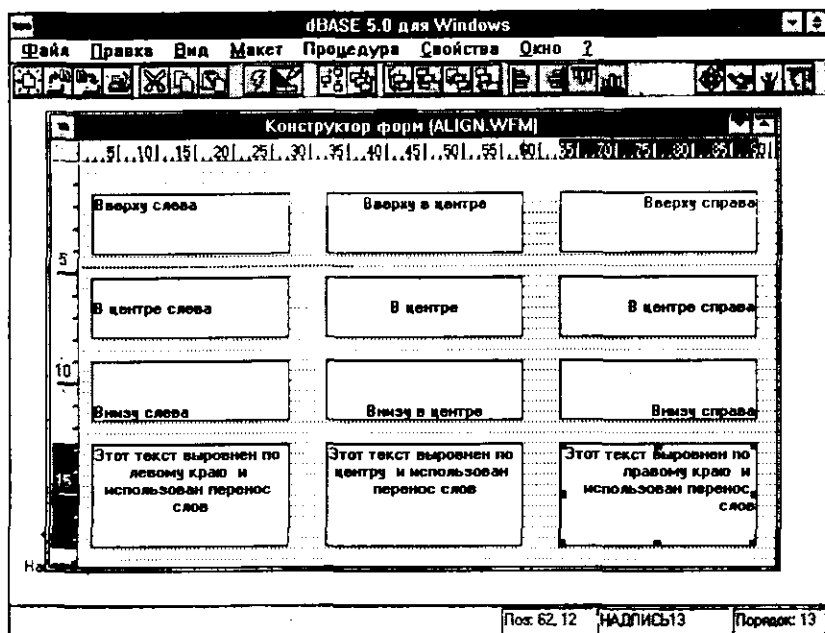


Рис. 8.18. Различные способы выравнивания текстовых объектов.

Вы можете изменить стандартные установки для шрифта, использованного в текстовом объекте, с помощью свойств категории **Шрифт**. Данная категория содержит следующие свойства:

Свойство	Значение по умолчанию	Описание
FontBold	.T.	Отображает символы полужирными
FontItalic	.F.	Отображает символы курсивом
FontName	MS Sans Serif	Определяет тип шрифта для отображения символов
FontSize	N/A	Определяет размер шрифта
FontStrikeOut	.F.	Отображает символы зачеркнутыми
FontUnderline	.F.	Подчеркивает отображаемые символы

Для свойства FontName определен инструмент, при выборе которого открывается стандартное окно диалога Windows (рис. 8.19), предназначенное для выбора шрифта.

В этом окне диалога вы можете выбрать не только тип шрифта, но и все его остальные атрибуты. В dBASE шрифт определяется для всего текстового объекта. Если вам необходимо использовать для отдельных частей текста различные шрифты, разделите текст на несколько текстовых объектов и определите для каждого из них свой шрифт. Вы можете сделать это непосредственно во время ввода текста или после его размещения.

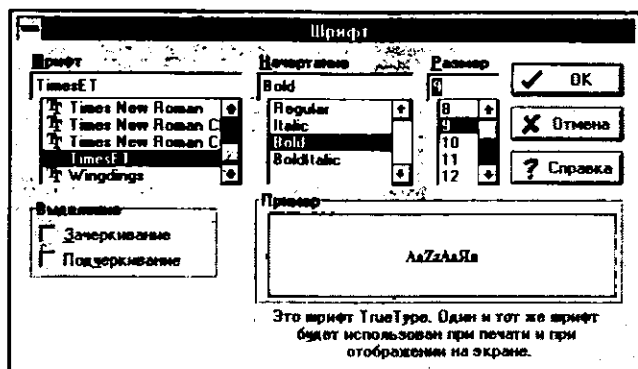


Рис. 8.19. Окно диалога выбора шрифта.

Текстовый объект наиболее часто является надписью к полям ввода. В этом случае вы можете определить для надписи назначенную клавишу, которая позволяет осуществить быстрый переход к этому полю во время ввода данных в форму. Для назначения быстрой клавиши напечатайте перед буквой, которую вы выбрали для быстрого доступа к полю, знак амперсанда (&). Как правило для этого выбирают первую букву надписи. Например, для быстрого перехода в поле "Адрес" напечатайте &Адрес. Назначенный символ будет отображаться на экране с признаком подчеркивания. Для перехода по назначенной клавише при вводе данных в таблицу ADDRESS нажмите комбинацию клавиш *Alt+выбранная буква* (в нашем случае буква А).



Каждая назначенная клавиша в форме должна быть уникальной. Если две надписи начинаются с одной и той же буквы, для одной из них вы должны использовать другую букву.

Настройка цвета объектов

Большинство объектов интерфейса в списке свойств объекта присутствует цвет. Вы можете задать цвет путем непосредственного ввода его значения в поле, но более удобным средством является определение цвета с помощью окна диалога "Выбор цвета" (рис. 8.20). Для открытия этого окна вам необходимо выбрать кнопку **Инструмент** в правой части поля ввода значения цвета.

В окне диалога "Выбор цвета" вы можете задать цвет текущего объекта. Для объектов, имеющих как цвет фона так и цвет текста, можно задавать отдельные значения для фона и текста.



Некоторые объекты имеют несколько наборов цветов. Например, поле ввода может иметь один набор цвета фона и текста, когда он не выбран, и другой набор цвета фона и текста при установке на него фокуса. Оба набора цветов являются свойствами объекта.

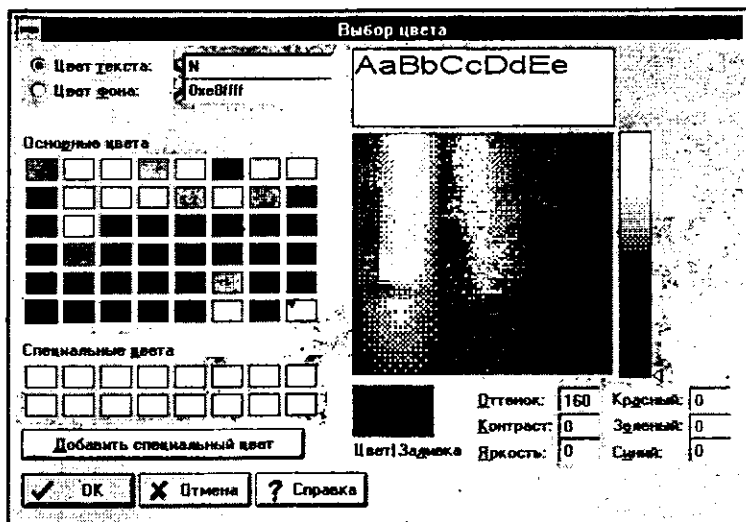


Рис. 8.20. Настройка цвета объекта.

В процессе настройки цвета вы можете переключать режимы установки цвета текста и цвета фона. Для этого используйте группу переключателей **Цвет текста** и **Цвет фона**. При выборе **Цвет текста** значения всех остальных параметров диалога относятся к цвету текста объекта. При выборе **Цвет фона** (если это возможно для данного объекта) значения параметров диалога относятся к цвету фона объекта. Справа от группы переключателей содержатся поля, содержащие текущие значения выбранного цвета. Эти значения определяются автоматически при выборе одного из образцов цвета. Вы можете и сами вводить значения в это поле, если вам известны коды цветов. Большинство цветов выражаются шестнадцатеричными числами (например, 0x8000ff для темно-розового цвета). Для некоторых цветов сохраняется преемственность с предыдущими версиями dBASE и для их обозначения применяются буквенные символы:

N	Черный	R	Красный
B	Синий	RB	Розовый
G	Зеленый	RG	Коричневый
GB	Голубой	W	Белый

Повышенная яркость текста обозначается знаком плюс (+), который следует за кодом. Повышенная яркость фона обозначается знаком звездочка (*), следующим за кодом. Например, красный цвет для текста и для фона обозначается R, ярко-красный для текста обозначается R+, а ярко-зеленый для фона обозначается R*.

В большинстве случаев цвет выбирается из палитры "Основные цвета", которая содержит набор стандартных цветов. Для выбора цвета установите курсор мыши внутрь прямоугольника с подходящим цветом и нажмите кнопку мыши. Код выбранного цвета появится в поле значения выбранного цвета. Для объектов, имеющих как цвет фона так и цвет текста, вам необходимо выбрать подходящую комбинацию этих цветов. Для этого при выборе цвета вы можете постоянно переключаться из режима установки цвета фона в режим установки цвета текста. При переходе из одного режима в другой выбранный в предыдущем режиме цвет продолжает действовать. Поэтому в области примера вы можете видеть как будет выглядеть выбранная комбинация.

Для более тонкой настройки цвета используйте поле цветов, расположенное правее палитры “Основные цвета”. Это поле содержит набор всех доступных цветов. Цвет поля определяется его оттенком, контрастностью и яркостью. Оттенок, собственно, и является “настоящим” цветом (например, красным, желто-зеленым или фиолетовым), описывающим цвет без контрастности и яркости. Код оттенка отображается в поле **Оттенок**.

Контрастность характеризует наличие в цвете серого цвета. Значение контрастности изменяется от 0 (среднее количество серого) до 240 (чистый цвет), текущее значение которого отображается в поле **Контраст**. Значение оттенка в поле цветов меняется слева направо, а сверху вниз меняется контрастность цвета. Текущий цвет отмечается перекрестьем.

Вы можете сохранить выбранный цвет в виде специальной палитры и использовать его при определении цвета других объектов. Для этого выберите цвет в поле цветов и нажмите кнопку **Добавить специальный цвет**. В результате выбранный цвет помещается на палитру “Специальные цвета”. Для выбора цвета из этой палитры установите указатель мыши в прямоугольник с подходящим цветом и нажмите кнопку мыши.

При выборе цвета из основной или специальной палитры, а также из поля цветов вы можете дополнительно определить яркость цвета. Яркость цвета может принимать значения от 0 (черный) до 240 (белый). Столбец справа от поля цветов отражает диапазон яркостей для текущего цвета. Для выбора яркости, передвигайте стрелку управления яркостью вверх или вниз. Значение яркости отображается в поле **Яркость**. При изменении значения яркости, изменяются значения полей **Красный**, **Зеленый** и **Синий**, которые содержат соотношение основных цветов (красного, зеленого и синего) в текущем цвете.



Контрастность определяет чистоту цвета. Яркость определяет насколько велика насыщенность цвета.

Поле **Цвет | Заливка** используется для выбора цвета заливки. Левая часть этого поля содержит текущий цвет, а правая часть поля — ближайший к нему цвет заливки. Чтобы сделать цвет заливки текущим, установите на него указатель и дважды нажмите кнопку мыши.

Размещение графических изображений

По наличию в форме графического изображения вы сразу можете сделать вывод о том, что имеете дело с приложением, работающим под Windows, а не под DOS. Графические изображения могут представлять собой оцифрованную фотографию, кнопку или просто отдельный символ. Эти изображения создаются с помощью графического редактора или снимаются с помощью сканера.

Размещение в форме dBASE графического изображения осуществляется в три этапа:

1. Создайте графическое изображение с помощью сканера или одного из имеющихся в вашем распоряжении графического пакета (например, Microsoft Paintbrush). Для размещения графического изображения вы можете использовать буфер Windows или выбрать файл в формате .BMP или .PCX. Большинство графических программ, включая Microsoft Paintbrush, использует для хранения графических файлов форматы .BMP.



Microsoft Paintbrush позволяет преобразовывать изображения из других форматов (например, .MSO и .PCX) в .BMP.

2. Определите на форме область, в которой должно отображаться графическое изображение. Нажмите на панели элементов кнопку **Рисунок**. В результате курсор мыши будет отображаться в виде перекрестья.

Далее разместите в документе рамку нужного размера. Для этого переместите курсор в один из углов области, в которой будет присутствовать графическое изображение. Нажмите кнопку мыши и, не отпуская ее, переместите курсор в противоположный угол. В результате на экране появится рамка, определяющая границы графического объекта.

3. Свяжите графический объект с источником данных, содержащим графическое изображение. Для этого выделите графический объект и откройте окно свойств объекта. Далее перейдите на поле ввода значения свойства **DataSource** и нажмите кнопку **Инструмент**. На экране откроется окно диалога “Выбор рисунка” (рис. 8.21).

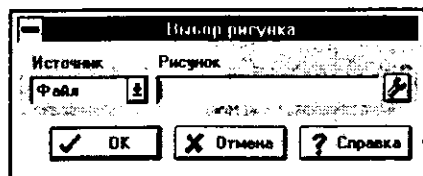


Рис. 8.21. Окно диалога “Выбор рисунка”.

В качестве источника данных могут использоваться:

- Рисунок, хранящийся в файле с расширением .BMP или .PCX
- Рисунок, хранящийся в файле ресурсов DBAS0019.DLL
- Рисунок, хранящийся в двоичном поле таблицы

Для выбора типа источника данных используйте список **Источник**. Размещение в форме рисунков из двоичных полей будет обсуждаться далее в настоящей главе, а сейчас рассмотрим размещение рисунков, хранящихся во внешнем файле и файле ресурсов.

Для размещения в форме рисунка из внешнего файла выберите из списка **Источник** значение “Файл”. Затем нажмите кнопку **Инструмент** справа от поля **Рисунок** в результате чего на экране появится окно диалога “Выбор рисунка” (рис. 8.22). При наличии в текущем каталоге графических файлов с допустимыми расширениями в окне будет отображен их список. Если требуемый файл отсутствует в текущем списке, воспользуйтесь областью “Каталог” для выбора каталога, содержащего требуемый файл. Если вы точно знаете наименование требуемого каталога, вы можете ввести его в строке ввода **Имя файла**.

После выбора нужного каталога в окне диалога будет отображаться список всех графических файлов, содержащихся в данном каталоге и имеющим один из допустимых форматов. Вам остается только указать требуемый файл и его содержимое будет скопировано в текущий графический объект.

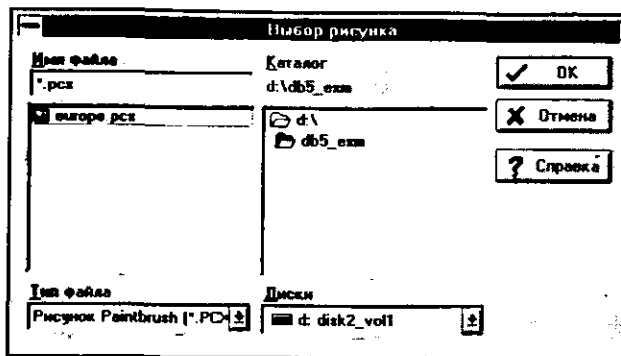


Рис. 8.22. Выбор файла, содержащего графическое изображение.

Для размещения в форме рисунка из файла ресурсов выберите из списка **Источник** значение "Ресурсы". Затем нажмите кнопку **Инструмент** и на экране появится окно диалога "Рисунки dBASE для Windows" (рис. 8.23). Это окно диалога содержит все рисунки из файла DBAS0019.DLL, которые используются в самой системе dBASE для Windows. Поле **Код** содержит идентификационный номер рисунка в файле .DLL, в поле **Рисунок** отображается выбранный в поле **Код** рисунок. Для размещения выбранного рисунка в графическом объекте выберите кнопку **OK**.

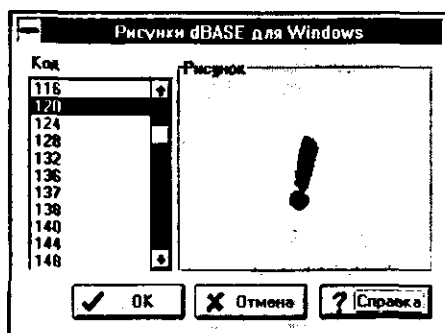


Рис. 8.23. Окно диалога "Рисунки dBASE для Windows".

При заполнении графического объекта dBASE вы можете использовать три способа размещения:

- По размеру рамки
- По верхнему и левому краям
- По центру

Способ размещения задается значением свойства **Alignment**. На рис. 8.24 представлены все три возможных способа размещения рисунка.

В карточке "События" для графического объекта вы можете определить действия, выполняемые при наступлении разнообразных событий. Таким образом, в форме могут отображаться различные графические объекты в зависимости от значения, содержащегося в таблице, или действий пользователя.

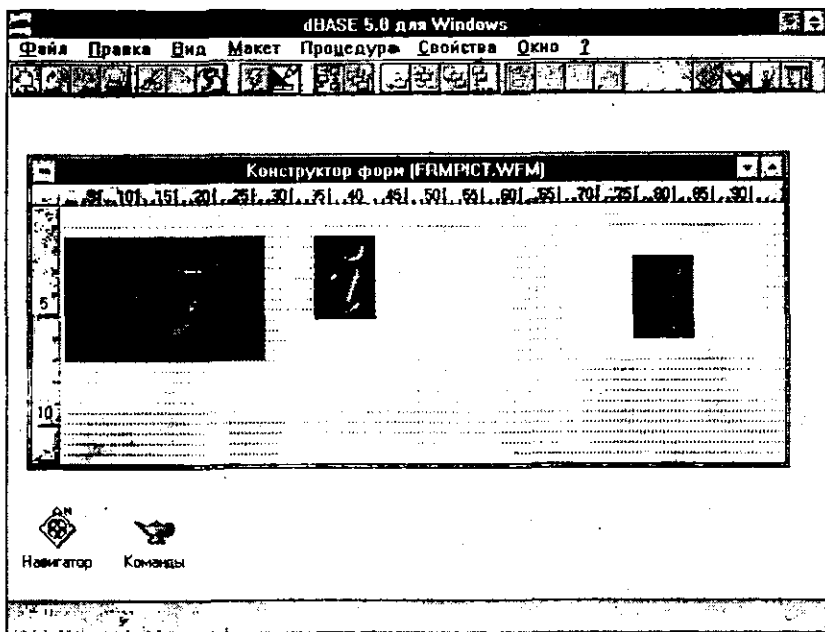


Рис. 8.24. Размещение рисунка в графическом объекте.



Графические изображения являются мощным средством. Однако не старайтесь включать их в формы только потому, что они есть. Например, не имеет смысла помещать товарный знак фирмы на форму, предназначенную для ввода данных, тем самым ежедневно напоминая сотрудникам где они работают.

Пример 8.3.

Давайте рассмотрим пример практического использования в форме графического изображения, с помощью которого мы попытаемся продемонстрировать полезность этого средства.

На рис. 8.25 показана карта Европы, на которую нанесены столицы некоторых из государств. Рядом с названиями столиц отображена температура воздуха в этих городах на дату, которая показана в правом верхнем углу карты. Вы можете пролистать даты с помощью кнопок, расположенных на панели инструментов. При изменении даты соответствующим образом будут изменяться значения температуры.

Для того чтобы создать такую форму, вам необходимы два элемента — графическое изображение и таблица, содержащая необходимые данные. Для создания графического изображения вы можете использовать графический редактор или любые другие средства, с помощью которых вы можете создать требуемое изображение. В данном случае графическое изображение было взято из библиотеки, поставляемой вместе с пакетом Draw Perfect. Собственный формат этого пакета не совместим в Microsoft Paintbrush, однако, с помощью команды **Export** вы можете экспортировать исходное изображение в формат .PCX. После данный файл доступен для чтения Microsoft Paintbrush (рис. 8.26).

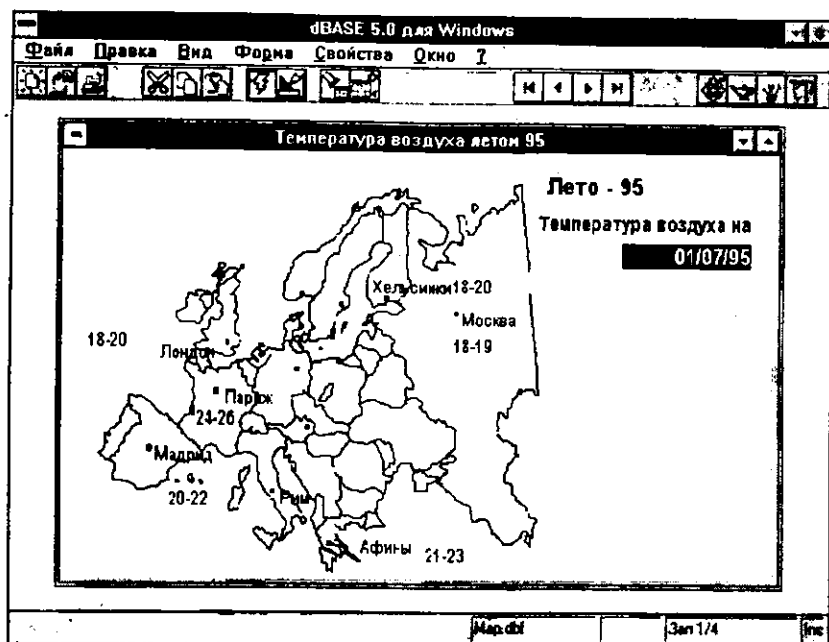


Рис. 8.25. Температура воздуха в столицах некоторых из европейских государств летом 95.

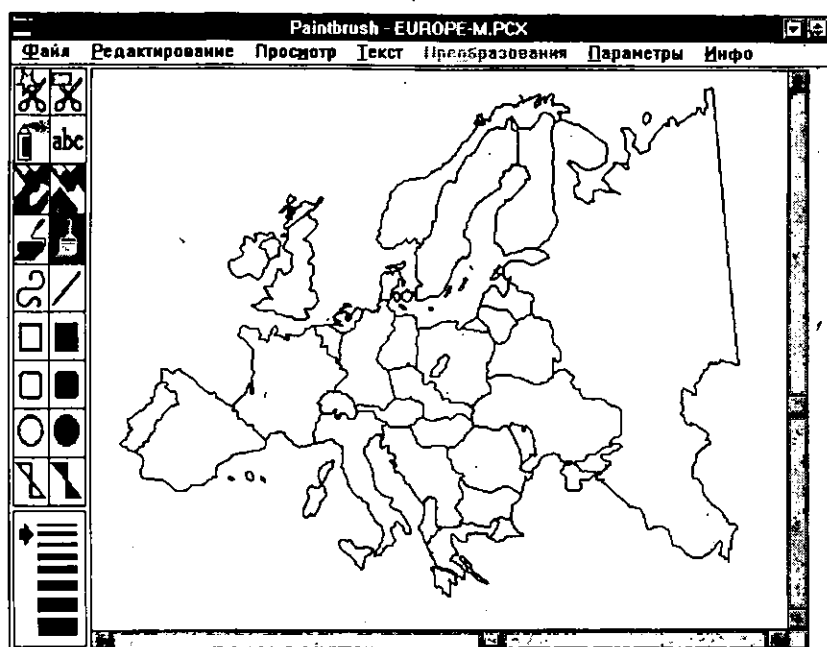


Рис. 8.26. Исходное изображение в окне графического редактора Microsoft Paintbrush.

Второй исходный компонент — таблица. На рис. 8.27 показаны структура и фрагмент таблицы MAP. Вы видите, что эта таблица содержит поле дат с названием "Дата", в которое занесены даты. Остальные поля соответствуют названию тех из столиц, которые приведены в форме (рис. 8.25).

Структура таблицы [MAP.DBF]					
Имя: MAP.DBF		Тип: DBASE			
Обновление: 08.07.95		Использовано байт: 44			
Записи: 4		Осталось байт: 32.723			
№	Имя	Тип	Размер	Дробная	Индекс
1	ДАТА	Дата	8	0	По возрастанию
2	АФИНЫ	Текстовый	5	0	Нет
3	ЛОНДОН	Текстовый	5	0	Нет
4	МАДРИД	Текстовый	5	0	Нет
5	МОСКВА	Текстовый	5	0	Нет
6	РИМ	Текстовый	5	0	Нет
7	ПАРИЖ	Текстовый	5	0	Нет
8	ХЕЛЬСИНКИ	Текстовый	5	0	Нет

Таблица [MAP.DBF]								
№	ДАТА	АФИНЫ	ЛОНДОН	МАДРИД	МОСКВА	РИМ	ПАРИЖ	ХЕЛЬСИНКИ
1	07.01.95	21-23	18-19	24-26	20-22	21-23	22-23	18-20
2	07.02.95	21-23	20-21	24-27	20-22	23-24	23-24	18-20
3	07.03.95	23-26	20-21	25-27	22-23	23-24	23-24	22-24
4	07.04.95	23-25	22-24	26-27	22-24	19-20	19-20	23-24

Рис. 8.27. Структура и фрагмент таблицы MAP.

После того, как вы определились с исходными компонентами вы можете приступить к созданию формы.

1. Начните с того, что создайте новую форму для таблицы MAP.
2. Разместите в форме графический объект, а затем определите его, воспользовавшись любым из изложенных выше способов.
3. Разместите в форме все необходимые надписи. В данном случае мы самостоятельно нанесли на карту название столиц и места их расположения (весьма приблизительно). Кроме того, вы можете поместить заголовок формы или любые другие надписи.
4. Разместите в форме все требуемые поля таблицы. В данном случае были размещены все поля исходной таблицы. Размещение полей будет рассмотрено далее в настоящей главе.
5. В заключение по своему усмотрению измените шрифты, цвет и другие атрибуты. Завершив работу на формой, нажмите кнопку Запуск на панели инструментов и вы увидите форму, подобную той, что изображена на рис. 8.25.

Использование графического объекта значительно улучшает восприятие информации. Если вы ограничите пользователя таблицей, данные, с которыми он будет работать останутся теми же, однако, работать с информацией, представленной в таком виде, неинтересно. Кроме цифровых данных вы можете самостоятельно поместить в форму дополнительные графические объекты, которые могут отображать солнечную погоду, пасмурную, дождь и т.д. Тщательно продумывая интерфейс ваших приложений вы можете сделать работу пользователя менее утомительной и даже приятной.

Размещение OLE-объекта

OLE-объект (Object Linking and Embedding) используется при обмене данными с другими приложениями, работающими под Windows. Механизм работы с составными документами (OLE) вместе с динамическим обменом данными (DDE) будет подробно рассмотрен в главе 16.

Размещение кнопки

В своей работе с различными приложениями, работающими под Windows, вам приходится пользоваться кнопками в самых различных ситуациях. dBASE в этом смысле не является исключением. Кнопки, расположенные на панели инструментов, кнопки раскрытия всевозможных списков, кнопка Поиск и многие другие кнопки, без которых вы просто не сможете эффективно взаимодействовать с dBASE.

Вы можете разместить стандартную кнопку и в своей собственной форме. Однако у вас должен возникнуть вопрос: как задать инструкции, которые должны выполняться при нажатии кнопки? Вы можете сделать это только с использованием языка dBASE и макроопределений. *Макроопределение* представляет собой тип данных, предназначенный для хранения небольших групп команд или выражений. Макроопределения заключаются в фигурные скобки ({}). Поскольку рассмотрение языка dBASE выходит за рамки настоящей книги, а макроопределения будут рассмотрены во второй части книги, в данной главе мы не будем останавливаться на размещении стандартных кнопок. Вместо этого рассмотрим размещение специальных кнопок управления, для которых уже определены действия, выполняемые при нажатии на них.

Размещение специальных объектов интерфейса

При создании форм вы можете использовать не только минимально необходимый набор стандартных объектов интерфейса, но и целый ряд специальных объектов. Для некоторых из них заданы не только отображение на экране, но и все связанные с ним действия. В качестве специальных объектов могут использоваться объекты, созданные в среде dBASE (с расширением .CC), объекты Visual Basic (с расширением .VBX) и динамические библиотеки (с расширением .DLL). Применение этих объектов интерфейса значительно расширяет возможности создаваемых форм.

Специальные объекты интерфейса находятся в карточке “Специальные” панели элементов. Для добавления новых специальных объектов интерфейса выполните команду **Файл | Выбор элементов управления**. Эта команда открывает окно диалога “Выбор элементов управления”, являющееся обычным окном диалога выбора файла, в котором вы задаете тип файла, его имя и расположение. В поставку dBASE входит файл **BUTTON.CC**, который содержит наиболее часто используемые в приложениях кнопки управления. Вы можете выбрать этот файл из каталога **SAMPLES**, нажать кнопку **ОК**, в результате чего в карточке “Специальные” появятся объекты управления, содержащиеся в этом файле (рис. 8.28).

Существует альтернативный способ загрузки специальных элементов управления с помощью навигатора. Для этого в окне навигатора выберите категорию “Специальные”, а в поле ввода расширений для выборки файлов напечатайте шаблон поиска (например, *.CC, *.VBX). В области “Специальные файлы” появится список всех файлов с указанным расширением. Выберите требуемый файл и выполните любое из следующих действий:

- Выполните команду **Навигатор | Загрузить элементы управления основного меню**

- Выполните команду Навигатор | Загрузить элементы управления контекстного меню
- Нажмите клавишу *F2*
- Дважды нажмите кнопку мыши
- Перенесите с помощью мыши файл со специальными объектами интерфейса на рабочую поверхность dBASE или в окно команд

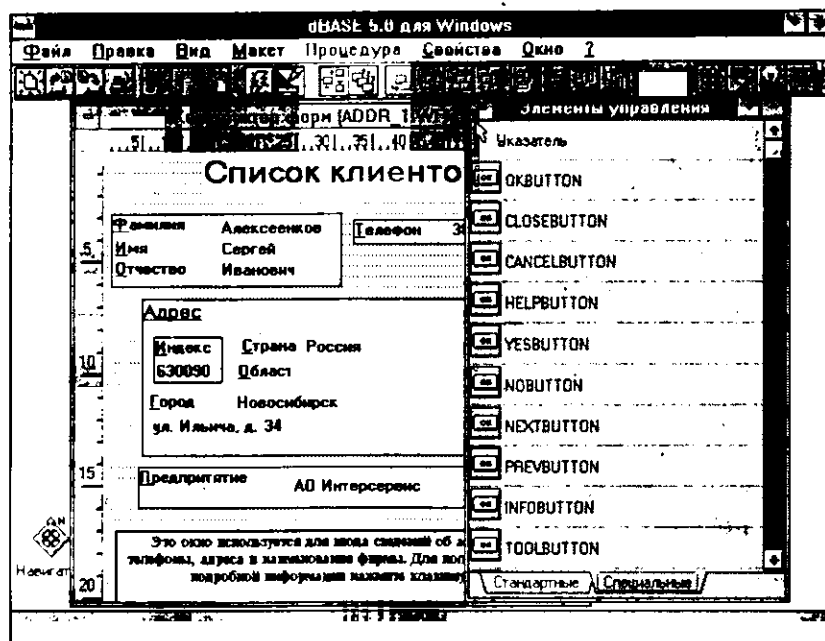


Рис. 8.28. Специальные объекты интерфейса.

Пример 8.4.

В данном примере вам предстоит добавить специальные кнопки управления записями в форму, созданную в примере 8.1.

1. Откройте форму ADDR_1.FSL, которую вы создали в примере 8.1 с помощью эксперта.
2. Измените расположение полей и текстовые поля так, как показано на рис. 8.28.
3. Разместите в форме специальные кнопки перехода на следующую и предыдущую записи. Для этих кнопок определены действия при нажатии на кнопку мыши, поэтому от вас не потребуется никакого программирования этих кнопок.
4. Разместите в форме специальную кнопку закрытия формы. Откройте карточку "События" окна свойств объекта и обратите внимание, что для события OnClick определено макроопределение.

5. Завершив работу на формой, сохраните ее, затем нажмите кнопку **Запуск** на панели инструментов и вы увидите форму, подобную той, что изображена на рис. 8.29.

Рис. 8.29. Использование специальных кнопок.

Группировка объектов

В dBASE имеется объект типа **Рамка группы**, который может использоваться для логического объединения нескольких объектов формы. Логическое объединение улучшает читабельность формы, подчеркивая общность сгруппированных объектов. Кроме того, при создании группы переключателей рамка группы используется и для выбора одного из значений и присвоения его полю таблицы. Рамка группы представляет собой прямоугольник, имеющий заголовок в левом верхнем углу. Для размещения рамки группы в форме выберите кнопку **Рамка группы** на панели элементов, установите указатель мыши на то место, где вы хотите создать рамку, и удерживая кнопку мыши в нажатом состоянии, переместите курсор по диагонали так, чтобы получилась рамка необходимого вам размера.

Для задания заголовка рамки выберите свойство **Text** и введите в поле ввода этого свойства требуемый текст. Для рамки можно задать не только заголовок, но и фон рамки.

0 - Отсутствует	
1 - Косая (///)	
2 - Сетка (###)	
3 - Сетка (XXX)	
4 - Косая (\\)	
5 - Горизонтальная	
6 - Вертикальная	

По умолчанию фон однотонный, но вы можете воспользоваться любым из доступных в dBASE оформлений. Для этого выберите свойство **PatternStyle** категории **Оформление** и укажите подходящее оформление в раскрываемом списке.

- 0 - Стандартный
- 0 - Стандартный
- 1 - Приподнятый
- 2 - Утопленный

Аналогичным образом вы можете выбрать и тип границы рамки. Для этого также выберите наиболее приемлемое значение свойства **BorderStyle** из раскрывающегося списка.

Пример объединения элементов в группу приведен на рис. 8.30.



Если рамка группы располагается поверх группы объектов, входящих в группу, то эти объекты будут невидимы. Для их отображения на экране выберите рамку, а затем выполните команду **Макет | На задний план**.

Рис. 8.30. Объединение группы объектов интерфейса.

Размещение полей таблицы

Основное назначение экранных форм состоит в том чтобы, обеспечить доступ к данным, содержащимся в таблицах dBASE. Прежде чем приступить к размещению полей таблицы, вы должны связать с формой источник данных, в качестве которого может использоваться отдельная таблица или созданный ранее запрос. Для этого выберите форму в качестве текущего объекта и в окне свойств объекта определите значение свойства **View**. Если вы помните точное название таблицы или запроса, можете ввести его непосредственно в поле ввода значения. В противном случае нажмите на кнопку **Инструмент** и выберите имя файла с помощью стандартного окна диалога выбора файла.

Для размещения полей таблицы в форме используется целый ряд кнопок:



Поле ввода



Флажок



Поле со списком



Переключатель



Счетчик



Редактор



Список



Горизонтальный регулятор



Вертикальный регулятор



Рисунок

Вы можете подобрать для каждого типа поля соответствующую ему тип объекта интерфейса. Для размещения поля в форме выберите объект из панели элементов, поместите его в требуемое место и определите размер объекта. После этого необходимо связать объект с источником данных. Для этого выберите свойство **DataLink** и определите для него имя поля таблицы. При определении поля таблицы вы можете использовать связанный с данным полем инструмент.



Источниками данных этих объектов интерфейса могут быть не только поля таблиц, но и значения переменных, массивов, данные из файлов и структура таблицы.

Ваши дальнейшие действия по определению поля зависят от его типа, поэтому рассмотрим отдельно определение полей каждого типа.

Определение полей ввода

Поля ввода можно связывать с полями любого вида. Наиболее часто они используются для ввода текстовых и числовых полей. Для того чтобы определить поле ввода, выберите кнопку **Поле ввода** на панели элементов, поместите в нужное место формы и установите размеры поля.

Связывание поля ввода с полем таблицы

Для связывания поля ввода с полем таблицы вы предварительно должны убедиться в том, что для данной формы уже определено значение свойства **View**. Далее выберите свойство **DataLink** в окне свойств и нажмите кнопку **Инструмент**. На экране появится окно диалога выбора поля (рис. 8.31). Область "Таблицы" содержит список таблиц, определенных свойством **View**. Для многотабличного запроса эта область содержит список всех таблиц, используемых при создании запроса. В случае выбора таблицы при определении свойства **View**, область "Таблицы" содержит имя этой таблицы.

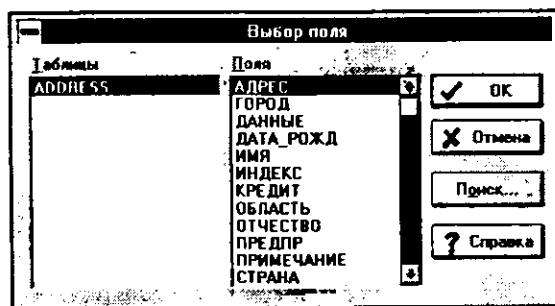


Рис. 8.31. Окно диалога выбора поля.

Область "Поля" содержит список полей выбранной таблицы. Выберите требуемое поле и нажмите кнопку ОК в окне диалога. Имя выбранного поля будет присвоено свойству **DataLink**. Теперь значения, вводимые в данное поле формы, автоматически будут сохраняться в соответствующем поле таблицы.



После размещения поля ввода желательно добавить рядом с ним текстовое поле, которое является для него надписью.

Определение значения по умолчанию

Свойство **Value** определяет значение поля, которое будет присваиваться ему при добавлении новой записи. Вы можете ввести значение этого свойства непосредственно в поле ввода значения данного свойства. Для числовых полей вы можете использовать счетчик, связанный с этим свойством.

Определение доступа к полю

В форме часто используются поля, значения которых могут быть просмотрены, но не могут быть изменены. Например, при изменении адреса клиента вы можете отображать его фамилию, имя и отчество, при этом запретив их редактирование. Для этого установите свойство **Enabled** категории **Доступ** равным (.F.). В этом случае переход на это поле будет запрещен, и вы можете только просматривать его значение.

Категория **Доступ** содержит еще одно свойство **TabStop**, которое управляет переходом на это поле с помощью клавиш **Tab** и **Shift+Tab**. Если вы установите значение этого свойства равным (.F.), то не сможете перейти на это поле нажатием клавиш **Tab** и **Shift+Tab**. Тем не менее для перехода на это поле можно воспользоваться мышью. Данное свойство, как правило, используется для кнопок управления, позволяя вам избежать лишнего перехода на них при вводе данных.

Определение шрифта и оформления поля

Шрифт поля определяется так же, как и для текстового объекта. Категория **Оформление** также аналогична одноименной категории текстового объекта, но в отличие от нее содержит два цвета. Один из них, **ColorHighLight**, используется для выбранного поля, (т.е. поля на который установлен фокус), а другой цвет, **ColorNormal**, — для остальных полей.

Автоматическое выделение текущего элемента

При переходе на поле ввода dBASE устанавливает курсор в начало поля, предоставляя вам возможность редактировать его значение. Этот режим установлен по умолчанию. Если же вам необходимо каждый раз вводить новое значение в поле, этот режим будет не очень удобен, поскольку вам необходимо предварительно выделить все поле и удалить из него данные. В этом случае выберите свойство **SelectAll** категории **Отображение** и установите его значение равным (.T.). В дальнейшем при переходе на это поле значение поля будет выделено, и при вводе первого же символа все выделенное значение удаляется.

Пример 8.5.

В данном примере вам предстоит создать новую форму и поместить в нее поле ввода.

1. Выполните команду **Файл | Создать | Форма** для создания пустой формы.
2. Определите для формы свойство **View**. Используйте в качестве источника данных формы таблицу **ADDRESS**.
3. Разместите в форме поле ввода. Определите свойство **DataLink**, с помощью которого свяжите это поле с полем "Фамилия" таблицы **ADDRESS**.
4. Разместите в форме надпись к полю ввода, которая содержит текст **Фамилия**.
5. Завершите на этом работу над формой и сохраните ее. Затем нажмите кнопку **Запуск** на панели инструментов, в результате чего на экране появится форма, подобная той, что изображена на рис. 8.32.

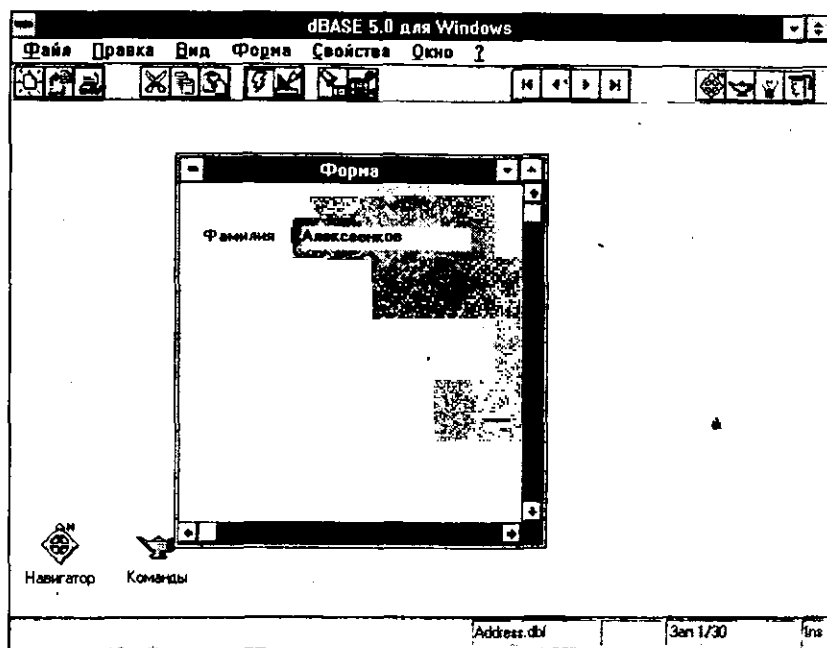


Рис. 8.32. Размещение в форме поля ввода.

Определение полей типа Редактор

Для редактирования символьных полей большого размера и Мемо-полей очень удобны объекты типа **Редактор**. Кроме того, вы можете использовать это поле для ввода текстовых файлов.

Для размещения объекта в форме выберите кнопку **Редактор** на панели элементов, установите указатель мыши на то место, где вы хотите добавить поле, и удерживая кнопку мыши в нажатом состоянии, переместите курсор по диагонали так, чтобы получилась рамка требуемого размера.

Связывание с полем таблицы или файлом

Для связывания поля с полем таблицы или файлом выберите свойство **DataLink** и нажмите кнопку **Инструмент** в правой части поля ввода значения свойства. На экране появляется окно диалога "Выбор связи данных" (рис. 8.33). Из раскрывающегося

списка Тип выберите значение "Поле" или "Файл". Далее используйте кнопку Инструмент для определения значения поля Источник данных.

При выборе типа "Поле" для выбора источника данных используется окно диалога "Выбор поля", с которым мы уже познакомились в предыдущем разделе при связывании поля ввода. При выборе типа "Файл" для выбора источника данных используется окно диалога "Выбор источника данных", которое представляет собой обычное окно выбора файла.

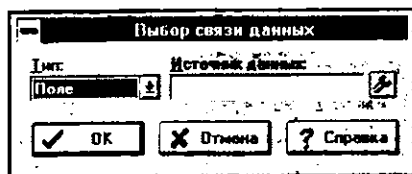


Рис. 8.33. Окно диалога "Выбор связи данных".

Определение параметров редактирования

Цвет и шрифт объекта, а также управление доступом к объекту определяются аналогично действиям, рассмотренным ранее. Особенностью объекта типа Редактор является возможность прокручивания больших текстов, если они не помещаются в поле. Для прокручивания текста вы можете использовать стрелки вверх и вниз, а также полосы прокрутки. По умолчанию для поля редактирования определены полосы прокрутки и они отображаются снизу и справа от области ввода поля. Для управления отображением полос прокрутки используется свойство ScrollBar категории Оформление, которое может принимать одно из следующих значений:

Значение	Описание
Отображать	Полосы прокрутки отображаются всегда.
Не отображать	Полосы прокрутки не отображаются никогда.
Автоматически	Полосы прокрутки отображаются только в том случае, если текст не помещается в поле.
Недоступны	Полосы прокрутки отображаются, но недоступны для управления.

По умолчанию текст в поле редактирования отображается строка за строкой в том виде, в каком он хранится в поле таблицы или файле. Если длина текста превышает ширину поля редактирования, часть строки будет скрыта и вам придется прокручивать текст для его просмотра. Вместо этого вы можете установить признак автоматического переноса строк, присвоив для этого свойству Wrap категории Отображение значение, равное (.T.). На рис. 8.34 представлены оба режима просмотра текста.

Для запрета редактирования поля ввода мы использовали свойство Enabled. Для поля редактирования вы также можете использовать это свойство. Однако в данном случае использование этого свойства имеет свои недостатки. Если текст не помещается в поле редактирования, то вы не сможете перейти на него и просмотреть содержимое поля. Поэтому для поля редактирования определено новое свойство Modify, значение (.T.) которого разрешает редактирование поля, а значение (.F.) запрещает его редактирование, но позволяет вам переходить на это поле и прокручивать его для просмотра.

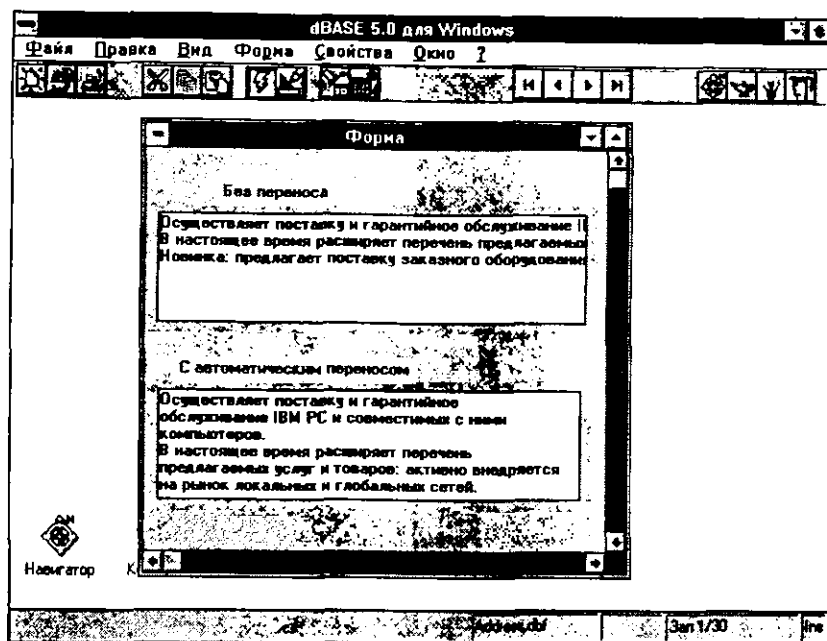


Рис. 8.34. Автоматический перенос текста.

Определение полей типа Рисунок

В настоящей главе вы уже познакомились с размещением рисунков из файлов в формате .PCX или .BMP, а также из файла ресурсов. Графическое поле типа **Рисунок** добавляется аналогичным образом, но при выборе источника для поля вы должны указать "Двоичное поле", а затем выбрать требуемое поле из списка полей таблицы двоичного типа.

Пример 8.6. Размещение полей редактирования и рисунка

В данном примере мы продолжим создание формы ввода списка клиентов. Вам предстоит разместить в ней поле редактирования и рисунок.

1. Выполните команду **Файл | Открыть** из основного меню и откройте форму ADDR_1.WFM, созданную в примере 8.4.
2. Удалите из формы текстовый объект, созданный в форме в примере 8.2 и разместите в ней поле редактирования. Свяжите это поле с полем таблицы "Примечание".
3. Добавьте в форму графическое поле и свяжите его с полем "Фотография" таблицы ADDRESS. Это поле содержит фотографии клиентов.
4. Теперь расположите поля таблицы в форме в удобном для ввода месте и определите их размеры.
5. Результат выполнения этих действий представлен на рис. 8.35. Сохраните форму под именем ADDR_2.WFM и закройте ее.

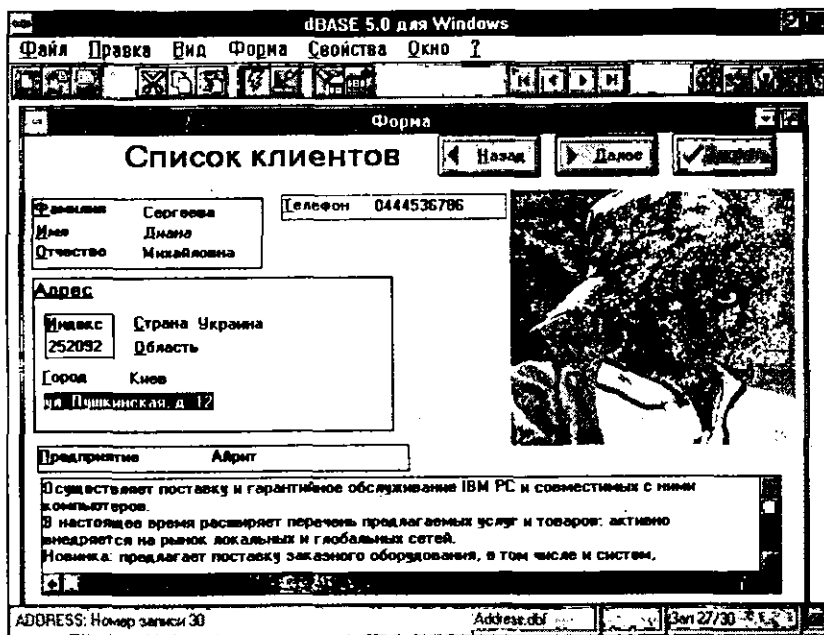


Рис. 8.35. Размещение в форме полей редактирования и рисунка.

Определение Флажка

Для индикации состояния, которое может иметь только одно из двух допустимых значений используются флажки. Они могут использоваться по одному или небольшими группами. Например, вы можете добавить в таблицу ADDRESS поле "Постоянный", которое будет содержать значения (.T.) для постоянных клиентов и (.F.). При установке флажка значение будет соответствовать (.T.), а при снятии установки - значению (.F.).



Флажки могут связываться только с логическими полями таблиц.

Пример 8.7. Создание объекта типа Флажок

В данном примере создадим объект типа Флажок для редактирования поля "Постоянный" таблицы ADDRESS, которое может принимать значения (.T.) или (.F.).

1. Выполните команду **Файл | Открыть** из основного меню и откройте форму ADDR_1.WFM.
2. Выберите кнопку **Флажок** на панели инструментов, установите курсор в форме в удобном для ввода месте и определите размеры создаваемого объекта.
3. Свяжите объект с полем "Постоянный" таблицы ADDRESS. Это поле содержит признак постоянного клиента.

4. Установите свойство Text объекта равным символьной строке "Постоянный". Этот текст служит для пояснения к полю и по существу является надписью к полю.
5. На этом работу завершите над формой и сохраните ее с именем ADDR_3.WFM. Нажмите кнопку Запуск на панели инструментов и вы увидите форму, подобную той, что изображена на рис. 8.36.

Рис. 8.36. Размещение в форме поля типа Флажок.

Определение группы переключателей

Объекты типа **Переключатель** позволяют вам выбрать одно из допустимых значений и присвоить его связанному с этим объектом полю таблицы. В английском варианте этот объект называется **radio button**. Название объекта произошло от кнопок выбора, используемых в радиоприемниках и даже их вид напоминает об этом. Переключатели широко используются как в dBASE, так и в любых других приложениях Windows.

Для создания группы переключателей вы размещаете в форме количество переключателей, равное количеству вариантов допустимых значений, а затем объединяете их в группу с помощью объекта **Рамка группы**. Каждый переключатель группы должен быть связан с одним и тем же полем таблицы. Для первого элемента группы установите свойство **Group** категории **Доступ** равным (.Т.), а для всех остальных элементов — равным (.F.).



Группы переключателей могут связываться только с текстовыми полями таблиц.

Пример 8.8. Создание группы переключателей

Допустим, что вы ввели в таблицу ADDRESS поле "Тип_связи", которое может принимать одно из значений: Телефон/Факс/Телекс. В данном примере вам предстоит создать группу переключателей для просмотра и редактирования этого поля.

1. Выполните команду **Файл | Открыть** из основного меню и откройте форму ADDR_3.WFM, созданную в предыдущем примере.
2. Выберите кнопку **Переключатель** на панели элементов, установите курсор в форме в удобном для ввода месте и определите размеры создаваемого переключателя.
3. Свяжите объект с полем "Тип_связи" таблицы ADDRESS.
4. Установите свойство **Text** объекта равным символьной строке "Телефон". Этот текст служит для пояснения к полю и по существу является надписью к полю.
5. Аналогичным образом создайте переключатели для определения значений "Факс" и "Телекс".
6. Перенесите поле ввода значения телефона ниже переключателей и удалите надпись к этому полю.
7. Объедините переключатели и поле ввода номера телефона в одну группу. Для этого разместите в форме рамку группы таких размеров, чтобы все переключатели и поле ввода оказались внутри нее. Определите для рамки группы заголовок "Связь с клиентом". Для размещения рамки группы на заднем плане выполните команду основного меню **Макет | На задний план**.

Рис. 8.37. Размещение в форме группы переключателей.

8. На этом завершите работу на формой и сохраните ее под именем ADDR_4. Нажмите кнопку **Запуск** на панели инструментов и вы увидите форму, подобную той, что изображена на рис. 8.37.

Определение поля со списком

В dBASE для Windows существуют несколько вариантов отображения одного и того же поля на экране. Например, вы можете отобразить поле "Тип_связи" как в виде группы переключателей, так и в виде поля со списком. Выбор типа объекта определяется вашими пожеланиями и наличием свободного места на экране.

Для размещения в форме поля со списком выберите на панели элементов кнопку **Поле со списком**, установите курсор мыши в требуемое место и определите размеры объекта. Связь с полем таблицы осуществляется как обычно с помощью свойства **DataLink**.

Допустимый список значений, которые пользователь может выбирать при вводе значения поля, определяется с помощью свойства **DataSource**. Для определения значения этого поля вы можете использовать кнопку **Инструмент**, при нажатии на которую на экране появляется окно диалога "Выбор источника данных" (рис. 8.38). Прежде всего из раскрывающегося списка **Тип** вы выбираете тип источника данных. В данном разделе нас интересует возможность использования в качестве источника данных поля таблицы. Остальные типы источников данных, как правило, требуют использования языка программирования dBASE и в данной книге рассматриваться не будут.

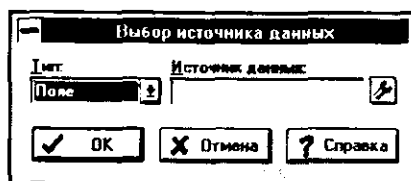


Рис. 8.38. Окно диалога "Выбор источника данных".

При выборе типа "Поле" вы можете выбрать таблицу, а затем поле этой таблицы, данные из которого будут отображаться в поле со списком. Необходимо помнить разницу между полем таблицы, связанным с полем со списком, и полем таблицы, используемым в качестве источника данных. При изменении значения поля со списком, результат сохраняется в связанном поле таблицы. Значение источника данных не может редактироваться и доступно только для чтения. Мы используем данные из поля таблицы, являющегося источником данных, и записываем их в поле таблицы, связанное с полем со списком. Например, в качестве источника данных поля со списком, связанного с полем "Тип_связи" таблицы ADDRESS может использоваться поле "Тип_связи" таблицы CONNTYPE.



Использование поля со списком позволяет настраивать список возможных вариантов. В случае необходимости вы можете добавить новые значения в таблицу, используемую в качестве источника данных. При этом в форме ввода изменится количество возможных вариантов, и это не потребует изменения самой формы.

Вы можете выбрать один из трех возможных стилей отображения списка, краткое описание которых приведено ниже, а внешний вид показан на рис. 8.39.

Стиль	Описание
Простой	Это стандартный стиль, который состоит из текстового поля ввода и раскрывающегося списка. Данный стиль можно использовать для отображения списка с небольшим количеством вариантов.
Раскрывающийся	Раскрывающееся поле со списком отображается в виде текстового поля ввода. Для раскрытия списка необходимо нажать стрелку, расположенную правее поля. Этот стиль экономит место в форме.
Список	Данный стиль отличается от предыдущего только тем, что не позволяет вам вводить данные, которые отсутствуют в источнике данных.

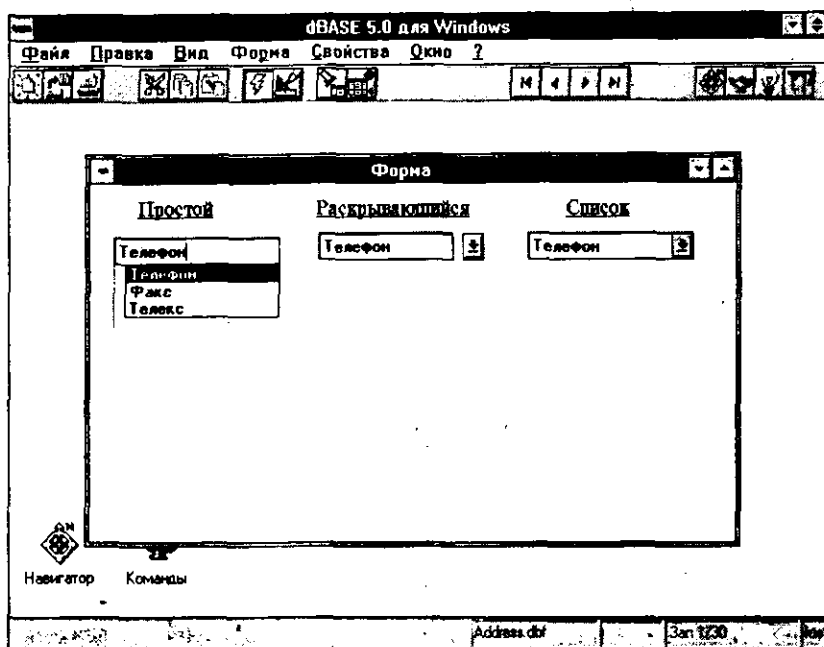


Рис. 8.39. Стили поля со списком.

Свойство **Sorted** категории **Связывание данных** предназначено для отображения значений списка в отсортированном порядке. Отсортированные значения появляются по возрастанию в алфавитном, числовом или хронологическом порядке.

Пример 8.9. Создание объекта типа “Поле со списком”

В данном примере создадим объект типа **Поле со списком** для редактирования поля “**Тип_связи**”, которое может принимать одно из значений: **Телефон/Факс/Телекс**. В качестве источника данных будем использовать таблицу **CONNTYPE**, которая содержит наименования всех типов связей. Если потребуется добавить новый тип связи, например, электронную почту, вам достаточно будет добавить в эту таблицу новую запись.

1. Выполните команду **Файл | Открыть** из основного меню и откройте форму ADDR_4.WFM, созданную в примере 8.8.
2. Удалите созданную в предыдущем примере группу переключателей, но сохраните поле ввода номера телефона.
3. Выберите на панели элементов кнопку **Поле со списком**, установите курсор в форме в удобном для ввода месте и определите размеры создаваемого поля.
4. Свяжите объект с полем "Тип_связи" таблицы ADDRESS.
5. Определите в качестве источника данных поле "Тип_связи" таблицы CONNTYPE.
6. Выберите для свойства **Style** стиль **Раскрывающийся**.
7. Сохраните форму под именем ADDR_5. Нажмите кнопку **Запуск** на панели инструментов и вы увидите на экране форму, подобную той, что изображена на рис. 8.40.

The screenshot shows a dBASE 5.0 for Windows window with a menu bar (Файл, Правка, Вид, Форма, Свойства, Окно, ?) and a toolbar. The main form is titled 'Список клиентов' and has a title bar 'Форма'. It contains several fields and buttons:

- Buttons:** 'Назад', 'Далее', 'Закрыть'.
- Fields:**
 - Фамилия:** Алексеев
 - Имя:** Сергей
 - Отчество:** Иванович
 - Предприятие:** АО Интерсервис
 - Адрес:** A list of addresses, including 'ул. Ильича, д. 34'.
 - Факс:** A list of phone numbers, including '3832243124'.
- Form Content:**
 - Адрес:** A list of addresses, including 'ул. Ильича, д. 34'.
 - Факс:** A list of phone numbers, including '3832243124'.
 - Text:** 'Осуществляет поставку и гарантийное обслуживание IBM PC и совместимых с ними компьютеров. В настоящее время расширяет перечень предлагаемых услуг и товаров: активно внедряется на рынок локальных и глобальных сетей.'

Рис. 8.40. Размещение в форме поля со списком.

Определение полей для ввода числовых значений и дат

Ввод числовых значений и дат может осуществляться с помощью поля ввода. Кроме того, для ввода данных этих типов вы можете использовать счетчики, а также вертикальный и горизонтальный регуляторы.

Счетчик

Счетчик удобен при вводе значений, которые имеют заданный диапазон изменения и определенную дискретность. Счетчик отличается от поля ввода наличием

120000

справа от поля ввода кнопок, которые могут быть использованы для изменения значения поля. Кроме того, вы можете ввести значение как и в обычное поле ввода.

Для размещения в форме счетчика выберите на панели элементов кнопку **Счетчик**, установите курсор мыши в требуемое место и определите размеры объекта. Далее свяжите его с полем таблицы, используя свойство **DataLink**. Для счетчика вы можете определить дискретность изменения значения (свойство **Step**), а также нижнюю (свойство **RangeMin**) и верхнюю (свойство **RangeMax**) границу допустимого диапазона значений.

По умолчанию dBASE позволяет непосредственно вводить данные в поле счетчика, которые не обязательно должны совпадать с допустимым шагом изменения значения. Если вы уверены в том, что в поле будут вводиться значения с заданной дискретностью, вы можете запретить непосредственный ввод в поле счетчика. В этом случае для ввода можно использовать только кнопки изменения значения счетчика. Для этого установите свойство **SpinOnly** равным (.T.).

Ползковые регуляторы

Ползковые регуляторы в отличие от счетчика не допускают ввод с клавиатуры и для них не определен шаг изменения значения. Они, как и счетчик, имеют допустимый диапазон изменения значений и могут связываться с полем таблицы.

Пример 8.10. Создание счетчика для отображения числовых данных

В данном примере создадим объект типа **Счетчик** для ввода и редактирования значения кредита.

1. Выполните команду **Файл | Открыть** из основного меню и откройте форму **ADDR_5.WFM**, созданную в примере 8.9.
2. Выберите на панели элементов кнопку **Счетчик**, установите курсор в форме в удобном для ввода месте и определите размеры создаваемого поля.
3. Свяжите объект с полем "Кредит" таблицы **ADDRESS**.
4. Установите диапазон изменения значения счетчика от 100 000 до 1 000 000 и шаг изменения значения равным 100 000.
5. Определите надпись "Кредит" для создаваемого поля.
6. На этом завершите работу над формой и сохраните ее с именем **ADDR_6**. Далее нажмите кнопку **Запуск** на панели инструментов и вы увидите форму, подобную той, что изображена на рис. 8.41.

Определение вычисляемых полей

Вычисляемые поля предназначены для вычислений над данными, содержащимися в полях исходной таблицы. При вычислениях могут быть использованы поля различных типов (текстовые, числовые, даты), однако в выражение для вычисления должны одновременно входить поля только одного типа.

Рис. 8.41. Размещение в форме счетчика.

Для того чтобы отобразить в форме вычисляемое поле, вам необходимо предварительно создать запрос, который будет содержать данное вычисляемое поле, а затем выбрать этот запрос в качестве источника данных формы.



Сложные выражения для вычисления могут содержать некоторые из функций языка dBASE. При этом вы можете ввести в вычисляемое поле любое выражение языка dBASE, результатом которого является отдельное значение.



Вычисляемые поля являются временными полями. Содержащиеся в них результаты не сохраняются и существуют на экране до тех пор, пока отображается содержащая их форма.

С созданием вычисляемых полей в запросах вы познакомились в главе 7. Теперь рассмотрим несколько примеров использования вычисляемых полей в формах.

Пример 8.11. Определение стоимости партии проданных товаров

В данном примере обратимся к таблице ORDSALE, которая содержит данные о каждой партии проданных товаров:

- индекс или код товара (поле "Код_товара");
- количество товаров, проданных в каждой партии (поле "Продано");
- цена единицы изделия (поле "Цена").

Определим стоимость каждой партии проданных товаров.

1. Создайте запрос, в котором определите вычисляемые поля для расчета стоимости каждой партии проданных товаров. Сохраните запрос под именем OS_COST.

- Создайте новую форму. В качестве источника данных формы (свойство View) используйте созданный на предыдущем шаге запрос OS_COST.
- Разместите поля в форме, как показано на рис. 8.42.

Конструктор форм [OS_COST.WFM]

Учет заказов и продаж

Покупатель Товар

Цена

Заказано		
Дата	Количество	Стоимость
02.04.95	5	3050.00

Продано		
Дата	Количество	Стоимость
12.04.95	5	3050.00

События Методы

Рис. 8.42. Размещение в форме полей запроса.

Форма

Учет заказов и продаж

Покупатель Товар

Цена

Заказано		
Дата	Количество	Стоимость
13.02.95	500	302500.00

Продано		
Дата	Количество	Стоимость
28.02.95	450	272250.00

Order.dat 30.06/92

Рис. 8.43. Просмотр информации о заказах и продажах покупателей, включая стоимость каждой партии.

4. Сохраните форму под именем OS_COST, после чего перейдите в режим просмотра формы.

Законченная форма показана на рис. 8.42, а на рис. 8.43 — ее использование для просмотра данных.

Пример 8.12. Определение вычислений над текстовыми полями

В данном примере создадим форму, предназначенную для просмотра данных в таблице ADDRESS, в которой в одно поле объединены фамилия, имя и отчество клиента, а в другое поле — компоненты его адрес.

1. Создайте запрос, в котором определите одно вычисляемое поле для объединения фамилии, имени и отчества клиента, а в другое поле — для объединения компонент адреса. Сохраните запрос под именем ADDR_F.
2. Создайте новую форму. В качестве источника данных формы (свойство View) используйте созданный на предыдущем шаге запрос ADDR_F.
3. Разместите в форме оба поля запроса.
4. Сохраните форму под именем ADDR_F.WFM, после чего перейдите в режим просмотра формы (8.44).

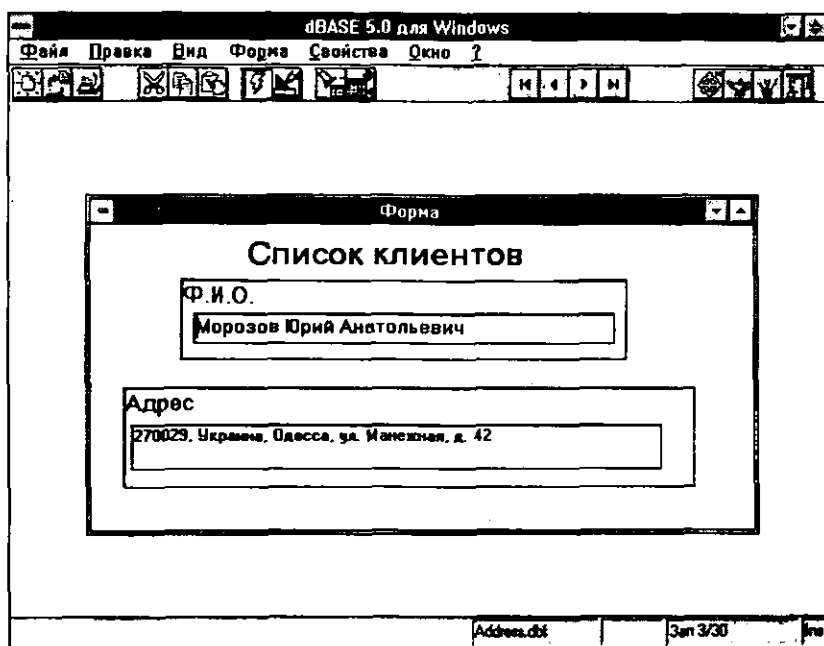


Рис. 8.44. Использование в вычислениях текстовых полей.

Размещение объекта типа “Список”

В настоящей главе мы рассмотрели поля со списком и использовали их для ввода данных в поля таблицы. В dBASE имеется еще один тип объекта интерфейса — Сп

сок, который также содержит перечень допустимых вариантов. Но в отличие от поля со списком вы не можете связать этот объект с полями таблицы. Поэтому его обработка требует использования языка dBASE.

В качестве источника для элементов списка могут использоваться:

- Массив
- Значения поля таблицы
- Список файлов текущего каталога
- Имена полей таблицы
- Список таблиц текущего каталога

Особенностью списка является возможность одновременного выбора нескольких элементов. Эта возможность может использоваться для обработки группы элементов, однако ее реализация невозможна без языка dBASE.

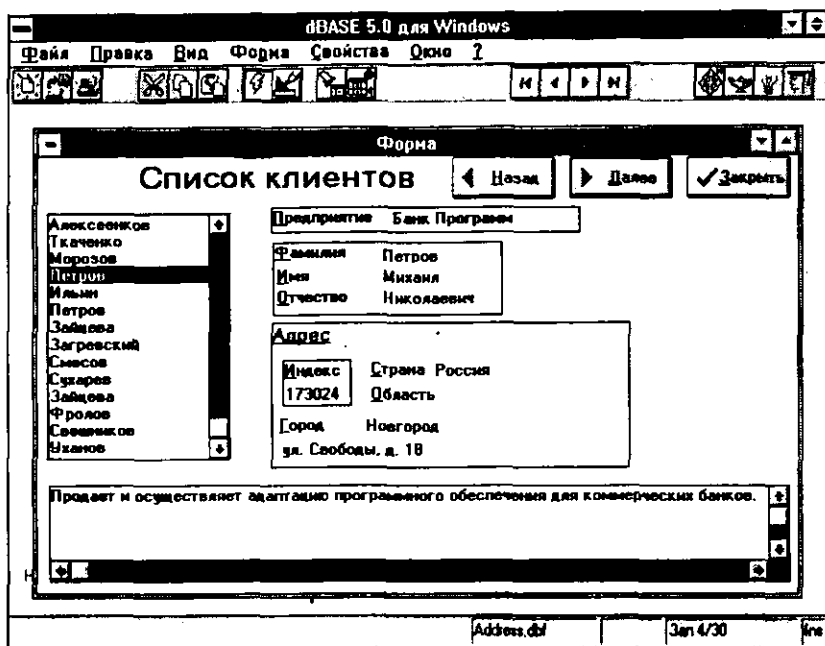


Рис. 8.45. Размещение в форме объекта типа Список.

Пример 8.13. Создание объекта типа “Список”

В данном примере определим объект типа Список для создания формы, в которой приведен список фамилий клиентов из таблицы ADDRESS, а также подробные сведения о каждом из них. При перемещении по списку изменяется номер записи, в результате чего изменяется информация о клиенте. Таким образом мы создаем средство для быстрого пролистывания сведений о клиентах.

1. Выполните команду **Файл | Открыть** из основного меню и откройте созданную ранее форму ADDR_1.WFM.

2. Выберите на панели элементов кнопку **Список**, установите курсор в форме в левом верхнем углу формы и определите размеры создаваемого поля.
3. Определите в качестве источника данных поле “**Фамилия**” таблицы **ADDRESS**.
4. Переместите остальные поля формы в правую часть формы.
5. Сохраните форму под именем **ADDR_7.WFM**, после чего нажмите кнопку **Запуск** на панели инструментов и вы увидите форму, подобную той, что изображена на рис. 8.45.

Размещение вспомогательной информации

В предыдущих разделах вы познакомились с размещением объектов интерфейса, связанных с полями таблицы и предназначенных для ввода в них данных. Кроме этого вы можете разместить в форме дополнительную информацию, которая может оказаться полезной при работе с формой. Например, вы можете показать в форме номер текущей записи, общее количество записей в таблице, текущую дату, время и т.д. В конструкторе форм отсутствуют средства для непосредственного размещения таких данных, однако вы можете использовать в качестве источника данных формы не таблицу, а запрос.

Включите в запрос все поля таблицы, которые вы предполагаете отображать в форме и некоторую дополнительную информацию. Для определения дополнительной информации используйте вычисляемые поля, содержащие вызовы соответствующих функций. Ниже приведен список некоторых из них:

Функция	Назначение
recno()	Определяет номер текущей записи.
reccount()	Определяет общее количество записей в таблице.
date()	Определяет текущую дату.
time()	Определяет текущее время.

Пример 8.14. Размещение вспомогательной информации

В данном примере разместим в форме номер текущей записи, общее количество записей в таблице, а также текущую дату и время.

1. Создайте для таблицы **ADDRESS** запрос, в котором установите метки выбора всех полей таблицы, за исключением поля “**Фотография**” и “**Звук**”.
2. Создайте в запросе вычисляемые поля для определения номера текущей записи, общего количества записей в таблице, текущей даты и текущего времени (рис. 8.46).
3. Проверьте правильность описания запроса, сохраните его под именем **ADD.QBE** и закройте окно запроса.
4. Выполните команду **Файл | Открыть** из основного меню и откройте форму **ADDR_4.WFM**, созданную ранее в этой главе.

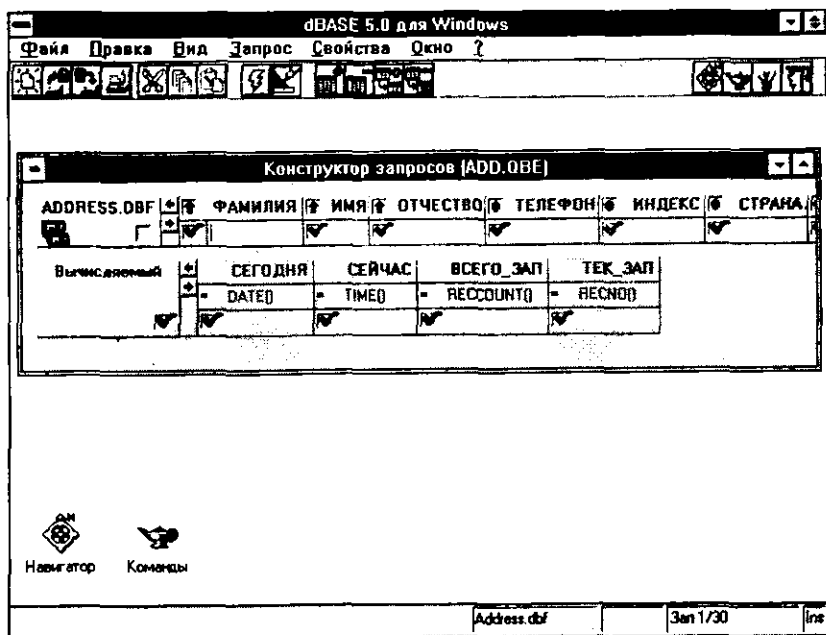


Рис. 8.46. Создание запроса для определения дополнительной информации.

5. Измените источник данных формы. Для этого выберите свойство **View** формы и выберите в качестве источника данных запрос, созданный на предыдущем шаге. Обратите внимание на то, что dBASE автоматически связал размещенные в поле объекты с новым источником данных.
6. Разместите в форме вычисляемые поля, созданные в запросе и определите для них соответствующие надписи.
7. Сохраните форму под именем ADDR_8, после чего нажмите кнопку **Запуск** на панели инструментов и вы увидите форму, подобную той, что изображена на рис. 8.47.

Удаление поля

Для удаления одного или нескольких полей выделите их, а затем выберите **Удалить** или **Вырезать** из меню **Правка**. Первая из команд удаляет поле из формы, а вторая — удаляет поле и помещает его в буфер обмена. Поэтому, если вы случайно удалили поле, воспользовавшись командой **Правка | Вырезать**, для его восстановления воспользуйтесь командой **Правка | Вставить**. Вместо команд из меню **Правка** вы можете воспользоваться кнопками панели элементов **Удалить в буфер** и **Вставить из буфера** или оперативными клавишами **Shift+Del** и **Ctrl+Ins**.

Изменения форматов полей

После размещения поля в форме вы можете изменить для него формат отображения данных. Для этой цели предназначены свойства **Function** и **Picture**. Оба эти свойства содержат кнопку **Инструмент**, при выборе которой вызывается окно диалога "Выбор шаблона". Действия по определению форматов полей полностью совпадают с теми, которые были изложены в главе 4.

The screenshot shows the 'Forma' window in dBASE 5.0 for Windows. The title bar reads 'dBASE 5.0 для Windows'. The menu bar includes 'Файл', 'Правка', 'Вид', 'Форма', 'Свойства', and 'Ожю ?'. The toolbar contains various icons for file operations. The main window is titled 'Форма' and contains a form titled 'Список клиентов'. The form has several sections: 'Текущая запись' (Current record) with value 1 and 'Всего записей' (Total records) with value 30; 'Сегодня' (Today) with date 30.07.95 and 'Время' (Time) with value 12:06:19; 'Преимущество' (Advantage) with value 'АО Интерсервис'; 'Адрес' (Address) section with fields for 'Индекс' (Index) 630030, 'Страна' (Country) Россия, 'Область' (Region), 'Город' (City) Новосибирск, and 'ул. Ильича, д. 34'; 'Связь с клиентом' (Contact with client) section with radio buttons for 'Постоянный' (Permanent), 'Телефон' (Telephone), 'Телекс' (Telex), and 'Факс' (Fax), with the fax number 3832243124; and a text area at the bottom describing services: 'Осуществляет поставку и гарантийное обслуживание IBM PC и совместимых с ними компьютеров. В настоящее время расширяет перечень предлагаемых услуг и товаров: активно внедряется на рынок локальных и глобальных сетей.'

Рис. 8.47. Размещение в форме дополнительной информации.

Результат выбора формата сохраняется в обоих свойствах, независимо от того, из какого свойства вы вызвали окно диалога. Например, вы можете определить формат отображения номера телефона, в котором первые три цифры заключены в круглые скобки, а при отображении остальных семи цифр для их разделения используется дефис. Этот формат уже обсуждался в главе 4. Для вызова окна диалога "Выбор шаблона" нажмите кнопку **Инструмент** для любого из свойств **Function** или **Picture** и установите описанный выше формат. В результате определения формата свойство **Function** будет иметь значение "R" а свойство **Picture** — значение "(999) 999-99-99".

Размещение объекта-таблицы

Форма может содержать связанные или несвязанные таблицы. Таблицы, размещенные в форме, выглядят как обычные таблицы dBASE. Вы можете управлять их свойствами так же, как делали это в главе 4.

Основное применение объекты-таблицы находят в многотабличных формах. К обсуждению этого средства мы вернемся в главе 14.

Управление группами объектов

До настоящего момента мы настраивали свойства отдельных объектов. Такой подход оправдан до тех пор, пока вы работаете с отдельными объектами. В тех случаях, когда вам нужно изменить свойства сразу нескольких объектов, такой подход оказывается слишком трудоемким. В настоящем разделе мы остановимся на средствах, предназначенных для управления группами объектов.

Прежде чем вы приступите к настройке свойств группы объектов, выделите их так, как было показано ранее в настоящей главе в разделе "Выбор объектов". Далее перейдите в окно свойств. Обратите внимание, что свойства в этом случае приведены в алфавитном порядке и отсутствуют кнопки вызова инструментов. При изменении какого-либо из свойств, свойства всех объектов, которые содержат одноименное свойство, будут изменены.

Порядок расположения объектов по слоям и порядок обхода объектов формы

Обычно форма содержит несколько объектов интерфейса. Каждый из них размещается в своей плоскости, которая находится впереди или позади плоскости другого объекта. Можно сказать, что объекты расположены в форме слоями. Расположение объектов по слоям влияет на их отображение в форме и порядок обхода объектов формы.

Порядок расположения объектов по слоям называется расположением объектов по Z-оси макета. Этот порядок определяет, какой объект появляется впереди (сверху) других объектов. Даже если объекты расположены рядом друг с другом, каждый из них находится в своем собственном слое формы, т. е. каждый объект занимает единственную позицию в порядке расположения объектов по слоям. В предыдущих разделах вы уже встречались с расположением объектов по слоям при размещении рамки группы. Если объект расположен сверху другого объекта, он накрывает его полностью или частично, в результате чего вы можете не увидеть на экране расположенный ниже объект.

Порядок обхода объектов формы определяет порядок выбора элементов при нажатии клавиши *Tab*.

Каждый объект интерфейса имеет номер, определяющий его положение в порядке расположения объектов по слоям. Самый последний объект ("самый глубокий") имеет номер 1. Следующий объект, расположенный выше, получает номер 2 и т.д.



Обычно объекты имеют одинаковый номер в порядке расположения объектов по слоям и в порядке обхода. Однако, если один или несколько объектов интерфейса не могут получать фокус (например, рамка группы или надпись поля), то в порядке расположения объектов по слоям окажется больше объектов, чем в порядке обхода.

Вы можете изменить как порядок расположения объектов как по слоям, так и порядок их обхода. При этом изменение порядка расположения объектов по слоям меняет и порядок их обхода, а также наоборот.

Порядок объектов изменяется в режиме макета или в режиме упорядочивания окна конструктора форм. Для выбора режима используются команды *Режим макета* и *Режим упорядочивания* из меню *Вид*, а также одноименные кнопки на панели инструментов.

Для изменения порядка расположения объектов по слоям в режиме макета вы можете использовать следующие команды меню:

- Макет | На задний план
- Макет | На передний план
- Макет | На один слой вниз
- Макет | На один слой вверх

а для изменения порядка их обхода — следующие кнопки панели инструментов:

Кнопка	Название	Назначение
	Установить первым	Устанавливает выбранный объект первым в порядке обхода объектов формы.

Кнопка	Название	Назначение
	Установить последним	Устанавливает выбранный объект последним в порядке обхода объектов формы.
	Установить за следующим	Устанавливает выбранный объект за следующим объектом в порядке обхода объектов.
	Установить перед предыдущим	Устанавливает выбранный объект перед предыдущим объектом в порядке обхода объектов.

При переходе в режим упорядочивания изменяются меню и панель инструментов. На панели инструментов появляется счетчик порядка, закрываются панель элементов и окно свойств, изменяется режим отображения объектов интерфейса (рис. 8.48). В этом режиме для каждого объекта приводится его наименование, а также порядок следования объекта в форме.

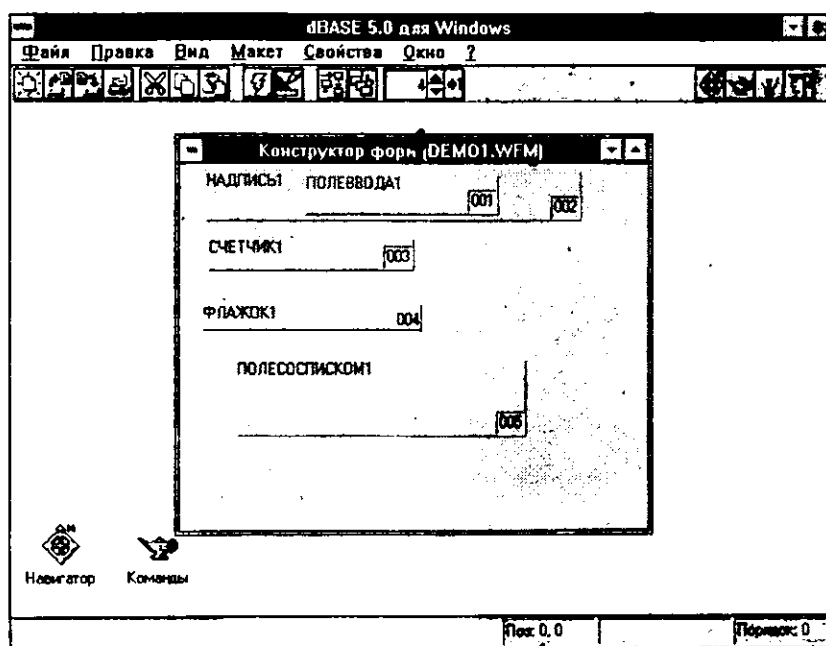


Рис. 8.48. Режим упорядочивания окна конструктора форм.

Для изменения порядка вы можете использовать счетчик порядка, расположенный на панели инструментов, а также нумерованные элементы объектов интерфейса.

Счетчик порядка панели инструментов состоит из поля отображения значения счетчика, кнопок изменения значения, а также кнопки установки значения счетчика равным 1. Значение счетчика является номером, который будет присвоен объекту интерфейса, выбранному следующим на рабочей поверхности. Для изменения порядка следования объектов, измените этот номер, затем выберите объект который должен получить этот номер.

Копирование объектов

Для копирования в форме отдельных объектов и групп вы можете использовать приведенные ниже средства.

- Команды **Копировать** и **Вставить** из меню **Правка**
- Кнопки **Копировать в буфер** и **Вставить из буфера**
- Клавиши *Ctrl+Ins* и *Shift+Del*

Выравнивание объектов

Вы можете изменить размеры и расположить на форме каждый из независимых объектов. Кроме того, dBASE предоставляет в ваше распоряжение более эффективные средства, которые позволяют выровнять объекты. Эти средства позволяют вам делать эскиз формы, не обращая внимания на точное расположение объектов. В дальнейшем вы сможете быстро выровнять их форме. Для выравнивания используются следующие команды и соответствующие им кнопки панели инструментов:

Кнопка	Команда	Назначение
	Выровнять по левому краю	Выравнивает выбранные объекты по левому краю самого левого из выбранных объектов.
	Выровнять по правому краю	Выравнивает выбранные объекты по правому краю самого правого из выбранных объектов.
	Выровнять по верхнему краю	Выравнивает выбранные объекты по верхнему краю самого верхнего из выбранных объектов.
	Выровнять по нижнему краю	Выравнивает выбранные объекты по нижнему краю самого нижнего из выбранных объектов.
	Выравнивание по центру	Выравнивает центральную ось выбранных объектов по центральной оси формы (абсолютное выравнивание) или по центральной оси относительно всех выбранных объектов (относительное выравнивание).

Команда **Выравнивание по центру** имеет подменю, команды которого уточняют вариант выравнивания.

Команда	Назначение
Абсолютное горизонтальное	Выравнивает центральную горизонтальную ось выбранных объектов по центральной горизонтальной оси формы.
Относительное горизонтальное	Выравнивает центральную горизонтальную ось выбранных объектов по центральной горизонтальной оси относительно всех выбранных объектов.
Абсолютное вертикальное	Выравнивает центральную вертикальную ось выбранных объектов по центральной вертикальной оси формы.
Относительное вертикальное	Выравнивает центральную вертикальную ось выбранных объектов по центральной вертикальной оси относительно всех выбранных объектов.

Для того чтобы продемонстрировать действие этих команд, выровняем поля ввода, расположенные произвольным образом (рис. 8.49). На рис. 8.50 приведен результат выравнивания по левому краю (с первого по третий) и по верхнему краю (с четвертого по шестой).

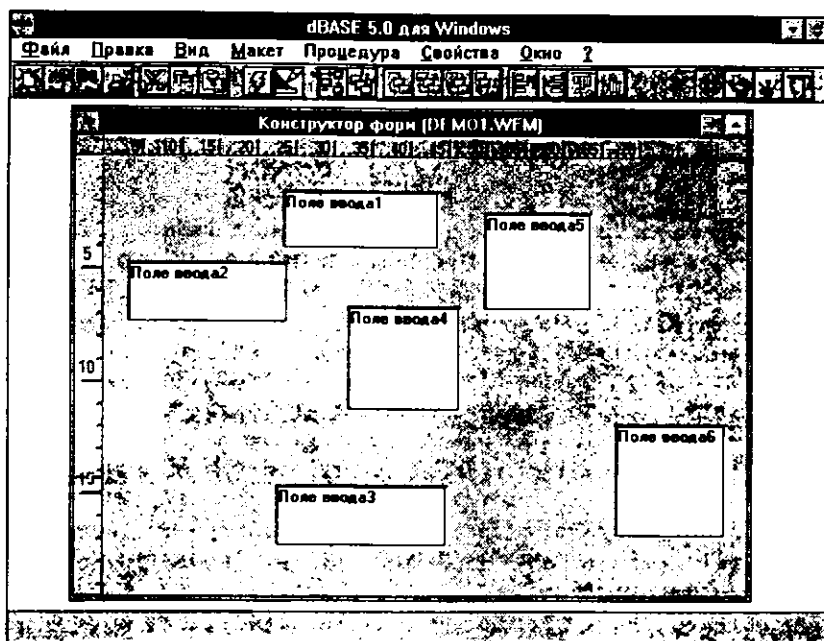


Рис. 8.49. Исходная форма с полями ввода.

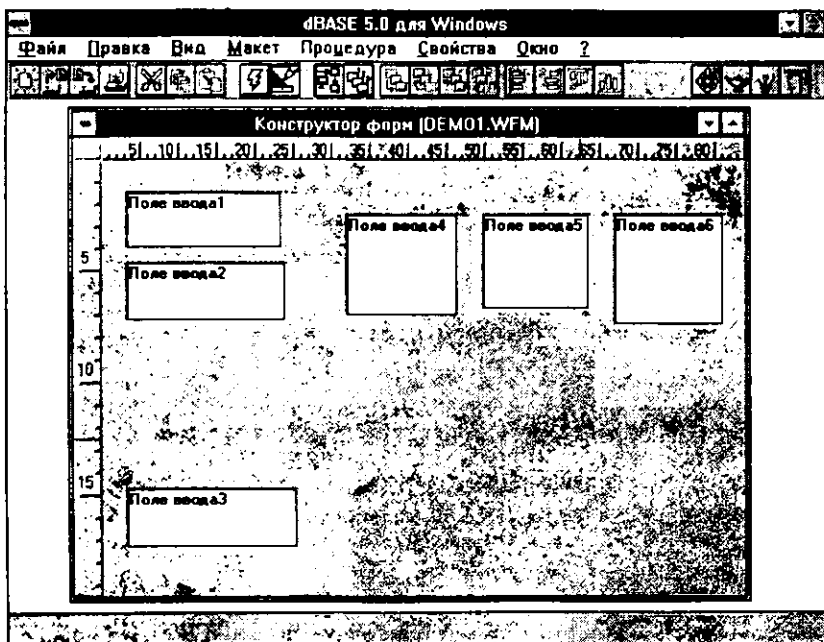


Рис. 8.50. Результат выравнивания полей.

Дополнительные замечания по созданию форм

В настоящем разделе мы дадим ряд рекомендаций по использованию средств, с которыми вы познакомились в настоящей главе. Эти замечания помогут вам при создании экранных форм для собственных приложений.

Цветовая схема

При создании экранных форм вы можете установить цвета практически для всех объектов, размещенных на форме. Умелое использование цветов значительно улучшает внешний вид формы и облегчает работу с формой. В то же время чрезмерное увлечение яркими цветами или наличие большого числа различных цветов в форме может вызвать у пользователя отрицательные эмоции в процессе работы с формой. Поэтому при выделении цветом различных областей формы важно соблюдать меру. Вы можете положиться на собственный вкус и опыт работы либо воспользоваться советами дизайнера или каким-либо руководством по дизайну.

Следование определенным правилам при установке цветов для полей может значительно облегчить работу с формой. Вы можете воспользоваться цветом для выделения различных характеристик полей. Например, вы можете выделить цветом поля, для которых установлено одно из правил по проверки достоверности данных (см. главу 4). Установив красный цвет фона для полей, которые не могут быть пустыми, вы тем самым напомните пользователю о том, что прежде чем покинуть это поле в него необходимо ввести значение. Другим, удачным на наш взгляд, примером является выделение цветом нередатируемых полей. Это сразу предостережет пользователя от попытки ввода данных в эти поля.

Альтернативный способ обратить внимание пользователя на поле заключается в использовании для его заголовка яркого цвета.

Если вы создаете приложение для продажи, не пренебрегайте той частью пользователей, в распоряжении которых имеются монохромные мониторы. После того, как вы выберете на цветном мониторе подходящие цветовые схемы обязательно убедитесь в том, что они обладают достаточной контрастностью и яркостью на монохромном мониторе. Постарайтесь найти такие цветовые схемы, которые могут быть использованы без дополнительной настройки при работе как с цветными, так и монохромными мониторами.

Старайтесь быть консервативным в хорошем смысле этого слова. Используйте для экранных форм, выполняющих одинаковые функции, одну и ту же цветовую комбинацию. Например, все формы для просмотра и ввода данных целесообразно выполнять в едином стиле, хотя стили могут различаться для форм, имеющих различное назначение. Это касается также и других элементов ваших экранов (например, меню, кнопок и т.д.).

Найдя хорошее цветовое решение формы, используйте его в последующих приложениях.

Вспомогательные тексты и заголовки полей

При разработке форм вы могли убедиться, что работать с экранными формами значительно удобнее, чем с таблицами. В частности информативные тексты, раскрывающие содержание полей, могут облегчить работу пользователя. Кроме того, форма может содержать вспомогательный текст, объясняющий работу с ней. Вы можете создавать формы, взаимодействие с которыми не требует дополнительных пояснений.

Старайтесь делать заголовки полей настолько понятными, насколько это возможно. Например, какую информацию следует вводить в поле "Количество"? Из

заголовка непонятно имеется ввиду количество единиц заказанного или проданного товара. В тоже время заголовок "Количество заказанного товара" будет занимать слишком много места. Принимая во внимание, что заказанный и проданный товары, кроме количества, характеризуются и другими параметрами (цена и стоимость) целесообразно выделить две логические группы, одна из которых содержит информацию о заказанных товарах, а другая — о проданных. В этом случае заголовки групп "Заказано" и "Продано" дадут пользователю исчерпывающую информацию о назначении каждой из групп, и заголовок "Количество" уже не вызывает вопросов.

Заголовки полей в форме могут не совпадать с наименованиями полей в таблице. В случае необходимости вы можете не ограничивать себя несколькими символами, а использовать текст произвольной длины. Однако при этом не следует забывать о том, что для размещения в форме вспомогательных текстов потребуется дополнительный размер.

Сокращения позволяют сэкономить размер, требуемый для заголовка. Используйте в своих формах только такие сокращения, которые понятны пользователю. Например, понятно, что "Ф. И. О." означает фамилию, имя и отчество клиента. Однако старайтесь избегать сокращений типа "ВНИПИЭИ Леспром" (Всесоюзный научно-исследовательский проектный институт экономики и информатики), если у вас нет полной уверенности в том, что пользователь, для которого предназначается форма, знаком с этой аббревиатурой.

Наличие заголовков полей не является обязательным условием. Например, в форме, предназначенной для просмотра, совершенно необязательно помещать заголовки "Индекс", "Страна", "Адрес", "Город" и т.п., поскольку содержимое полей очевидно.

Если в различных формах одного приложения размещено одно и то же поле, используйте для него одинаковые заголовки. В противном случае пользователю может быть непонятно имеется в виду одно и то же поле или нет ("Код покупателя" и "Покупатель_идент.>").

Не пренебрегайте (но и не увлекайтесь!) специальными символами. Например, знак доллара подскажет пользователю, что цены даны в долларовом эквиваленте. В этом случае вам не нужно будет использовать пояснительный текст.

dBASE и Windows используют в заголовках смешанный или верхний регистры. Эти сочетания являются наиболее удачными. Как правило, заголовки, выполненные только в нижнем регистре, не используются.

В заголовках старайтесь использовать пропорциональные шрифты (Pragmatica или разновидности Times), а для полей — равноширинные шрифты (разновидности Courier). В этом случае экранная форма пользователя будет напоминать заполненный бланк, на котором заголовки выполнены типографским способом, а содержимое полей напечатано на пишущей машинке. И, конечно, старайтесь избегать экзотических шрифтов, которые скорее утомят пользователя, чем сделают форму привлекательной.

Формат

От формата формы полностью зависит насколько удобно будет работать пользователю с вашим приложением. dBASE предоставляет вам широчайшие возможности по разработке собственных форматов, однако, как вы ими воспользуетесь полностью определяется вашими представлениями о экранной форме.

При работе с Windows и dBASE попытайтесь понять, в чем состоит привлекательность форм этих продуктов. Отметьте для себя расположение объектов и цветовые схемы. Заимствуя все лучшее, создавайте свой стиль. Однако не пытайтесь использовать лучшие качества различных продуктов. Сочетание оптимальных элементов не всегда позволяет получить оптимальный результат.

Для того чтобы выделить объекты, используйте рамки. Отметьте, что рамки и цвет очень хорошо выделяют группы объектов. Законченные логические блоки на форме делают ее более информативной по сравнению с формой, содержащей только отдельные поля.

Старайтесь выравнивать логические группы и отдельные поля. По возможности делайте блоки одинаковой длины и располагайте их в форме симметрично.

Все данные нами рекомендации не носят обязательный характер. Помните, что пользователю придется взаимодействовать с вашими формами длительное время. Попробуйте облегчить его работу и, по возможности, сделайте ее не нудной, а интересной.

Тестирование форм

После того, как вы закончили работу над формой, тщательно проверьте ее. Попробуйте ввести несколько записей и убедитесь, что длина полей в форме адекватна данным из таблиц. Обратите внимание на цветовую схему, читабельность шрифтов, используемых для отображения текста и специфические особенности вашей формы.

Вы можете перейти непосредственно из окна конструктора форм к просмотру данных через форму, воспользовавшись одним из следующих способов:

- Выберите Вид | Форма
- Нажмите кнопку Запуск на панели элементов
- Нажмите клавишу F2

Находясь в окне формы, вы можете быстро перемещаться по записям формы. Как и большинство операций, перемещение может осуществляться различными способами. Наиболее простой способ предполагает использование кнопок, расположенных на панели элементов. Кроме этого вы можете воспользоваться командами меню **Форма**, выполняющими аналогичные функции.

Если после окончания тестирования формы вам потребуется внести в нее какие-либо изменения, возвратитесь в конструктор форм, воспользовавшись любым из следующих способов:

- Выберите Вид | Конструктор форм
- Нажмите кнопку Конструктор на панели элементов
- Нажмите клавишу F2



Сохранение форм

Для того чтобы сохранить форму, существует несколько способов. Конкретный способ, с помощью которого вы сохраняете форму, зависит от текущего контекста работы.

1. Выберите **Файл | Сохранить**. В этом случае dBASE сохраняет в файле текущее состояние окна конструктора. Мы рекомендуем в процессе работы над формой время от времени сохранять текущее состояние формы. Это убережет вас от потери данных при аварийных ситуациях с компьютером.
2. Выберите **Файл | Сохранить как**. dBASE предложит указать наименование нового файла, в котором будет сохранена форма. Сохранение текущего состояния формы, также как и в предыдущем случае, выполняется без выхода из окна кон-

структора. Данное средство позволяет вам использовать одну из уже готовых форм в качестве исходной и разрабатывать на ее основе новые формы.

3. Нажмите кнопку закрытия окна конструктора. Если в текущую форму были внесены какие-либо изменения, dBASE откроет окно диалога, с помощью которого предложит вам указать как поступить со внесенными изменениями (рис. 8.51). При выборе Да dBASE сохранит информацию. Выбор Нет будет иметь результатом потерю всех сделанных изменений. При выборе Отмена, вы возвратитесь в окно конструктора форм. Для получения контекстной справки выберите Справка.



Отказ от сохранения созданной формы является не лучшим завершением текущего сеанса работы. Как правило, сохранение созданной формы и внесение в нее необходимых изменений потребуют значительно меньших затрат времени, чем создание формы заново.

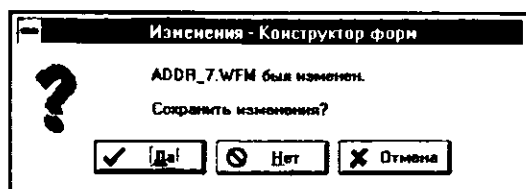


Рис. 8.51. Предупреждение dBASE о необходимости сохранения формы.

Печать формы

Основным назначением форм является просмотр информации на экране. Однако вы можете напечатать форму, находясь непосредственно в окне форм. Для этого выберите **Файл | Печать** или нажмите кнопку **Печать** на панели элементов. В результате на экране появится окно диалога "Печать" (рис. 8.52). Далее, если вам необходимо напечатать несколько экземпляров, укажите нужное число экземпляров. В заключение выберите **ОК**. Более подробно печать документов будет рассмотрена в следующей главе.

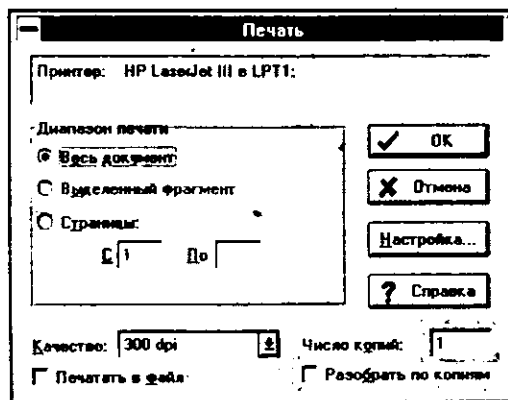


Рис. 8.52. Окно диалога "Печать".

Заключение

- Для ввода, редактирования и просмотра данных в режиме формы вы можете использовать специальные формы.
- Для того чтобы приступить к созданию новой формы, выберите **Файл | Создать | Форма**.
- Окно конструктора форм является основным средством для создания форм. В этом окне выполняется большинство операций по размещению, перемещению, удалению и модификации всех экранных объектов.
- На рис. 8.9 показан внешний вид панели элементов в окне конструктора форм, а назначения кнопок приведены в табл. 8.1.
- Для выбора требуемого объекта воспользуйтесь кнопкой **Указатель** на панели элементов, командой **Правка | Выделить все** или клавишными командами.
- Все объекты dBASE характеризуются рядом параметров, называемых свойствами. Для того чтобы получить доступ к свойствам объекта, выделите нужный объект и откройте окно свойств объекта.
- Для размещения объектов интерфейса используйте соответствующие кнопки, расположенные на панели элементов.
- Для определения источника данных формы задайте свойство **View** формы.
- Для объединения в группу нескольких объектов используйте объект типа **Рамка группы**.
- dBASE предоставляет в ваше распоряжение эффективные средства выравнивания объектов в форме.
- Линейки и сетка предоставляют вам дополнительные возможности по точному позиционированию и выравниванию объектов в форме.
- Для того чтобы сохранить форму, выберите **Сохранить** или **Сохранить как** из меню **Файл**.
- Для того чтобы напечатать форму, выберите **Файл | Печать** или нажмите кнопку **Печать** на панели элементов, находясь в окне формы.

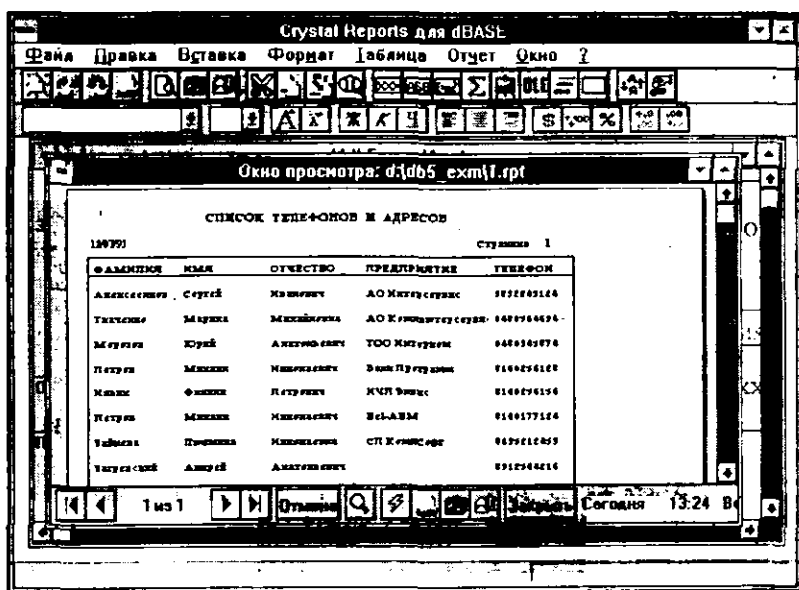
Глава 9. Отчеты

В процессе работы с dBASE вы уже научились создавать таблицы, вводить и редактировать в них данные, выполнять сортировку и выбор записей из таблиц. Вы можете оперативно обрабатывать информацию, содержащуюся в таблицах, и отображать результаты на экране.

В настоящей главе вы познакомитесь со средствами dBASE, предназначенными для создания отчетов. Под *отчетом* понимается форматированное представление данных, выводимое на принтер или на экран.

При знакомстве со средствами dBASE для создания специальных форм вы могли убедиться в их гибкости и мощности. Аналогичные средства dBASE предоставляются в ваше распоряжение для создания отчетов. Вы можете разработать формат отчета с помощью *конструктора отчетов*. Конструктор отчетов позволяет создавать отчеты в различных форматах (табличный, свободный или почтовые наклейки). На рис. 9.1 и 9.2 показаны фрагменты табличного отчета и отчета в свободной форме для таблицы ADDRESS.

При создании отчетов вы можете прибегнуть к помощи инструктора. С помощью инструктора вы сможете изучить все стадии создания простейшего отчета на примере таблицы-образца или собственной таблицы. Внести изменения в отчет вы сможете в конструкторе отчетов Crystal Reports.



The screenshot shows the 'Crystal Reports для dBASE' application window. The menu bar includes 'Файл', 'Правка', 'Вставка', 'Формат', 'Таблица', 'Отчет', and 'Окно'. The toolbar contains various icons for report creation and editing. The main window title is 'Окно просмотра: d:\db5\exam\1.rpt'. The report content is titled 'СПИСОК ТЕЛЕФОНОВ И АДРЕСОВ' and 'Страница 1'. It displays a table with 5 columns: 'Ф.А.И.', 'ИМЯ', 'ОТЧЕСТВО', 'ПРЕДПРИЯТИЕ', and 'ТЕЛЕФОН'. The table contains 7 rows of data. At the bottom, there are navigation buttons and a status bar showing '1 из 1' and 'Сегодня 13:24 В'.

Ф.А.И.	ИМЯ	ОТЧЕСТВО	ПРЕДПРИЯТИЕ	ТЕЛЕФОН
Александров	Сергей	Иванович	АО Интелсервис	8022803124
Тихонов	Михаил	Михайлович	АО Консалтингсервис	8408964494
Метель	Юрий	Александрович	ТОО Интелсервис	8408020774
Петров	Михаил	Николаевич	ЗАО Прогресс	8100256124
Михаил	Филипп	Петрович	ИЧП Финанс	8100296154
Петров	Михаил	Николаевич	Вел-АВМ	8100177124
Таблица	Пример	Иванович	СПК Комиссия	8079612459
Таблица	Андрей	Александрович		8910944214

Рис. 9.1. Табличный отчет.

Табличный отчет представляет собой напечатанную таблицу, в которой данные упорядочены по столбцам и строкам. Каждый из столбцов отчета содержит поле исходной таблицы, а строка представляет собой запись.

Табличный отчет позволяет напечатать данные из таблиц в наиболее простом и естественном виде. Однако табулированное представление данных в отчете имеет свои недостатки. На практике в ряде случаев (наклейки, чеки, письма и т.д.) поля исходной таблицы должны располагаться в специально отведенных для них местах отчета. Очевидно, что табличный отчет не пригоден для этих целей.

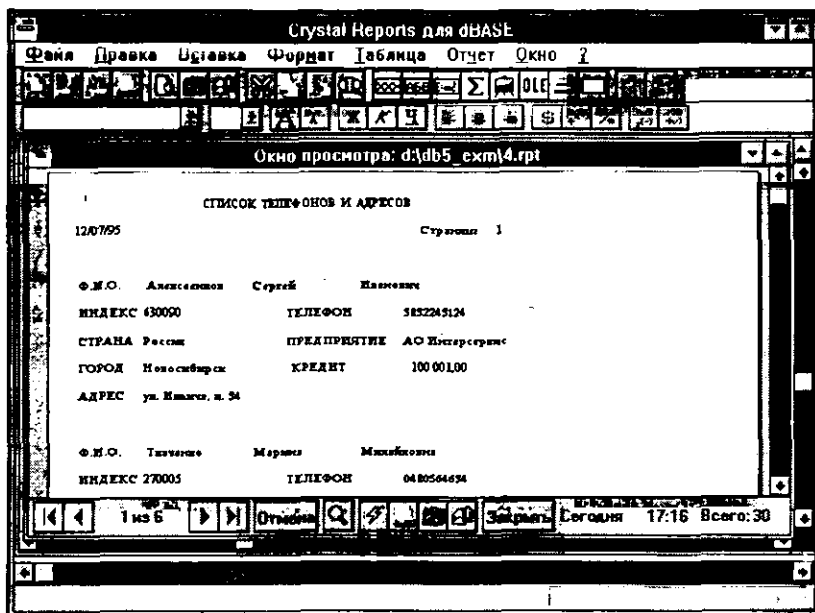


Рис. 9.2. Отчет в свободной форме.

Отчеты в свободной форме позволяют устранить ограничения, свойственные табличным отчетам. С помощью конструктора отчетов вы можете разработать специальный формат отчета, где поля исходной таблицы будут расположены в требуемых местах отчета.

Между экранными формами и отчетами существует много общего. Отчеты, как и формы, являются конструкторскими документами и служат для представления информации, содержащейся в базах данных. Большая часть принципов и средств, используемых при создании экранных форм, применимы при разработке отчетов.

В настоящей главе вы познакомитесь с конструктором отчетов Crystal Reports. Мы обсудим основные средства, используемые при создании табличных отчетов и отчетов в свободной форме. Кроме того, вы узнаете, как напечатать отчет в соответствии с разработанным форматом, а также как улучшить формат отчета, не выводя его на печать. В настоящей главе в качестве примера для создания отчетов будет использована таблица ADDRESS.

Во второй части (глава 14) мы вернемся к проблемам, связанным с разработкой отчетов. Там вы познакомитесь с созданием отчетов для связанных таблиц.

Рекомендации, предшествующие созданию отчета

Прежде чем приступить к созданию отчета, вы должны дать ответ на следующие вопросы:

- С какой целью создается настоящий отчет и чем он будет вам полезен?
- Какая информация и из каких таблиц должна быть представлена в отчете?
- Отчет какого вида вы предполагаете создать (табличный, в свободной форме или наклейки)?
- Предполагается ли группировка данных?

Четкие ответы на поставленные вопросы облегчат вашу работу при создании отчета.

Мы рекомендуем вам, по крайней мере при создании первых отчетов, прежде чем приступить к работе с конструктором отчетов, прорисовать формат отчета на бумаге. Это позволит избежать ошибок и сократит время на разработку отчета. По мере приобретения навыков работы с конструктором вы сможете обходиться без эскиза формата на бумаге, улучшая отчет прямо на экране.

Создание отчета

В настоящем разделе будет рассмотрено создание отчета с помощью конструктора отчетов Crystal Reports.

Создание полей

Для того чтобы приступить к созданию нового отчета выберите **Файл | Создать | Отчет** (рис. 9.3).

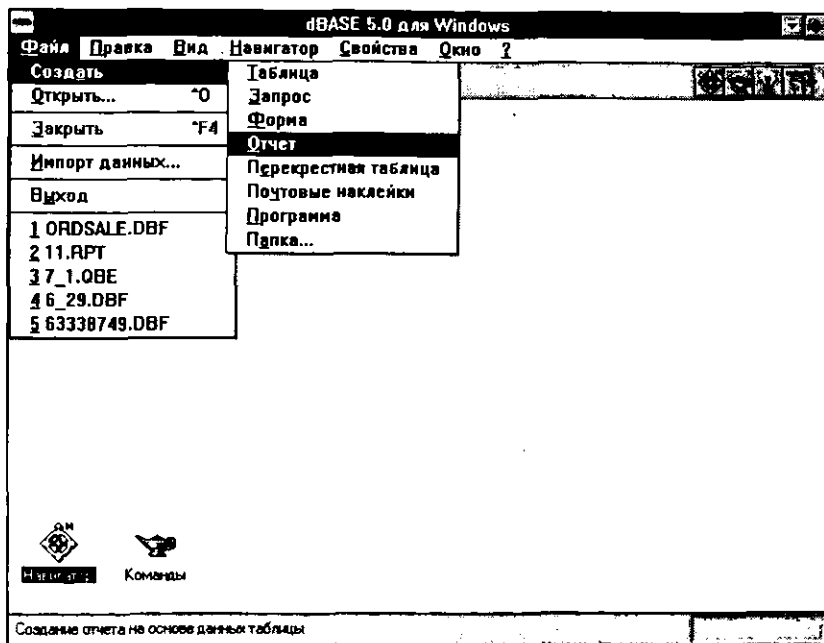


Рис. 9.3. Создание нового отчета.

Если таблица, для которой вы хотите создать отчет не открыта, на экране откроется окно диалога “Выбор таблицы” (рис. 9.4).

В настоящей главе мы ограничимся созданием отчетов для одной таблицы. Многотабличные отчеты будут рассмотрены в главе 14. Сейчас просто укажите имя требуемой таблицы из списка, помещенного в левой части окна диалога. В качестве примера выберите таблицу ADDRESS и нажмите ОК. В результате dBASE откроет окно конструктора отчетов Crystal Reports для однотабличного документа с открытым окном диалога “Вставка поля базы данных”, содержащим список полей открытой таблицы (рис. 9.5).

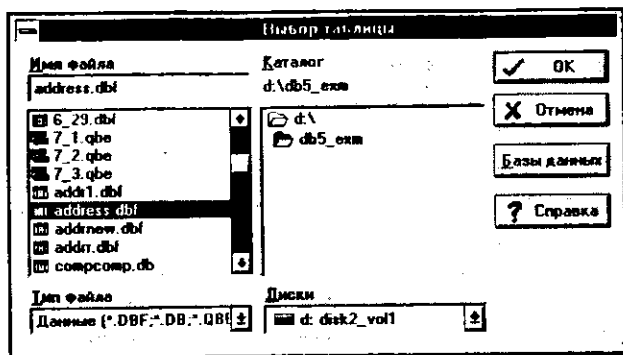


Рис. 9.4. Окно диалога "Выбор таблицы".

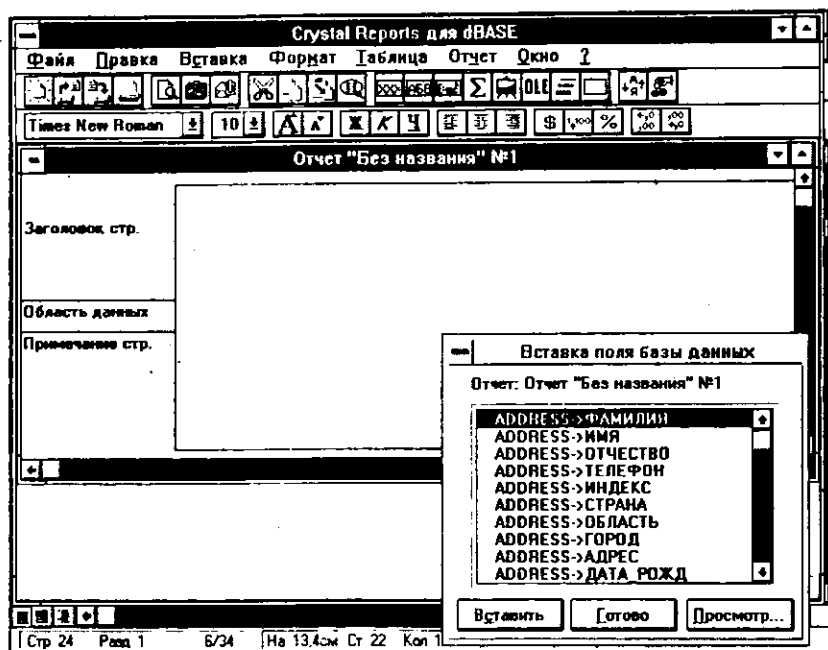


Рис. 9.5. Окно конструктора отчетов Crystal Reports.



Для открытия данного окна диалога используется также команда меню Вставка | Поле таблицы или кнопка Вставка поля из файла данных на панели инструментов.

В нижней части окна диалога расположены три кнопки:

Наименование кнопки	Выполняемое действие
Вставить	Вставляется в отчет поле, выделенное в списке.
Готово	Закрывает окно диалога после завершения вставки полей данных в отчет.
Просмотр	Открывает окно диалога для просмотра типа данных, длины и содержимого любого поля из списка.

При нажатии кнопки **Просмотр** на экране открывается окно диалога, заголовком которого будет являться полное название поля (рис. 9.6). Это окно диалога содержит следующие элементы:

- Тип данных выбранного поля (текстовый, числовой, логический, дата)
- Число символов, выделенных под выбранное поле в исходной таблице
- Список значений выбранного поля

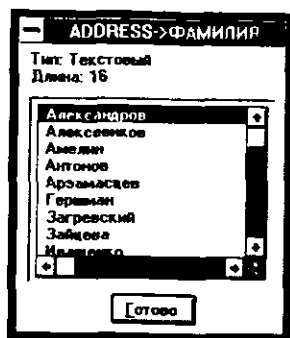


Рис. 9.6. Окно диалога просмотра типа данных, длины и значений поля.

Для размещения полей в отчете выполните следующую последовательность действий:

1. Выберите в списке нужное поле данных.
2. Нажмите кнопку **Вставить**. Указатель мыши принимает вид прямоугольника, имеющего размер, соответствующий настоящей ширине выбранного поля данных.
3. Переместите прямоугольник в нужное место отчета и нажмите левую кнопку мыши для вставки поля данных в отчет.
4. Повторите шаги 1 — 3 до тех пор, пока все необходимые поля данных не будут вставлены в отчет.
5. Для закрытия окна диалога нажмите кнопку **Готово**.

Процесс переноса полей в отчет можно ускорить, одновременно вставляя в отчет сразу несколько предварительно выделенных полей данных. Выделение непрерывной группы полей осуществляется нажатием кнопки мыши при нажатой клавише *Shift*, а выделение нескольких отдельных полей — при нажатой клавише *Ctrl*. Выделенные поля будут вставлены в отчет в том порядке, в котором они были указаны в списке окна диалога “Вставка поля базы данных”.

Определение параметров страницы отчета

После перехода в окно конструктора прежде всего целесообразно определить формат страницы при печати отчета и/или его отображении. Для этого выберите **Файл | Поля страницы**, в результате чего на экране появится окно диалога “Поля страницы”. На рис. 9.7 показано окно диалога “Поля страницы”, содержащее стандартные установки.

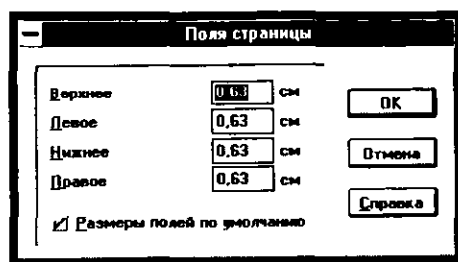


Рис. 9.7. Окно диалога “Поля страницы”.

Данное окно диалога предназначено для задания размеров полей страницы отчета, то есть пустого пространства между краем страницы и областью отображаемых данных.

Для установки нестандартных размеров полей страницы введите требуемые значения в поля **Верхнее**, **Нижнее**, **Левое** или **Правое**. При изменении значения любого из этих полей флажок **Размеры полей по умолчанию** автоматически снимается.

Для восстановления стандартных значений размеров полей установите флажок **Размеры полей по умолчанию**.



Установленные стандартные значения полей определяются размерами непечатаемой области страницы выбранного по умолчанию принтера. Теоретически допускается задание полей страницы, меньших чем непечатаемая область, однако при этом часть данных может быть потеряна.

Использование областей в отчетах

Для того чтобы понять структуру отчетов, давайте обратимся к рис. 9.8 и 9.9, на которых представлены отчеты в конструкторе отчета Crystal Reports, показанные в окне просмотра на рис. 9.1 и 9.2, соответственно.

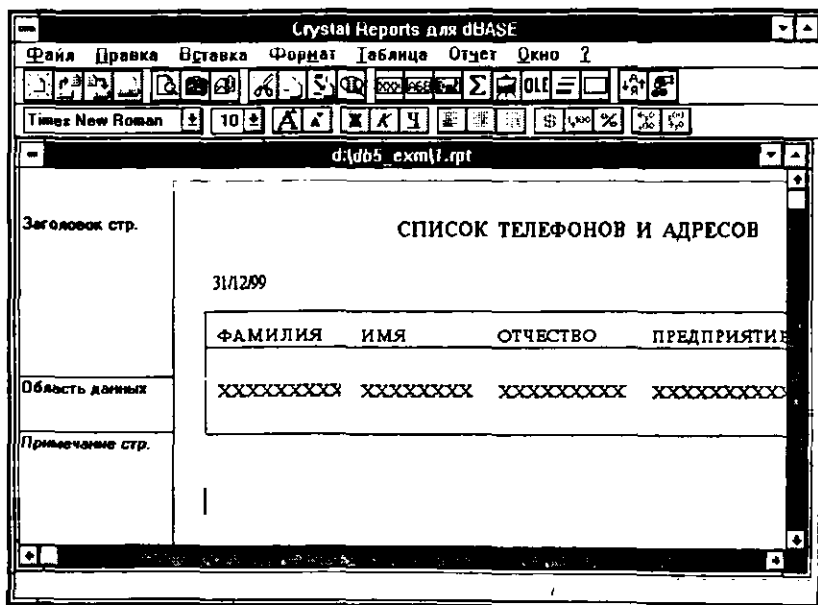


Рис. 9.8. Табличный формат отчета.

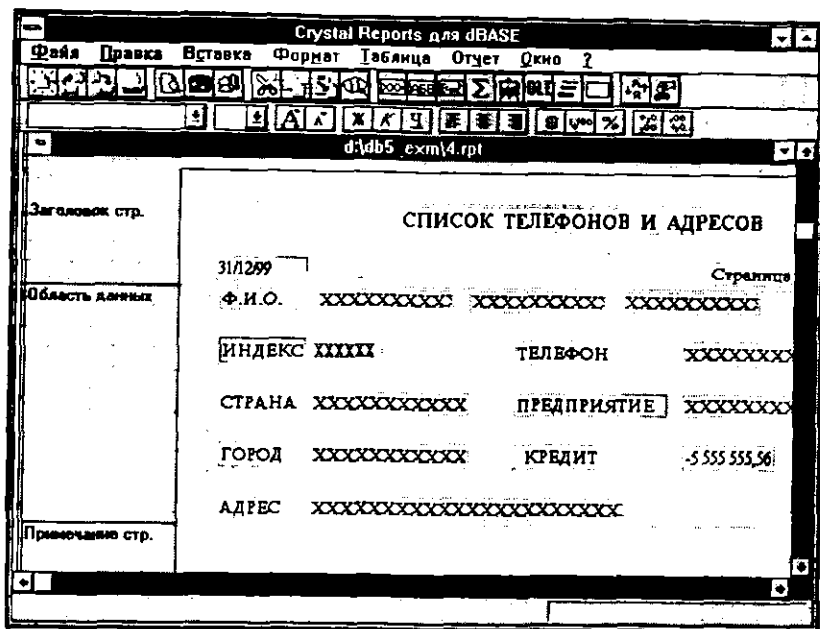


Рис. 9.9. Свободный формат отчета.

Рабочая область окна конструктора отчетов ограничена контурной рамкой, внутри которой должны располагаться все объекты. Изменить размеры контурной рамки можно с помощью окна диалога "Поля страницы" (рис. 9.7), установив нужные размеры в поля **Верхнее**, **Нижнее**, **Левое** или **Правое**.

Рабочую область окна конструктора отчетов пересекают горизонтальные линии. Эти линии не присутствуют в напечатанном отчете. Однако относительное расположение этих линий и объектов отчета однозначно определяет, сколько раз и в каком месте отчета будет напечатан каждый объект.

Линии разделяют отчет на три части: **Заголовок страницы**, **Область данных** и **Примечание страницы**. Разделы **Заголовок страницы** и **Примечание страницы** составляют область страницы.

В главе 14 при создании отчетов для связанных таблиц вы познакомитесь еще с одним типом областей — областью группы. С помощью областей групп вы можете сгруппировать данные в отчете по различным признакам. Области группы, как и области страницы, разделены внутренними областями на верхний и нижний слои, называемые **заголовком** и **примечанием группы**, которые располагаются вокруг области данных или области другой группы.

При добавлении в отчет итоговых полей в отчете добавляется еще одна область — **Общая сумма**.

Давайте посмотрим, как dBASE отображает в отчете содержимое объектов, расположенных в каждой из областей отчета.

Раздел заголовка страницы

Информация, расположенная в заголовке страницы, будет напечатана один раз для каждой страницы отчета. Обычно, в начале страницы печатают краткий заголовок отчета, текущую дату или время, номер страницы. Кроме того, заголовок страницы может содержать названия предприятия и/или подразделения, выпускающих отчет, и графические изображения фирменных и/или товарных знаков.

В соответствии с форматом отчета, показанным на рис. 9.8, в начале страницы будет напечатано "Список телефонов и адресов". Эта область содержит еще объекты для отображения текущей даты и номера страницы, а также текст "Страница". Эта информация представлена в верхней части каждой страницы отчета (рис. 9.1).

Область данных

Самым внутренним слоем в отчете является область данных. Эта область представляет собой основную часть отчета.

В области данных располагаются поля, которые содержат табличные данные. В табличном отчете заголовки и данные из исходных таблиц располагаются в столбцах. При этом заголовки столбцов печатаются один раз сразу вслед за информацией из верхнего колонтитула. Ниже заголовков располагается область, в которой dBASE отображает столько записей, содержащих данные из исходной таблицы, сколько их может поместиться на каждой странице отчета.

В области данных для отчета в свободной форме заголовки и поля могут располагаться в произвольном месте отчета. При печати отчета в свободной форме для каждой из записей в отчете будут отображены заголовок поля и данные из соответствующего поля исходной таблицы (рис. 9.2). Аналогично отчету в табличном виде dBASE, отобразит в области данных столько записей, сколько их сможет поместиться на одной странице отчета.

Помимо полей исходной таблицы вы можете расположить в области данных вычисляемые поля, что позволит отобразить в отчете результат вычисления над полями исходной таблицы по каждой записи. С отчетами, содержащими вычисляемые поля, вы познакомитесь в разделе "Определение вычисляемых полей".

Таким образом, взаимное расположение слоев в табличных отчетах и отчетах в свободной форме аналогично. Структуры указанных отчетов различаются только основным слоем — областью данных, в которой размещаются поля исходной таблицы и, возможно, вычисляемые поля.

Раздел примечания страницы

Раздел примечания страницы позволяет отобразить в отчете групповые значения по каждой странице отчета. При печати последней страницы отчета dBASE вначале напечатает информацию из раздела *Общая сумма*, а затем из раздела примечания страницы.

Область группы

При создании отчетов в ряде случаев бывает полезно группировать данные из различных записей и поместить в отчете групповую информацию по каждой из групп. Например, вам требуется не просто распечатать таблицу ADDRESS, а напечатать список абонентов отдельно для каждого города и отобразить в отчете, сколько абонентов проживает в каждом из городов. Для решения этой и аналогичных ей задач в отчет вводится одна или несколько областей групп. С помощью областей групп вы можете разбить записи таблицы на группы, объединив в одну группу записи, которые имеют одинаковые значения в указанном поле.

Каждая область группы содержит разделы заголовка и примечания, информация из которых отображается в отчете перед и после соответствующей группы. Примечание группы может быть использовано для отображения в отчете групповых значений по группе.

Отчеты, использующие группировку данных, будут рассмотрены в главе 14.

Раздел итоговых значений

Раздел **Общая сумма** предназначен для вставки в конце отчета поля общей суммы (или среднего значения по отчету, числа значений и других общих итогов). В этом же разделе вы можете разместить текстовую информацию, относящуюся ко всему отчету.

Изменение области

При создании отчета dBASE автоматически размещает области страницы и данных. Вы можете по своему усмотрению увеличить или уменьшить размер любой из этих областей. Изменение ширины области отчета осуществляется с помощью мыши или меню свойств.

Для изменения ширины области с помощью мыши установите курсор мыши на горизонтальную линию, определяющую границу областей, внутри страницы. В результате курсор изменит свою форму на двустороннюю вертикальную стрелку, указывая на то, что вы можете переместить вверх или вниз границу выделенной области.

Для изменения ширины области с помощью контекстного меню установите курсор мыши на любое место в пределах области темного цвета, где располагается название области, и нажмите правый курсор мыши. Для увеличения ширины области выберите команду меню **Добавить строку**, а для уменьшения — **Удалить последнюю строку**.

Для удаления пустой строки из отчета существует еще один способ. Установите текстовый указатель в пустую удаляемую строку и нажмите кнопку мыши, в результате чего появляется текстовый курсор. Нажмите клавишу *Backspace*, при этом будет удалена строка, на который был расположен текстовый курсор.

Контекстное меню разделов отчета

Каждая область отчета содержит контекстное меню, которое позволяет вам быстро выполнить наиболее часто используемые действия в данной области. Для вызова контекстного меню какой-либо области отчета, укажите на любое место в пределах области темного цвета, где располагается название области, и нажмите правую кнопку мыши.

Состав команд меню зависит от конкретной области. Ниже приведены команды контекстного меню, общие для всех областей:

Команда	Назначение
Скрыть	Скрывает текущий раздел. Для скрытого раздела этот пункт меню изменяется на Показать .
Показать/Скрыть раздел	Открывает окно диалога "Отображение/скрытие разделов".
Формат	Открывает окно диалога "Формат раздела" для текущего раздела.
Добавить строку	Добавляет одну строку в раздел.
Удалить последнюю строку	Удаляет последнюю строку раздела.

Кроме этого область группы содержит две дополнительные команды:

Команда	Назначение
Изменить группу	Изменяет выражение для группировки данных.
Удалить	Удаляет группу из отчета.

Форматирование раздела

Для форматирования разделов отчета и установки общих параметров используется команда **Формат | Раздел**, открывающая окно диалога "Формат раздела" (рис. 9.10). Данная команда позволяет:

- Скрыть раздел отчета (не печатать его)
- Выводить промежуточные суммы или групповые значения только внизу страницы
- Вставлять перевод страницы перед выводом раздела
- Вставлять перевод страницы после вывода раздела
- Начать нумерацию страниц с единицы после вывода группового значения
- Не допускать разделения данных одной записи на несколько страниц
- Подавлять вывод пустых строк
- Создавать несколько столбцов в отчете
- Форматировать почтовые наклейки и другие документы аналогичного типа

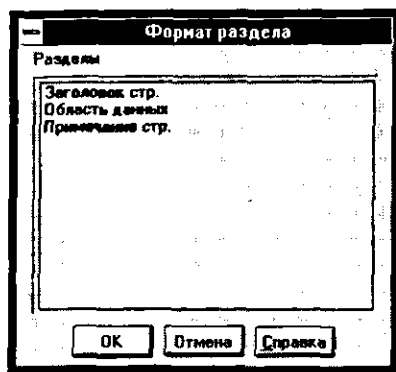


Рис. 9.10. Окно диалога "Формат раздела".

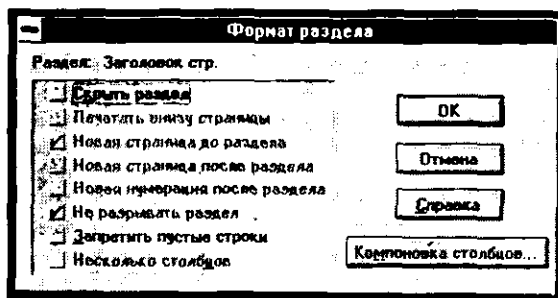


Рис. 9.11. Окно диалога "Формат раздела" с опциями.

После выбора требуемого раздела отчета нажмите кнопку ОК, в результате чего откроется окно диалога “Формат раздела” (рис. 9.11) с параметрами форматирования, наименования которых и выполняемые ими функции приведены в таблице 9.1.

Табл. 9.1. Параметры форматирования

Наименование флажка	Выполняемое действие
Скрыть раздел	Запрещает вывод на печать выбранного раздела.
Печатать внизу страницы	Предназначен для вывода групповых значений только внизу страницы, сохраняя при этом нормальное расположение областей данных.
Новая страница до раздела	Применяется для форматирования областей заголовков и примечаний групп, общей суммы и области данных. При установке этого флажка перед указанным разделом будет вставлен перевод страницы. Это позволяет начать печатать с новой страницы: — группу, если флажок установлен для области заголовков групп; — каждую запись отчета, если флажок установлен для области данных. Если в области примечаний группы расположено поле промежуточной суммы или поле итога, то установка данного флажка для области примечаний группы приведет к выводу этих полей на следующей странице после областей данных. Заголовок и примечания страницы будут выведены на каждой странице.
Новая страница после раздела	Вставляет перевод страницы после вывода указанного раздела.
Новая нумерация после раздела	После вывода групповых значений начинает нумерацию следующих страниц с единицы. Если данный параметр используется совместно с параметром “Печатать внизу страницы”, то каждая группа выводится на отдельной странице, внизу страницы выводятся групповые значения, и следующая страница снова получает номер 1. Этот флажок применяется, когда требуется распечатать несколько отчетов (например счетов) из одного файла, причем каждый отчет должен начинаться с первой страницы.
Не разрывать раздел	Все строки раздела выводятся вместе, либо на текущей странице (если на ней достаточно места), либо на следующей (если места не хватает).
Запретить пустые строки	Вызывает подавление вывода в отчете пустых строк, образующихся в результате запрещения вывода некоторых полей (например, полей с нулевыми значениями, с совпадающими значениями, а также скрытых полей).

Наименование флажка	Выполняемое действие
Несколько столбцов	При установке данного флажка становится доступной кнопка Компоновка столбцов . При нажатии этой кнопки на экран выводится окно диалога "Компоновка столбцов", позволяющее настроить формат вывода отчета в несколько столбцов. Допускается задание нескольких столбцов по ширине и высоте страницы, установка последовательности вывода текста в столбцах, а также задание интервала между столбцами.

Управление объектами интерфейса

Окно конструктора отчетов аналогично окну конструктора форм. В нем отображается формат разрабатываемого отчета, объекты интерфейса, которые могут быть использованы при работе над отчетом, а также средства разработки отчета и управления объектами интерфейса. Давайте начнем знакомство с окном конструктора отчетов с объектов интерфейса и средств управления ими. На рис. 9.12 показана панель инструментов в окне конструктора отчетов. Названия кнопок панели инструментов, поясняющие их назначение, приведены в табл. 9.2.



Рис. 9.12. Панель инструментов в окне конструктора отчетов.

Табл. 9.2. Кнопки панели инструментов

Кнопка	Название	Кнопка	Название
	Создать новый отчет		Открыть существующий отчет
	Сохранить отчет		Печать
	Вывести отчет в окне просмотра		Экспортировать отчет в файл
	Переслать отчет по электронной почте		Вырезать выделенный текст и поместить его в буфер обмена
	Копировать выделенный текст в буфер обмена		Вставить текст из буфера обмена
	Выделить поля		Вставить поле таблицы
	Вставить надпись		Построить выражение
	Вставить поле итога (сумма, счетчик) для выделенного поля		Вставить рисунок из файла

Кнопка	Название	Кнопка	Название
	Вставить новый объект OLE		Провести линию в отчете
	Нарисовать рамку в отчете		Задать порядок сортировки записей
	Задать критерии отбора записей		

Выбор объектов

Прежде чем создавать, модифицировать или перемещать объект (или группу объектов) в отчете вы должны выбрать его. Для выделения одного объекта достаточно установить курсор в пределах объекта и нажать кнопку мыши.

После того как вы выберете некоторый объект, вокруг него будут отображаться маркеры, с помощью которых вы можете перемещать выбранный объект, а также изменять его размеры и форму. Если вы укажете мышью на один из маркеров, ее курсор изменит свою форму, показывая в каких направлениях можно перемещать данный маркер. Например, для того чтобы изменить длину объекта, укажите на любой из маркеров, расположенных в середине вертикального отрезка. Курсор мыши примет форму горизонтальной двунаправленной стрелки. После чего переместите маркер до установки требуемого размера. Если вы захотите изменить сразу оба размера, воспользуйтесь одним из маркеров, расположенных по углам выбранного объекта. Для перемещения выбранного объекта укажите на область внутри объекта (но не на маркер) и переместите объект в нужное место.

Если вам требуется управлять сразу несколькими объектами, вам необходимо выбрать их одновременно. После этого вы можете перемещать выделенные объекты, выравнивать их размеры и интервалы между ними, изменять их свойства и т.д. Для выделения нескольких изолированных объектов, нажмите клавишу *Shift* и укажите на первый из объектов. Затем, не отпуская клавиши *Shift*, укажите на следующий из объектов и так далее до тех пор, пока не будут выбраны все необходимые вам объекты.

Если объекты, которые необходимо выделить, можно расположить в пределах некоторой прямоугольной области, используется другой способ. Воспользуйтесь кнопкой **Выделить** поля на панели инструментов или командой меню **Правка | Выделить** поля. Укажите курсором мыши на один из углов воображаемой области. Далее не отпуская кнопки, переместите курсор мыши в противоположный угол области. В процессе перемещения курсора мыши рамка указывает на положение области. После того как все нужные объекты будут находиться внутри рамки, отпустите кнопку мыши. В результате все объекты, находящиеся внутри рамки, окажутся выделенными.

Если вам необходимо удалить выделение с какого-либо из объектов, нажмите клавишу *Shift*, и укажите на требуемый объект. В результате маркеры объекта исчезнут. Если клавиша *Shift* не будет нажата, выделение будет снято со всех выделенных объектов, за исключением того, на который вы укажете.

После выделения одного или более объектов вы можете выполнять с ними различные действия. Например, для того чтобы переместить выделенные объекты, достаточно поместить курсор мыши внутрь одного из этих объектов, нажать кнопку мыши и, не отпуская ее, переместить группу объектов в требуемое место.

Свойства объектов

Аналогично объектам экранных форм, объекты отчетов имеют свойства, которыми вы можете управлять. Для того чтобы отобразить на экране меню свойств объекта, выделите нужный объект и нажмите правую кнопку мыши

Каждый из объектов имеет набор свойств со стандартными параметрами. Вы можете установить по своему усмотрению параметры для каждого из объектов отчета. Например, для того чтобы изменить тип шрифта, установленный по умолчанию для текста, выберите пункт меню **Изменить шрифт** и в открывшемся окне диалога “Шрифт” установите нужные параметры шрифта.

Размещение прямоугольников, линий

При создании отчетов вы можете использовать средства для размещения простейших графических изображений — прямоугольников, линий. Такими средствами являются кнопки **Нарисовать рамку в отчете**, **Провести линию в отчете**. Для того чтобы разместить в отчете какой-либо из объектов указанного типа, выполните следующие действия:

1. Выберите на панели инструментов требуемую кнопку. В результате курсор мыши примет форму карандаша.
2. Установите курсор на начальную точку линии или угол рамки.
3. Для того чтобы зафиксировать начальную точку объекта, нажмите кнопку мыши. Удерживая кнопку мыши, переместите ее в конечную точку линии или противоположный угол рамки. Для того чтобы зафиксировать объект, отпустите кнопку мыши.

Размещая объект, вы можете не особенно заботиться о его точном местоположении, форме и размерах. В дальнейшем вы всегда сможете изменить эти параметры. Для этого выделите требуемый объект, после чего с помощью соответствующего маркера установите требуемую форму или размеры.

Рамка
Изменить формат...
Поместить на задний план
Удалить рамку
Заккрыть меню

Каждый из объектов указанного типа имеет набор свойств, которые позволяют вам изменить параметры отображения объекта.

Опции меню **Изменить формат** позволяют изменить цвет, ширину, стиль объекта. В частности, при выборе этой опции для рамки из меню свойств на экране откроется окно диалога “Формат рамки” (рис. 9.13). Данное окно диалога предоставляет в ваше распоряжение следующие средства:

Опция	Назначение
Заливка	Устанавливает цвет внутри рамки, используя для этого цветовую палитру.

Опция	Назначение
Граница	Устанавливает цвет границы рамки, используя для этого цветовую палитру.
Ширина	Изменяет толщину линии. По умолчанию используется тонкая линия. Если вам потребуется изменить этот параметр, выберите необходимый вариант из предложенных.
Стиль	Позволяет вам использовать один из трех возможных типов пунктирных линий.

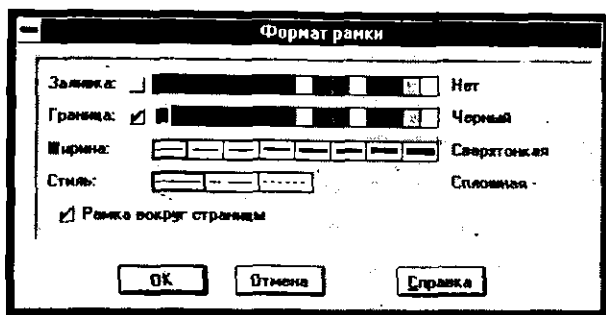


Рис. 9.13. Окно диалога "Формат рамки".

Размещение текста

Кнопка **Вставить надпись** на панели инструментов и команда меню **Вставка | Надпись** предназначены для размещения в отчете текстовых объектов. Вы можете использовать текстовые объекты в качестве заголовка страницы, подписей к полям, а также для отображения в отчете вспомогательной информации.

Для создания текстового объекта нажмите кнопку **Вставить надпись** на панели инструментов. На экране откроется окно диалога "Редактирование надписи" (рис. 9.14). В текстовом поле **Введите текст** напечатайте необходимую надпись и нажмите кнопку **Готово**. Окно диалога закроется и в отчете в месте расположения курсора появится прямоугольный объект, содержащий надпись. Установите курсор мыши в место отображения текста и зафиксируйте его положение нажатием курсора мыши. Не заботьтесь о точном расположении и размерах области, вы всегда сможете в дальнейшем изменить их с помощью маркеров.

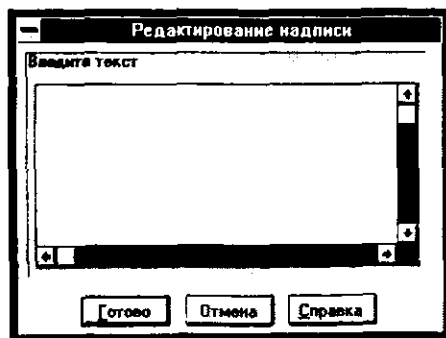


Рис. 9.14. Окно диалога "Редактирование надписи".

Для того чтобы получить доступ к свойствам текстового объекта, поместите курсор мыши внутрь объекта и нажмите правую кнопку. Измените шрифт текстового объекта с помощью команды меню **Формат | Шрифт** или свойства объекта **Изменить шрифт**. Скорректируйте размер текстового объекта с помощью бокового маркера, чтобы в нем помещалась вся надпись.

Размещение графических изображений

Вы можете разместить в отчете графические изображения, которые содержат необходимую информацию (например, чертежи или рисунки) или выполняют иллюстративные функции, улучшающие восприятие документа. Графические изображения могут представлять собой оцифрованную фотографию или просто отдельный символ. Эти изображения создаются с помощью графического редактора или снимаются с помощью сканера.

Для вставки графического изображения используется команда меню **Вставка | Рисунок** и кнопка панели инструментов **Вставить рисунок из файла**.

Размещения OLE-объекта

OLE-объекты создаются в процессе обмена информацией с другими приложениями Windows при использовании механизма работы с составными документами. Вы можете воспользоваться этим средством для размещения в отчетах dBASE графических изображений, текстовых документов и т.д., созданных с помощью других приложений Windows (графических и/или текстовых редакторов). Для вставки OLE-объектов используется команда меню **Вставка | Объект** и кнопка панели инструментов **Вставить новый объект OLE**.

Размещение полей

Основное назначение отчетов состоит в отображении данных, содержащихся в таблицах dBASE, поэтому поля можно считать наиболее важными из объектов. Кроме полей из таблиц dBASE вы можете использовать в отчетах вычисляемые, групповые и служебные поля.

Размещение полей таблицы в отчете осуществляется в два этапа — сначала выбирается поле из таблицы или запроса, используемых в качестве источника данных. Затем выбранное поле размещается в отчете и устанавливается его положение и размер. Для создания объекта используется кнопка панели инструментов **Вставить поле таблицы**.

Определение полей таблицы

Если вы предполагаете разместить поле, выберите требуемое наименование из списка полей таблиц, включенных в источник данных.

В качестве примера рассмотрим объединение полей в табличном отчете.

Пример 9.1. Объединение полей в табличном отчете

Табличный формат, который вы используете при разработке отчета в качестве исходного, содержит указанные вами поля исходной таблицы. Вы можете разместить каждое из полей в собственном столбце. Однако такое представление информации в напечатанном отчете не всегда читабельно. Например, в отчете для таблицы ADDRESS адрес каждого абонента будет отображен в пяти различных столбцах: "Индекс", "Страна", "Область", "Город", "Адрес". Отображение этих полей в одном

столбце сделает отчет более читабельным. В качестве примера поместим поля “Индекс”, “Страна”, “Город” и “Адрес” в одном столбце таблицы ADDRESS.

1. Выберите **Файл | Создать | Отчет** или нажмите мышью кнопку **Открыть** на панели инструментов, после чего выберите **Отчет**.
2. В открывшемся окне диалога “Выбор таблицы” выберите из списка таблицу ADDRESS и нажмите **ОК**. В результате на экране появится окно конструктора отчетов Crystal Reports с открытым окном диалога “Вставка поля базы данных”.
3. В разрабатываемом отчете ограничимся полями “Фамилия”, “Имя”, “Отчество”, “Предпр”, “Телефон”, “Индекс”, “Страна”, “Город”, “Адрес”. Сначала перенесем поля “Фамилия”, “Имя” и “Отчество”. Можно было бы выбрать и перенести сразу все нужные поля, но так как все они не поместятся по ширине отчета, то часть из них окажется потерянной и вам придется еще раз выбирать и переносить недостающие поля. Держа клавишу *Shift* нажатой, выберите курсором требуемые поля.
4. Нажмите кнопку **Вставить**. Указатель мыши принимает вид стопки документов.
5. Переместите “стопку документов” в место расположения фамилии в отчете и нажмите левую кнопку мыши для вставки полей данных “Фамилия”, “Имя” и “Отчество” в отчет.
6. Увеличьте ширину области данных так, чтобы друг под другом поместилось несколько полей. Для этого поместите курсор мыши на линию, ограничивающую снизу область данных. Курсор примет вид вертикальной двунаправленной стрелки. Нажмите кнопку мыши и, удерживая ее, переместите курсор вниз.
7. Выделите курсором мыши поле под надписью “Имя” и перенесите его, расположив под полем “Фамилия”. Так же перенесите поле “Отчество”, расположив его ниже.
8. Удалите надписи “Имя” и “Отчество”. Для этого их выделите и нажмите клавишу *Del*.
9. Выберите курсором мыши поле “Телефон” и нажав кнопку **Вставить**, перенесите его в отчет, расположив рядом с полем “Фамилия”.
10. Держа нажатой клавишу *Ctrl* (так как поля расположены не подряд), выберите курсором поля “Индекс”, “Страна”, “Город”, “Адрес”. Нажмите кнопку **Вставить** и перенесите их в отчет, расположив рядом с полем “Телефон”.
11. Перенесите поля “Страна”, “Город” и “Адрес”, расположив их под полем “Индекс”. Удалите лишние надписи.
12. Выберите курсором мыши поле “Предпр” и нажав кнопку **Вставить**, перенесите его в отчет, расположив рядом с полем “Индекс”.
13. Закройте окно диалога, нажав кнопку **Готово**.
14. Измените надпись “Фамилия” на “Абонент”. Для этого выделите ее курсором и выполните команду **Правка | Надпись**. В открывшемся окне диалога

“Редактирование надписи” в поле коррекции замените “Фамилия” на “Абонент” и нажмите кнопку **Готово**.

15. Аналогичным образом измените надпись “Индекс” на “Адрес”.
16. Выводяйте расположение полей и промежутки между ними.
17. В заключение измените начертание шрифтов всех объектов с обычного на жирный. Окончательный формат отчета показан на рис. 9.15.

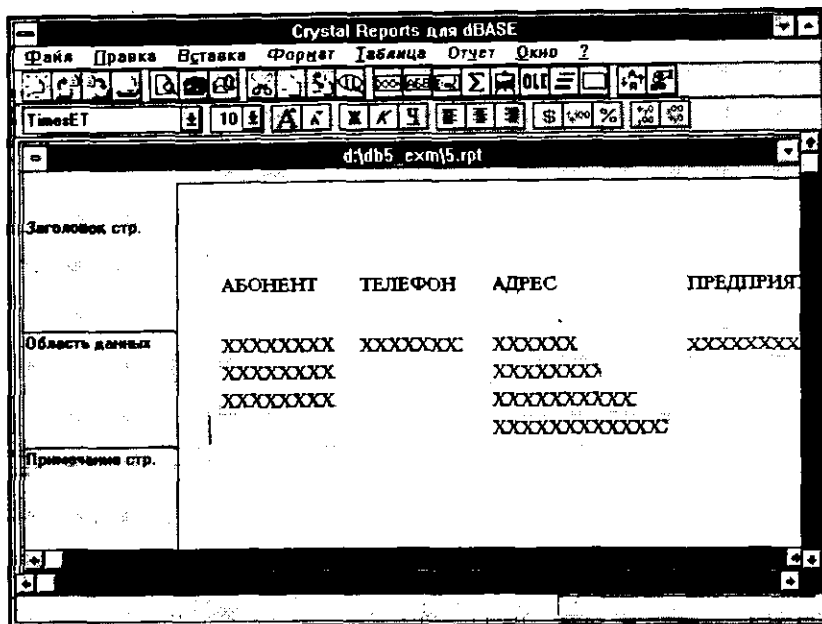


Рис. 9.15. Формат отчета в окне конструктора отчетов.

18. Просмотрите внешний вид отчета на экране, выполнив команду **Файл | Печать | Просмотр** или нажав кнопку **Вывести отчет** в окне просмотра на панели инструментов. На рис. 9.16 показана часть отчета, которая будет отображена на экране.



Обратите внимание на то, что в окне просмотра каждый объект отчета можно корректировать (изменять его размеры, местоположение). Эти изменения сохраняются после выхода из окна просмотра.

Определение вычисляемых полей

Вычисляемые поля позволяют выполнять математические операции над полями исходной таблицы и отображать результат в отчете. При этом вычисления могут производиться как горизонтально над полями одной записи, так и вертикально над данными поля для вычисления групповых значений. При вычислениях могут быть использованы текстовые и числовые поля, а также поля дат и времени. При этом выражение для вычисления может содержать любое допустимое выражение dBASE.

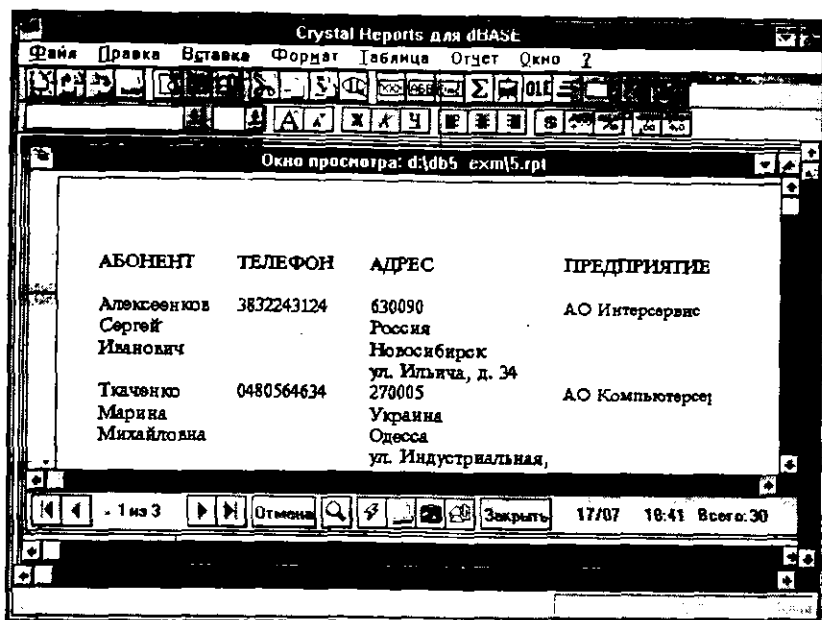


Рис. 9.16. Размещение нескольких полей в одном столбце.

Для того чтобы определить вычисляемое поле, выберите из меню команду **Вставка | Поле выражения**. На экране появится окно диалога “Вставка выражения” (рис. 9.17).

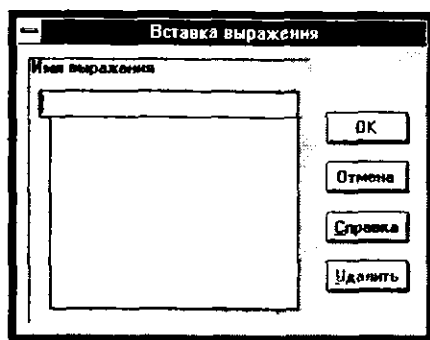


Рис. 9.17. Окно диалога “Вставка выражения”.

В текстовое поле **Имя выражения** введите имя выражения, которое будет использоваться для ссылки на это выражение. Имя выражения может содержать любую комбинацию букв, цифр и знаков. Нажмите кнопку **ОК**. На экране появится окно диалога “Построение выражений” (рис. 9.18).

Для определения вычисляемого поля можно также использовать кнопку панели инструментов **Построить выражение**.

Простейшее выражение для вычисления состоит из переменных и констант, соединенных арифметическими операторами (“+”, “-”, “*”, “/”). Для изменения порядка выполнения арифметических операций в выражении для вычисления используются круглые скобки.

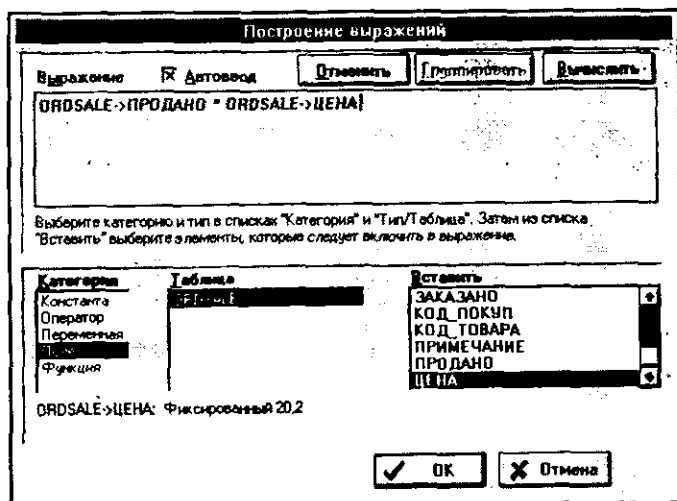


Рис. 9.18. Окно диалога "Построение выражений".

Выражения для вычислений могут содержать константы всех типов (текстовые строки, числовые, даты, время). В качестве переменных используются наименования полей.



Вычисляемые поля являются временными полями. Содержащиеся в них результаты не сохраняются и существуют во время отображения на экране или выводе на принтер содержащий их отчет.

При выполнении горизонтальных вычислений над каждой записью таблицы вычисляемые поля должны располагаться в области данных.

Пример 9.2.

В качестве примера воспользуемся вычисляемыми полями для получения дополнительной информации о продажах каждого из товаров. Исходные данные о количестве товара, проданного в каждой партии (поле "Продано"), и о цене товара каждого вида (поле "Цена") содержатся в таблице ORDSALE. Отобразим в отчете данные о стоимости товаров, проданных в каждой партии. В этом случае вычисляемое поле отображает результат вычисления над двумя полями одной записи исходной таблицы и должно располагаться в области данных.

1. Выберите **Файл | Создать | Отчет** или нажмите мышью кнопку **Открыть** на панели инструментов, после чего выберите **Отчет**.
2. В открывшемся окне диалога "Выбор таблицы" выберите из списка таблицу ORDSALE и нажмите **OK**. В результате на экране появится окно конструктора отчетов Crystal Reports с открытым окном диалога "Вставка поля базы данных".
3. В разрабатываемом отчете ограничимся полями, содержащими сведения о покупках клиентов: "Код_покуп", "Код_товара", "Дата_прод", "Продано", "Цена". Держа нажатой клавишу *Shift*, выберите курсором требуемые поля.

4. Нажмите кнопку **Вставить**. Указатель мыши принимает вид стопки документов. Переместите стопку в место расположения первого поля в отчете и нажмите левую кнопку мыши для вставки полей данных в отчет.
5. Нажмите кнопку **Готово** для закрытия окна диалога.
6. Для создания вычисляемого поля выберите команду **Вставка | Поле выражения**.
7. В открывшемся окне диалога "Вставка выражения" в текстовое поле **Имя выражения** введите имя выражения, которое будет использоваться для ссылки на это выражение. Нажмите кнопку **ОК**.
8. Для вычисления стоимости товаров, проданных в каждой партии, вам нужно ввести вычисляемое поле с выражением:

Продано*Цена

В списке **Вставить** укажите на поле "Продано". Дважды нажмите кнопку мыши, и указанное вами поле будет скопировано в строку ввода выражения.
9. Для переноса в вычисляемое выражение знака умножения в списке **Категория** выберите значение **Оператор**, а в списке **Вставить** — значение **Умножение** и двойным нажатием мыши перенесите его в выражение.
10. В списке **Категория** выберите значение **Поле**, а в списке **Вставить** — поле "Цена" и также перенесите его в выражение. Определение вычисляемого поля завершите нажатием **ОК**.
11. Создайте надпись для вычисляемого поля "Стоимость партии", выполнив команду меню **Вставка | Надпись** и расположив его над созданным полем.

СПИСОК ПРОДАЖ

КОД ПОКУПАТЕЛЯ	КОД ТОВАРА	ДАТА ПРОДАЖИ	ПРОДАНО	ЦЕНА	СТОИМОСТЬ ПАРТИИ
555 555,56	35 555,56	31/12/99	5 555,56	555 555,56	5 555 555,56

Рис. 9.19. Вычисляемое числовое поле.

Crystal Reports для dBASE

Файл Правка Вставка Формат Таблица Отчет Окно ?

Окно просмотра: d:\db5_ект\6.rpt

СПИСОК ПРОДАЖ

КОД ПОКУПАТЕЛЯ	КОД ТОВАРА	ДАТА ПРОДАЖИ	ПРОДАНО	ЦЕНА	СТОИМОСТЬ ПАРТИИ
1,00	2,00	12/04/95	5,00	610,00	3 050,00
2,00	1,00	12/01/95	2,00	2 600,00	5 200,00
2,00	2,00	4/01/95	15,00	625,00	9 375,00
2,00	2,00	11/02/95	7,00	630,00	4 410,00

1 из 2 Отмена Закрыть Сегодня 09:39 Всего: 43

Рис. 9.20. Отчет с вычисляемым числовым полем.

- Поскольку надписи занимают много места, разместите в две строки заголовки "Код покупателя", "Код товара", "Дата продажи" и "Стоимость партии", создав две надписи, и уменьшите ширину столбцов с указанными заголовками.
- Поместите заголовок "Список продаж" в Заголовок страницы отчета.
- Вставьте рамки и разделительные линии. В результате ваш отчет будет иметь вид, представленный на рис. 9.19. На рис. 9.20 показан просмотр разработанного отчета.

Окно просмотра содержит собственную панель инструментов, которая содержит дополнительные кнопки, предназначенные для управления просмотром отчета:

	Первая страница		Последняя страница
	Предыдущая страница		Следующая страница
	Просмотр отчета в уменьшенном формате		Обновление данных отчета

Кроме того, данная панель содержит кнопки, которые выполняют свои стандартные функции.



В отличие от окна формы, в окне отчета указанные кнопки предназначены для перемещения не по записям формы, а по страницам отчета. При этом на каждой странице могут находиться несколько записей.



Если вам требуется напечатать в отчете большое число полей или достаточно длинные Мемо-поля, имеет смысл воспользоваться отчетом в свободном формате или "наклейками".

Определение групповых полей

Групповые поля, также как и вычисляемые поля, являются результатом вычисления над данными, содержащимися в полях исходной таблицы. Однако, если вычисляемые поля предназначены для вычислений над полями одной записи, групповые поля вычисляют статистические данные по полям исходной таблицы. Например, для таблицы ORDSALE вы можете с помощью групповых полей отобразить в отчете общее количество продаж, количество проданных товаров и т.д.

При создании групповых полей вы можете выполнять следующие групповые операции:

Операция	Назначение	Типы полей
Сумма	Сумма значений	Числовые
Среднее	Среднее по значениям	Числовые
Минимум	Минимальное значение	Текстовые, числовые, даты и время
Максимум	Максимальное значение	Текстовые, числовые, даты и время
Подсчет	Количество ненулевых значений	Все
Несмещенная дисперсия	Несмещенная дисперсия	Числовые
Несмещенное стандартное отклонение	Несмещенное стандартное отклонение	Числовые
Смещенная дисперсия	Смещенная дисперсия	Числовые
Смещенное стандартное отклонение	Смещенное стандартное отклонение	Числовые
Специальный подсчет	Количество неповторяющихся значений	Все

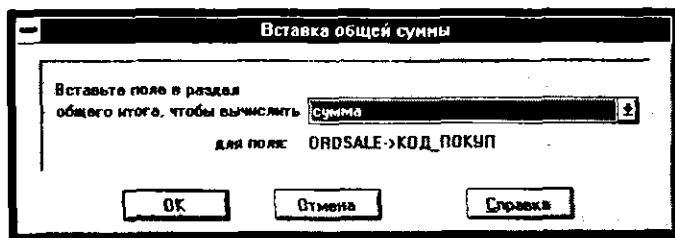


Рис. 9.21. Окно диалога "Вставка общей суммы".

Для определения групповых полей используется окно диалога “Вставка общей суммы” (рис. 9.21). Выделите в отчете поле таблицы, по значениям которого хотите создать групповое поле. Выполните команду меню **Вставка | Поле итога | Общая сумма**, в результате чего откроется окно диалога “Вставка общей суммы”. Нажмите кнопку раскрытия списка. dBASE отобразит на экране список групповых операторов, с помощью которых можно выполнить групповые вычисления над значениями полей таблицы.

Выберите из списка требуемый тип групповой операции и нажмите кнопку **ОК**, в результате dBASE возвратится в окно конструктора отчетов.

В качестве примера на рис. 9.22 и 9.23 показан отчет, содержащий два групповых поля. В обоих полях для группировки использовалось поле исходной таблицы “Код_покуп”. Однако при размещении первого поля была использована операция **Специальный подсчет**, а при размещении второго — **Подсчет**.



Обратите внимание на то, что в отчете появился дополнительный раздел **Общая сумма**, в который dBASE поместил созданные групповые поля.

Размещение специальных полей

Специальные поля содержат информацию о создаваемом отчете и текущей дате. Для их создания используется команда меню **Вставка | Специальное поле**. К категории специальных полей относятся следующие поля:

Дата	Текущая дата системных часов вашего компьютера в формате Windows.
Номер страницы	Номер текущей страницы.
Номер записи	Номер текущей записи в области данных.
Номер группы	Номер группы отчета.



Вы не можете изменить значения этих полей.

Для того чтобы разместить в отчете любое из этих полей, вам необходимо выполнить стандартные действия по определению и размещению поля.

Вы можете воспользоваться полем номера записей для отображения в отчете нумерации напечатанных записей, полями даты и номера страницы для расположения в отчете текущей даты и нумерации страниц отчета.

Для корректного отображения в отчете текущей даты часы вашего компьютера должны быть установлены правильно. Если при просмотре на экране вы выяснили, что часы компьютера показывают неверную дату, скорректируйте ее. Для этого с помощью **Alt+Tab** перейдите в Диспетчер Программ Windows. Далее раскройте группу **Главная**, выберите **Панель Управления**, а затем **Дата/Время**. Введите правильную дату и возвратитесь в dBASE.

Удаление поля

Для удаления одного или нескольких полей выделите их и нажмите клавишу **Del**. В результате поле исчезнет из отчета.

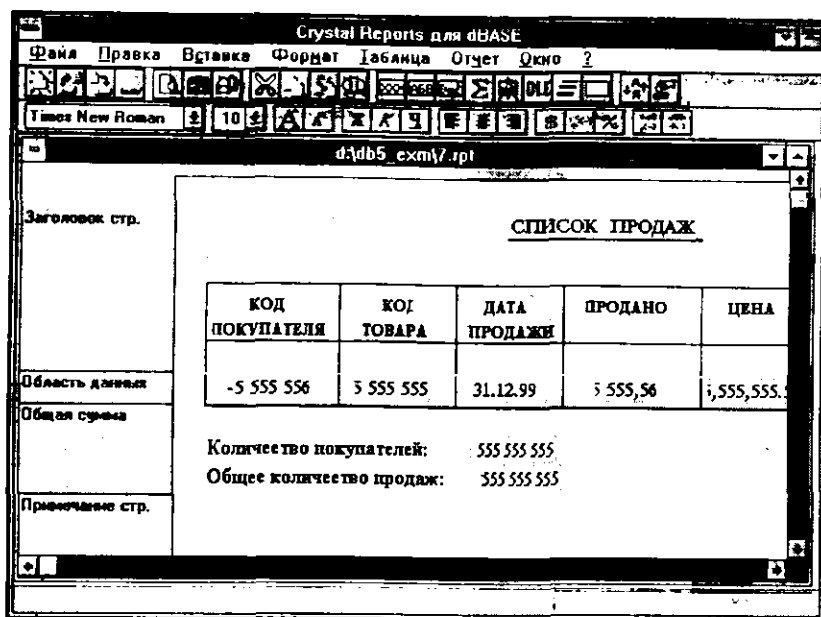


Рис. 9.22. Групповые поля, размещенные в отчете

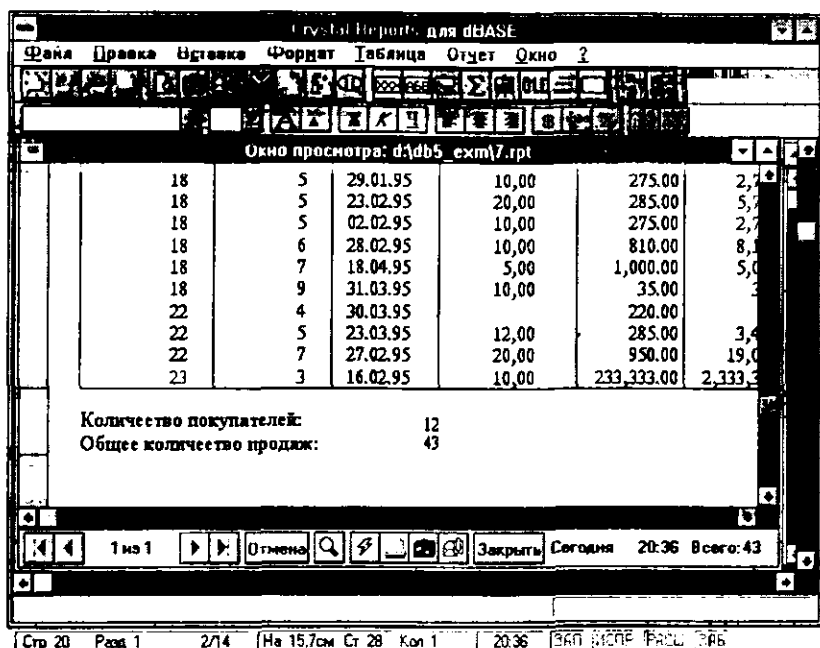


Рис. 9.23. Фрагмент отчета в окне просмотра с групповыми полями.

Изменения форматов полей

После размещения поля в отчете вы можете изменить для него формат отображения данных. Для этой цели предназначено свойство **Изменить формат** в меню свойств поля. Формат отображения определяется типом данных, поэтому список опций, который появится на экране в результате выбора свойства **Изменить формат**,

будет иметь различный вид в зависимости от типа указанного поля. В качестве примера на рис. 9.24 представлено окно диалога “Числовой формат”.

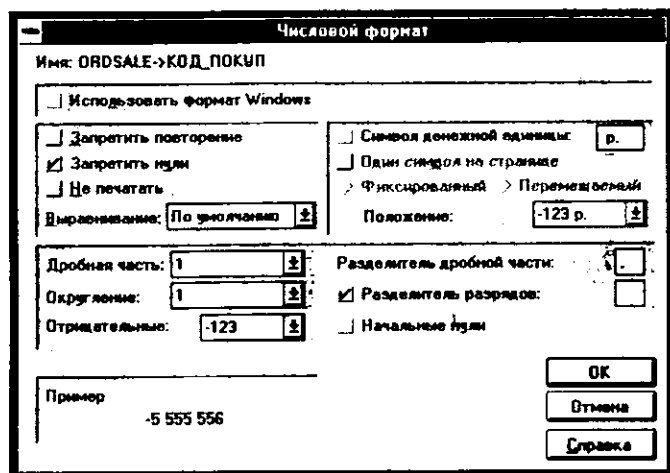


Рис. 9.24. Окно диалога “Числовой формат”.

Данное окно диалога позволяет для числовых полей задать вид представления чисел в отчете: количество разрядов в дробной части, разделитель разрядов, разделитель дробной части, будут ли печататься нулевые значения и т.д.

Изменение вычисляемых и групповых полей

Для того чтобы внести изменения в вычисляемое поле, вызовите меню его свойств и с помощью опции Редактировать выражение откройте окно диалога “Построение выражения”. Дальнейшие действия по редактированию поля полностью аналогичны его размещению.

Для того чтобы внести изменения в итоговое поле, вызовите меню его свойств и с помощью опции Изменить итоговую операцию откройте окно диалога “Поле итога” (рис. 9.25). Далее выберите новый тип итоговой операции.

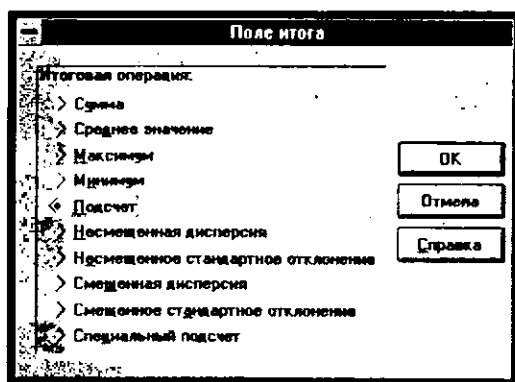


Рис. 9.25. Окно диалога “Поле итога”.

Изменение размеров объектов

Для того чтобы изменить размеры объекта, выполните следующие действия:

1. Выберите объект, в результате чего вокруг него появятся маркеры.
2. Установите курсор мыши на один из маркеров. При этом курсор изменит свою форму, показывая в каком направлении вы можете изменять размеры объекта.
3. Нажмите кнопку мыши и переместите маркер в соответствующее место. Для завершения изменения размера отпустите кнопку мыши.

Вы можете также переместить объект. Для этого после выбора объекта расположите курсор мыши в его пределах и нажмите кнопку мыши. Далее, удерживая ее, переместите объект в требуемое место, после чего отпустите кнопку мыши.

Меню свойств объекта

Кроме приведенных выше средств, предназначенных для управления объектом, для настройки его параметров вы можете воспользоваться меню свойств объекта, представленное на рис. 9.26.

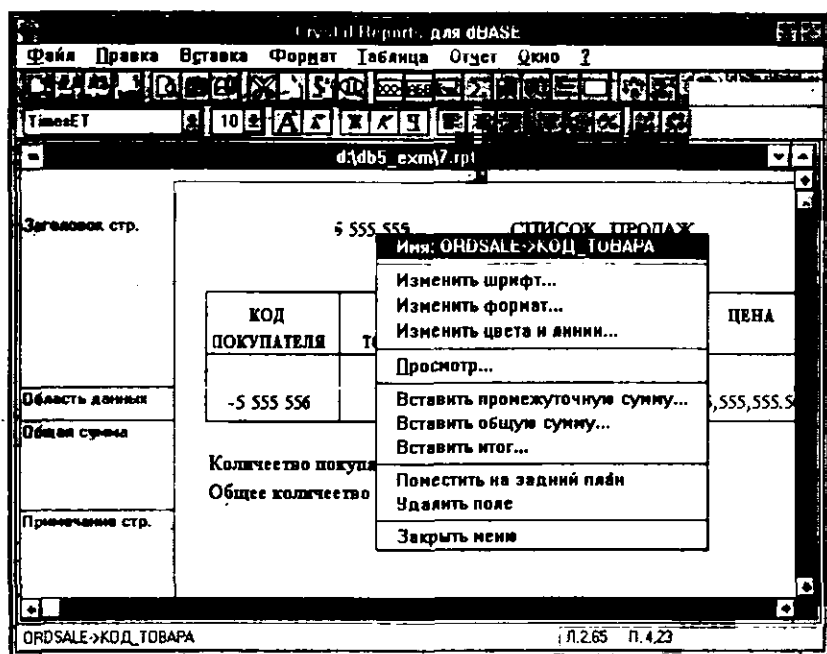


Рис. 9.26 Меню свойств объекта.

Ниже приведены состав и назначение опций меню свойств объекта.

Опция	Назначение
Изменить шрифт	Устанавливает шрифт поля.
Изменить формат	Устанавливает формат отображения данных.

Опция	Назначение
Изменить цвета и линии	Устанавливает цвет фона поля, цвет шрифта, стиль и цвет линии контура.
Просмотр	Позволяет просмотреть значения поля.
Вставить промежуточную сумму	Открывает окно диалога "Вставка промежуточной суммы".
Вставить общую сумму	Открывает окно диалога "Вставка общей суммы".
Вставить итог	Открывает окно диалога "Вставка итога".
Поместить на задний план	Помещает объект на задний план.
Удалить поле	Удаляет поле.

Отображение полей в двух или более строках

Отображение в отчете текстового поля большой длины может вызвать определенные неудобства. Например, в таблице присутствует поле шириной 60 символов. Если вы поместите такое поле на одной строке отчета, то он окажется неприемлемо широким. Попытка уменьшить ширину поля может привести к тому, что часть данных из этого поля не будет отображена в отчете.

Опция Печатать на нескольких строках пункта Изменить формат меню свойств объекта позволяет автоматически отобразить значения текстовых полей в нескольких строках. Используя режим с переносом слов, вы сможете уменьшить ширину отчета и отобразить в нем всю информацию поля.

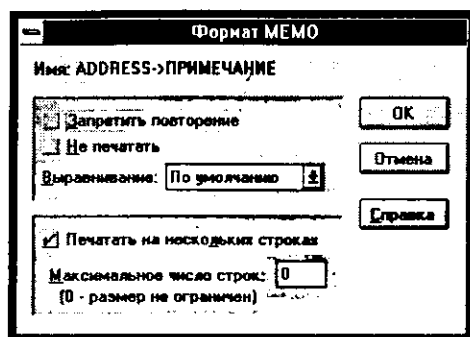


Рис. 9.27. Окно диалога "Формат МЕМО".

dBASE автоматически использует столько строк, сколько необходимо для отображения всей информации, содержащейся в Мемо-поле. На рис. 9.27 показано окно диалога "Формат МЕМО" для Мемо-поля. На рис. 9.28 вы видите просмотр отчета, содержащего такое поле.

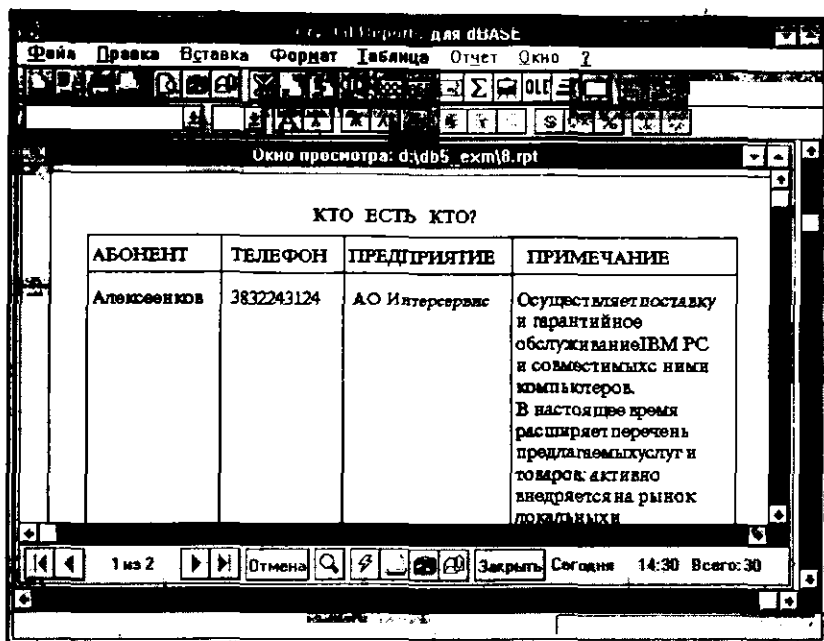


Рис. 9.28. Представление поля "Примечание" в нескольких строках.

Печать отчетов и отображение отчетов на экране

Просмотр отчета на экране

Обычно процесс создания формата отчета носит итеративный характер. Для того чтобы просмотреть, как выглядит отчет, вы можете напечатать отчет или отобразить его на экране. Для просмотра отчета на экране нажмите кнопку **Вывести отчет** в окне просмотра на панели инструментов. В качестве альтернативных средств вы можете также использовать команду **Файл | Печать | Просмотр**.

dBASE каждый раз выводит на экран только одну страницу отчета. Для того чтобы перейти к нужной странице отчета, воспользуйтесь кнопками перемещения, расположенными на панели инструментов окна просмотра. dBASE облегчает идентификацию текущей страницы отчета, отображая ее номер в правой части строки состояния.

Во время просмотра отчета на экране в панели инструментов присутствует кнопка **Заккрыть**. Нажатие этой кнопки приведет к завершению просмотра отчета и возврату в окно конструктора отчетов.

Печать отчета

Для того чтобы напечатать весь отчет, выберите **Файл | Печать | Принтер** или нажмите кнопку **Принтер** на панели инструментов. Оба эти средства доступны в окне конструктора отчетов и окне просмотра отчета. Независимо от того, каким из средств вы воспользуетесь, на экране появится окно диалога "Печать" (рис. 9.29).

По умолчанию dBASE предлагает вам напечатать отчет полностью в одном экземпляре. Если вам требуется напечатать одну или несколько страниц отчета, выберите опцию **Страницы**, после чего укажите диапазон печатаемых страниц. При этом вы

можете предложить dBASE напечатать страницы в обратном порядке. Для этого введите большее число в строку ввода С, а меньшее — в строку ввода По.

Строка ввода Копий позволяет напечатать за один раз требуемое число экземпляров, а опция Разобрать по копиям устанавливает порядок печати экземпляров.

После того как вы определитесь с параметрами печати, выберите ОК.

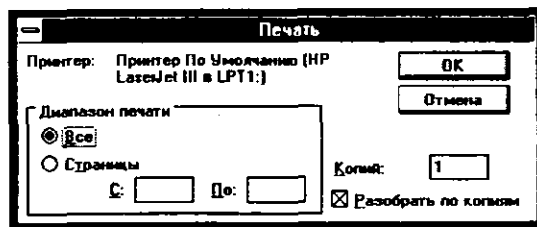


Рис. 9.29. Окно диалога "Печать".

Управление принтером

Подобно большинству приложений, работающих под Windows, установка параметров принтера осуществляется не средствами самого dBASE, а Windows. Это означает, что вы можете использовать для печати любой из списка принтеров, установленный на вашем компьютере для работы под Windows, и выполнить настройки принтера, доступные из Windows.

Для выбора принтера или изменения его настройки выберите **Файл | Выбор принтера**, в результате на экране появится окно диалога "Выбор принтера" (рис. 9.30). В верхней части содержится список принтеров, установленных в вашей системе. Принтер, используемый по умолчанию выделен более интенсивной окраской или цветом. Вы можете выбрать любой принтер из списка. Если требуемого принтера не окажется в списке, выберите **Настройка**, в результате будет открыто окно диалога "Настройка принтера" утилиты Панель Управления Windows (рис. 9.31).

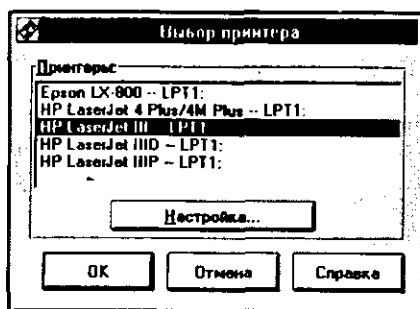


Рис. 9.30. Окно диалога "Выбор принтера".

Для изменения параметров настройки принтера воспользуйтесь раскрывающимися списками **Разрешение**, **Размер Бумаги**, **Источник Бумаги** и **Память**. Кроме того, вам доступны опции для выбора картриджа и настройки шрифтов.

Опция **Ориентация** позволяет вам указать, как должен быть напечатан на странице отчет: вертикально (**Портрет**) или горизонтально (**Ландшафт**).

Для получения подробной информации о настройке параметров принтера в окне диалога "Настройка принтера" обратитесь к документации по Windows и руководству по вашему принтеру.

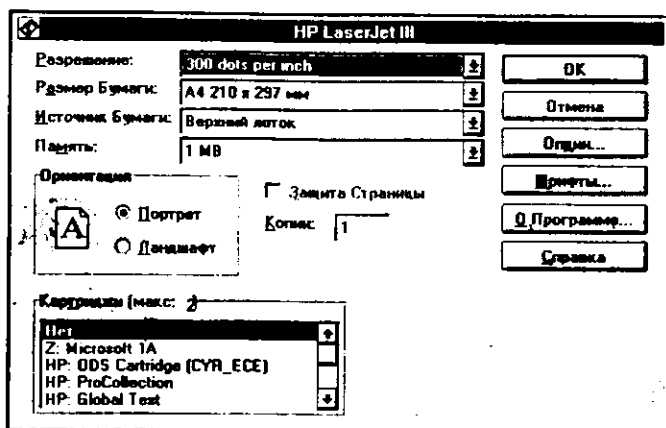


Рис. 9.31. Окно диалога “Настройка принтера” утилиты Панель Управления Windows.



Если вам не требуется высокое качество печати, но желательно быстро напечатать отчет вы можете для лазерных принтеров уменьшить разрешение с 300 до 150 и даже до 75.

Сохранение формата отчета

Для сохранения разработанного отчета воспользуйтесь командой **Файл | Сохранить как...** dBASE запишет отчет в файл с указанным вами именем и расширением .RPT.



Если для разработки требуемого формата отчета вам потребуется выполнить достаточно большой объем работы, не забывайте время от времени выполнять команду **Файл | Сохранить**. Это позволит вам сохранить результаты при различных внештатных ситуациях.

Отказ от текущего формата отчета

Вы можете покинуть конструктор отчетов без сохранения текущего формата отчета. Для этого воспользуйтесь кнопкой системного меню. dBASE откроет окно, с помощью которого предложит вам подтвердить отказ от разработанного макета отчета (рис. 9.32).

Вы можете сохранить внесенные изменения, выбрав **Да**. В случае выбора **Нет** dBASE выполнит отказ и перейдет в главное окно. Если вы решили остаться в окне конструктора отчетов, выберите **Отмена**.

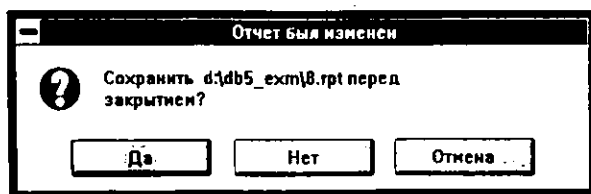


Рис. 9.32. Предупреждение dBASE о необходимости сохранения отчета.



Мы не рекомендуем вам злоупотреблять выходом из окна конструктора отчетов без сохранения результатов работы. Даже если в разработанном отчете присутствуют ошибки, часто оказывается легче их исправить, чем начать все с нуля.

Модификация созданных форматов отчетов

После сохранения текущего формата отчета dBASE переходит в главное окно. Для того чтобы продолжить работу над форматом отчета, воспользуйтесь командой **Файл | Открыть** или кнопкой **Открыть существующий отчет** на панели инструментов. Независимо от того, каким из этих средств вы воспользуетесь, dBASE откроет на экране окно диалога “Открытие файла”, с помощью которого предложит указать требуемый отчет (рис. 9.33).

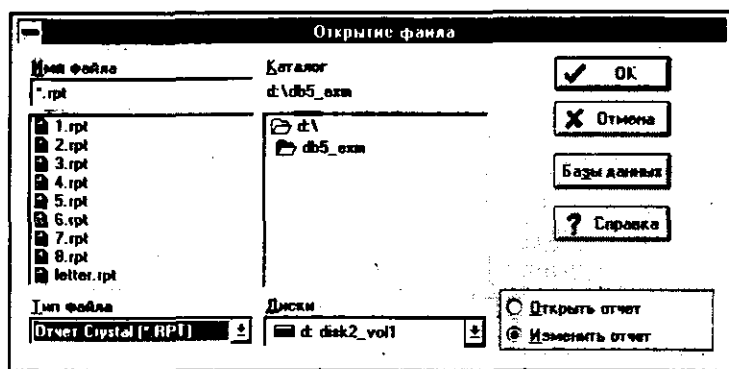


Рис. 9.33. Окно диалога “Открытие файла”.

Раздел “Имя файла” содержит список отчетов, расположенных в рабочем каталоге. Вы можете непосредственно ввести имя нужного отчета в строке ввода или выбрать его из списка. В том случае, если в рабочем каталоге требуемый вам отчет отсутствует, вы можете ввести полное имя нужного каталога перед именем файла в строку ввода **Имя файла** или выбрать нужный каталог используя область выбора каталога.

Примеры создания специальных отчетов

Пример 9.3. Печать отсортированных таблиц

В качестве первого примера напечатаем табличный отчет для таблицы ADDRESS, в котором записи по полям “Фамилия”, “Имя” и “Отчество” будут упорядочены в порядке возрастания значений.

1. Откройте любой из созданных отчетов для таблицы ADDRESS.
2. Выполните команду меню **Отчет | Порядок сортировки записей**. Откроется окно диалога “Порядок сортировки записей” (рис. 9.34).
3. Укажите поля для сортировки (“Фамилия”, “Имя” и “Отчество”), перенеся их с помощью кнопки **Добавить** из списка **Поля отчета** в список **Сортировка по полям**.

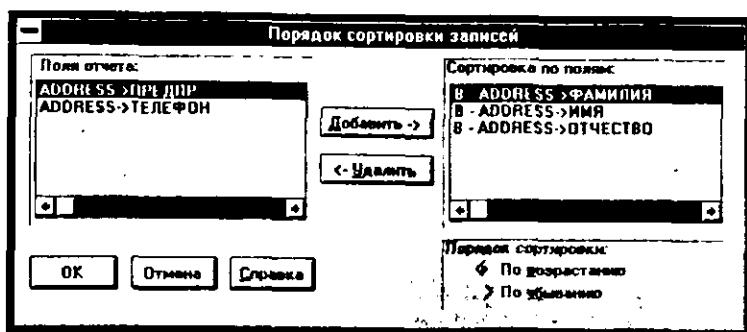


Рис. 9.34. Окно диалога “Порядок сортировки записей”.

4. Установите переключатель **Порядок сортировки**, выбрав нужный вариант, и нажмите кнопку **OK**.
5. Просмотрите на экране требуемый отчет (рис. 9.35). Если вы удовлетворены результатом просмотра, подготовьте принтер к работе и выполните команду **Файл | Печать | Принтер**.

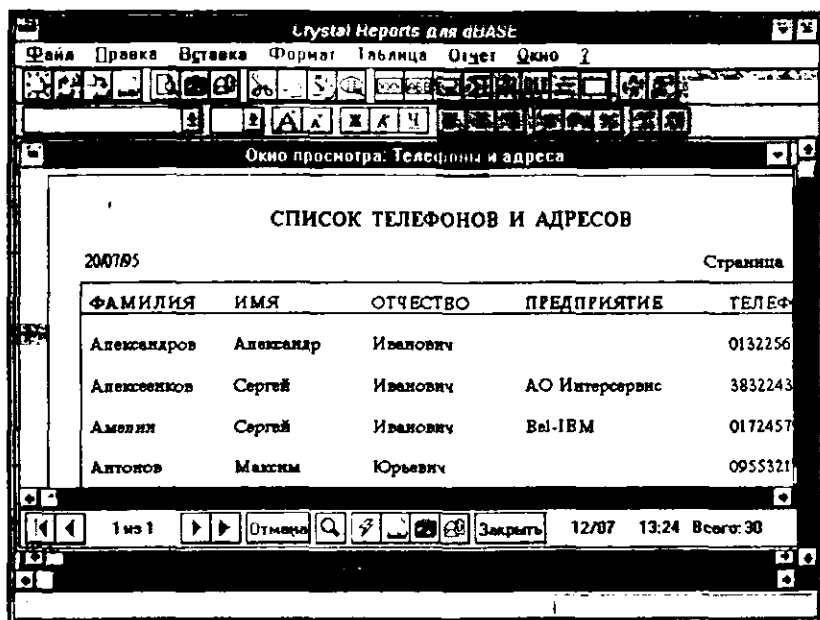


Рис. 9.35. Отчет с упорядоченными в порядке возрастания данными по значениям “Фамилия”, “Имя” и “Отчество”.

Пример 9.4. Создание отчета с групповыми значениями по вычисляемым полям

Запрос не является единственным средством, которое позволяет отобразить в отчете групповые значения по вычисляемым поля. Это позволяет сделать и конструктор отчетов Crystal Reports.

В качестве примера создадим отчет, в котором отобразим стоимость всех проданных товаров (таблица ORDSALE). На рис. 9.23 показан отчет, в котором представлена стоимость товаров по каждой проданной партии. Для вычисления суммы по всем значениям вычисляемого поля, воспользуемся конструктором отчетов.

СПИСОК ПРОДАЖ

КОД ПОКУПАТЕЛЯ	КОД ТОВАРА	ДАТА ПРОДАЖИ	ПРОДАНО	ЦЕНА	СТОИМОСТЬ ПАРТИИ
555 555,56	35 555,56	31/12/99	3 555,56	555 555,56	5 555 555,56
Общая стоимость проданных товаров:					5 555 555,56

Рис. 9.36. Отчет для вычисления стоимости всех проданных товаров.

За основу возьмем отчет, разработанный в примере 9.2. В нем содержатся сведения о покупках клиентов: “Код покупателя”, “Код товара”, “Дата продажи”, “Продано”, “Цена” и вычисляемое поле “Стоимость партии” со значением:

Продано*Цена

1. Откройте созданный ранее отчет.
2. Выделите в отчете поле “Стоимость партии”. Выполните команду меню Вставка | Поле итога | Общая сумма.
3. В открывшемся окне диалога “Вставка общей суммы” нажмите кнопку раскрытия списка. dBASE отобразит на экране список групповых операторов. Выберите из списка значение “Сумма” и нажмите кнопку ОК.
4. Создайте надпись “Общая стоимость проданных товаров” и расположите ее рядом с созданным полем (рис. 9.36).
5. Просмотрите созданный отчет в окне просмотра (рис. 9.37). В нижней части отчета вы видите значение, представляющее стоимость всех продаж.

Crystal Reports для dBASE

Файл Правка Вставка Формат Таблица Отчет Окно ?

Окно просмотра: d:\db5\exm\6.rpt

18,00	3,00	30/03/95	,00	1 120,00	,00
18,00	5,00	29/01/95	10,00	275,00	2 750,00
18,00	5,00	23/02/95	20,00	285,00	5 700,00
18,00	5,00	2/02/95	10,00	275,00	2 750,00
18,00	6,00	28/02/95	10,00	810,00	8 100,00
18,00	7,00	18/04/95	5,00	1 000,00	5 000,00
18,00	9,00	31/03/95	10,00	35,00	350,00
22,00	4,00	30/03/95	,00	220,00	,00
22,00	5,00	23/03/95	12,00	285,00	3 420,00
22,00	7,00	27/02/95	20,00	950,00	19 000,00
,00	,00		,00	233 333,00	,00
Общая стоимость проданных товаров:					512 625,00

1 из 1 Отмена Закрыть 19/07 09:39 Всего: 43 10

Рис. 9.37. Фрагмент отчета в окне просмотра.

Пример 9.5. Печать наклеек

В настоящем примере вам предстоит создать почтовые наклейки.

Почтовые наклейки

Тип наклеек: Специальные наклейки

Последовательность печати

Слева направо

Сверху вниз

Число наклеек

По ширине страницы 2

По вертикали 5

Макет почтовых наклеек:

Полов страницы

Верхнее 1,27 см

Нижнее 1,27 см

Левое 0,00 см

Правое 0,00 см

Размер наклейки

Ширина 9,19 см

Высота 4,54 см

Интервал между наклейками

По горизонтали 0,00 см

По вертикали 0,00 см

OK Отмена Справка

Рис. 9.38. Окно диалога "Почтовые наклейки".

1. Выберите **Файл | Создать | Почтовые наклейки**. В окне диалога "Выбор файла данных" укажите таблицу ADDRESS, в которой содержатся данные для создания наклеек, после чего нажмите OK.
2. На экране откроется окно диалога "Почтовые наклейки" (рис. 9.38). Нажмите кнопку раскрытия списка **Тип наклеек**. dBASE предусматривает создание почто-

вых наклеек самых разных типов. Вам только остается выбрать из большого списка устраивающий вас тип.

- Каждый тип наклеек по умолчанию имеет свои размеры, однако dBASE позволяет скорректировать их. Установите необходимые вам размеры наклейки и нажмите кнопку ОК. Вы перейдете в окно конструктора отчетов (рис. 9.39).

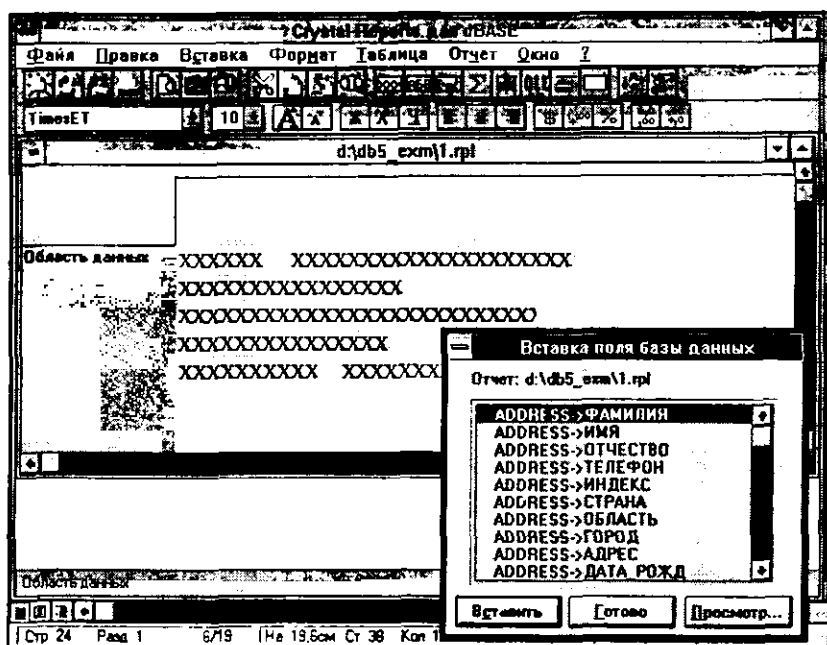


Рис. 9.39. Расположение данных при создании почтовых наклеек.

- В окне конструктора отчетов Crystal Reports из окна диалога "Вставка поля базы данных" перенесите в поле данных следующие поля: "Фамилия", "Имя", "Отчество", "Индекс", "Страна", "Город" и "Адрес". Выбор полей завершите нажатием Готово.
- В окне конструктора переместите поля так, как это принято для почтовых наклеек в конкретной стране. В нашем случае будем считать, что почтовые наклейки изготавливаются для рассылки в пределах СНГ, поэтому расположим их так, как показано на рис. 9.39.
- Нажмите кнопку **Вывести отчет** в окне просмотра на панели инструментов. В результате вы увидите на экране отчет, показанный на рис. 9.40.
- Если вам потребуется изменить количество наклеек, которые будут напечатаны на одной странице, или порядок их расположения, выберите **Макет наклеек** из меню свойств объекта и в окне диалога "Почтовые наклейки" (рис. 9.38) задайте новые параметры.
- Сохраните отчет, выбрав **Файл | Сохранить как**.
- Если работа над отчетом закончена напечатайте его, воспользовавшись командой **Файл | Печать | Принтер**.

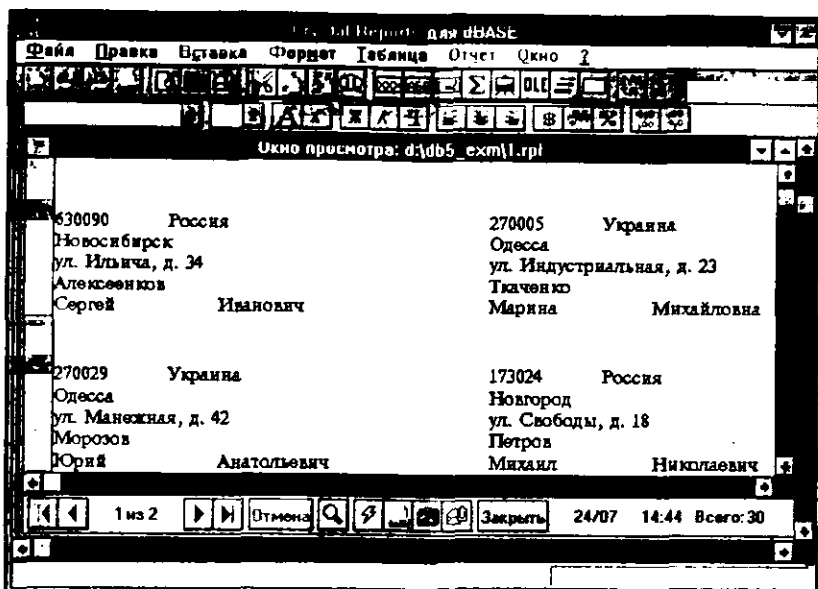


Рис. 9.40. Просмотр почтовых наклеек.

Пример 9.6. Подготовка писем

При подготовке стандартного письма для рассылки клиентам воспользуемся отчетом в свободном формате для таблицы ADDRESS.

1. Выполните команду **Файл | Создать | Отчет** и укажите ADDRESS в качестве единственной таблицы, которую следует включить в отчет.
2. Поскольку нам необходимо напечатать на каждой странице только одно письмо, увеличьте ширину области **Заголовок** страницы и **Область данных**.
3. Поместите в области заголовка страницы текстовые объекты:

Дата :

Адресат :

От : Генерального директора компании Bel-IBM Костина Г.В.

Содержание : Информация о новых компьютерах

Целесообразно разместить надписи "От :", "Генерального директора компании Bel-IBM Костина Г.В.", "Содержание :" и "Информация о новых компьютерах" как различные объекты. В дальнейшем это облегчит выравнивание полей и объектов отчета.

4. Разместите поля "Фамилия", "Имя" и "Отчество" после надписи "Адресат:". После надписи "Дата :" разместите специальное поле "Дата".
5. Оставьте перед обращением достаточно свободного места, разместите поля "Имя" и "Отчество", перед которыми поместите надпись "уважаемый".

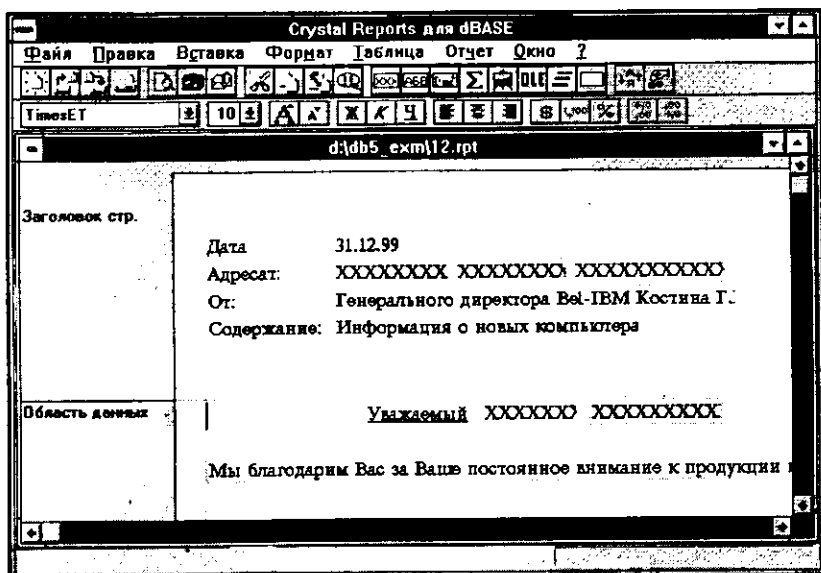


Рис. 9.41. Отчет информационного письма.

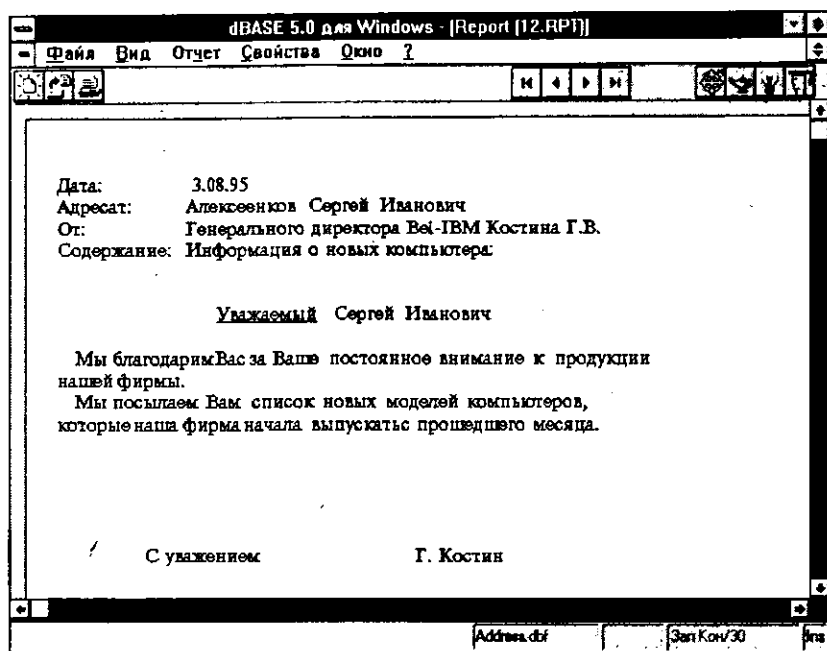


Рис. 9.42. Информационное письмо.

6. Поместите в области данных надпись, содержащую требуемый текст, используя dBASE в качестве текстового редактора. Далее выровняйте текст. Вид отчета показан на рис. 9.41.
7. Нажмите кнопку Вывести отчет в окне просмотра на панели инструментов и просмотрите внешний вид отчета. При просмотре отчета на экране каждое из писем

будет иметь вид, представленный на рис. 9.42. Каждое письмо будет содержать одинаковый текст и разные фамилии, имена и отчества для каждого из клиентов.

8. Если вы удовлетворены внешним видом отчета, напечатайте его, воспользовавшись командой **Файл | Печать | Принтер**.

После завершения работы над отчетом не забудьте сохранить его (команда **Файл | Сохранить как**).

Очевидно, мы рассмотрели решение простейших практических задач, которыми вы сможете воспользоваться в более сложных случаях. В заключение нам хотелось бы сделать несколько замечаний о создании отчетов.

Дополнительные замечания по созданию отчетов

Так что же отличает хороший отчет от посредственного? Хороший отчет помогает усвоить содержащуюся в нем информацию, не отвлекая внимания на второстепенные детали. Хороший отчет выглядит “читабельным”, а содержащаяся в нем информация легко усваивается и не требует понимания проблематики баз данных.

dBASE предоставляет в ваше распоряжение мощные средства для создания по настоящему профессиональных отчетов. Однако то, как вы ими распорядитесь, целиком зависит от вас. Например, если вы разрабатываете отчет, содержащий деловую информацию, и для каждой записи поместите графическое изображение, то вместо хорошего отчета вы получите альбом картинок. Возможно, вам могут понадобиться и такие отчеты, но только не при составлении деловой информации. Графика, как правило носит иллюстративный характер, и может отвлекать внимание на малозначительные детали.

Прежде чем составлять свои собственные отчеты, обратите внимание на те отчеты, с которыми работают сотрудники вашего учреждения, или для которого вы разрабатываете специализированное приложение. Выясните, какие отчеты для них более удобны. В некоторых случаях может сказываться привычка. Возможно, вы разработали отчет, который по информативности превосходит все то, с чем работали сотрудники до настоящего времени. Но ... внешний вид отчета непривычен и потребует значительного времени, чтобы полностью освоиться с ним. Разработав собственный отчет, не пренебрегайте мнением сотрудников, которым придется с ним работать.

По возможности, снабжайте отчет заголовком. В противном случае может оказаться затруднительным объяснить, какой именно отчет вы хотели бы получить. Помимо этого, желательно, чтобы заголовок отчета содержал текущую дату.

Учтите, что просмотр отчета — это лишь промежуточная стадия создания документа. Как правило, отчет выводится на принтер. Большинство доступных принтеров не поддерживают цветной печати. В этом случае вы можете использовать в отчете только градации серого, но не цветовую схему. Прибегайте к цветовым решениям только в том случае, если уверены, что отчет будет выводиться на цветном принтере. Однако и в этом случае вы должны выбирать цветовую схему крайне осторожно, чтобы не отвлекать внимание пользователя на великолепное сочетание подобранных вами цветов в ущерб основной информации отчета.

dBASE допускает размещение в отчете, полей как с заголовками, так и без них. Используйте заголовки только в том случае, когда в этом действительно есть необходимость. Например, заголовок “Фамилия” является избыточным, и его присутствие в каждой записи наряду с фамилией будет только загромождать отчет. Однако при отображении в отчете даты рождения сотрудника и даты приема на фирму, вы просто обязаны дать поясняющие подписи.

Использование специальных символов, раскрывающих назначение данных, также имеет особенности. Например, столбец “Стоимость партии”. Вы обязаны указать в каких единицах отображается стоимость — в тысячах (миллионах) рублей или долларовом эквиваленте. Наиболее простой способ — изменить формат отображения значений для указанного поля. Однако на наш взгляд такой формат загромождает отчет. Поэтому более целесообразно указать единицу измерения в заголовке столбца.

К использованию сокращенных названий, заголовков и т.п. следует прибегать только в том случае, если вы знаете, что используемые сокращения будут понятны работающему с отчетом человеку. Сокращение “ВНИПИЭИ Леспром” (Всесоюзный научно-исследовательский проектный институт экономики и информатики) может быть приемлемо для сотрудников и клиентов, имеющих дело с этой организацией, но приведет в замешательство большинство сторонних пользователей.

Если в различных отчетах одного приложения размещено одно и то же поле, используйте для него одинаковые заголовки. В противном случае у пользователя могут возникнуть сомнения, имеется в виду одно и то же поле или нет (“Код покупателя” и “Покупатель_идент.”).

Специальные шрифты, безусловно, привлекают внимание пользователя. В одних случаях это необходимо (например, в заголовке отчета или при отображении статистических данных), но использование целой палитры шрифтов в теле отчета безусловно рассеивает внимание и отвлекает от основного содержания отчета. Какие шрифты лучше использовать в отчетах? С нашей точки зрения следует отдавать предпочтение пропорциональным шрифтам типа Times. Равноширинные шрифты (например, Courier) могут быть использованы в заголовке отчета. Однако, если вы используете достаточно много пропорциональных шрифтов, отчет также будет трудночитаем.

Старайтесь выравнивать логические группы и отдельные поля. По возможности делайте блоки одинаковой длины и располагайте их в отчете симметрично.

Выделение информации имеет важное значение. Например, целесообразно выделить итоговые данные. В ряде случаев необходимость в просмотре сведений по всей группе покупателей может отсутствовать, вместо этого требуются статистические данные (например, стоимость товаров, проданных за определенный срок). В этом случае выделение итоговых значений будет оправдано. Для выделения используйте стили с подчеркиванием или сами разместите одинарную или двойную линию, чтобы выделить требуемые данные. Полезность этих средств мы продемонстрируем в главе 14.

Будьте консервативными в хорошем смысле этого слова. Найдя читабельный стиль, следуйте ему при разработке последующих отчетов. При этом помните о том, что почерк может быть хорошим и плохим. Критерий полезности и читабельности — это мнение тех людей, которые будут использовать ваши отчеты.

Все данные нами рекомендации не носят обязательный характер. Помните, что пользователю придется взаимодействовать с вашими отчетами длительное время.

Заключение

- Конструктор отчетов позволяет создавать отчеты в различных форматах (табличный, свободный или почтовые наклейки), которые могут быть выведены на принтер или на экран.
- Для того чтобы приступить к созданию нового отчета, выберите **Файл | Создать | Отчет** или нажмите кнопку **Создать новый отчет** на панели инструментов.
- Окно диалога “Выбор файла данных” предназначено для задания наименования таблицы или запроса, используемых при создании отчета, а также для установки связи между связанными таблицами.

- Окно конструктора Crystal Reports является основным средством для создания отчетов. В этом окне выполняется большинство операций по размещению, перемещению, удалению и модификации всех объектов.
- На рис. 9.12 показан внешний вид панели инструментов в окне конструктора отчетов, а назначения кнопок приведены в табл. 9.2.
- Для выбора требуемого объекта воспользуйтесь кнопкой **Выделить** поля на панели инструментов или командой меню **Правка | Выделить** поля.
- Все объекты dBASE характеризуются рядом параметров, называемых свойствами. Для того чтобы получить доступ к свойствам объекта, выделите нужный объект и нажмите правую кнопку мыши.
- Для размещения объектов интерфейса используйте соответствующие кнопки, расположенные на панели инструментов.
- Для объединения в группу нескольких объектов выделите требуемые объекты. После этого вы можете переместить или удалить группу как единый объект.
- Для отображения значений текстовых полей в нескольких строках выберите опцию **Перенос слов** из меню объекта.
- Для того чтобы сохранить отчет, выберите **Сохранить** или **Сохранить как** из меню **Файл**.
- Для того чтобы напечатать отчет, выберите **Файл | Печать | Принтер** или нажмите кнопку **Принтер** на панели инструментов, находясь в окне отчета.
- Для выбора принтера или изменения его настройки выберите **Файл | Выбор принтера**.

Глава 10. Управление таблицами dBASE

В первой части книги вы научились создавать таблицы и разрабатывать для них формы, отчеты и запросы, которые являются объектами dBASE. Многие наиболее часто используемые операции с таблицей можно выполнить, используя навигатор dBASE, включая изменение его свойств, изменение структуры таблицы и ее удаление. В настоящей главе мы рассмотрим команду **Операции**, которая присутствует как в меню **Таблица**, так и в меню **Форма**. Эта команда содержит дополнительные операции управления таблицами dBASE. Кроме того, вы познакомитесь со средствами импорта и экспорта данных.

Меню команды Таблица

При выборе команды **Таблица | Операции** на экране появится следующее меню:

И ндексы...
О бновить индексы
З акрыть таблицу З акрыть все таблицы
П ометить для удаления... С нять пометки... У далить помеченные У далить все записи
Д обавить из файла... Э кспорт записей... С ортировка записей... С оздать записи...
П одсчет записей... В ычисление значений...

Часть команд этого меню была рассмотрена в предыдущих главах (создание индексов, сортировка таблиц и т.д.). В данной главе мы остановимся на командах, которые ранее не были рассмотрены.

Команда	Назначение
Индексы	Задаёт индекс, который становится основным индексом. Кроме того, с помощью данной команды можно создавать, изменять и удалять индексы.
Обновить индексы	Перестраивает все открытые индексы.
Закрыть таблицу	Закрывает текущую таблицы.
Закрыть все таблицы	Закрывает все открытые таблицы и индексы.
Пометить для удаления	Помечает группу записей для удаления.

Команда	Назначение
Снять пометки	Снимает метки удаления с заданной группы записей.
Удалить помеченные	Физически удаляет все помеченные для удаления записи из таблицы.
Удалить все записи	Удаляет (а не помечает для удаления) все записи из таблицы.
Добавить из файла	Добавляет данные в текущую таблицу из других таблиц в формате dBASE и из файлов других форматов.
Экспорт записей	Выполняет экспорт данных в другие форматы.
Создать записи	Добавляет в текущую таблицу записи со случайными данными.
Сортировка записей	Сортирует записи по заданному критерию.
Подсчет записей	Подсчитывает количество записей таблицы, удовлетворяющих заданному условию.
Вычисление значений	Выполняет финансовые и статистические вычисления с числовыми полями.

Выбор основного индекса

Как вы уже знаете, каждая таблица может иметь более одного индекса. Для открытой таблицы в любой момент времени один из этих индексов является основным и при отображении таблицы в окне таблицы или формы порядок расположения записей таблицы определяется этим индексом. Выбирая основной индекс вы выбираете и порядок расположения записей в отображаемой таблице. Для выбора основного индекса используйте команду Таблица | Индексы, при выполнении которой открывается окно диалога “Индексы”, которое содержит список всех индексов таблицы (рис. 10.1).

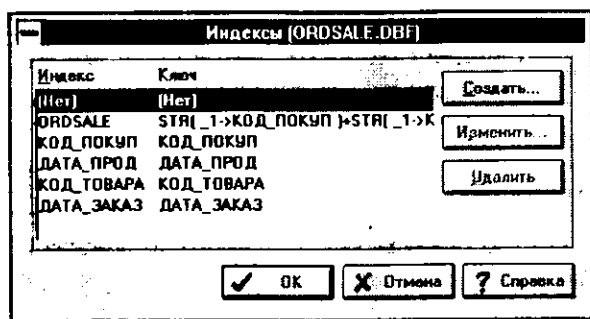


Рис. 10.1. Окно диалога “Индексы”.

Индекс, который в данный момент является основным, в окне диалога выделяется цветом или яркостью. По умолчанию при открытии таблицы для нее не устанавливается основной индекс и данные расположены в таблице в порядке ввода. Для выбора основного индекса достаточно выбрать строку с требуемым индексом и нажать кнопку ОК. В качестве примера, на рис. 10.2 приведен результат выбора в качестве основного индекса “ДАТА_ПРОД” таблицы ORDSALE.

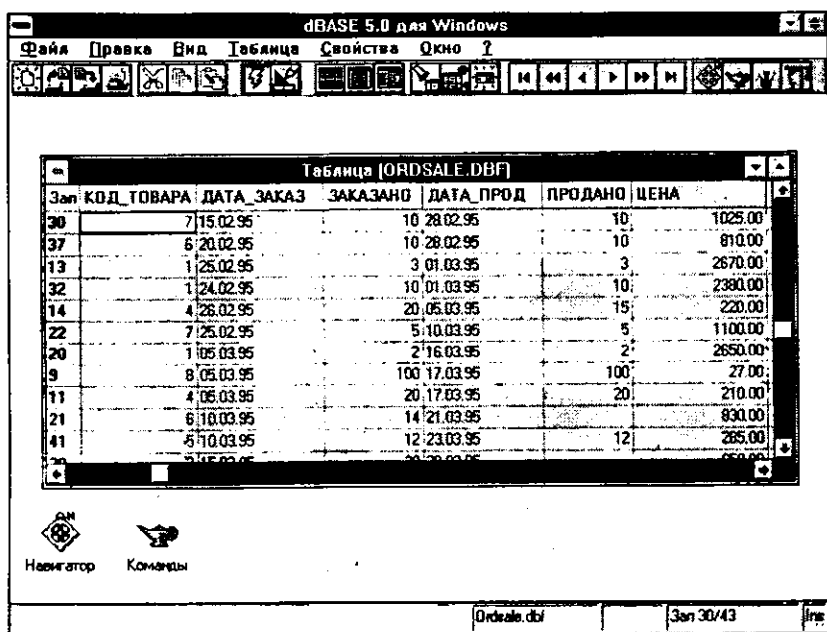


Рис. 10.2. Выбор индекса "ДАТА_ПРОД" в качестве основного.

Обновление индексов

Создаваемые в dBASE индексы таблицы, хранимые в одном файле с расширением .MDX, автоматически открывается при открытии таблицы. В процессе работы с таблицей они непрерывно обновляются. Если при работе с таблицей произойдет отключение электропитания или пользователь случайно выключит компьютер не выйдя из dBASE, может произойти повреждение индекса. В этом случае вы можете перестроить поврежденные индексы с помощью команды Таблица | Обновить индексы.

Отключение электропитания может привести к такому повреждению индексного файла, что его уже будет невозможно восстановить данной командой. Кроме этого отключение электропитания может привести к повреждению и файла таблицы. Поэтому настоятельно рекомендуется осуществлять регулярное резервное копирование файлов таблиц и индексов. В этом случае вы всегда можете восстановить ваши данные на момент копирования.

Действия над записями в таблицах

В этом разделе вы научитесь решить следующие проблемы:

- Добавлять в текущую таблицу записи из другой таблицы или файла
- Добавлять в текущую таблицу записи со случайными данными
- Определять количество записей таблицы, удовлетворяющих заданному условию
- Выполнять финансовые и статистические вычисления с числовыми полями

Добавление записей в текущую таблицу

Команда **Добавить из файла** позволяет добавить данные в текущую таблицу из других таблиц в формате dBASE и Paradox, а также из текстовых файлов в системном

формате данных (SDF) и с разделителями. Данные добавляются в конец текущей таблицы в том порядке, как они хранятся в исходном файле.

При добавлении данных из таблицы в формате dBASE, содержащей записи, помеченные для удаления, режим их добавления зависит от значения флажка **Скрыть удаленные записи** окна диалога "Свойства среды dBASE". Если этот флажок снят, то исходные записи, помеченные для удаления, будут добавлены в текущую таблицу и помечены в ней для удаления. Если же этот флажок установлен, то исходные записи, помеченные для удаления, в текущую таблицу не добавляются.

При добавлении данных в таблицу в формате Paradox с первичным индексом, добавляются только записи с уникальным ключом. Любые записи, дублирующие существующий ключ, записываются в отдельную таблицу в формате Paradox.

Для того чтобы добавить записи в текущую таблицу из внешнего файла, выполните приведенную ниже последовательность действий.

1. Выберите **Добавить из файла** из меню **Операции**, в результате dBASE откроет окно диалога "Добавление записей из файла" (рис. 10.3).

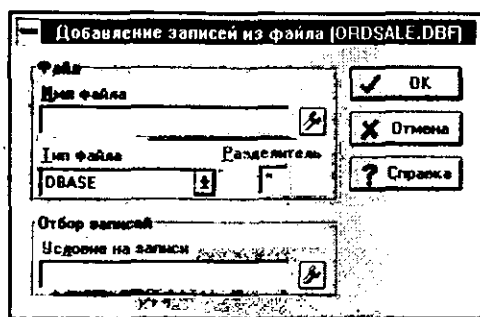


Рис. 10.3. Окно диалога "Добавление записей из файла".

2. Установите курсор в строку ввода **Имя файла** и введите имя файла, из которого будут добавляться данные. Если вы не помните точное название файла, нажмите кнопку **Инструмент** правее этого поля и выберите подходящий файл из окна диалога "Импорт файла" (рис. 10.4).

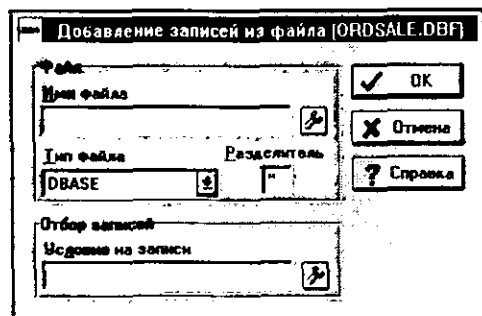


Рис. 10.4. Окно диалога "Импорт файла".

3. Выберите из раскрывающегося списка **Тип файла** один из допустимых типов.

Тип	Описание
DBASE	Таблица dBASE в формате .DBF.
PARADOX	Таблица Paradox в формате .DB.
SDF	Текстовый файл в системном формате данных. Записи в файле SDF имеют фиксированную длину, конец записи отмечается знаками возврата каретки и перехода на новую строку. Когда данные в формате SDF вставляются в таблицу в формате dBASE, оставшееся свободное пространство в полях заполняется пробелами.
DELIMITED	Текстовый файл с разделителями. Поля в этом файле разделяются запятыми. Для выделения конца записи используются знаки возврата каретки и перехода на новую строку. Для выделения значений символьных полей по умолчанию используются двойные кавычки (""). При выборе этого типа становится доступным поле ввода Разделитель , в котором вы можете задать символ, являющийся разделителем полей в текстовом файле. Если разделитель не указан, то им считается пробел.

- При добавлении записей вы можете задать условие отбора записей исходной таблицы, которые будут добавляться в текущую таблицу. Для этого перейдите в поле **Условие на записи** и введите условие отбора записей. Более предпочтительным является использование построителя выражений, для вызова которого нажмите кнопку **Инструмент** правее поля ввода.
- Нажмите кнопку **ОК** и данные из внешнего файла будут добавлены в текущую таблицу.

Пример 10.1. Добавление записей в архивную таблицу

Создадим таблицу O&S95, которая будет являться архивной таблицей продаж за 1995 год. Данные в эту таблицу дополняются раз в месяц из таблицы ORDSALE. Для добавления данных будем использовать команду **Операции | Добавить из файла**.

- Выполните команду **Файл | Открыть** и укажите наименование O&S95 в качестве таблицы, которую следует открыть.
- Выберите **Добавить из файла** из меню **Операции** и в окне диалога "Добавление записей из файла" укажите имя файла ORDSALE и тип файла DBASE.
- Введите условие выбора записей. Для этого нажмите кнопку **Инструмент** правее поля ввода **Условие на записи** и в окне построителя выражения введите условие отбора записей за апрель 1995 года (рис. 10.5).
- Вернитесь в окно диалога "Добавление записей из файла" и нажмите кнопку **ОК**. Убедитесь, что данные за апрель месяц добавлены в таблицу O&S95.

Добавление записей со случайными данными

В dBASE существует возможность заполнения таблицы записями, которые содержать значения, формируемые случайным образом. Данное средство может исполь-

зоваться при тестировании приложения и облегчает процесс заполнения таблицы данными.

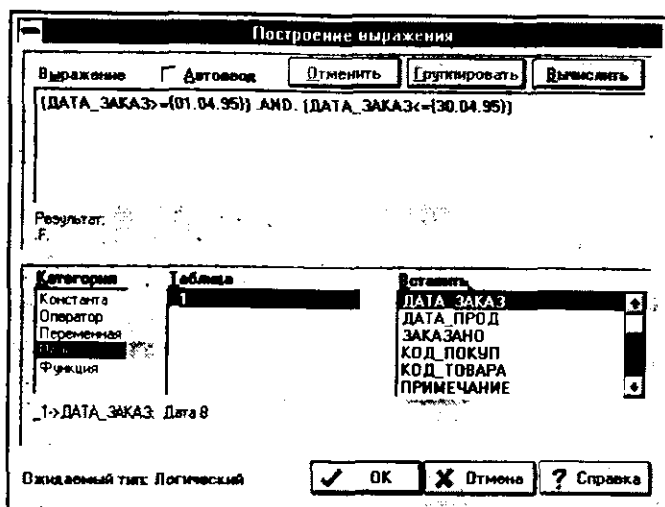


Рис. 10.5. Определение условия отбора записей.

Добавление записей со случайными данными осуществляется командой **Таблица | Операции | Создать запись**, при выполнении которой открывается окно диалога "Создание записей" (рис. 10.6) в котором вы задаете единственный параметр — количество добавляемых записей. При выполнении этой команды будут заполнены все поля записей, за исключением Мемо-полей, OLE и двоичных полей.



Количество создаваемых записей не должно превосходить 1 миллиард. Кроме того, объем создаваемых записей должен быть менее двух миллиардов байт.

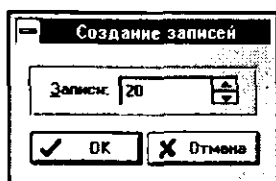


Рис. 10.6. Окно диалога "Создание записей".

Определение количества записей таблицы, удовлетворяющих заданному условию

Команда **Таблица | Операции | Подсчет записей** позволяет вам достаточно просто подсчитать количество записей таблицы, удовлетворяющих заданному условию. С помощью этой команды вы можете узнать количество продаж за заданный интервал времени, количество покупок указанного покупателя и т.п. При выполнении этой команды на экране появляется окно диалога "Подсчет записей" (рис. 10.7), в котором задается условие отбора записей.

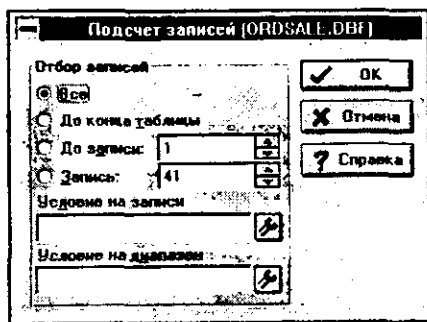


Рис. 10.7. Окно диалога "Подсчет записей".

После ввода условия отбора нажмите кнопку ОК и количество подсчитанных записей появится в окне диалога "Подсчитанные записи" (рис. 10.8). Если в окне диалога "Свойства среды dBASE" установлен флажок Скрыть удаленные записи, данная команда не включает записи, помеченные для удаления.

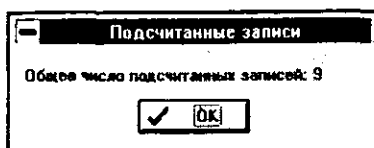


Рис. 10.8. Окно диалога "Подсчитанные записи".

Финансовые и статистические вычисления

Более мощной по сравнению с предыдущей является команда Таблица | Операции | Вычисление значений, которая выполняет финансовые и статистические вычисления с числовыми полями фиксированного формата и с числовыми полями с плавающей точкой в текущей таблице в формате dBASE или с числовыми, денежными или целочисленными полями текущей таблицы в формате Paradox. При выполнении этой команды открывается окно диалога "Вычисление записей" (рис. 10.9).

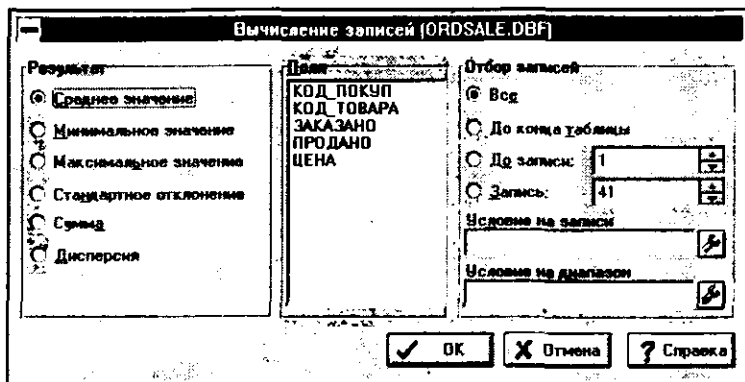


Рис. 10.9. Окно диалога "Вычисление записей".



Если в текущей таблице нет числовых полей фиксированного формата или числовых полей с плавающей точкой, на экране появится предупреждение.

Тип выполняемого вычисления задается группой переключателей области “Результат”:

Тип	Назначение
Среднее значение	Вычисляет среднее значение указанного числового выражения фиксированного формата или числового выражения с плавающей точкой и возвращает значение в формате с плавающей точкой.
Минимальное значение	Вычисляет минимальное значение указанного поля и возвращает результат того же типа, что и указанное поле.
Максимальное значение	Вычисляет максимальное значение указанного поля и возвращает результат того же типа, что и указанное поле.
Стандартное отклонение	Вычисляет стандартное отклонение указанного числового поля фиксированного формата или числового поля с плавающей точкой и возвращает значение в формате с плавающей точкой.
Сумма	Вычисляет суммарное значение указанного поля и возвращает значение в формате с плавающей точкой.
Дисперсия	Вычисляет дисперсию указанного числового поля фиксированного формата и возвращает значение в формате с плавающей точкой.

Вы можете осуществить вычисления по одному или нескольким полям таблицы. Список Поля содержит список полей, по которым возможны вычисления. Выделите из этого списка поля, по которым вы собираетесь провести вычисления. Для выбора нескольких подряд следующих полей нажмите клавишу *Shift* и, удерживая ее, нажмите левую кнопку мыши. Для выбора отдельно расположенных полей нажимайте клавишу *Ctrl* и левую кнопку мыши. Дополнительно вы можете определить условие отбора записей. После выбора типа вычислений, вычисляемых полей и условия отбора нажмите кнопку ОК и на экране появится окно диалога “Результаты вычислений”, которое содержит вычисленные значения выбранных полей (рис. 10.10).

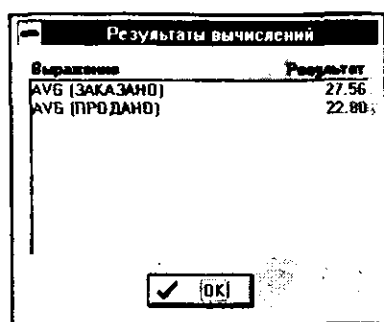


Рис. 10.10. Окно диалога “Результаты вычислений”.

Экспорт данных

Команда Таблица | Операции | Экспорт записей предназначена для экспорта данных из всей таблицы dBASE или ее части в другие файлы в формате dBASE, Paradox или в текстовые файлы. Экспорт записей в форматы dBASE и Paradox позволяет вам копировать таблицу или ее часть в другую таблицу. В этом случае экспорт выполняет

роль копирования данных. В ряде случаев вам может потребоваться осуществить обмен данными с программами, которые не поддерживают форматы dBASE и Paradox. Например, при передаче данных в базу данных СУБД Oracle или Informix. Однако большинство СУБД имеют средства для экспорта и импорта файлов в текстовом формате. В такой ситуации для обмена вы можете воспользоваться текстовыми файлами.

Если для экспортируемой таблицы установлен основной индекс, то записи новой таблицы или файла размещаются в порядке, соответствующем этому индексу.

Если для экспортируемой таблицы в формате dBASE установлен флажок **Скрыть удаленные записи**, то исходные записи, помеченные для удаления, не экспортируются в результирующую таблицу.

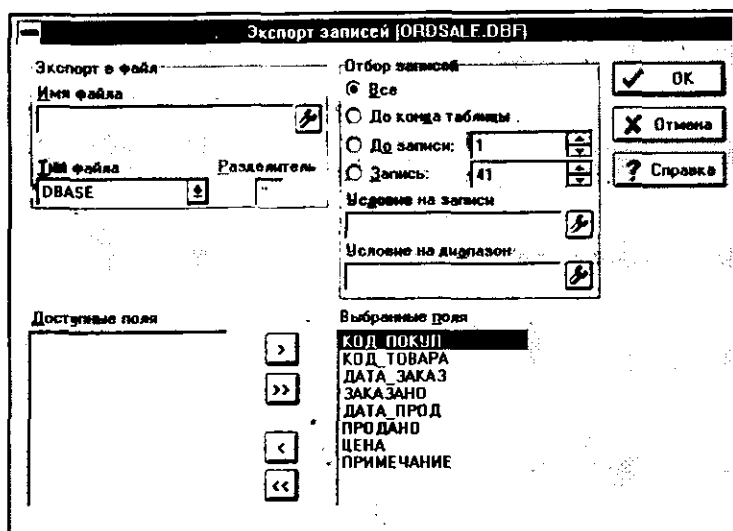


Рис. 10.11. Окно диалога “Экспорт записей”.

Для того чтобы экспортировать записи из текущей таблицы во внешний файл выполните следующую последовательность действий.

1. Выберите **Экспорт записей** из меню **Операции**, в результате dBASE откроет окно диалога “Экспорт записей” (рис. 10.11). В этом окне диалога укажите имя и тип файла, в который будут экспортироваться данные, список экспортируемых полей и условие отбора экспортируемых записей.
2. Установите курсор в строку ввода **Имя файла** и введите имя файла в который будут экспортироваться данные. Вы можете воспользоваться кнопкой **Инструмент**, расположенной правее этого поля для определения каталога и имени результирующего файла.
3. Выберите из раскрывающегося списка **Тип файла** один из возможных типов:

Тип	Описание
DBASE	Таблица dBASE в формате .DBF.
PARADOX	Таблица Paradox в формате .DB.

Тип	Описание
SDF	Текстовый файл в системном формате данных. Записи в файле SDF имеют фиксированную длину, конец записи отмечается знаками возврата каретки и перехода на новую строку. Когда данные в формате SDF вставляются в таблицу в формате dBASE, оставшееся свободное пространство в полях заполняется пробелами.
DELIMITED	Текстовый файл с разделителями. Поля в этом файле разделяются запятыми. Для выделения конца записи используются знаки возврата каретки и перехода на новую строку. Для выделения значений символьных полей по умолчанию используются двойные кавычки (""). При выборе этого типа становится доступным поле ввода Разделитель, в котором вы можете задать символ, являющийся разделителем полей в текстовом файле. Если разделитель не указан, то им считается пробел.
DBMemo3	Формат dBASE III PLUS для базы данных и Мемо-полей.

4. Определите список экспортируемых полей таблицы, которые приведены в списке **Выбранные поля**.

При открытии окна диалога этот список содержит перечень всех полей таблицы, доступных для экспорта. Поэтому ваша задача состоит в выборе полей, которые не предполагается экспортировать, и перемещении их в список **Доступные поля**. Выберите поле из списка **Выбранные поля** и нажмите кнопку **Удалить (<)**. В результате этих действий выбранное поле перемещается в список **Доступные поля**. Для перемещения всех полей из списка **Выбранные поля** в список **Доступные поля** вы можете использовать кнопку **Удалить все (<<)**. Для добавления полей в список **Выбранные поля** используйте кнопки **Добавить (>)** и **Добавить все (>>)**.

1	219950402	519950412	5	610.00
2	119950102	219950112	2	2600.00
2	219950103	1519950104	15	625.00
2	219950201	1119950211	7	630.00
2	219950205	1019950220	10	645.00
2	219950213	50019950228	450	605.00
2	319950101	1019950112	8	1025.00
2	519950412	1019950422	0	270.00
2	819950305	10019950317	100	27.00
6	219950105	2019950123	20	620.00
7	419950305	2019950317	20	210.00
7	819950105	5019950123	50	27.00
10	119950225	319950301	3	2670.00
10	419950226	2019950305	15	220.00
10	519950210	2019950223	15	270.00
11	319950210	119950219	1	1100.00
11	419950110	2019950121	20	220.00
12	119950305	219950316	2	2650.00
12	619950310	1419950321		830.00
12	719950225	519950310	5	1100.00

Рис. 10.12. Экспорт записей из таблицы ORDSALE в текстовый файл.

5. При экспортировании записей вы можете задать условие отбора записей исходной таблицы в области "Отбор записей". При определении условия обычно используют построитель выражений, вызов которого осуществляется с помощью кнопки **Инструмент** правее поля ввода.
6. Нажмите кнопку **ОК** и данные из исходного файла будут экспортированы в результирующий файл.

На рис. 10.12 приведен пример экспортирования таблицы **ORDSALE** в текстовый файл с разделителями.



dBASE не позволяет экспортировать в текстовый файл значения из двоичных и OLE-полей.

Импорт данных из электронных таблиц

dBASE позволяет импортировать данных из файлов электронных таблиц Quattro Pro для Windows и Lotus 1-2-3.



Некоторые электронные таблицы (например, Excel) могут воспринимать файлы с расширением .DBF без предварительного преобразования данных и сохранять созданные электронные таблицы в формате .DBF.

Перед импортированием данных из электронной таблицы вы должны предварительно ее подготовить в программе управления электронной таблицей. Прежде всего решите, собираетесь ли вы использовать заголовки столбцов электронной таблицы в качестве имен полей таблицы базы данных. В случае утвердительного ответа разместите их в первой строке электронной таблицы. В противном случае, удалите их из электронной таблицы, так как иначе они будут считаться обычными данными. Кроме этого удалите из электронной таблицы начальные пустые строки и столбцы и убедитесь, что все столбцы содержат данные одного типа, одинаково отформатированные.

Для импортирования данных из перечисленных выше электронных таблиц вам необходимо выполнить следующую последовательность действий:

1. Выберите **Файл | Импорт данных**, в результате dBASE откроет окно диалога "Импорт" (рис. 10.13).

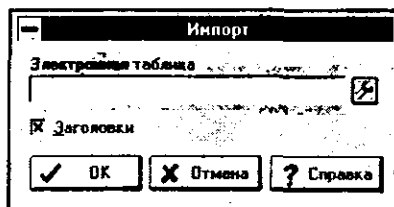


Рис. 10.13. Окно диалога "Импорт".

2. Установите флажок **Заголовки**, если вы собираетесь использовать заголовки столбцов электронной таблицы в качестве имен полей таблицы базы данных.

- Укажите импортируемый файл, для чего введите имя этого файла в строку ввода или выберите его с помощью кнопки **Инструмент**. При нажатии кнопки **Инструмент** открывается окно диалога "Выбор электронной таблицы" (рис. 10.14) из которого вы можете выбрать требуемую электронную таблицу. В качестве примера выберем электронную таблицу SALE_Y.WK1 в формате Lotus 1-2-3, которая содержит сведения о продажах товаров за 1994 год.

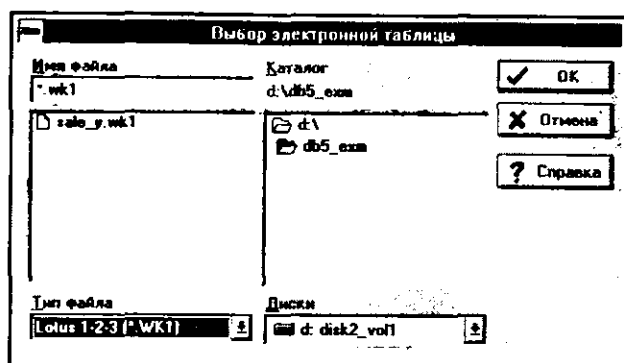


Рис. 10.14. Выбор электронной таблицы.

- Выберите **OK**. В результате будут создана новая таблица с тем же именем, что и электронная таблица. Если файл с таким именем уже существует, система dBASE запрашивает подтверждение на замену файла. На рис. 10.15 вы видите таблицу SALE_Y, которая получена из электронной таблицы SALE_Y.WK1.

dBASE 5.0 для Windows

Файл Правка Вид Таблица Свойства Окно ?

Навигатор Команды

Таблица [SALE_Y.DBF]

Зам.	№	CP_35_SX	CP_40_SX	CP_55_LS	CP_55_SX	ИТОГО
1	1	10300.00	30000.00	100000.00	200000.00	78075.00
2	2	33330.00	45000.00	120000.00	250000.00	106875.00
3	3	30000.00	36000.00	200000.00	120000.00	96500.00
4	4	70000.00	50000.00	100000.00	100000.00	80000.00
5	5	30000.00	43000.00	180000.00	80000.00	83250.00
6	6	40000.00	100000.00	190000.00	200000.00	132500.00
7	7	60000.00	70000.00	160000.00	210000.00	125000.00
8	8	20000.00	45000.00	154000.00	190000.00	102250.00
9	9	35000.00	23000.00	210000.00	320000.00	147000.00
10	10	20000.00	67000.00	90000.00	200000.00	94250.00
11	11	10000.00	78000.00	80000.00	230000.00	99500.00
12	12	30000.00	57000.00	180000.00	123000.00	97500.00

SALE_Y.dbf Monopol Jan 1/12

Рис. 10.15. Результат импорта из электронной таблицы.

Формат файлов электронных таблиц не имеет типов полей. При импортировании данных dBASE автоматически устанавливает наиболее ограничивающий тип поля,

которому удовлетворяют все значения. правила преобразования столбцов электронных таблиц в поля dBASE приведены в табл. 10.1.

Табл. 10.1. Преобразование данных при импорте из электронных таблиц

Значение в электронной таблице	Тип поля dBASE
Текстовые строки	Текстовый
Числа (целые)	Числовой
Числа (с дробной частью)	Числовой
Числа (в денежном формате)	Числовой
Числа (в формате даты)	Дата

Если столбец содержит хотя бы одну текстовую величину, он преобразуется в текстовое поле. В электронных таблицах для разделения группы чисел часто используют текстовые строки (например, строки дефисов). В этом случае все столбцы электронной таблицы, содержащие хотя бы один дефис, будут преобразованы в текстовые поля. В текстовое поле будет также преобразован столбец, содержащий числовые данные и даты.



Если вас не устраивает структура таблицы, полученной в результате импорта, то ее можно изменить средствами dBASE.

Выбор языкового драйвера

Языковой драйвер таблицы определяет порядок сортировки таблицы и допустимый набор символов. Языковые драйверы, используемые по умолчанию для таблиц dBASE и Paradox, устанавливаются с помощью программы настройки параметров.

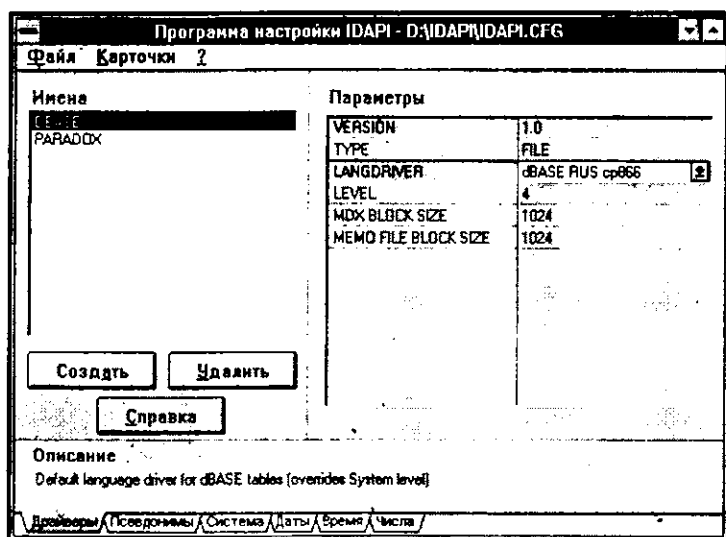


Рис. 10.16. Выбор языкового драйвера.



Если вам потребуется изменить используемый по умолчанию языковой драйвер, запустите программу настройки параметров IDAPI, которая запускается выбором пиктограммы "Настройка IDAPI" из группы dBASE для Windows. После запуска программы на экране появляется окно диалога "Программа настройки IDAPI", карточка "Драйверы" которого содержит параметр LANGDRIVER, который и определяет используемый языковой драйвер (рис. 10.16).

В заключение выберите требуемый драйвер из раскрывающегося списка, после чего сохраните файл конфигурации IDAPI.



Изменение языкового драйвера может привести к потере специальных символов, характерных для прежнего драйвера.

Заключение

- Для выбора основного индекса используйте команду **Таблица | Индексы**.
- Для обновления поврежденных индексов используйте команду **Таблица | Обновить индексы**.
- Для добавления данных в текущую таблицу из других таблиц в формате dBASE и Paradox, а также из текстовых файлов в системном формате данных (SDF) и с разделителями выполните команду **Добавить из файла**.
- Для добавления в текущую таблицу записей со случайными данными выполните команду **Таблица | Операции | Создать записи**.
- Для подсчета количества записей таблицы, удовлетворяющих заданному условию, выполните команду **Таблица | Операции | Подсчет записей**.
- Для выполнения финансовых и статистических вычислений используйте команду **Таблица | Операции | Вычисление значений**.
- Для экспорта данных из всей таблицы dBASE или ее части в другие файлы в формате dBASE, Paradox или в текстовые файлы выполните команду **Таблица | Операции | Экспорт записей**.
- Для импортирования данных из файлов электронных таблиц Quattro Pro для Windows и Lotus 1-2-3 используйте команду **Файл | Импорт данных**.
- Для изменения языкового драйвера запустите программу настройки параметров IDAPI и выберите требуемый драйвер из раскрывающегося списка.

Ч а с т ь 2

МНОГОТАБЛИЧНЫЕ БАЗЫ ДАННЫХ

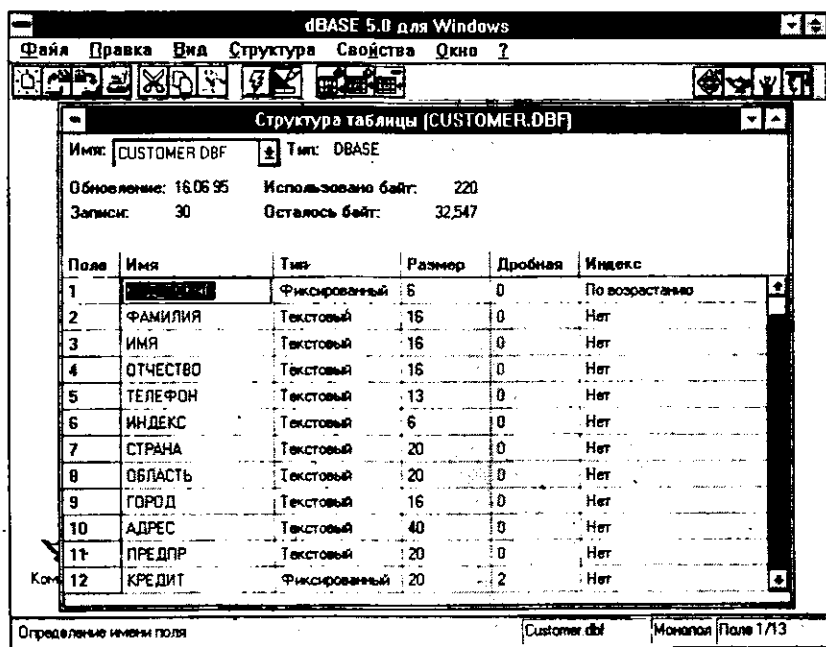
Глава 11. Введение в многотабличные базы данных

В первой части книги вы научились создавать в dBASE базу данных, состоящую из одной таблицы, и работать с ней. Возможно, что при создании простейших приложений вы не выйдете за границы однотабличных баз данных. Однако большинство практических задач требует наличия в базе данных нескольких связанных между собой таблиц. Для управления данными, содержащимися в связанных таблицах, используются те же средства, что и при работе с однотабличными базами данных. Кроме того, для связывания таблиц предназначены специальные средства, изучению которых будет посвящена вторая часть книги.

В настоящей главе мы обсудим, как спроектировать многотабличную базу данных и эффективно распределить данные между таблицами.

Реляционные базы данных

Реляционные базы данных состоят из нескольких таблиц, связь между которыми устанавливается с помощью совпадающих полей. Каждая из таблиц содержит информацию о каких-либо объектах одного типа (группы). В качестве примера обратимся к базе данных, предназначенной для учета заказов покупателей, которая состоит из двух таблиц: CUSTOMER и ORDSALE. Таблица CUSTOMER (Покупатель) содержит информацию о покупателях (фамилия, имя, отчество, адрес, телефон и т.п.). На рис. 11.1 и 11.2 показаны структура этой таблицы и ее содержимое.



Структура таблицы [CUSTOMER.DBF]

Имя: CUSTOMER.DBF Тип: DBASE

Обновление: 16.06.95 Использовано байт: 220

Записи: 30 Осталось байт: 32,547

Поле	Имя	Тип	Размер	Дробная	Индекс
1		Фиксированный	6	0	По возрастанию
2	ФАМИЛИЯ	Текстовый	16	0	Нет
3	ИМЯ	Текстовый	16	0	Нет
4	ОТЧЕСТВО	Текстовый	16	0	Нет
5	ТЕЛЕФОН	Текстовый	13	0	Нет
6	ИНДЕКС	Текстовый	6	0	Нет
7	СТРАНА	Текстовый	20	0	Нет
8	ОБЛАСТЬ	Текстовый	20	0	Нет
9	ГОРОД	Текстовый	16	0	Нет
10	АДРЕС	Текстовый	40	0	Нет
11	ПРЕДПР	Текстовый	20	0	Нет
12	КРЕДИТ	Фиксированный	20	2	Нет

Определение имени поля

Customer.dbf Monahan Page 1/13

Рис. 11.1. Структура таблицы CUSTOMER.

Таблица ORDSALE содержит информацию о заказах и покупках различных покупателей. Структура этой таблицы и сама таблица показана на рис. 11.3 и 11.4 соответственно.

dBASE 5.0 для Windows

Файл Правка Вид Таблица Свойства Окно ?

Таблица (CUSTOMER.DBF)

Зап	КОД_ПОКУП	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ТЕЛЕФОН	ИНДЕКС	Ст
1		1 Алексеев	Сергей	Иванович	(363)224-3124	630080	Ро
2		2 Ткаченко	Марина	Михайловна	(048)056-4634	270005	Ук
3		3 Морозов	Юрий	Анатолеевич	(048)034-5874	270029	Ук
4		4 Петров	Михаил	Николаевич	(816)025-6128	173024	Ро
5		5 Ильин	Филипп	Петрович	(816)025-6156	173003	Ро
6		6 Петров	Михаил	Николаевич	(816)017-7124	173003	Ро
7		7 Зайцева	Людмила	Николаевна	(863)521-2453	346415	Ро
8		8 Загребский	Андрей	Анатолеевич	(831)256-4216	603006	Ро
9		9 Слесарь	Филипп	Петрович	(831)245-3123	630056	Ро
10		10 Сухарев	Андрей	Анатолеевич	(863)573-4543	357736	Ро
11		11 Зайцева	Дарья	Ивановна	(257)212-2453	142480	Ро
12		12 Фролов	Виктор	Михайлович	(095)576-5456	140180	Ро
13		13 Семенов	Сергей	Иванович	(095)453-5985	142432	Ро
14		14 Уланов	Максим	Сергеевич	(812)623-4456	184230	Ро
15		15 Морозов	Андрей	Михайлович	(343)232-4542	620151	Ро
16		16 Райский	Максим	Сергеевич	(846)253-4542	443115	Ро
17		17 Антонов	Максим	Юрьевич	(095)532-1532	125187	Ро
18		18 Морозов	Андрей	Петрович	(095)131-7714	117334	Ро
19		19 Александров	Александр	Иванович	(013)225-6541	226010	Па

Customer.dbf Jan 1/30 line

Рис. 11.2. Таблица CUSTOMER.

dBASE 5.0 для Windows

Файл Правка Вид Структура Свойства Окно ?

Структура таблицы (ORDSALE.DBF)

Имя: ORDSALE.DBF Тип: DBASE

Обновление: 26.07.95 Использовано байт: 315

Записи: 43 Осталось байт: 32,452

Поло	Имя	Тип	Размер	Дробная	Индекс
1	КОД_ПОКУП	Фиксированный	6	0	По возрастанию
2	КОД_ТОВАРА	Фиксированный	6	0	Нет
3	ДАТА_ЗАКАЗ	Дата	8	0	Нет
4	ЗАКАЗАНО	Фиксированный	6	0	Нет
5	ДАТА_ПРОД	Дата	8	0	Нет
6	ПРОДАНО	Фиксированный	6	0	Нет
7	ЦЕНА	Фиксированный	20	2	Нет
8	ПРИМЕЧАНИЕ	Текстовый	254	0	Нет

Ordsale.dbf Monopol Page 1/8

Рис. 11.3. Структура таблицы ORDSALE.

Вы можете получить информацию о каждом клиенте (таблица CUSTOMER) и сделанных им покупках (таблица ORDSALE). Каждая запись в таблицах идентифицирует один объект группы (покупатель или сделанный заказ). Отношение между объектами определяет отношение между таблицами. Предполагается, что один и тот же

покупатель может сделать несколько заказов и покупок. Таким образом, между покупателями и сделанными ими заказами существует отношение один-ко-многим. Связь таблиц осуществляется на основании данных в совпадающих полях "Код_покуп".

Рис. 11.4. Таблица ORDSALE.

dBASE поддерживает четыре типа отношений между таблицами: один-к-одному, один-ко-многим, много-к-одному, много-ко-многим.

Прежде чем перейти к вопросам проектирования реляционных баз данных, остановимся подробнее на каждом из типов отношений.

Отношение один-к-одному

Отношение один-к-одному означает, что каждая запись в одной таблице соответствует только одной записи в другой таблице.

В качестве примера рассмотрим отношение между группами PERSON и EMPLOYEE, представленными в таблицах PERSON (Личность) и EMPLOYEE (Служащий). Структура этих таблиц представлена на рис. 11.5 и 11.6 соответственно.

Обе таблицы содержат информацию о сотрудниках компании Bel-IBM. В таблице PERSON содержатся данные о личности сотрудника, а в таблице EMPLOYEE — профессиональные сведения. Между таблицами PERSON и EMPLOYEE существует отношение один-к-одному, поскольку для одного человека может существовать только одна запись, содержащая профессиональные сведения.

Связь между этими таблицами поддерживается при помощи совпадающих полей: "Идент. N" (таблица PERSON) и "Таб. N" (таблица EMPLOYEE). Обратите, что эти поля имеют разные наименования. Связь между таблицами устанавливается на основании значений совпадающих полей, но не их наименований.

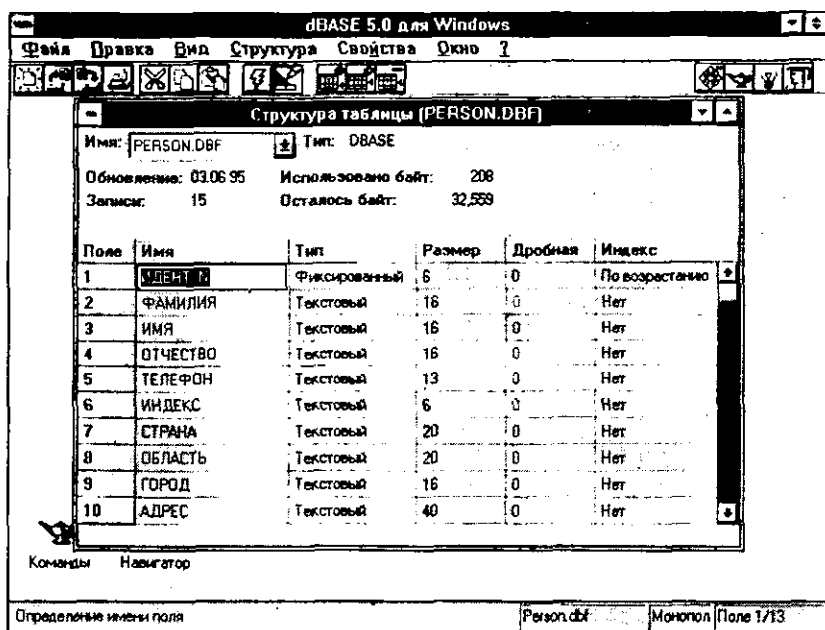


Рис. 11.5. Структура таблицы PERSON.

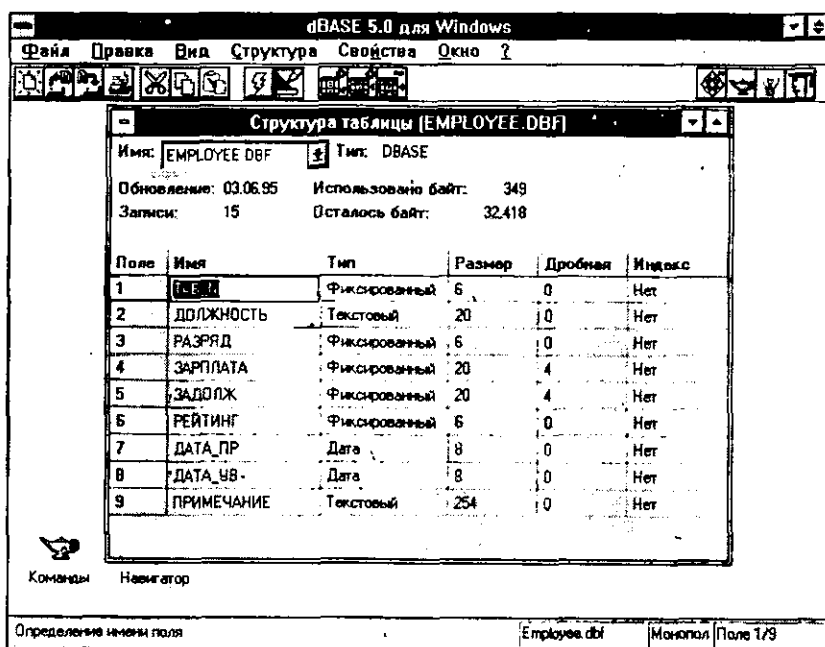


Рис. 11.6. Структура таблицы EMPLOYEE.

Отношение один-ко-многим

В качестве иллюстрации данного типа отношения обратимся к рассмотренной ранее базе данных CUSTOMER — ORDSALE. Связь между таблицами осуществляется на основании значений совпадающих полей "Код_покуп". В этом случае совпа-

дающие поля в обеих таблицах имеют одинаковые наименования. Поле "Код_покуп" является единственным индексным полем в таблице CUSTOMER. Таблица ORDSALE имеет составной индекс, одним из полей которого является поле "Код_покуп".

В качестве других примеров могут быть рассмотрены отношения между предприятием и работающими на нем сотрудниками. Аналогичный тип отношения существует между компьютером и входящими в него компонентами. Как правило, при иерархической организации данных тип отношения один-ко-многим является наиболее общим.

Отношение много-к-одному

Отношение много-к-одному аналогично рассмотренному ранее типу один-ко-многим. Тип отношения между объектами зависит от вашей точки зрения. Например, если вы будете рассматривать отношение между сделанными заказами и клиентами, то получите отношение много-к-одному.

Отношение много-ко-многим

Отношение много-ко-многим возникает между двумя таблицами в тех случаях, когда:

- Одна запись из первой таблицы может быть связана более чем с одной записью из второй таблицы.
- Одна запись из второй таблицы может быть связана более чем с одной записью из первой таблицы.

Поле	Имя	Тип	Размер	Дробная	Индекс
1	КОД_ПОСТАВ	Фиксированный	6	0	Нет
2	КОД_ТОВАРА	Фиксированный	6	0	По возрастанию
3	ЦЕНА	Фиксированный	20	4	Нет
4	МИН_ПАРТ	Фиксированный	6	0	Нет
5	СРОК_ПОСТ.	Фиксированный	6	0	Нет

Рис. 11.7. Структура таблицы SUPPGOOD.

В качестве примера обратимся к магазину оптовой торговли. Рассмотрим две группы объектов.

1. Список цен на товары, производимые предприятиями-поставщиками (таблица SUPPGOOD).
2. Список товаров, заказанных потребителями (таблица ORDSALE).

Структура таблицы SUPPGOOD представлена на рис. 11.7.

Между таблицами SUPPGOOD и ORDSALE существует отношение много-ко-многим, так как на каждый поставляемый товар может быть более одного заказа. Аналогично каждый заказанный товар может производиться более чем одним предприятием. Связь между таблицами устанавливается на основании значений в совпадающих полях "Код_товара".

Проектирование нормализованных баз данных

При проектировании реляционной базы данных вам необходимо решить вопрос о наиболее эффективной структуре данных. Основные цели, которые при этом преследуются:

- Обеспечить быстрый доступ к данным в таблицах.
- Исключить ненужное повторение данных, которое может являться причиной ошибок при вводе и нерационального использования дискового пространства вашего компьютера.
- Обеспечить целостность данных таким образом, чтобы при изменении одних объектов автоматически происходило соответствующее изменение связанных с ними объектов.

Процесс уменьшения избыточности информации в базе данных называется *нормализацией*. В теории нормализации баз данных разработаны достаточно формализованные подходы по разбиению данных, обладающих сложной структурой, на несколько таблиц. Эти вопросы детально освещаются в специальной литературе. Мы обсудим некоторые практические аспекты нормализации таблиц, не вдаваясь в их теоретическое обоснование.

Теория нормализации оперирует с пятью нормальными формами таблиц (от первой до пятой включительно). Эти формы предназначены для уменьшения избыточной информации от первой до пятой нормальной формы. Поэтому каждая последующая нормальная форма должна удовлетворять требованиям предыдущей формы и некоторым дополнительным условиям. При практическом проектировании баз данных четвертая и пятая формы, как правило, не используются, поэтому мы ограничимся рассмотрением первых трех нормальных форм.

Воспользуемся результатами теории при проектировании многотабличной базы данных с эффективной структурой. В качестве примера рассмотрим таблицу SALES, которая содержит следующую информацию:

- Сведения о покупателях.
- Дату заказа и количество заказанного товара.
- Дату выполнения заказа и количество проданного товара.
- Характеристику проданного товара (наименование, стоимость, категория).

Структура таблицы SALES приведена в таблице 11.1, а часть таблицы — на рис. 11.8.

Табл. 11.1. Структура таблицы SALES

N	Наименование	Тип
1	Код_покуп	Numeric(10)
2	Фамилия	Character(16)
3	Имя	Character(16)
4	Отчество	Character(16)
5	Телефон	Character(13)
6	Индекс	Character(10)
7	Страна	Character(20)
8	Область	Character(20)
9	Город	Character(20)
10	Адрес	Character(60)
11	Предпр	Character(20)
12	Руководит	Character(30)
13	Кредит	Numeric(12.2)
14	Примечание	M(10)
15	Код_товара	Numeric(10)
16	Дата_заказ	D
17	Заказано	Numeric(10)
18	Дата_прод	D
19	Продано	Numeric(10)
20	Цена	Numeric(12.2)
21	Прим_заказ	Character(254)
33	Категория	Numeric(10)
23	Наим_тов	Character(60)

В данной таблице приведены обозначения типов полей, соответствующие оригинальной версии dBASE для Windows:

Character	Текстовый
Numeric	Числовой
D	Даты
M	Мемо

Таблицу SALES можно рассматривать как однотабличную базу данных. Основная проблема состоит в том, что в ней содержится значительное количество повторяющейся информации. Например, сведения о каждом покупателе повторяются для каждого сделанного им заказа. Такая структура данных является причиной следующих проблем, возникающих при работе с базой данных:

- Вам придется тратить значительное время на ввод повторяющихся данных. Например, для всех заказов, сделанных одним из покупателей, вам придется каждый раз вводить одни и те же данные о покупателе.
- При изменении адреса или телефона покупателя, необходимо корректировать все записи, содержащие сведения о заказах этого покупателя.
- Наличие повторяющейся информации приведет к неоправданному увеличению размера базы данных. В результате снизится скорость выполнения запросов.

Кроме того, повторяющиеся данные нерационально используют дисковое пространство вашего компьютера.

- Любые внештатные ситуации потребуют от вас значительного времени для получения требуемой информации. Например, при многократном вводе повторяющихся данных возрастает вероятность ошибки. При больших размерах таблиц поиск ошибок будет занимать значительное время.

The screenshot shows the dBASE 5.0 for Windows interface. It displays four tables from the SALES.DBF database:

Зеп	КОД_ПОКУП	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ТЕЛЕФОН	ИНДЕКС	СТРАНА
14	10	Сухарев	Андрей	Анатолеевич	(863)573-4543	357736	Россия
15	10	Сухарев	Андрей	Анатолеевич	(863)573-4543	357736	Россия

Зеп	ОБЛАСТЬ	ГОРОД	АДРЕС	ПРЕДПР	РУКОВОДИТ
14		Кисловодск	ул. Севастопольская, д. 60	Стиггер	Сухарев
15		Кисловодск	ул. Севастопольская, д. 60	Стиггер	Сухарев

Зеп	КРЕДИТ	ПРИМЕЧАНИЕ	КОД_ТОВАРА	ДАТА_ЗАКАЗ	ЗАКАЗНО	ДАТА_ПРОД	ПРОДАНО
14	400000.00			1 25.02.95		3 01.03.95	3
15	400000.00			4 26.02.95		20 05.03.95	15

Зеп	КАТЕГОРИЯ	НАИМ_ТОВ	ЦЕНА	ПРИМ_ЗАКАЗ
14	2	ASTI 386	2670	
15	3	Turbo Pascal 6.0	220	

At the bottom of the window, it shows 'Sales.dbf' and 'Зеп 14/42'.

Рис. 11.8. Таблица SALES.

Воспользуемся практическими рекомендациями теории нормализации для разработки на основании таблицы SALES многотабличной базы данных с эффективной структурой.

Первая нормальная форма таблицы

Приведенная на рис 11.8 таблица является ненормализованной. Таблица в первой нормальной форме должна удовлетворять следующим требованиям:

1. Таблица не должна иметь повторяющихся записей.
2. В таблице должны отсутствовать повторяющиеся группы полей.
3. Строки должны быть не упорядочены.
4. Столбцы должны быть не упорядочены.

Для удовлетворения условия 1 каждая таблицы должна иметь уникальный индекс. Таблица SALES не содержит уникального индекса, что допускает наличие в таблице повторяющихся записей. Для выполнения условия 1 создадим уникальный индекс, содержащий поле "Код_покуп".

В таблицах dBASE, содержащих индексы, записи упорядочены в соответствии со значениями индексного выражения. Поэтому в dBASE требование 3 не может быть удовлетворено.

Требование 2 постулирует устранение повторяющихся групп. Поскольку каждый покупатель может заказать несколько товаров, нам необходимы две таблицы. Каждая запись одной таблицы будет содержать сведения об одном из покупателей, а второй таблицы — информацию о каждом из заказов. Поэтому мы разобьем таблицу SALES на две отдельные таблицы (CUSTOMER и ORDSALE) и определим “Код_покуп” в качестве совпадающих полей для связывания таблиц. Между таблицами CUSTOMER и ORDSALE будет отношение один-ко-многим.

Структуры таблиц CUSTOMER и ORDSALE приведены на рис. 11.9. Таблица CUSTOMER содержит данные о покупателях. Для данной таблицы определим уникальный индекс с полем “Код_покуп”. Таким образом, для таблицы CUSTOMER решена проблема повторяющихся групп.

Customer			OrdSale		
N	Наименование	Тип	N	Наименование	Тип
1	Код_покуп	Numeric (10)	1	Код_покуп	Numeric (10)
2	Фамилия	Character (16)	2	Код_товара	Numeric (10)
3	Имя	Character (16)	3	Дата_заказ	Date (8)
4	Отчество	Character (16)	4	Заказано	Numeric (10)
5	Телефон	Character (13)	5	Дата_прод	Date (8)
6	Индекс	Character (10)	6	Продано	Numeric (10)
7	Страна	Character (20)	7	Цена	Numeric (12.2)
8	Город	Character (20)	8	Прим_заказ	Character (254)
9	Область	Character (20)	9	Категория	Numeric (10)
10	Адрес	Character (60)	10	Наим_тов	Character (60)
11	Предпр	Character (20)			
12	Руководит	Character (30)			
13	Кредит	Numeric (12.2)			
14	Примечание	Мемо (10)			

Рис. 11.9. Первая нормальная форма.

Таблица ORDSALE содержит сведения о заказанных товарах. Для исключения повторяющихся записей вы можете воспользоваться одним из следующих способов:

1. Добавьте в таблицу новое уникальное индексное поле “Номер_заказ”, что позволит вам однозначно идентифицировать каждый из заказов. Однако при работе с dBASE это далеко не лучший метод. В последующих главах вы узнаете, что при разработке многотабличных форм и отчетов связь между таблицами осуществляется посредством индексов. Ввод нового индексного поля в таблицу ORDSALE не позволит вам связать эти таблицы по значениям поля “Код_покуп” при создании многотабличных форм и отчетов.
2. На рис. 11.9 вы видите реализацию второго способа исключения повторяющихся записей. В этом случае использован уникальный составной индекс, состоящий из полей “Код_покуп”, “Код_товара” и “Дата_заказ”.

После того как вы разделили повторяющиеся объекты и определили поля, которые образуют уникальный индекс в каждой таблице, считается, что таблица находится в первой нормальной форме. Поэтому с учетом ограничений (условия 3 и 4) мы можем считать, что таблицы CUSTOMER и ORDSALE находятся в первой нормальной форме.

Вторая нормальная форма таблицы

О таблице говорят, что она находится во второй нормальной форме, если:

1. Она удовлетворяет условиям первой нормальной формы.
2. Любое поле, не входящее в индекс, однозначно идентифицируется выражением индекса.

Из приведенного выше определения следует, что понятие второй нормальной формы применимо только к таблицам, имеющим составной индекс. В рассматриваемом примере такой таблицей является ORDSALE, в которой составной индекс образуют поля "Код_покуп", "Код_товара" и "Дата_прод". Данная таблица не является таблицей во второй нормальной форме, поскольку поля "Наименование товара", "Цена" и "Категория" однозначно определяются только одним из индексных полей ("Код_товара").

Для приведения таблицы ко второй нормальной форме выделим из таблицы ORDSALE таблицу GOODS, которая будет содержать информацию о товарах каждого типа. Результат представлен на рис. 11.10. Для связывания таблиц ORDSALE и GOODS используется поле "Код_товара".

Customer		OrdSale		Goods	
N	Наименование	Тип	N	Наименование	Тип
1	Код_покуп	Numeric (10)	1	Код_покуп	Numeric (10)
2	Фамилия	Character (16)	2	Код_товара	Numeric (10)
3	Имя	Character (16)	3	Дата_заказ	Date (8)
4	Отчество	Character (16)	4	Заказано	Numeric (10)
5	Телефон	Character (13)	5	Дата_прод	Date (8)
6	Индекс	Character (18)	6	Продано	Numeric (10)
7	Страна	Character (28)	7	Цена	Numeric (12.2)
8	Город	Character (28)	8	Прим_заказ	Character (254)
9	Область	Character (28)			
10	Адрес	Character (68)	N	Наименование	Тип
11	Предпр	Character (28)	1	Код_товара	Numeric (10)
12	Руководит	Character (38)	2	Категория	Numeric (10)
13	Кредит	Numeric (12.2)	3	Наим_тов	Character (68)
14	Примечание	Мемо (18)			

Рис. 11.10. Вторая нормальная форма.

Третья нормальная форма таблицы

О таблице говорят, что она находится в третьей нормальной форме, если:

1. Она удовлетворяет условиям второй нормальной формы.
2. Ни одно из полей таблицы, не входящих в индекс, не идентифицируется с помощью другого поля, не входящего в индекс.

Сведение таблицы к третьей нормальной форме предполагает разделение таблицы с целью помещения в отдельную таблицу (или несколько таблиц) столбцов, которые не зависят от выражения индекса. В результате такого разбиения каждое из полей, не входящих в индекс, должно оказаться независимым от какого-либо другого неиндексного поля.

Обратимся к таблице CUSTOMER. Поле "Руководит" этой таблицы содержит имена руководителей компаний, которые однозначно определяются значением поля "Предпр". Поскольку поле ("Руководит"), не входящее в индекс, однозначно опреде-

ляется другим неиндексным полем ("Предпр"), таблица CUSTOMER не является таблицей в третьей нормальной форме. Для приведения этой таблицы к третьей нормальной форме создадим новую таблицу MANAGER (рис. 11.11).

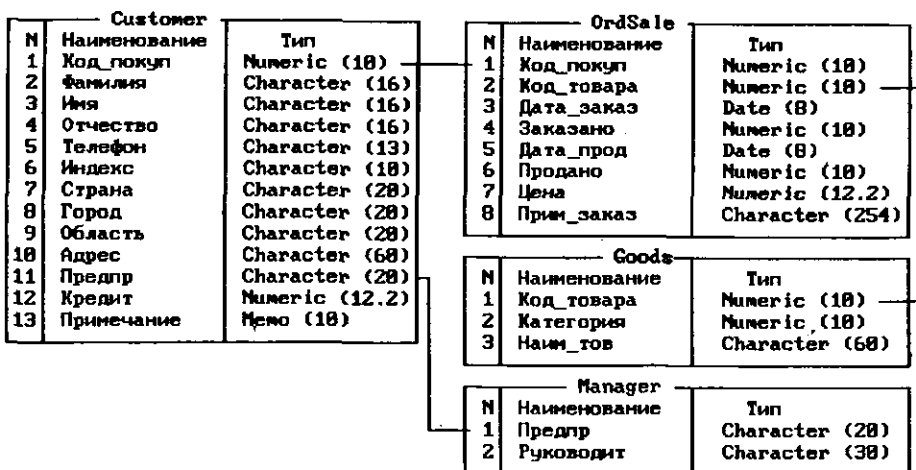


Рис. 11.11. Третья нормальная форма.

Требование, чтобы все таблицы были нормализованы, не является обязательным в dBASE. Однако в рассмотренном примере при проектировании многотабличной базы данных вы могли отметить определенные преимущества использования нормализованных таблиц. На практике вы не сможете в полном объеме выполнить все требования, предъявляемые к нормализованным таблицам. Это обусловлено ограничениями самой системы dBASE, но не доставит вам каких-либо серьезных проблем при практической работе.

После определения структуры таблиц, отношений между ними и совпадающих полей, которые будут использованы для связывания отдельных таблиц, вы готовы к созданию многотабличной базы данных в dBASE. Каждая из таблиц создается независимо от других, и на этой стадии связи между таблицами отсутствуют. Вам потребуется установить связи между таблицами только при создании объектов приложений (запросов, многотабличных форм и отчетов и т.д.). Последующие главы второй части посвящены специальным средствам создания и использования многотабличных баз данных в dBASE.

Глава 12. Запросы к многотабличным базам данных

В главе 11 вы познакомились с основными принципами построения многотабличных баз данных. В настоящей главе мы обсудим выполнение запросов по образцу для связанных таблиц. Далее мы познакомим вас с языком SQL, который является мощным средством выборки данных.

Связывание таблиц

В dBASE вы можете связать таблицы, бланки запросов которых открыты в окне запроса. При связывании двух таблиц одна из них должна быть главной, а другая — подчиненной. В главной таблице вы указываете поле или задаете выражение, которое используется для связи с подчиненной таблицей. В подчиненной таблице для связи используется индекс. Выражение, задаваемое в главной таблице, должно соответствовать выбранному индексу по типу значения выражения и по его смыслу.

Такой подход означает, что вы можете связывать две таблицы только по имеющимся для подчиненной таблицы индексам. Очевидно, что в случае необходимости вы можете создать для связывания таблиц новый индекс.

Установка связи

Для связывания двух таблиц, бланки запроса которых открыты в окне запроса, выберите команду **Запрос | Установить связь** или нажмите кнопку **Связать таблицы** на панели инструментов. На экране откроется окно диалога “Определение связи” (рис. 12.1).

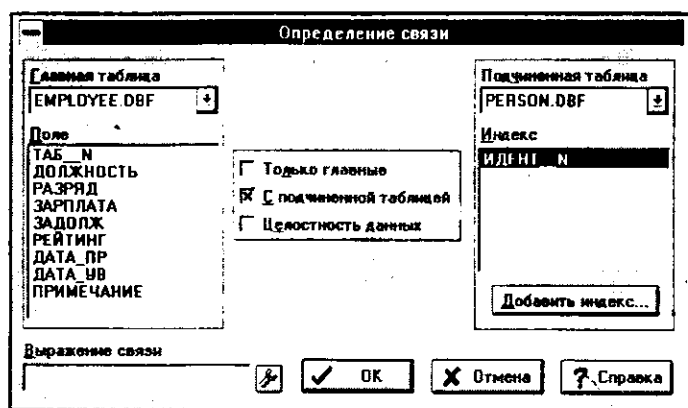


Рис. 12.1. Окно диалога “Определение связи”.

По умолчанию в качестве главной таблицы устанавливается текущая таблица запроса, т.е. таблица, на которую в данный момент установлен фокус. Для изменения главной таблицы нажмите стрелку около имени таблицы и выберите требуемое имя из раскрывающегося списка. Этот список содержит имена всех таблиц запроса.



Текущая таблица не может быть главной по отношению к самой себе, т.е. таблицу нельзя связать с самой собой.

Подчиненная таблица выбирается аналогично главной таблице.

Если связь между таблицами осуществляется по одному полю главной таблицы, вы можете выбрать это поле из списка полей главной таблицы. Выбранное вами поле появится в поле "Выражение связи" окна диалога. Если же связь между таблицами осуществляется по нескольким полям, вам необходимо ввести выражение связи в поле "Выражение связи". Формирование этого выражения осуществляется с помощью окна диалога "Выражение связи" (рис. 12.2). Для открытия данного окна нажмите кнопку, расположенную правее поля "Выражение связи".

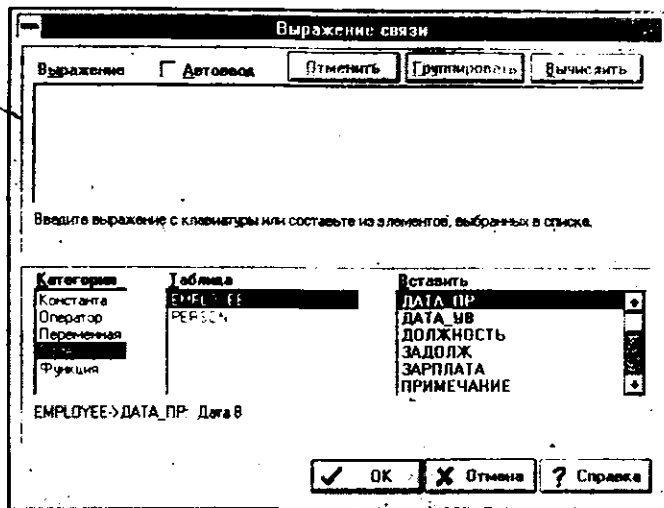


Рис. 12.2. Окно диалога "Выражение связи".

Так как связь между таблицами осуществляется по выражению из главной таблицы и выбранному индексу подчиненной таблицы, вам необходимо определить для подчиненной таблицы только индекс, используемый для связи. Выберите существующий индекс из списка индексов, приведенных в разделе "Индекс". Если же требуемый для установления связи индекс еще не создан, вы можете создать его из данного окна диалога. Для этого нажмите кнопку **Добавить индекс** и на экране откроется окно диалога "Создание индекса", в котором в подчиненную таблицу добавляется новый индекс или изменяется существующий.

Существует и другой способ установления связи между таблицами. Вы можете нажать значок  в бланке запроса главной таблицы и перенести его с помощью мыши к такому же значку в бланке запроса подчиненной таблицы. При нажатии на этот значок в главной таблице около указателя мыши появляется такой же значок и возникает линия связи, которая соединяет главную таблицу с указателем мыши. После перемещения к значку подчиненной таблицы и освобождения кнопки мыши указатель dBASE восстанавливает свой вид и открывается окно диалога "Определение связи".

После установления связи между таблицами расположение бланков запроса этих таблиц в окне запроса таблицы, в случае необходимости, изменяется, чтобы разместить их друг под другом.

Отображение полей в результирующей таблице

Флажки **Только главные** и **С подчиненной таблицей** позволяют вам установить режимы отображения полей в результирующей таблице.

Если флажок установлен **Только главные**, то каждая запись главной таблицы будет выбрана в результирующую таблицу независимо от того, соответствует ли она каким-либо записям в подчиненной таблице. Если этот флажок снят, в результирующей таблице будут только те записи главной таблицы, для которых имеются соответствующие записи в подчиненной таблице.

При установленном флажке **С подчиненной таблицей** все записи подчиненной таблицы, которые соответствуют записи главной таблицы, появляются в представлении. Если этот флажок снят, появляется только первая соответствующая запись из подчиненной таблицы.

Проверка целостности данных

В главе 7 вы познакомились с редактированием данных в исходных таблицах, используя для этого результирующую таблицу. Аналогичная возможность сохраняется в многотабличных запросах. При этом в ваше распоряжение добавляется новое средство по отслеживанию целостности вводимых данных.

Для этой цели используется флажок **Целостность данных**. Данный флажок определяет, что произойдет при выполнении действия, влияющего на связь между таблицами.



Флажок **Целостность данных** действует только для таблиц в формате dBASE.

Когда флажок **Целостность данных** установлен, при попытке удаления записи главной таблицы, для которой в подчиненной таблице присутствует одна или несколько записей, появляется предупреждение о необходимости отменить удаление или удалить и главную и все подчиненные записи. Таким образом исключается наличие записей в подчиненной таблице, которые не имеют ссылки на главную таблицу.

Кроме этого при установленном флажке при добавлении новой записи в подчиненную таблицу автоматически передается ключевое значение текущей записи главной таблицы, которое доступно только для чтения. Таким образом автоматически обеспечивается связь между главной и подчиненной таблицей.

Воспользуемся таблицами, разработанными в главе 11, для демонстрации создания запросов для таблиц с различными типами отношений между ними.

Запросы к связанным таблицам с отношением один-к-одному

В качестве таблиц с отношением один-к-одному воспользуемся таблицами PERSON и EMPLOYEE.

Пример 12.1.

Давайте на основании этих таблиц получим список всех программистов, работающих на предприятии Bel-IBM.

1. Откройте окно запроса для таблицы PERSON. Поскольку поля "Фамилия", "Имя" и "Отчество" должны присутствовать в результирующей выборке, поместите в них метки.

2. Все бланки запроса должны располагаться в одном и том же окне. Поэтому добавьте в окно запроса бланк запроса для таблицы EMPLOYEE и задайте условие для выбора из таблицы списка всех программистов.



Для того чтобы в окно запроса добавить бланки запроса таблицы EMPLOYEE, выберите **Запрос | Добавить таблицу** или нажмите кнопку **Добавить таблицу** на панели инструментов.

dBASE откроет окно диалога “Выбор файла”, в котором вам необходимо указать одну или несколько таблиц, которые вы предполагаете добавить к запросу. При этом вы можете включать в запрос таблицы, находящиеся в различных каталогах. После нажатия **ОК** требуемая таблица будет добавлена к запросу.



Удаление ненужных бланков запроса из окна запроса осуществляется аналогичным образом. Для этой цели используется команда **Запрос | Удалить таблицу** или кнопка **Удалить таблицу** на панели инструментов.



Вы можете перемещаться между бланками запроса, используя мышь или клавиши **F3** и **F4**. Для перемещения по полям внутри бланка запроса, как обычно, используется мышь или клавиши **Tab** и **Shift+Tab**.

3. В рассматриваемом примере установите связь между полем “Таб_Н” главной таблицы EMPLOYEE и индексом “Идент_Н” подчиненной таблицы PERSON и для выполнения запроса нажмите кнопку **Выполнить запрос** или клавишу **F2**.

Зан	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО
1	Бобочникова	Светлана	Дмитриевна
2	Волков	Александр	Степанович
3	Григорьева	Светлана	Михайловна
4	Карловская	Людмила	Петровна
5	Подушкин	Александр	Владимирович
6	Сосипетров	Сергей	Владимирович

Рис. 12.3. Запрос к связанным таблицам с отношением один-к-одному.

На рис. 12.3 представлены заполненные бланки запроса и результирующую таблицу. Если вам потребуется дополнительная информация из таблицы EMPLOYEE (например, дата поступления на работу и оклад), поместите метки в требуемые поля

бланка запроса для таблицы EMPLOYEE и выполните запрос. Заполненные бланки запроса и результирующая таблица, содержащая дополнительную информацию, показаны на рис. 12.4.

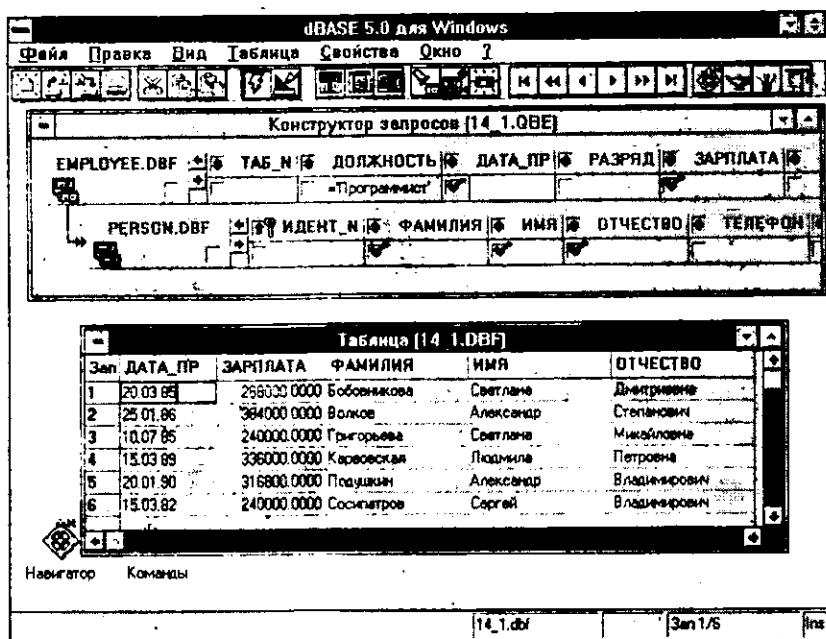


Рис. 12.4. Получение дополнительной информации.

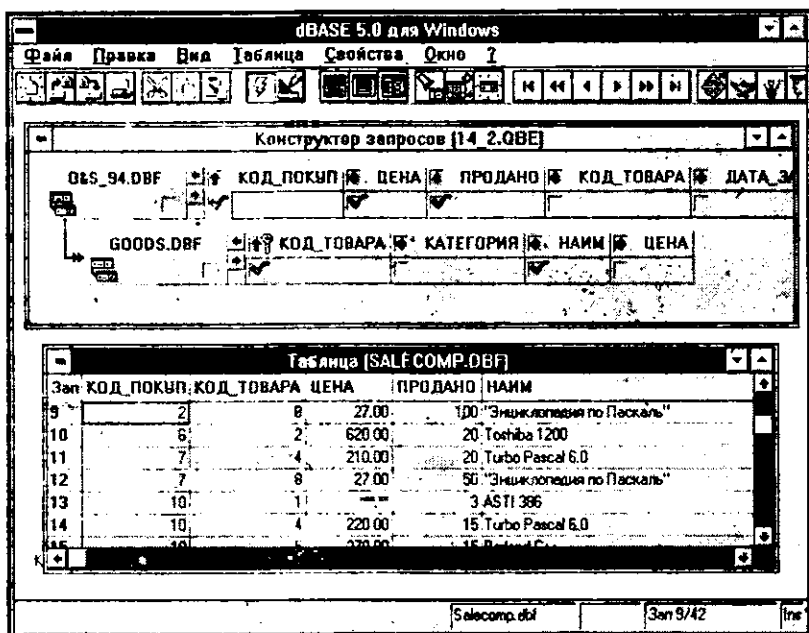


Рис. 12.5. Получение таблицы SALECOMP.

Таким образом, процедуры создания запросов для связанных и одиночных таблиц аналогичны. Отличие состоит в том, что кроме заполнения бланка запроса для каждой из таблиц вы должны связать эти таблицы.

На рис. 12.5 приведен еще один пример запроса для связанных таблиц с отношением один-к-одному. Данный запрос позволяет вам получить таблицу SALECOMP, которая содержит данные о продажах 1994 г. O&S_94 является архивной таблицей, структура которой совпадает со структурой таблицы ORDSALE.

Запросы к связанным таблицам с отношением ОДИН-КО-МНОГИМ

Для демонстрации построения запросов для таблиц с отношением один-ко-многим воспользуемся таблицами CUSTOMER, ORDSALE и GOODS.

Пример 12.2.

Предположим, что вам требуются сведения о всех невыполненных заказах предприятия. Признаком невыполненного заказа является пустое значение поля "Продано" в таблице ORDSALE.

1. Откройте окно запроса и поместите в него бланки запроса для таблиц CUSTOMER, ORDSALE и GOODS.
2. Поместите метки в поля "Фамилия", "Имя" и "Отчество" бланка запроса для таблицы CUSTOMER.
3. Поместите метки в поля "Дата_заказ" и "Заказано" бланка запроса для таблицы ORDSALE.
4. В качестве условия выбора введите $\neq 0$ в поле "Продано".
5. Поместите метки в поля "Категория" и "Наим" бланка запроса для таблицы GOODS.
6. Теперь вы должны связать все три таблицы.



Для установки связи воспользуемся кнопкой **Связать таблицы** на панели инструментов.

Нажмите кнопку **Связать таблицы**, после чего на экране появится окно диалога "Определение связи". Выберите поле "Код_покуп" главной таблицы CUSTOMER и индекс CD_CUST подчиненной таблицы ORDSALE.

Аналогичным образом установите связь между главной таблицей ORDSALE и подчиненной таблицей GOODS.

Заполненные бланки запроса и результирующая таблица показаны на рис. 12.6. В результате выполнения запроса из таблицы CUSTOMER произведена выборка всех покупателей, сведения о невыполненных заказах которых содержатся в таблице ORDSALE. Если для какого-либо из клиентов не выполнены несколько заказов, запись о каждом из них появится в результирующей таблице.

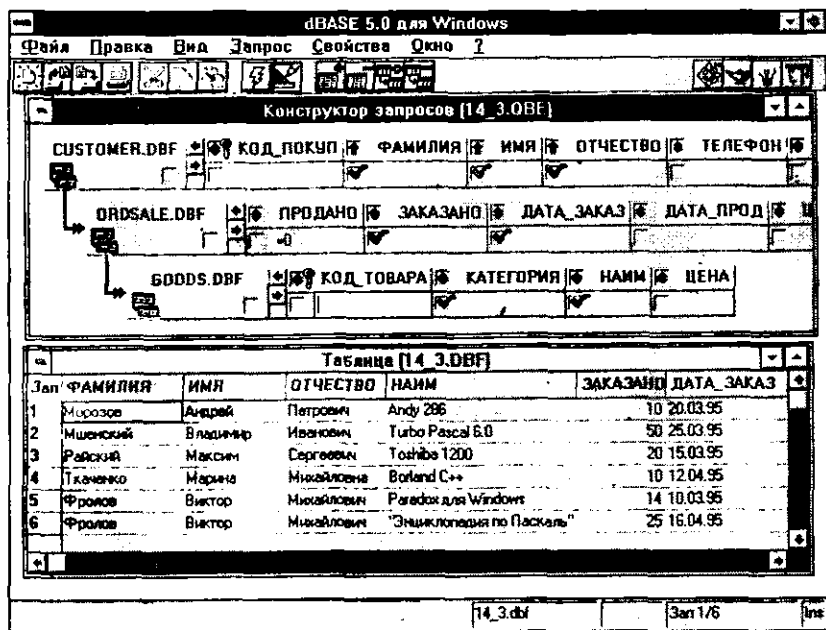


Рис. 12.6. Запрос к связанным таблицам с отношением один-ко-многим.

В настоящем примере было использовано только одно условие выбора записей. В следующем разделе мы рассмотрим запрос с несколькими условиями выбора, введенными в разные таблицы.

Использование логического И/ИЛИ в многотабличных запросах

Логическое И/ИЛИ в многотабличных запросах работает точно так же, как в однотабличных. Условия выбора, образующие выражение для логического И, должны располагаться в одной строке. При задании выражения для логического ИЛИ вы можете расположить условия в разных строках или воспользоваться оператором OR в области условия.

Пример 12.3.

В качестве примера получим информацию о товарах, проданных в феврале 1995 г. клиентам, проживающим в России. Эти данные содержатся в таблицах CUSTOMER, ORDSALE и GOODS.

1. Откройте окно запроса и поместите в него бланки запроса для указанных таблиц.
2. Пометьте поля, которые вы хотите отобразить.
3. Для задания условий выбора, образующих в данном случае логическое И, введите Россия в поле "Страна" бланка запроса для таблицы CUSTOMER и $\geq \{01.02.95\}$, $\leq \{28.02.95\}$ — в поле "Дата_прод" бланка запроса для таблицы ORDSALE.

4. Свяжите таблицы.

Результирующая таблица вместе с бланками запроса представлены на рис. 12.7.

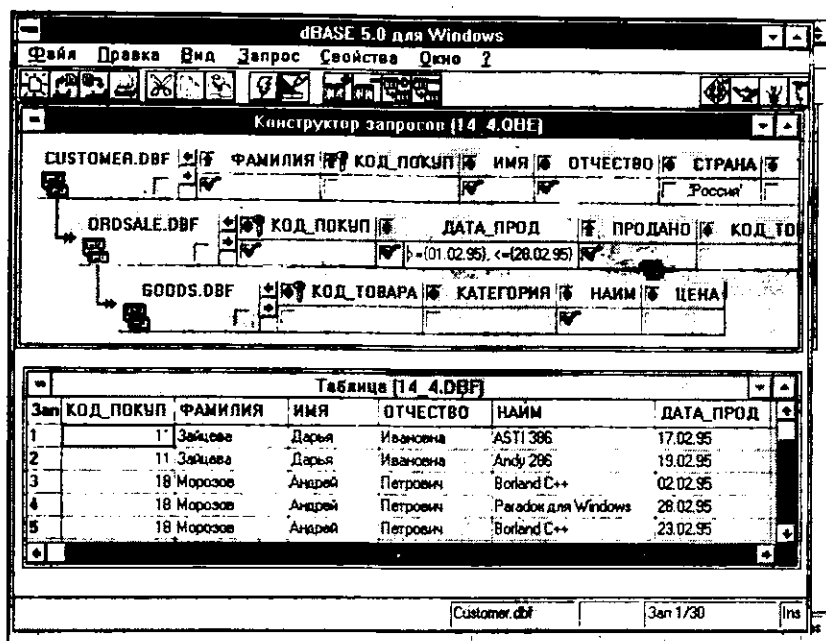


Рис. 12.7. Запрос с условиями выбора.

SQL — язык управления данными

Кроме выборки данных из таблиц с использованием запросов по образцу в dBASE вы дополнительно можете использовать язык высокого уровня SQL, который является стандартным средством доступа к данным в реляционных СУБД. Язык SQL может использоваться как с локальными базами данных, так и с базами данных, размещенных на серверах баз данных. SQL может использоваться для манипуляции с данными (выборка и модификация), обработки структуры базы данных (создание и удаление таблиц и индексов) и администрирования базы данных (определение списка пользователей, назначение прав доступа).

Использование SQL имеет множество преимуществ, наиболее важными из которых являются:

- SQL поддерживается многими поставщиками СУБД. Программы, написанные с использованием языка SQL, будут работать практически на всех реляционных СУБД.
- SQL прост в изучении, поскольку в нем для описания действия используются простые английские слова, такие, как SELECT, INSERT, UPDATE, DELETE, COMMIT и ROLLBACK.

В данной главе мы рассмотрим только основные конструкции SQL, используемые для выборки данных и их редактирования.

Формирование запросов в SQL

Для формирования запросов в SQL используется конструкция SELECT. Результат выполнения запроса хранится во временном буфере или в указанной вами таблице. Вы можете использовать выбранные данные для просмотра, печати отчетов, формирования графиков или даже добавить их в постоянные таблицы базы данных. Знакомство с выборками начнем с простого примера: выбор кодов, фамилий и имен покупателей, проживающих в Москве. В терминах SQL этот запрос может быть выражен следующим образом:

```
SELECT КОД ПОКУП, ИМЯ, ФАМИЛИЯ  
FROM CUSTOMER  
WHERE ГОРОД="Москва"
```

Для выполнения конструкций SQL в dBASE имеется специальная функция SQLEXEC, которой задаются SQL-конструкция в виде символьной строки и имя результирующей таблицы. Для вызова этой функции удобно использовать простейшую программу, которая состоит из вызова функции SQLEXEC.

Для создания программы выполните команду Создать | Программа и сохраните ее с именем SQL_TEST.PRG. Для выполнения вышеприведенного запроса в окне редактора напечатайте вызов функции SQLEXEC, которой задайте в качестве параметров конструкцию SQL и имя результирующей таблицы ANSWER.DBF. Результат запроса будет сохранен в таблице ANSWER.DBF. Для удобства тестирования запросов добавим в программу команды открытия этой таблицы (USE) и ее просмотра (BROWSE). Текст программы приведен на рис. 12.8.

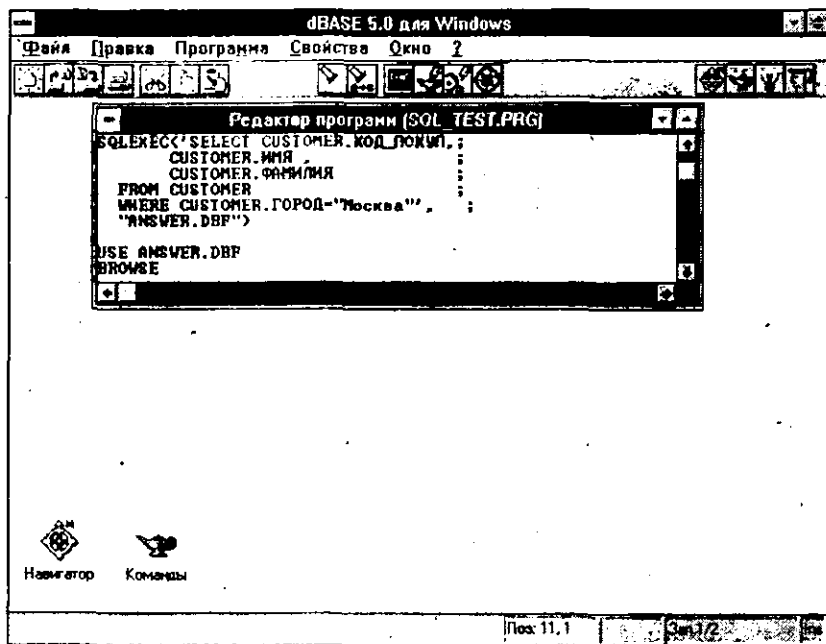


Рис. 12.8. Программа выполнения запроса и отображения результата в окне таблицы.

Мы разместили конструкцию SQL на нескольких строках, так при этом значительно улучшается ее читабельность. В качестве признака продолжения строки в

dBASE используется символ точка с запятой (;). Для выполнения запроса, а затем открытия результирующей таблицы и отображения ее в окне таблицы вам достаточно выполнить программу. Для этого выполните команду Программа | Выполнить. Результат выполнения этой программы приведен на рис. 12.9. В дальнейшем в программе мы будем изменять только SQL-конструкцию, оставляя остальные команды без изменения.

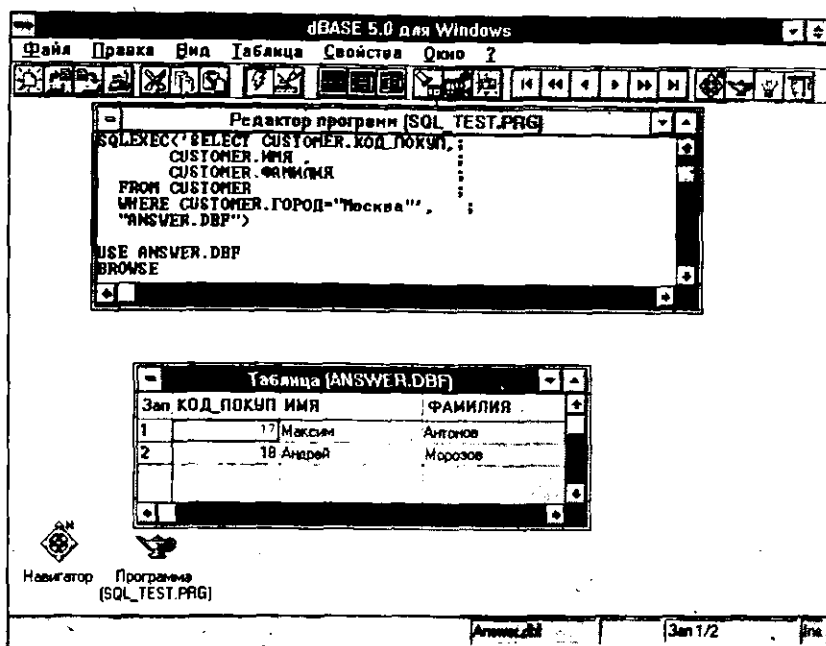


Рис. 12.9. Выполнение запроса и отображение результирующей таблицы.

Данный пример иллюстрирует наиболее простую форму конструкции SELECT в языке SQL:

SELECT	списокВыбираемыхПолей
FROM	списокТаблиц
WHERE	условиеВыборки

Заметим, что результатом запроса является другая таблица, которая получается из указанных в выборке таблиц. Таким образом, в реляционной СУБД пользователь всегда действует в рамках простой табличной структуры. Данное обстоятельство является весьма привлекательной особенностью таких систем.

При формировании запросов вы можете использовать уточненные имена полей (например, CUSTOMER.КОД_ПОКУП):

```
SELECT CUSTOMER.КОД_ПОКУП,
        CUSTOMER.ИМЯ,
        CUSTOMER.ФАМИЛИЯ
FROM CUSTOMER
WHERE CUSTOMER.ГОРОД="Москва"
```

Для выполнения более сложных выборов используется общая форма конструкции SELECT:

```
SELECT [DISTINCT] списокВыбираемыхПолей
FROM списокТаблиц
[WHERE условиеВыборки ]
[GROUP BY условиеГруппировки
[HAVING условиеВыборкиГруппы] ]
[ORDER BY условиеУпорядочивания ];
```

Теперь перейдем к иллюстрации с помощью ряда примеров основных особенностей этой конструкции.

Простая выборка

Выберем коды всех поставленных товаров из таблицы ORDSALE:

```
SELECT КОД_ТОВАРА
FROM ORDSALE
```

При выполнении данного запроса в результат выборки (рис. 12.10) будут включены все дубликаты товаров. Реляционная СУБД не исключает дубликатов из результата конструкции SELECT, если пользователь явно не потребует это сделать с помощью ключевого слова DISTINCT (различные), как будет показано в следующем примере.

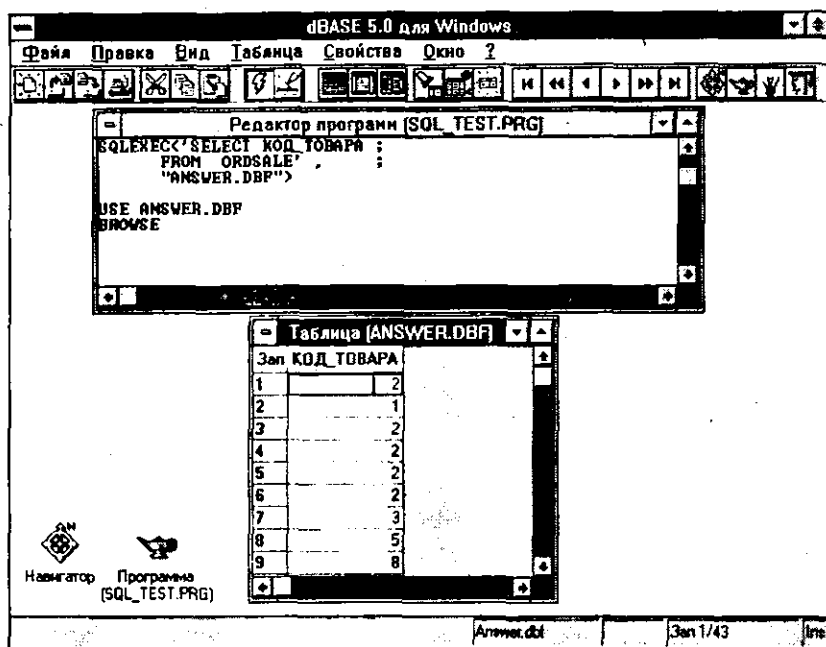


Рис. 12.10. Простая выборка. таблицы.

Выборка уникальных значений

Выберем из таблицы ORDSALE коды всех поставленных товаров, исключая повторяющиеся значения.


```
SELECT DISTINCT КОД_ТОВАРА
FROM ORDSALE
```

Результат выборки приведен на рис. 12.11.

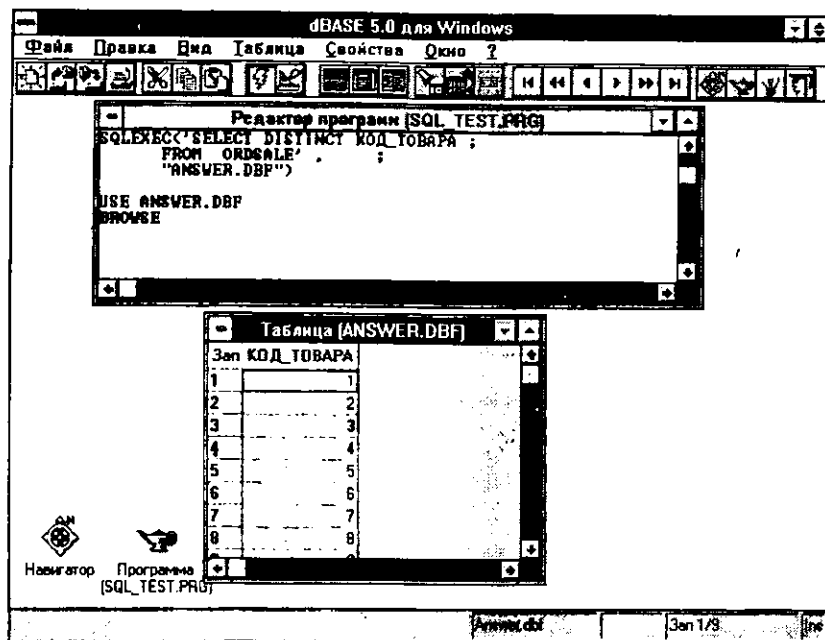


Рис. 12.11. Выборка уникальных значений.

Выборка вычисляемых значений

Результирующая таблица может включать не только поля исходных таблиц, но и результат вычисления над полями таблицы. В выражениях могут использоваться арифметические операторы и функции. Перечень допустимых функций определяется используемым диалектом языка SQL.



При переносе приложений с одного сервера данных на другой наиболее узким местом являются функции.

В качестве примера выберем коды и стоимость каждого товара с учетом налога на добавленную стоимость (НДС), предполагая, что в таблице GOODS стоимость не включает НДС:

```
SELECT КОД_ТОВАРА, ЦЕНА*(1+0.23)
FROM GOODS
```

Результат выборки приведен на рис. 12.12.

Конструкция SELECT может включать арифметические выражения, а также прос-тые имена полей. Кроме этого, вы можете добавить константы в результат выборки.

Например:

```
SELECT  КОД ТОВАРА, "Стоимость с учетом НДС",  
        ЦЕНА*(1+0.23)  
FROM    GOODS
```

Результат этой выборки содержит три поля: код товара, текстовое поле, которое для всех записей выборки имеет одинаковое значение, и стоимость товара с учетом НДС.

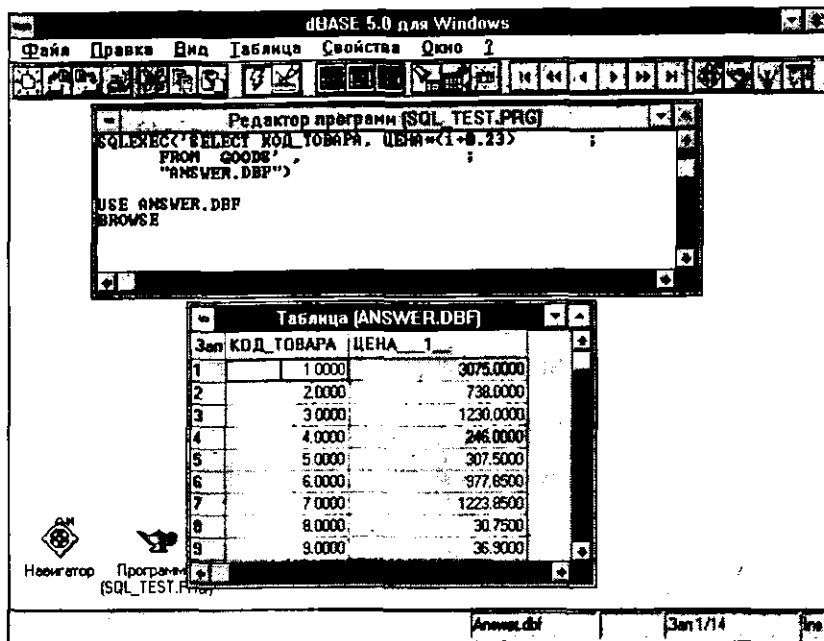


Рис. 12.12. Выборка вычисляемых значений.

Ограничение выборки

Выберем коды покупателей, которые живут в России и имеют кредит превышающий 200 000:

```
SELECT  КОД ПОКУП, ИМЯ, ФАМИЛИЯ  
FROM    CUSTOMER  
WHERE   СТРАНА="Россия"  
AND     КРЕДИТ>200000
```

Результат выполнения данного запроса приведен на рис. 12.13.

Условие, следующее за ключевым словом WHERE, может включать:

- Операторы сравнения =, <> (не равно), >, >=, < и <=
- Логические операторы AND (и), OR (или) и NOT (нет)
- Скобки, определяющие порядок вычислений

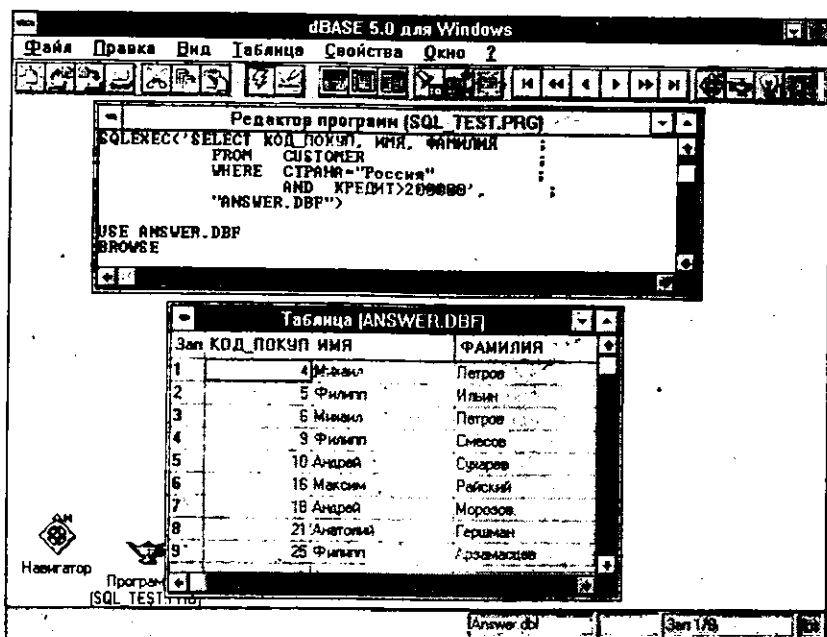


Рис. 12.13. Ограничение выборки.

При выполнении запроса условия выборки числа сравниваются алгебраически: отрицательные числа считаются меньше, чем положительные, независимо от их абсолютной величины. Строки сравниваются в соответствии с их представлением в коде OEM. При сравнении двух строк, имеющих разные длины, предварительно более короткая строка дополняется справа пробелами для того, чтобы обе строки имели одинаковую длину.

Выборка с упорядочением

Теперь выберем номера и кредит покупателей, проживающих в Москве, в порядке убывания их кредита:

```

SELECT    КОД_ПОКУП,
          ИМЯ, ФАМИЛИЯ, КРЕДИТ
FROM      CUSTOMER
WHERE     СТРАНА="Россия"
ORDER BY  КРЕДИТ DESC
  
```

Результат выполнения выборки приведен на рис. 12.14.

В выборках без указания критерия упорядочивания результирующая таблица будет упорядочена в соответствии с внутренними алгоритмами осуществления выборки. В данном случае явно указано, что результат выборки должен быть организован в определенной последовательности. Упорядочение может быть осуществлено по любому полю результирующей таблицы:

```

имяПоля [упорядочение]
[, имяПоля [упорядочение] ], ...,
  
```

где аргумент упорядочение может принимать значение ASC (возрастание) или DESC (убывание). По умолчанию устанавливается значение ASC. В качестве аргументов имяПоля могут использоваться только поля результирующей таблицы. Поэтому недопустима следующая конструкция:

```
SELECT КОД_ПОКУП, ФАМИЛИЯ
FROM CUSTOMER
ORDER BY ГОРОД
```

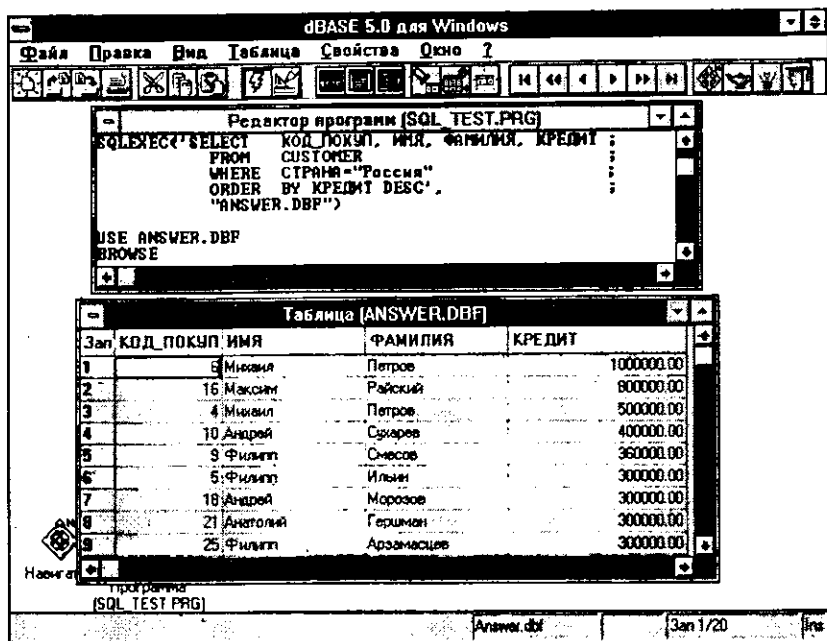


Рис. 12.14. Выборка с упорядочением.

Для идентификации полей, по которым осуществляется упорядочивание, вы можете использовать не только наименования полей результирующей таблицы, но и их номера (номер поля указывает порядковую позицию данного поля в результирующей таблице запроса). Благодаря этому результат можно упорядочить на основе вычисляемых полей, которые не обладают именами. В качестве примера упорядочивается результат выборки по возрастанию стоимости товара с учетом НДС:

```
SELECT КОД_ТОВАРА, ЦЕНА*(1+0,23)
FROM GOODS
ORDER BY 2
```

Здесь число 2 ссылается на второй столбец результирующей таблицы.

Выборка с использованием оператора BETWEEN (между)

Выберем сведения о товарах, стоимость которых находится в диапазоне от 20 000 до 100 000 включительно:

```
SELECT КОД_ТОВАРА, НАИМ, ЦЕНА
FROM GOODS
WHERE ЦЕНА BETWEEN 20000 AND 100000
```

Кроме этого вы можете использовать условие NOT BETWEEN (не принадлежит диапазону между), например:

```
SELECT КОД_ТОВАРА, НАИМ, ЦЕНА
FROM GOODS
WHERE ЦЕНА
      NOT BETWEEN 20000 AND 100000
```

Выборка с использованием оператора IN (принадлежит)

Выберем товары, стоимость которых равна 100 000, 200 000 или 500 000:

```
SELECT КОД_ТОВАРА, НАИМ, ЦЕНА
FROM GOODS
WHERE ЦЕНА IN(100000, 200000,
              500000)
```

Оператор IN в действительности является просто краткой записью условия, представляющего собой последовательность отдельных сравнений, соединенных операторами OR (или). Предыдущая конструкция SELECT эквивалентна конструкции, приведенной ниже:

```
SELECT КОД_ТОВАРА, НАИМ, ЦЕНА
FROM GOODS
WHERE ЦЕНА=100000
      OR ЦЕНА=200000
      OR ЦЕНА=500000
```

Вы можете использовать также оператор NOT IN (не принадлежит), например:

```
SELECT КОД_ТОВАРА, НАИМ, ЦЕНА
FROM GOODS
WHERE ЦЕНА
      NOT IN(100000, 200000, 500000)
```

Выборка с использованием шаблонов

Использование шаблонов в SQL расширяет возможности выборки данных, для которых вы не помните точные значения. В качестве примера выберем все товары, наименования которых начинаются с буквы "Т":

```
SELECT КОД_ТОВАРА, НАИМ, ЦЕНА
FROM GOODS
WHERE НАИМ LIKE "Т%"
```

Результат выполнения выборки приведен на рис. 12.15.
Обычно оператор LIKE имеет форму:

имяПоля LIKE строковаяКонстанта

Результат выполнения оператора принимает значение истина, если значение в указанном поле соответствует образцу, указанному аргументом строковаяКонстанта. Символы этой константы интерпретируются следующим образом:

- (пробел или подчеркивание) обозначает любой одиночный символ;
- % (процент) обозначает любую последовательность символов.

Все другие символы обозначают просто сами себя.

Следовательно, в приведенном примере конструкция SELECT будет осуществлять выборку записей из таблицы GOODS, для которых значение в поле "Наим" начинается с буквы "Т" и содержит далее любую последовательность символов (в том числе и отсутствие символов).

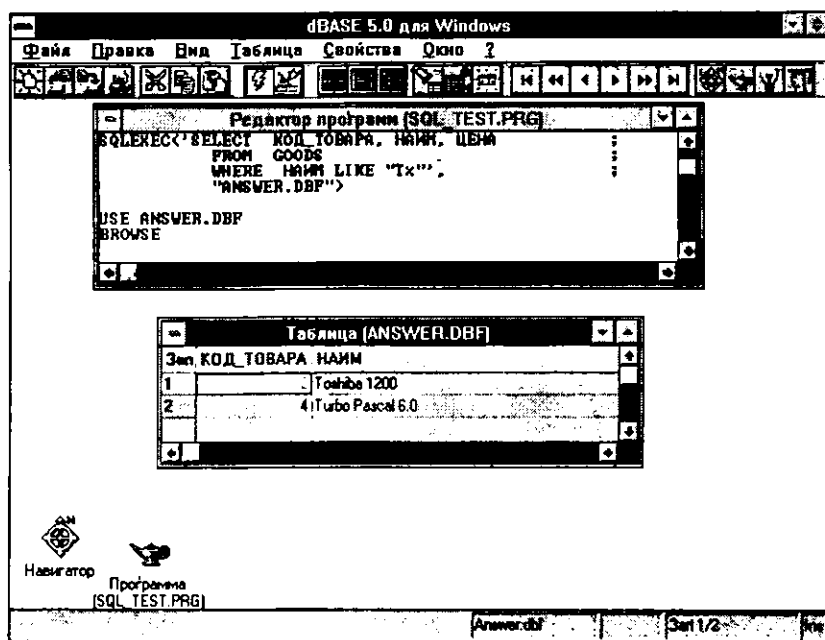


Рис. 12.15. Выборка с использованием шаблонов.

Выборки из связанных таблиц

Способность "соединять" две или более таблицы в одну представляет собой одно из наиболее мощных средств реляционных систем. Фактически, наличие операции соединения является едва ли не самым главным, что отличает реляционные системы от систем других типов.

Простое соединение

Выберем список наименований всех товаров, проданных покупателям:

```
SELECT ORDSALE.КОД_ПОКУП, GOODS.НАИМ
FROM ORDSALE, GOODS
WHERE ORDSALE.КОД_ТОВАРА=GOODS.КОД_ТОВАРА
```

Результат выполнения выборки приведен на рис. 12.16.

Отметьте, что в данном случае ссылки на поля во фразе WHERE должны уточняться именами содержащих их таблиц. Во избежание неопределенности в этой выборке два поля "Код товара" показаны явным образом: CUSTOMER.КОД_ТОВАРА и GOODS.КОД_ТОВАРА.

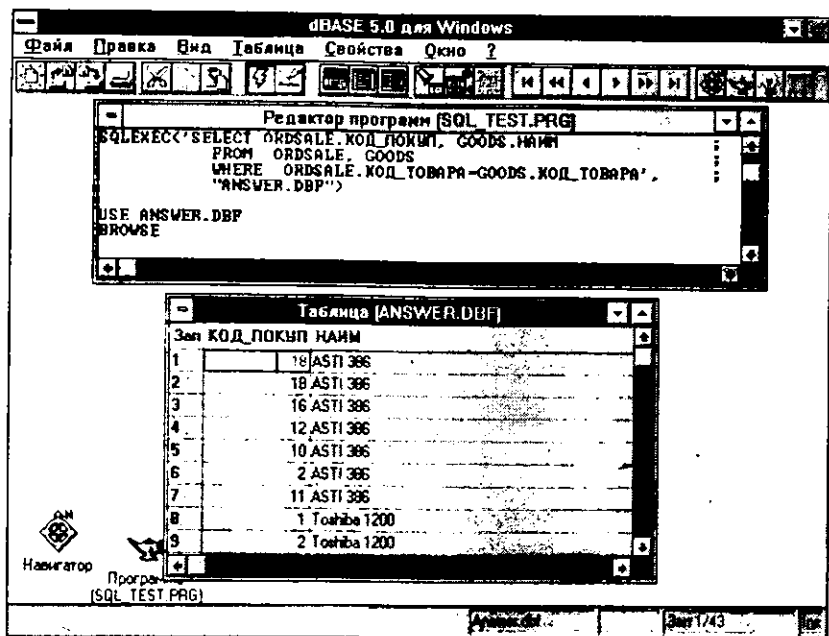


Рис. 12.16. Простое соединение..

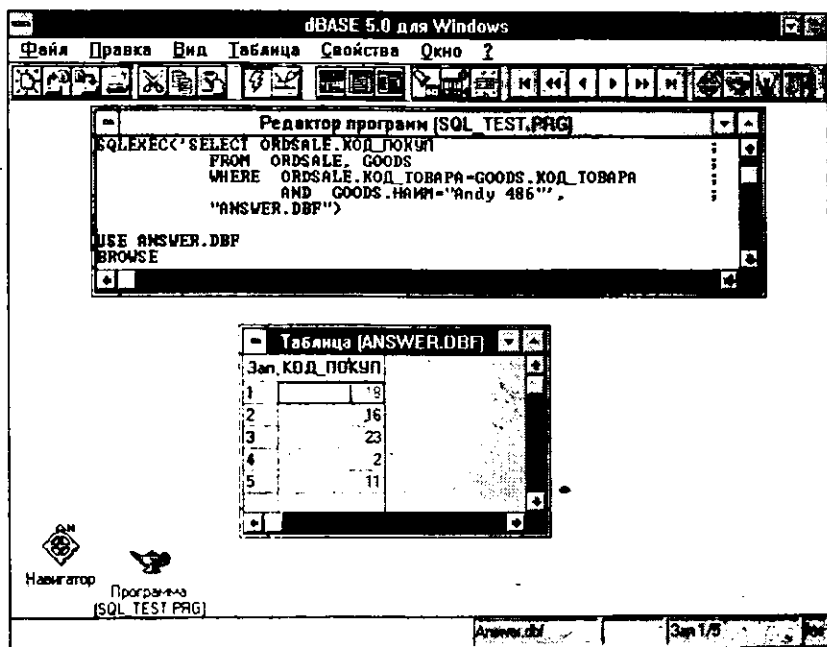


Рис. 12.17. Соединение с дополнительным условием.

Соединение с дополнительным условием

Выберем список покупателей, которым проданы компьютеры Andy 486.

```

SELECT ORDSALE.КОД ПОКУП
FROM ORDSALE, GOODS
WHERE ORDSALE.КОД ТОВАРА=GOODS.КОД ТОВАРА
AND GOODS.НАИМ="Andy 486"

```

Результат выборки приведен на рис. 12.17.

Соединение трех таблиц

Если вам необходимо выбрать не код покупателя, а его фамилию и имя соедините три таблицы: CUSTOMER, ORDSALE и GOODS:

```

SELECT DISTINCT CUSTOMER.ФАМИЛИЯ, CUSTOMER.ИМЯ
FROM CUSTOMER, ORDSALE, GOODS
WHERE ORDSALE.КОД ПОКУП=CUSTOMER.КОД ПОКУП
AND ORDSALE.КОД ТОВАРА=GOODS.КОД ТОВАРА
AND GOODS.НАИМ="Andy 486"

```

Результат выборки приведен на рис. 12.18.

В данном случае вы можете выбрать вместо номера покупателя его имя. Во многих случаях выборки из нескольких таблиц используются не только для ограничения выборки, но и для объединения данных из нескольких таблиц.

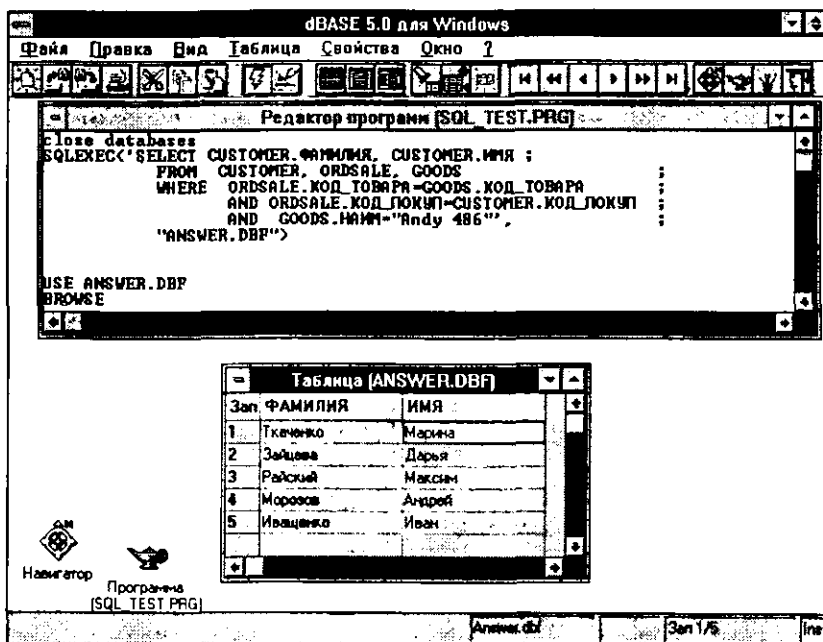


Рис. 12.18. Соединение трех таблиц.

Вычисление групповых значений

Конструкцию SELECT в том виде, в котором она использовалась до сих пор, невозможно применить для многих практических задач. Например, даже настолько простой запрос как "Сколько имеется покупателей?", нельзя сформировать, используя только введенные до сих пор конструкции. Для определения групповых значений

в SQL предусматривается ряд специальных стандартных функций, которые присутствуют во всех диалектах SQL:

Функция	Вычисляет
COUNT	Число значений в поле.
SUM	Сумма значений по полю.
AVERAGE	Среднее значение в поле.
MAX	Наибольшее значение в поле.
MIN	Наименьшее значение в поле.

За исключением COUNT, каждая из этих функций оперирует совокупностью значений в одном поле некоторой таблицы (которая может быть результатом выборки из заданных базовых таблиц) и вычисляет единственное значение.

Для функций SUM и AVERAGE рассматриваемый столбец должен содержать числовые значения. В общем случае аргументу функции может предшествовать ключевое слово DISTINCT, указывающее, что дублирующие значения должны быть исключены при выполнении функции. Однако для функций MAX и MIN ключевое слово DISTINCT должно быть опущено.

Использование функции COUNT

Определим общее количество покупателей.

```
SELECT COUNT(*)  
FROM CUSTOMER
```

Использование функций совместно с условием

Определим количество продаж товара "Andy 486".

```
SELECT COUNT(*)  
FROM ORDSALE  
WHERE КОД_ТОВАРА="Andy 486"
```

Выберем общее количество проданных "Andy 486".

```
SELECT SUM(ПРОДАНО)  
FROM ORDSALE  
WHERE КОД_ТОВАРА="Andy 486"
```

Использование функций совместно с подзапросом

Для создания более сложных запросов вы можете использовать вложенные запросы (*подзапросы*). Определим коды покупателей со значением поля "Кредит" меньшим, чем текущий средний кредит в таблице CUSTOMER.

```
SELECT КОД_ПОКУП  
FROM CUSTOMER  
WHERE КРЕДИТ <  
      (SELECT AVERAGE(КРЕДИТ)  
       FROM CUSTOMER)
```

Функция в коррелированном подзапросе

В некоторых, довольно часто встречающихся случаях необходимо выбрать данные из таблицы, основываясь на результатах дополнительных выборок из этой же таблицы. Такие выборки называются *коррелированными*. Для их выполнения используются псевдонимы таблиц, которые следуют непосредственно за именем таблицы в выборке. В приведенном ниже примере используются псевдонимы таблицы CUSTOMER: CUST_X и CUST_Y.

Выберем код покупателя, кредит и город для всех покупателей, у которых кредит больше или равен среднему для их конкретного города.

```
SELECT КОД ПОКУП, КРЕДИТ, ГОРОД
FROM CUSTOMER CUST_X
WHERE КРЕДИТ >=
      (SELECT AVERAGE(КРЕДИТ)
       FROM CUSTOMER CUST_Y
       WHERE CUST_Y.ГОРОД=CUST_X.ГОРОД)
```

Использование группировки данных

В приведенных выше примерах результатом выборки всегда являлась одна запись. Рассмотрим теперь примеры вычисления групповых значений для заданных групп. Предположим, что требуется вычислить общий объем покупок для каждого товара, т. е. для каждого покупаемого товара необходимо определить код этого товара и общий объем покупок:

```
SELECT КОД ТОВАРА, SUM(ПРОДАНО)
FROM ORDSALE
GROUP BY КОД_ТОВАРА
```

С концептуальной точки зрения, оператор GROUP BY (группировать по) перекомпоновывает таблицу, представленную фразой FROM, в разделы или группы таким образом, чтобы в каждой группе все строки имели одно и то же значение поля, указанного во фразе GROUP BY. Конечно, это не означает, что таблица физически перекомпоновывается в базе данных. В рассматриваемом примере строки таблицы ORDSALE группируются таким образом, что в одной группе содержатся все строки для одного вида товара.

Далее, к каждой группе перекомпонованной таблицы (а не к каждой записи исходной таблицы) применяется конструкция SELECT. Каждое выражение во фразе SELECT должно принимать единственное значение для группы, т. е. оно может быть либо самим полем, указанным в операторе GROUP BY, либо арифметическим выражением, включающим это поле, либо константой, либо такой функцией, как SUM, которая оперирует всеми значениями данного поля в группе и сводит эти значения к единственному значению.

Строки таблицы можно группировать по любой комбинации ее полей. Если поле, по значениям которого осуществляется группировка, содержит какие-либо неопределенные значения, то каждое из них порождает отдельную группу.

Фраза GROUP BY не предполагает ORDER BY (упорядочить по). Для упорядочения результата этого примера по кодам товаров, следует поместить фразу ORDER BY КОД_ТОВАРА после фразы GROUP BY.

Использование группировки данных совместно с условием

Выберем для каждого покупаемого товара его код и общий объем покупок, за исключением покупок покупателя с кодом 23.

```
SELECT КОД_ТОВАРА, SUM(ПРОДАНО)
FROM ORDSALE
WHERE КОД_ПОКУП<>23
GROUP BY КОД_ТОВАРА
```

Строки, не удовлетворяющие условию WHERE, исключаются перед группировкой данных.

Использование HAVING

Для ограничения записей, участвующих в группировке, используется оператор HAVING, который играет такую же роль для групп, что и оператор WHERE для записей. Иными словами, HAVING используется для того, чтобы исключать группы точно так же, как WHERE используется для исключения записей. Выражение во фразе HAVING должно принимать единственное значение для группы.

Выберем коды товаров, покупаемых более чем одним покупателем:

```
SELECT КОД_ТОВАРА
FROM ORDSALE
GROUP BY КОД_ТОВАРА
HAVING COUNT(*)>1;
```

Оператор HAVING не можете использоваться отдельно от оператора GROUP BY.

Данный пример может быть выполнен и с помощью коррелированных запросов. Например:

```
SELECT DISTINCT КОД_ТОВАРА
FROM ORDSALE O&S_X
WHERE 1<
  (SELECT COUNT(*)
   FROM ORDSALE O&S_Y
   WHERE O&S_Y.КОД_ТОВАРА=O&S_X.КОД_ТОВАРА)
```

Следующий вариант этого запроса, в котором вместо псевдонима O&S_X используется таблица GOODS, является, вероятно, более ясным:

```
SELECT КОД_ТОВАРА
FROM GOODS
WHERE 1<
  (SELECT COUNT(КОД_ПОКУП)
   FROM ORDSALE
   WHERE ORDSALE.КОД_ТОВАРА=
     GOODS.КОД_ТОВАРА)
```

Оба этих альтернативных варианта логически более понятны и не требуют дополнительных языковых конструкций, поэтому они более предпочтительны по сравнению с вариантом GROUP BY/HAVING..



Конструкции GROUP BY свойственно серьезное ограничение, которое заключается в том, что она работает только на одном уровне. Невозможно разбить каждую из этих групп на группы более низкого уровня, а затем применить некоторую стандартную функцию, например SUM или AVERAGE, на каждом уровне группировки.

Использование квантора существования в запросах

Оператор EXISTS (существует) представляет в SQL-запросах квантор существования — понятие, заимствованное из формальной логики. Пусть символ “х” обозначает некоторую произвольную переменную. Тогда в формальной логике оператор с примененным квантором существования EXISTS х (предикат-зависящий-от-х) принимает значение истина тогда и только тогда, когда “предикат-зависящий-от-х” имеет значение истина при каком-либо значении переменной х. Предположим, например, что переменная х обозначает любое целое число в диапазоне от 1 до 10. Тогда предикат:

```
EXISTS х(х<5)
```

принимает значение истина. Напротив, предикат:

```
EXISTS х(х<0)
```

принимает значение ложь.

В языке SQL предикат с квантором существования представлен выражением вида:

```
EXISTS (SELECT * FROM...)
```

Выражение считается истинным тогда и только тогда, когда результат вычисления подзапроса, представленного с помощью SELECT * FROM..., является непустым множеством. Иными словами, выражение истинно тогда и только тогда, когда существует какая-либо запись в таблице, указанной во фразе FROM подзапроса, которая удовлетворяет условию WHERE этого подзапроса.

В качестве примера выберем фамилии покупателей, которым продан компьютер “Andy 486”.

```
SELECT  ФАМИЛИЯ
FROM    CUSTOMER
WHERE   EXISTS
        (SELECT *
         FROM  ORDSALE
         WHERE ORDSALE.КОД_ПОКУП=
              CUSTOMER.КОД_ПОКУП
         AND  КОД_ТОВАРА="Andy 486")
```

В данном примере поочередно рассматривается каждое значение поля “Фамилия” и проверяется истинность условия существования. Предположим, что первое значение поля “Фамилия” — “Зайцева”, а соответствующее значение поля “Код_покуп” — 23. Если множество записей из ORDSALE, содержащих “Код_покуп”, равный 23, и “Код_товара”, равный “Andy 486”, не пусто, то “Зайцева” будет одним из результирующих значений. Аналогично выбираются и другие значения поля “Фамилия”.

Этот пример показывает иной способ формулировки запроса, по сравнению с рассмотренными ранее. Оператор EXISTS реализует одну из наиболее важных возможностей языка SQL. Фактически любой запрос, который может быть выражен с использованием IN, альтернативным образом может быть сформулирован также с помощью EXISTS.

Вы можете сконструировать отрицание существования, используя совместно операторы NOT и EXISTS.

В качестве примера выберем фамилии покупателей, которые не купили "Andy 486":

```
SELECT  ФАМИЛИЯ
FROM    CUSTOMER
WHERE   NOT EXISTS
        (SELECT  *
         FROM    ORDSALE
         WHERE   ORDSALE.КОД_ПОКУП=
                CUSTOMER.КОД_ПОКУП
         AND    КОД_ТОВАРА="Andy 486")
```

Обратите внимание на то, что в подзапросе осуществляется выборка из двух таблиц CUSTOMER и ORDSALE, но во фразе FROM перечисляется только ORDSALE.

Заключенный в скобки подзапрос, входящий в выражение EXISTS, вовсе не обязательно должен использовать предложение SELECT вида SELECT *. Например, можно использовать предложение следующего вида: SELECT имя-поля FROM... Однако на практике почти всегда используется вид SELECT *, как было продемонстрировано в приведенных выше примерах.

Объединение множеств

Объединением двух множеств называется множество всех элементов, принадлежащих какому-либо одному или обоим исходным множествам. Поскольку отношение является множеством записей, можно построить объединение двух отношений. Результатом будет множество, состоящее из всех строк, входящих в какое-либо одно или в оба первоначальных отношения. Однако, если этот результат сам по себе должен быть другим отношением, а не просто разнородной смесью строк, то два исходных отношения должны быть совместимыми по объединению, т. е. строки в обоих отношениях должны быть одной и той же формы. Что касается SQL, то две таблицы совместимы по объединению (и к ним может быть применен оператор объединения UNION) тогда и только тогда, когда:

1. Они имеют одинаковое число полей (например, m).
2. Для всех i ($i = 1, 2, \dots, m$) i -е поле первой таблицы и i -е поле второй таблицы имеют в точности одинаковый тип данных.

Пример запроса, использующего оператор UNION

В качестве примера выберем коды товаров, которые имеют стоимость более 1 000, либо приобретаются покупателем с кодом 23 (либо то и другое).

```
SELECT  КОД_ТОВАРА
FROM    GOODS
WHERE   ЦЕНА>1000
UNION
SELECT  КОД_ТОВАРА
FROM    ORDSALE
WHERE   КОД_ПОКУП=23
```

Из результата выборки, использующей оператор UNION, всегда исключаются повторяющиеся значения. Поэтому, хотя в рассматриваемом примере товар выбирает-

ся обеими составляющих предложений SELECT, в результирующей таблице каждый товар присутствует только один раз.

Вы можете соединить оператором UNION любое число конструкций SELECT.

Рассмотрим расширенный вариант данного примера, включив в него коды товаров ценой менее 500. Для этого дополним приведенный выше запрос следующей конструкцией:

```
UNION
  SELECT КОД_ТОВАРА
    FROM GOODS
   WHERE ЦЕНА<500
```

Аналогичный результат можно получить, добавив к первому из первоначальных конструкций SELECT оператор OR ЦЕНА<500.

Оператор ORDER BY в запросе с использованием оператора UNION может входить только в последнее предложение SELECT. При указании критерия упорядочивания используйте номера полей результирующей таблицы.

При использовании оператора UNION часто оказывается полезным включение константы в результирующую таблицу. Например, текстовую константу можно использовать в качестве пояснения условия выборки товаров:

```
SELECT КОД_ТОВАРА,
      "Стоимость товара > 1000 $"
  FROM GOODS
 WHERE ЦЕНА>1000
 UNION
 SELECT КОД_ТОВАРА,
      "Товар куплен покупателем 23"
  FROM ORDSALE
 WHERE КОД_ПОКУП=23
 ORDER BY 2,1
```

Модификация данных в таблицах

Познакомившись с возможностями выборки данных с помощью конструкции SELECT, перейдем теперь к модификации данных в таблицах. В SQL для этих целей используется конструкция UPDATE, которая имеет следующий синтаксис:

```
UPDATE   таблица
SET      поле=выражение
        [, поле=выражение] ...
[WHERE   условие]
```

В результате выполнения конструкции все записи в таблице, которые удовлетворяют условию, обновляются в соответствии с оператором присвоения поле=выражение.

Модификация единственной записи

Изменим наименование товара "Andy 486" на "Pentium" и увеличим стоимость на 400 000.00.

```
UPDATE GOODS
  SET  НАИМ="Pentium"
      ЦЕНА=ЦЕНА+400000.00
 WHERE НАИМ="Andy 486"
```

Для каждой записи, которая должна быть обновлена (т. е. для каждой записи, которая удовлетворяет условию WHERE, или для всех записей, если фраза WHERE опущена), ссылки во фразе SET на поля этой записи обозначают значения этих полей до их модификации.

Модификация множества записей

Удвоим кредит всех покупателей, находящихся в Киеве:

```
UPDATE CUSTOMER
SET КРЕДИТ=КРЕДИТ*2
WHERE ГОРОД="Киев"
```

Модификация с подзапросом

Установим скидку в 20% для всех покупателей, проживающих в Минске:

```
UPDATE ORDSALE
SET ЦЕНА=0.8*ЦЕНА
WHERE "Минск"=
(SELECT ГОРОД
FROM CUSTOMER
WHERE CUSTOMER.КОД_ПОКУП=
ORDSALE.КОД_ПОКУП)
```

В конструкции UPDATE может использоваться только одна таблица, таким образом вы не можете в рамках одного запроса изменить данные более чем в одной таблице.

При использовании подзапросов для модификации данных результат выборки должен возвращать только одно значение, а не несколько.

Удаление данных

Для удаления данных таблиц используется конструкция DELETE, которая имеет следующий синтаксис:

```
DELETE
FROM      таблица
[WHERE    условие]
```

В результате выполнения конструкции удаляются все записи в таблице, которые удовлетворяют условию.

Удаление единственной записи

Удалим покупателя с кодом 23:

```
DELETE
FROM CUSTOMER
WHERE КОД_ПОКУП=23
```

Удаление множества записей

Удалим из таблицы ORDSALE все покупки товара с кодом 34.

```
DELETE
FROM ORDSALE
WHERE КОД_ТОВАРА=34
```

Удаление всех записей

Удалим все записи из таблицы ORDSALE:

```
DELETE
FROM ORDSALE
```

В результате выполнения этой операции таблица доступна при дальнейшей работе, однако она пуста.

Удаление всех записей не приводит к уничтожению таблицы.

Удаление с подзапросом

Удалим все покупки клиентов, проживающих в Киеве.

```
DELETE
FROM ORDSALE
WHERE "Киев"=
      (SELECT ГОРОД
       FROM CUSTOMER
       WHERE CUSTOMER.КОД_ПОКУП=
             ORDSALE.КОД_ПОКУП)
```

Добавление записей

Для добавления записей используется конструкция INSERT, которая имеет два варианта синтаксиса:

```
INSERT
INTO   таблица [(поле [,поле]...)]
VALUES (константа [, константа]...)
```

или:

```
INSERT
INTO   таблица [(поле [,поле]...)]
подзапрос
```

В первом варианте в таблицу вставляется запись, имеющая заданные значения для указанных полей, причем i-я константа в списке констант соответствует i-му полю в списке полей. Во втором варианте формируется подзапрос, представляющий собой множество записей, которые добавляются в таблицу. При этом i-е поле результата подзапроса соответствует i-му полю в списке полей добавляемой таблицы. В обоих случаях отсутствие списка полей эквивалентно использованию всех полей таблицы.

Вставка единственной записи

Добавим в таблицу GOODS товар "Andy 486" стоимостью 3 000 000.00.

```
INSERT
INTO   GOODS
      (КОД_ТОВАРА, НАИМ, ЦЕНА, КАТЕГОРИЯ)
```


VALUES

(1001, "Andy 486", 3000000.00, 2)

В результате создается новая запись для товара с заданным номером, наименованием, стоимостью и категорией товара.

Вставка единственной записи с опущенными именами полей

В предыдущем примере опустим имена полей таблицы:

INSERT

INTO GOODS

VALUES (1001, "Andy 486", 3000000.00, 1)

Отсутствие полей эквивалентно перечислению списка всех полей таблицы в порядке слева направо так, как они были определены при создании таблицы.

Вставка множества записей

Если по каким-либо причинам из таблицы CUSTOMER были удалены все записи, вы можете добавить в нее всех покупателей из таблицы ORDSALE:

INSERT

INTO CUSTOMER (КОД_ПОКУП)

SELECT DISTINCT КОД_ПОКУП

FROM ORDSALE

Здесь конструкция SELECT выполняется точно так же, как обычно, но результат не возвращается пользователю, а копируется в таблицу CUSTOMER.

Заключение

- Для добавления в бланк запроса новой таблицы выберите Запрос | Добавить таблицу или нажмите кнопку Добавить таблицу.
- Для удаления ненужных бланков запроса из окна запроса используется команда Запрос | Удалить таблицу или кнопка Удалить таблицу.
- Для связывания двух таблиц, бланки запроса которых открыты в окне запроса, выполните команду Запрос | Установить связь.
- Вы можете установить режимы отображения полей в результирующей таблице используя флажки Только главные и С подчиненной таблицей.
- В dBASE для выборки данных вы можете использовать язык высокого уровня SQL, который является стандартным средством доступа к данным в реляционных СУБД.
- Для формирования запросов в SQL используется конструкция SELECT.
- Для добавления записей в SQL используется конструкция INSERT.
- Для модификации значений таблиц в SQL используется конструкция UPDATE.

Глава 13. Многотабличные формы

В главе 11 вы познакомились с основными принципами построения многотабличных баз данных, в главе 12 — с построением многотабличных запросов. В настоящей главе мы остановимся на вопросах создания форм для одновременной работы с несколькими таблицами на основе многотабличных запросов и покажем, как использовать эти формы для просмотра, редактирования и ввода данных. Мы обсудим проблему поддержания целостности данных, хранящихся в связанных таблицах, при их редактировании с помощью многотабличных форм.

Знакомство с многотабличными формами начнем с использования уже готовой формы для просмотра данных из нескольких связанных таблиц.

Просмотр данных из нескольких таблиц

В качестве примера обратимся к многотабличной форме, которая отображает информацию о покупателях из таблицы CUSTOMER и сделанных ими заказах из таблицы ORDSALE. В данном случае для связывания записей подчиненной таблицы (ORDSALE) с записями главной таблицы (CUSTOMER) использованы значения поля "Код_покуп".

дBASE 5.0 для Windows

Файл Правка Вид Форма Свойства Окно ?

ПОКУПАТЕЛЬ

ПОКУПАТЕЛЬ Александров
Сергей
Иванович
Код 1
Индекс 630090
Адрес пл. Ильича, д. 34
Примечание Осуществляет поставку и гарантийное обслуживание IBM PC и совместимых. В настоящее время расширяет перечень предлагаемых услуг и товаров: ак. Новинка: предлагает поставку заказного оборудования, в том числе и сист.

Телефон 13831224-3124
Предприятие АО Интерсервис
Кредит 100000.00

Код товара	Заказано	Дата заказа	Дата продажи	Продано	Цена
2	5	02.04.95	12.04.95	5	610.00

Customer.dbf 3 of 1/30

Рис. 13.1. Просмотр таблиц CUSTOMER и ORDSALE.

На рис. 13.1 вы видите многотабличную форму, в которой отображена информация из двух таблиц — CUSTOMER и ORDSALE. В верхней части отображены сведения о покупателе из таблицы CUSTOMER, а в нижней части, отображаемой в виде таблицы, — информация, содержащаяся в связанной с ней записи о покупке данного клиента в таблице ORDSALE. Фактически любая многотабличная форма представляет собой главную форму, отображающую данные из главной таблицы, в которую встроена одна или более форм, отображающих информацию из подчиненных таблиц.

Поэтому формы, размещенные в главной форме, называют *подчиненными*. Такая структура многотабличной формы позволяет вам с помощью одной формы просматривать информацию сразу из нескольких таблиц.

При появлении многотабличной формы на экране курсор первоначально располагается в главной форме. Для перемещения по полям формы вы можете воспользоваться мышью или клавиатурой (клавиши-стрелки, *Tab* и *Shift+Tab*). Перемещение по записям осуществляется с помощью кнопок, расположенных на панели управления. Кроме того, вы можете использовать клавиши *PgDn* и *PgUp* (перемещают курсор к последующей и предыдущей записи) и *Ctrl+PgDn* и *Ctrl+PgUp* (перемещают курсор в первую и последнюю запись таблицы).

Давайте воспользуемся указанными средствами для просмотра информации из таблиц *CUSTOMER* и *ORDSALE*. При перемещении курсора по записям таблицы *CUSTOMER* в нижней части будут отображены сведения о заказах данного покупателя. Если клиент не сделал ни одного заказа, данные в этой части формы будут отсутствовать.

Между таблицами *CUSTOMER* и *ORDSALE* существует отношение один-ко-многим, поэтому с каждой из записей таблицы *CUSTOMER* могут быть связаны несколько записей из таблицы *ORDSALE*. Если вы захотите просмотреть заказы текущего покупателя, перейдите в табличный образ, расположенный в нижней части формы. Для просмотра записей из таблицы *ORDSALE* используйте клавиши $\uparrow$ и $\downarrow$, *PgDn* и *PgUp* или воспользуйтесь вертикальной полосой прокрутки, расположенной в правой части образа. Для возврата в главную форму воспользуйтесь мышью или клавишей *Tab*.

Многотабличная форма очень удобна для просмотра информации о заказах клиентов. Находясь в главной форме, вы можете с помощью команды *Форма | Перейти к записи* или клавиш *Ctrl+G* перейти к определенному номеру записи и найти требуемого клиента. После этого перейдите в подчиненную форму и просмотрите сведения обо всех заказах данного клиента.

Построение запроса для многотабличной формы

При создании многотабличной формы в качестве источника данных используется не отдельная таблица, а многотабличный запрос. Поэтому на первом этапе создания многотабличной формы вам необходимо создать многотабличный запрос. Прежде, чем приступить к созданию запроса в *dBASE* определите главную и подчиненные таблицы, а также отношение между ними. Давайте в качестве примера создадим запрос для формы, показанной на рис. 13.1.

Выбор главной таблицы многотабличной формы определяется задачами, для решения которых предназначена создаваемая форма. В данном случае нам требуется форма, которая позволит получить сведения обо всех заказах и покупках каждого из клиентов. Это требование обуславливает выбор *CUSTOMER* в качестве главной таблицы. В другом случае, возможно, вам потребуются данные обо всех заказах и покупках клиентов за определенный интервал времени. При решении этой задачи для выбора требуемых записей из таблицы *ORDSALE* вам придется воспользоваться запросом, в качестве главной таблицы которого будет выбрана таблица *ORDSALE*, а подчиненной таблицы — *CUSTOMER*.

На следующем этапе вам необходимо определить связи между подчиненными и главной таблицами. Для установки связи между главной и подчиненными таблицами в *dBASE* используется окно диалога "Установление связи". В данном случае связь устанавливаем по полю "Код покуп", которое содержат обе таблицы. При этом используется индекс подчиненной таблицы, в котором это поле является первым полем индексного выражения.



В таблицах CUSTOMER и ORDSALE совпадающие поля имеют одинаковые имена. Однако это требование не является обязательным в dBASE. Достаточно, чтобы совпадающие поля имели совместимые типы данных.



Наличие индексов главной таблицы не является обязательным требованием. В данном случае поле "Код_покуп" в таблице CUSTOMER может не входить ни в один из индексов.

В рассматриваемом примере оказалось достаточно просто установить соотношение между таблицами и определить совпадающие поля. Однако, на практике могут встретиться более сложные случаи, когда существует несколько подчиненных таблиц, каждая из которых в свою очередь может иметь подчиненные таблицы следующего уровня. В таких случаях при построении форм (по крайней мере, первых) мы рекомендуем нарисовать с помощью компьютера или на бумаге схему данных. Такой подход позволит вам сэкономить время при определении запроса в dBASE.

Создание многотабличной формы с помощью эксперта

Пример 13.1. Определение многотабличной формы с помощью эксперта

Для создания многотабличной формы вы можете использовать эксперта по созданию формы, который разместит в ней все указанные вами объекты интерфейса. При определении многотабличной формы с помощью эксперта вы должны выполнить следующие действия:

1. Используя таблицы CUSTOMER и ORDSALE, создайте запрос. Задайте CUSTOMER в качестве главной таблицы. Установите связь между таблицами по полю "Код_покуп". Сохраните созданный запрос.
2. Выберите **Файл | Создать | Форма** в меню главного окна dBASE. По умолчанию опция **Помощь эксперта** в окне диалога "Эксперт по разработке форм" установлена, поэтому нажмите кнопку **Далее** (рис. 13.2).

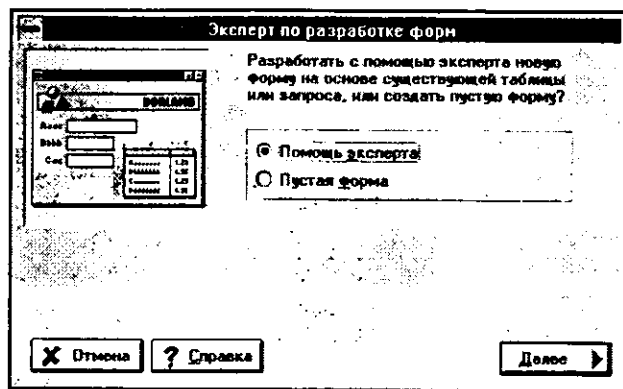


Рис. 13.2. Окно диалога "Эксперт по разработке форм".

3. На этом этапе вам необходимо указать какие из таблиц или запросов будут использованы в создаваемой форме. Для этого в следующем окне диалога выберите в области "Доступные файлы" запрос, созданный на шаге 1, который будет ис-

пользоваться при построении многотабличной формы (рис. 13.3). Нажмите кнопку *Далее*.

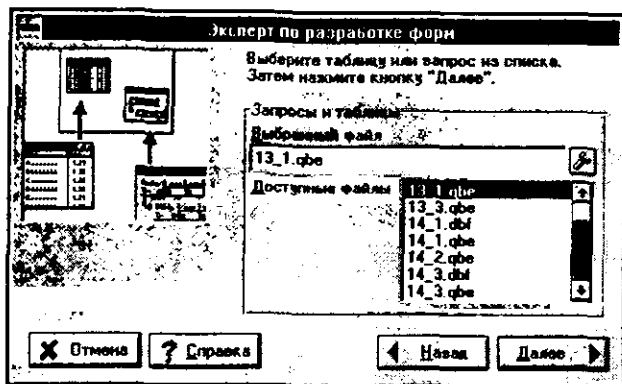


Рис. 13.3. Выбор запроса для создания многотабличной формы.

4. В следующем окне диалога выберите из области "Доступные" сначала поля таблицы CUSTOMER, а затем и таблицы ORDSALE, которые вы будете использовать при построении многотабличной формы, и перенесите их в область "Выбранные" (рис. 13.4). Завершите выбор нажатием кнопки *Далее*.

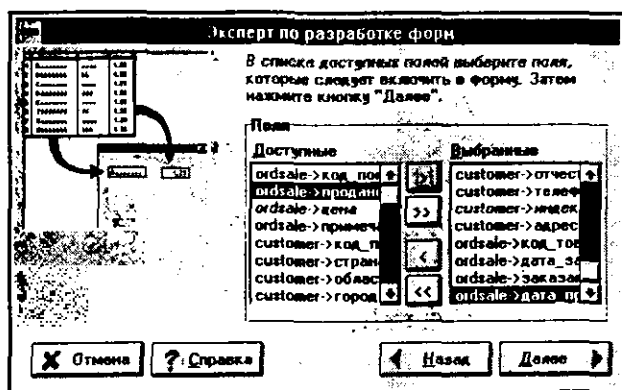


Рис. 13.4. Окно диалога "Эксперт по разработке форм" с выбранными полями таблиц CUSTOMER и ORDSALE.

5. Теперь вам необходимо определить тип макета представления данных в многотабличной форме. Так как тип отношения между таблицами CUSTOMER и ORDSALE один-ко-многим, каждой записи таблицы CUSTOMER будет соответствовать несколько записей таблицы ORDSALE. Установите опцию *С подчиненной таблицей* (рис. 13.5). Этот макет удобен для просмотра таблицы, в которой подчиненные записи открываются по мере продвижения по главным записям. После установки опции нажмите кнопку *Далее*.
6. В следующем окне диалога установите шрифты заголовка, надписей и элементов, а также цвет фона и текста. Создание формы завершите нажатием кнопки *Создать*.

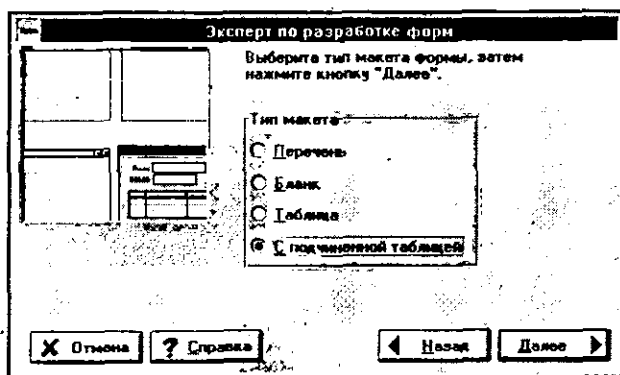


Рис. 13.5. Выбор типа макета.

- На экране появится созданная с помощью эксперта экранная форма для связанных таблиц **CUSTOMER** и **ORDSALE** (рис. 13.6). В верхней части формы располагаются поля таблицы **CUSTOMER**, в нижней части — объект-таблица, содержащая данные из таблицы **ORDSALE**. Сохраните форму, выполнив команду **Файл | Сохранить как**. Теперь вам осталось модифицировать форму, придав ей более наглядный вид.

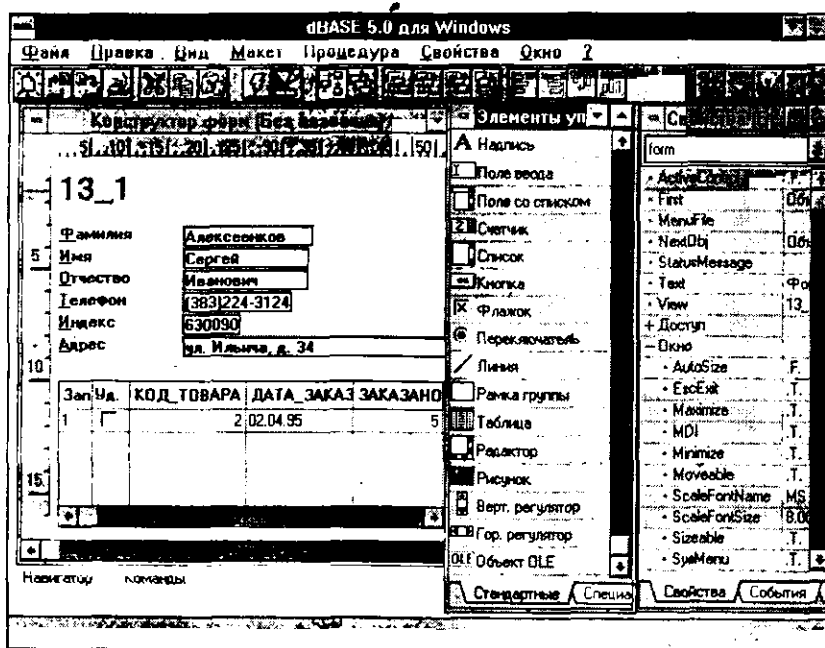


Рис. 13.6. Экранная форма, созданная с помощью эксперта.

Доработка многотабличной формы

Многотабличная форма, созданная с помощью эксперта, позволяет вам просматривать и редактировать данные входящих в нее таблиц. Однако для использования ее в законченном приложении может потребоваться дополнительная доработка. Прежде всего это касается оформления формы. Вы можете сгруппировать поля, а также распо-

ложить их в порядке, наиболее удобным для ввода. Надписи к полям в форме, созданной с помощью эксперта, содержат наименования полей таблицы. Вследствие ограничений в dBASE на наименования полей таблицы может потребоваться коррекция надписей к полям формы. Кроме этого вы можете определить форматы полей ввода, проверку достоверности ввода данных и т.п. Поэтому форму, созданную с помощью эксперта, можно считать эскизом законченной формы, которую вы совершенствуете в дальнейшем в конструкторе форм.

При работе над формой в окне конструктора в вашем распоряжении находятся все средства, с которыми вы познакомились в процессе создания однотабличных форм. Сейчас мы остановимся на специфических особенностях табличного формата, наиболее часто используемого при создании форм для таблиц, между которыми существует соотношение один-ко-многим.

Определение свойств объекта-таблицы

При выборе типа макета С подчиненной таблицей в окне диалога “Эксперт по разработке форм” в форме будет размещен объект-таблица. Эксперт устанавливает стандартные значения свойств объекта-таблицы. Вы можете изменить эти свойства, а также определить для таблицы перечень полей, отображаемых в таблице, их заголовки, форматы отображения данных и т.п. Для изменения свойств объекта-таблицы, как и для всех других объектов интерфейса, используется окно свойств объекта.

Объект-таблица может иметь заголовок, содержащий краткое описание этого объекта. Например, в форме, созданной в примере 13.1, вы можете определить для объекта таблицы заголовок “Список заказов и продаж”. Для этого в окне свойств выберите Text и в поле ввода значения текста напечатайте Список заказов и продаж. В результате у объекта-таблицы появится требуемый заголовок (рис. 13.7).

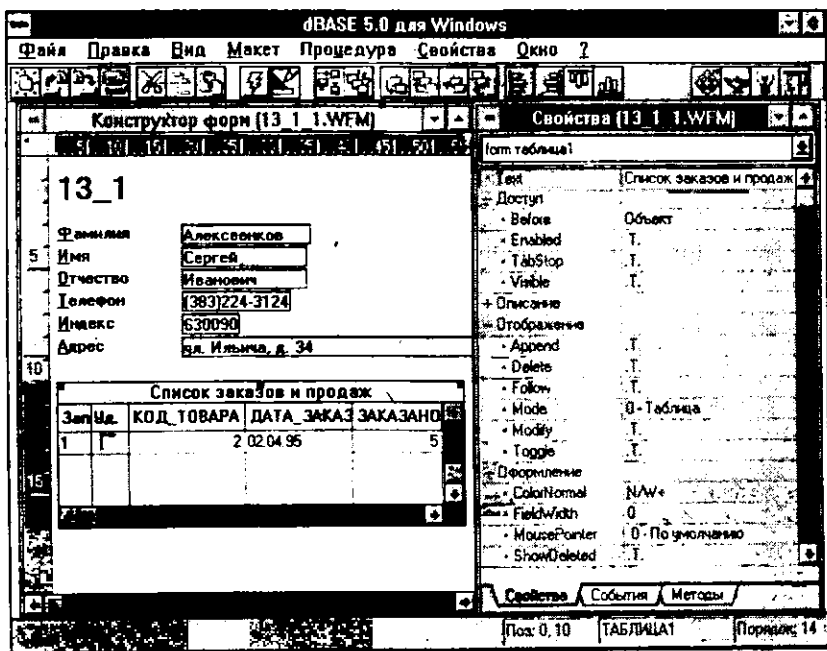


Рис. 13.7. Определение заголовка объекта-таблицы.

Для таблицы вы можете разрешить удаление, добавление и модификацию данных таблицы с помощью следующих свойств:

Свойство	Назначение
Append	Определяет, можно ли добавлять записи в таблицу.
Delete	Определяет, можно ли удалять записи из таблицы.
Modify	Определяет, можно ли изменять записи в таблице.

Свойство **Mode** определяет режим отображения данных объекта-таблицы: режим таблицы, перечня или бланка. По умолчанию установлен режим таблицы. Свойство **Toggle** определяет может ли пользователь переключаться между режимами отображения данных.

Для задания перечня полей, отображаемых в таблице, а также свойств полей объекта-таблицы используется свойство **Fields**, для которого существует кнопка **Инструмент**. При нажатии на эту кнопку на экране открывается уже знакомое вам окно диалога “Выбор полей” (рис. 13.8). В этом окне диалога вы можете удалить поле из списка выбранных полей, добавить в этот список новое поле, а также изменить порядок следования полей в списке.

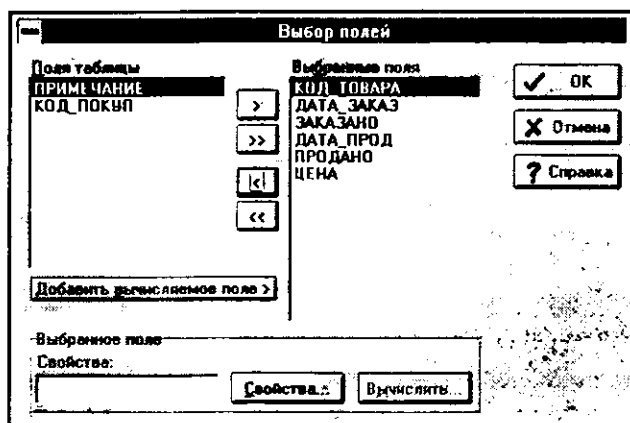


Рис. 13.8. Окно диалога “Выбор полей”.

Помимо выбора полей, которые будут размещены в объекте-таблице, данное окно диалога позволяет добавить вычисляемое поле, изменить текст заголовков, расположенных в верхней части каждого столбца таблицы, ширину столбца, нижнюю и верхнюю границы значений поля и условие проверки достоверности вводимых данных. Для этой цели служит кнопка **Свойства** в нижней части окна. Нажмите эту кнопку и для выбранного поля откроется окно диалога “Свойства поля” (рис. 13.9).

Определение свойств полей было рассмотрено в первой части книги при обсуждении отображения данных в окне таблицы. Обратим ваше внимание на тот факт, что вы можете добавить в объект-таблицу вычисляемое поле, которое может быть полезным при отображении результатов вычисления над полями таблицы.

По умолчанию в объект-таблицу включены столбцы “Зап.” и “Уд.”, которые содержат информацию о номере записи и метку удаления записи. Если данная информация не представляет для вас интерес, вы можете удалить их из объекта-таблицы, освободив таким образом место для более важной информации. Для этого в

окне свойств объекта-таблицы раскройте список **Оформление** и задайте значение (.F.) для свойств **ShowDeleted** и **ShowRecNo**.

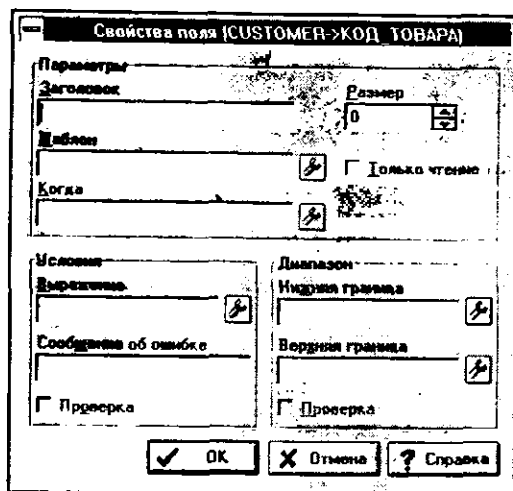


Рис. 13.9. Окно диалога “Свойства поля”.

Создание многотабличной формы

Мы рассмотрели создание многотабличной формы с помощью эксперта и дальнейшее совершенствование этой формы. Такой подход к созданию многотабличной формы является наиболее предпочтительным, однако при создании специфических форм использование эксперта может оказаться нецелесообразным. В этом случае вы должны открыть пустое окно конструктора форм, связать форму с источником данных и разместить объекты интерфейса в форме. При создании многотабличной формы в вашем распоряжении находятся все средства, используемые в однотабличной форме. Поэтому в этом разделе мы остановимся только на особенностях создания многотабличных форм.

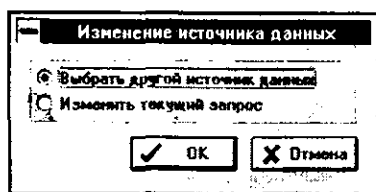


Рис. 13.10. Окно диалога “Изменение источника данных”.

Определение источника данных формы

В многотабличных формах в качестве источника данных используется многотабличный запрос. Вы можете использовать созданный ранее запрос или же создать новый запрос, не выходя из конструктора форм с помощью команды **Макет | Операции | Конструктор запросов**.

Для связывания запроса с формой используется свойство **View** формы. Если вы в процессе разработки формы решили изменить используемый в ней запрос, вновь выберите свойство **View** и нажмите кнопку **Инструмент**. На экране появится окно диалога “Изменение источника данных” (рис. 13.10), которое предоставляет вам воз-

возможность изменения текущего запроса или выбора другого источника данных. Если вы решили изменить текущий запрос, вызовите конструктор запросов, в окне которого откроется используемый в форме запрос. Вы можете модифицировать его, сохранить и выйти из конструктора запросов. После этого форма будет связана с модифицированным запросом.

Использование объекта-таблицы



Для размещения объекта-таблицы в форме откройте панель элементов, выберите на ней кнопку **Таблица** и разместите объект-таблицу в форме, определив положение объекта в форме и его размер.

Связывание объекта-таблицы с псевдонимом

При создании формы из многотабличного запроса с помощью эксперта объект-таблица определен для указанной вами подчиненной таблицы, список отображаемых полей соответствует выбранным вами полям, а заголовки столбцов совпадают с заголовками полей подчиненной таблицы.

Если вы создаете форму без помощи эксперта, то для связывания объекта-таблицы с таблицей базы данных используется свойство **Alias**. Для определения этого свойства нажмите кнопку **Инструмент** правее поля ввода значения свойства. На экране откроется окно диалога "Выбор псевдонима" (рис. 13.11).

Список **Таблицы** содержит все таблицы запроса, определенные в свойстве **View**. Таблица, выбранная из списка, связывается с текущим объектом-таблицей. Кнопка **Поиск** данного окна диалога открывает окно диалога "Поиск", в котором выбирается запрос или таблица для связывания с объектом-таблицей.

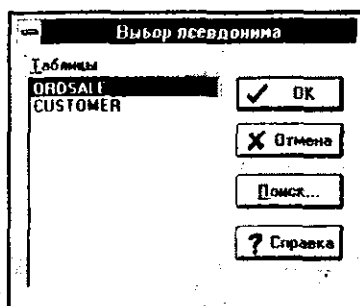


Рис. 13.11. Окно диалога "Выбор псевдонима".

Включение полей в объект-таблицу

По умолчанию в объект-таблицу включены все поля из связанной с объектом таблицы базы данных. dBASE позволяет вам не только изменять список входящих в объект-таблицу полей, но и предоставляет возможность определения свойств полей таблицы. Для этого используйте свойство **Fields** категории **Связывание данных**. При нажатии кнопки **Инструмент** данного свойства открывается окно диалога "Выбор полей", которое содержит наименования этих полей в том порядке, в котором они отображаются в объекте-таблице (рис. 13.8).

Пример 13.2. Построение многотабличной формы с вычисляемым полем

В данном примере создадим многотабличную форму, которая используется для просмотра информации о покупателях из таблицы **CUSTOMER** и сделанных ими

заказах из таблицы ORDSALE, не прибегая к помощи эксперта. В объект-таблицу, содержащую поля таблицы ORDSALE, добавим вычисляемое поле “Стоимость”, содержащее стоимость партии товаров. Определение стоимости партии осуществляется перемножением значений полей “Продано” и “Цена” из таблицы ORDSALE.

При построении многотабличной формы без помощи эксперта вы должны выполнить следующие действия:

1. Используя таблицы CUSTOMER и ORDSALE, создайте запрос. Задайте CUSTOMER в качестве главной таблицы. Установите связь между таблицами по полю “Код_покуп”. Сохраните созданный запрос.
2. Выберите **Файл | Создать | Форма** в меню главного окна dBASE. В окне диалога “Эксперт по разработке форм” (рис. 13.2) установите опцию Пустая форма и нажмите кнопку **Создать**.
3. Откройте окно свойств формы, выполнив команду **Вид | Окно свойств** или выбрав **Окно свойств** из контекстного меню формы.
4. Нажмите кнопку **Инструмент свойства View**. В окне диалога “Поиск” (рис. 13.12) выберите созданный для многотабличной формы запрос и нажмите кнопку **OK**. В поле ввода значения свойства View появляется имя запроса.

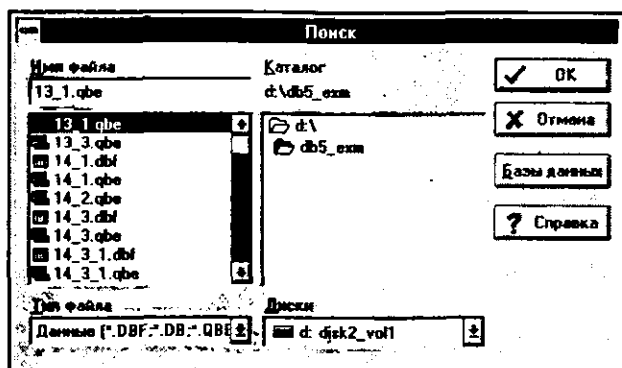


Рис. 13.12. Окно диалога “Поиск”.

5. Выберите кнопку **Поле ввода** на панели элементов и с помощью курсора мыши расположите ее в форме.
6. Для связывания созданного объекта с полем таблицы выберите его, откройте окно свойств объекта и нажмите кнопку **Инструмент свойства DataLink**.
7. В окне диалога “Выбор поля” (рис. 13.13) выберите поле “Фамилия” из списка **Поля** и нажмите кнопку **OK**. В поле ввода значения свойства **DataLink** появляется полное имя поля, включая псевдоним таблицы: CUSTOMER->ФАМИЛИЯ.
8. Повторите пункты 5 — 7 для каждого из полей таблицы (“Имя”, “Отчество”, “Код_покуп”, “Индекс”, “Адрес”, “Телефон”, “Предпр”, “Кредит”).
9. Для создания надписи “Покупатель”, описывающей назначение полей ввода “Фамилия”, “Имя” и “Отчество”, выберите кнопку **Надпись** на панели элементов и с помощью курсора мыши расположите ее в форме левее этих трех полей.

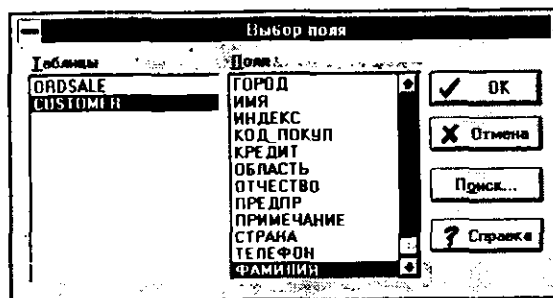


Рис. 13.13. Окно диалога "Выбор поля".

10. Откройте окно свойств надписи. В поле ввода свойства Text введите текст: «Покупатель».
11. Создайте надписи для остальных созданных полей ввода.
12. Для добавления в форму Метод-поля выберите кнопку Редактор на панели элементов и с помощью курсора мыши расположите его в форме ниже созданных полей.
13. Откройте окно свойств созданного объекта и нажмите кнопку Инструмент свойства DataLink. В открывшемся окне диалога "Выбор связи данных" (рис. 13.14) выберите из списка Тип значение "Поле", а из списка Источник данных поле "CUSTOMER->Примечание" и нажмите кнопку OK. В поле ввода значения свойства DataLink появляется полное имя поля: CUSTOMER->ПРИМЕЧАНИЕ.

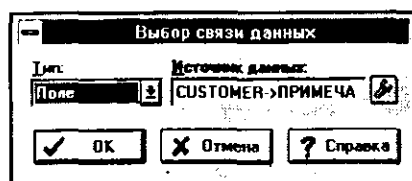


Рис. 13.14. Окно диалога "Выбор связи данных".

14. Создайте надпись для созданного объекта.
15. Для создания объекта-таблицы выберите кнопку Таблица на панели элементов и с помощью курсора мыши расположите его в нижней части формы.
16. Чтобы связать объект-таблицу с псевдонимом таблицы выберите его и перейдите в окно свойств объекта. Раскройте список Связывание данных и нажмите кнопку Инструмент свойства Alias. На экране откроется окно диалога "Выбор псевдонима". Из списка Таблицы выберите таблицу ORDSALE и нажмите кнопку OK.
17. Для того чтобы исключить из объекта-таблицы поля "Код_покуп" и "Примечание", в окне свойств нажмите кнопку Инструмент свойства Fields. Откроется окно диалога "Выбор полей". В этом окне диалога выберите по очереди указанные поля и нажатием кнопки перенесите их из списка  Выбранные поля в список Поля таблицы.

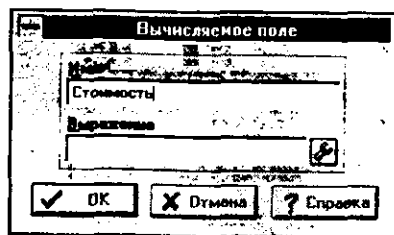


Рис. 13.15. Окно диалога “Вычисляемое поле”.

18. Для добавления в объект-таблицу вычисляемого поля “Стоимость” нажмите кнопку **Добавить вычисляемое поле**. В открывшемся окне диалога “Вычисляемое поле” (рис. 13.15) введите имя создаваемого поля в текстовое поле **Имя**. Для построения выражения для вычисляемого поля нажмите кнопку **Инструмент поля Выражение**. В окне диалога “Построение выражения” сформируйте выражение для вычисления поля `ORDSALE->ПРОДАНО*ORDSALE->ЦЕНА` и нажмите кнопку **OK**. Для возврата в окно диалога “Выбор полей” еще раз нажмите кнопку **OK**.
19. Для изменения текста заголовков полей объекта-таблицы, которые содержат в названиях символ подчеркивания, выберите курсором редактируемое поле и нажмите кнопку **Свойства** окна диалога “Выбор полей”. В открывшемся окне диалога “Свойства поля” введите новое название в поле **Заголовок**, а также установите размер поля. Нажмите кнопку **OK**. Повторите этот пункт для всех полей, которые требуют коррекции заголовка. После завершения коррекции заголовков полей и ширины столбцов нажмите кнопку **OK** для выхода из окна диалога “Выбор полей”.

Код товара	Заказано	Дата заказа	Дата продажи	Продано	Цена	Стоимость
2	5	02.04.95	12.04.95	5	610.00	3050.00

Рис. 13.16. Форма с вычисляемым полем “Стоимость”.

20. Чтобы скрыть находящиеся в объекте-таблице столбцы "Зап." и "Уд.", освободив таким образом место для более полезной информации, откройте окно свойств объекта-таблицы. В окне свойств объекта-таблицы раскройте список **Оформление** и установите значения (.F.) для свойств **ShowDeleted** и **ShowRecNo**.
21. Создаваемая форма предназначена для просмотра информации, поэтому необходимо запретить пользователю удалять, добавлять и изменять информацию. Для этого в окне свойств объекта-таблицы раскройте список **Отображение** и установите значение (.F.) для свойств **Append**, **Delete** и **Modify**.

Созданная форма, содержащая вычисляемое поле "Стоимость", представлена на рис. 13.16.



Не забывайте во время создания формы постоянно сохранять промежуточные результаты вашей работы, выполняя команду **Файл | Сохранить**.

Пример 13.3. Создание многотабличной формы с двумя объектами-таблицами

В данном примере создадим многотабличную форму, которая в отличие от предыдущей формы отображает информацию о покупателях в табличном виде.

1. Выберите **Файл | Создать | Форма** в меню главного окна dBASE. В окне диалога "Эксперт по разработке форм" установите опцию **Пустая форма** и нажмите кнопку **Создать**.
2. Установите в качестве источника данных запрос, использованный в предыдущем примере.
3. Для создания объекта-таблицы выберите кнопку **Таблица** на панели элементов и с помощью курсора мыши расположите его в верхней части формы.
4. Чтобы связать объект-таблицу с псевдонимом таблицы выберите его и перейдите в окно свойств объекта. Откройте список **Связывание данных** и нажмите кнопку **Инструмент свойства Alias**. На экране откроется окно диалога "Выбор псевдонима". Из списка **Таблицы** выберите таблицу **CUSTOMER** и нажмите кнопку **ОК**.
5. Создайте еще один объект типа **таблица**, расположите его в нижней части формы и свяжите с таблицей **ORDSALE**.
6. Определите список отображаемых полей в объектах-таблицах и их заголовки. Законченная форма представлена на рис. 13.17.
7. Сохраните форму и откройте ее для просмотра (рис. 13.18).

Создание многотабличных форм со сложными связями

dBASE позволяет создавать настолько простые и настолько сложные формы, насколько необходимо для отображения информации из вашей базы данных с различными типами отношений между ними. Ранее было отмечено, что одним из наибо-

лее распространенных типов отношений между таблицами является один-ко-многим. Вместе с тем в различных частных случаях вам может потребоваться форма для отображения данных с более сложными типами отношений.

The screenshot shows the dBASE 5.0 for Windows interface. The menu bar includes 'Файл', 'Правка', 'Вид', 'Макет', 'Процедура', 'Свойства', and 'Окно'. The toolbar contains various icons for file operations. The main window is titled 'Конструктор форм (13.7.WFM)'. It displays two tables:

КОД ПОКУП	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО
1	Алексеев	Сергей	Иванович
2	Ткаченко	Марина	Михайловна
3	Морозов	Юрий	Анатолеевич
4	Петров	Михаил	Николаевич

КОД ТОВАРА	ДАТА ЗАКАЗ	ЗАКАЗАНО	ДАТА ПРОД	ПРОДАНО	ЦЕНА
2	02.04.95		5	12.04.95	610.00

At the bottom, there is a status bar with 'Посл: 10', 'ТАБЛИЦА', and 'Порядок: 1'.

Рис. 13.17. Форма с двумя объектами-таблицами.

The screenshot shows the dBASE 5.0 for Windows interface. The menu bar includes 'Файл', 'Правка', 'Вид', 'Форма', 'Свойства', and 'Окно'. The toolbar contains various icons for form operations. The main window is titled 'Форма'. It displays two tables:

КОД ПОКУП	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО
1	Алексеев	Сергей	Иванович
2	Ткаченко	Марина	Михайловна
3	Морозов	Юрий	Анатолеевич
4	Петров	Михаил	Николаевич

КОД ТОВАРА	ДАТА ЗАКАЗ	ЗАКАЗАНО	ДАТА ПРОД	ПРОДАНО	ЦЕНА
1	02.01.95		2	12.01.95	2600.00
2	03.01.95		15	04.01.95	625.00
2	01.02.95		11	11.02.95	630.00

At the bottom, there is a status bar with 'Навигатор', 'Команды', 'Customer.dbf', and 'Стр: 2/30'.

Рис. 13.18. Просмотр формы с двумя объектами-таблицами.

Пример 13.4.

Давайте в качестве примера обратимся к таблицам CUSTOMER, ORDSALE и SUPPGOOD. Последняя из перечисленных таблиц содержит информацию о поставщиках и имеющихся у них в наличии товарах. Такая база данных позволит получить сведения о том какие товары заказаны каждым из клиентов, а также список поставщиков по каждому наименованию товара. Как вы помните, между таблицами CUSTOMER и ORDSALE существует отношение один-ко-многим. В то же время каждый из заказанных товаров может поставляться несколькими поставщиками, а каждый поставщик может предлагать к продаже товар, необходимый различным покупателям. Таким образом, тип отношения между таблицами ORDSALE и SUPPGOOD много-ко-многим. Окончательная схема данных описывается отношением один-ко-многим-ко-многим и является хорошей иллюстрацией вложенной много-табличной формы.

Для создания формы, которая позволит отобразить информацию из таблиц CUSTOMER, ORDSALE и SUPPGOOD, вы должны выполнить приведенную ниже последовательность действий.

1. В качестве исходных воспользуемся запросом и формой, разработанными в примере 13.2. Сохраните их под новыми именами, воспользовавшись командой **Файл | Сохранить как**.
2. Откройте запрос для таблиц CUSTOMER и ORDSALE. Командой меню **Запрос | Добавить таблицу** добавьте в запрос таблицу SUPPGOOD, содержащую информацию о поставщиках и имеющихся у них в наличии товарах. Свяжите таблицы ORDSALE и SUPPGOOD по полю "Код_товара" (рис. 13.19).

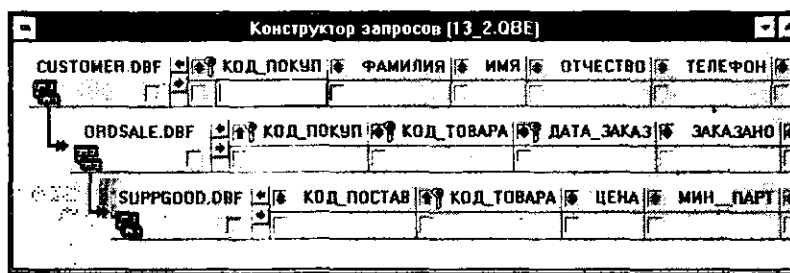


Рис. 13.19. Запрос для построения многотабличной формы.

3. Для того чтобы связать данную форму с другим запросом, откройте окно свойств формы, выполнив команду меню **Вид | Окно свойств** или выбрав **Окно свойств** из контекстного меню формы. Нажмите кнопку **Инструмент свойства View**. Откроется окно диалога "Изменение источника данных" с установленной опцией **Выбрать другой источник данных**. Нажмите кнопку **ОК** и в окне диалога "Поиск" осуществите выбор созданного в данном примере запроса.
4. Для переопределения псевдонима таблицы для объекта-таблицы выделите ее и откройте окно свойств. Раскройте список **Связывание данных** и нажмите кнопку **Инструмент свойства Alias**. На экране откроется окно диалога "Выбор псевдонима". В списке **Таблицы** выберите таблицу ORDSALE и нажмите кнопку **ОК**.
5. Из таблицы ORDSALE мы ограничимся информацией, содержащейся в полях "Код_товара", "Дата_заказа" и "Заказано". Удалите остальные поля существую-

шего объекта-таблицы. Для этого в окне свойств нажмите кнопку Инструмент свойства Fields. Откроется окно диалога "Выбор полей". Оставьте в списке Выбранные поля только поля "Код_товара", "Дата_заказа" и "Заказано". Скорректируйте при необходимости заголовки столбцов объекта-таблицы и их ширину.

6. Кроме того, удалите из формы поле "Примечание" таблицы CUSTOMER. Измените размер объекта-таблицы и придайте форме вид, показанный на рис. 13.20.

Рис. 13.20. Промежуточная форма для построения многотабличной формы.

7. Далее вам необходимо определить новый объект-таблицу. Для этого нажмите кнопку Таблица на панели элементов управления и с помощью курсора мыши расположите его в нижней правой части формы.
8. Чтобы связать объект-таблицу с псевдонимом таблицы выберите его и откройте окно свойств объекта. Раскройте список Связывание данных и нажмите кнопку Инструмент свойства Alias. На экране откроется окно диалога "Выбор псевдонима". В списке Таблицы выберите таблицу SUPPGOOD и нажмите кнопку ОК.
9. Укажите поля которые должны отображаться в определяемом объекте-таблице. В текущем примере поместите в список Выбранные поля все поля таблицы SUPPGOOD, за исключением "Код_товара" (это поле отобрано для таблицы ORDSALE). Для завершения определения нового объекта-таблицы нажмите ОК.
10. Измените заголовок нового объекта-таблицы так, чтобы он не выходил за пределы формы и установите для него свойства по своему усмотрению.
11. Определите для объектов таблиц заголовки "Заказы и продажи" и "Поставщики". Законченная форма показана на рис. 13.21.
12. Сохраните форму и откройте ее для просмотра (рис. 13.22).

dBASE 5.0 для Windows

Файл Правка Вид Макет Процедура Свойства Окно ?

Конструктор формы [13 ЗАВФМ]

ПОКУПАТЕЛЬ: Алексеев, Сергей Иванович

Телефон: 3831224-3124

Предприятие: АО Интерсервис

Кредит: 100000.00

Код: 1

Индекс: 630090

Адрес: пл. Ильича, д. 34

Заказы и продажи			Поставщики			
Код товара	Дата заказа	Заказано	Код пост.	Мин. партия	Срок поставки	Цена
	2/02/04/95	5	1	50	10	530.0000
			6	50	10	525.0000
			11	50	10	550.0000

Рис. 13.21. Законченная многотабличная форма.

dBASE 5.0 для Windows

Файл Правка Вид Форма Свойства Окно ?

ПОКУПАТЕЛЬ

ПОКУПАТЕЛЬ: Ивневский, Владимир Иванович

Телефон: 10441264-5627

Предприятие:

Кредит: 20000.00

Код:

Индекс: 252096

Адрес: пл. Мичуринская, д. 67

Заказы и продажи			Поставщики			
Код товара	Дата заказа	Заказано	Код пост.	Мин. партия	Срок поставки	Цена
4	25/03/95	50	3	50	5	175.0000
5	10/03/95	12	7	50	5	180.0000
7	15/02/95	20				

Customer.dbf 31/01/2000

Рис. 13.22. Просмотр многотабличной формы.

Очевидно, что описанный подход с использованием уже готовой формы и доопределением объекта не является единственным. Вы можете начать разработку формы "с нуля".



Процедура редактирования таблиц с использованием многотабличных форм полностью совпадает с редактированием таблиц с помощью однотобличных форм.

Сохранение целостности данных в многотабличных базах данных

При создании многотабличной формы dBASE не препятствует размещению совпадающих полей в записях подчиненной таблицы. В этом случае при редактировании данных может возникнуть следующая ситуация, для рассмотрения которой мы обратимся к таблицам CUSTOMER и ORDSALE.

При создании формы для таблиц CUSTOMER и ORDSALE в качестве совпадающего было использовано поле "Код_покуп". Предположим, что вам потребовалось изменить значение идентификатора клиента с 12 на 25. Если вы измените значение только в таблице CUSTOMER, логическая связь между таблицами будет нарушена. Теперь все значения в таблице ORDSALE для клиента с идентификатором 12 будут относиться к несуществующему покупателю. Далее, если в таблице CUSTOMER ранее отсутствовали записи о клиенте с идентификатором 25, то оказывается, что "новый" клиент не сделал ни одной покупки. По крайней мере информация о его заказах и покупках в таблице ORDSALE отсутствует. Вам придется затратить значительное время на изменение всех записей в таблице ORDSALE. При этом вы не гарантированы от ошибок.

Для того чтобы исключить возможность возникновения ситуаций, нарушающих логическую связь между таблицами, вам необходимо при создании запросов для многотабличных форм установить флажок **Целостность данных** в окне определения связи между таблицами. Когда этот флажок установлен, при попытке удалить запись главной таблицы, для которой существуют записи в подчиненной таблице, выдается предупреждение о необходимости отменить удаление или удалить и главную и все подчиненные записи (рис. 13.23). Кроме этого при добавлении записи в подчиненную таблицу значения полей подчиненной таблицы, используемые для связывания таблиц, автоматически получают значения соответствующих полей текущей записи главной таблицы. При этом поля подчиненной таблицы, используемые для связывания таблиц, доступны только для чтения.

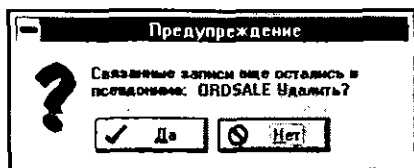


Рис. 13.23. Предупреждение о нарушении целостности данных.

Заключение

- Многотабличная форма создается на основании запроса.
- При работе над многотабличной формой в окне конструктора вы можете использовать все средства, с которыми вы познакомились в процессе создания однотобличных форм.

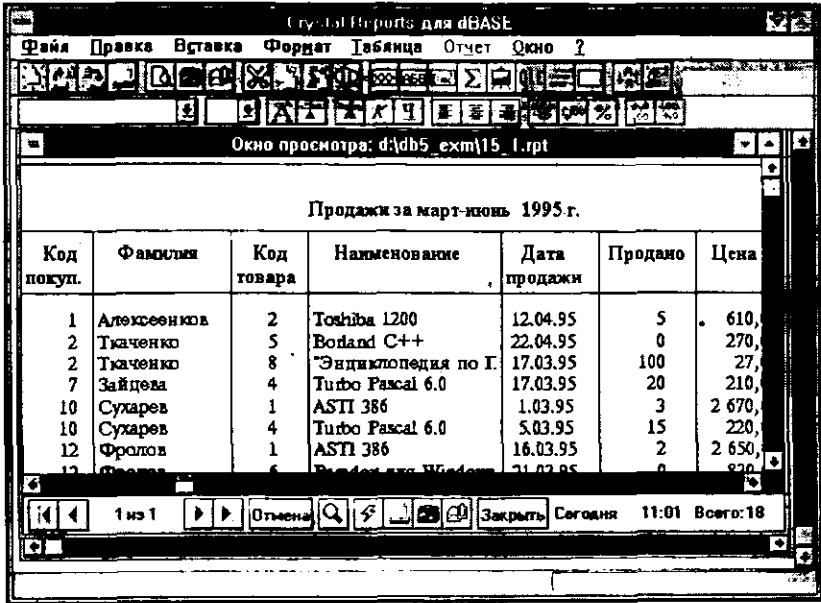
- Для построения многотабличной формы с помощью эксперта выберите **Файл | Создать | Форма** в меню главного окна dBASE с установленной опцией **Помощь эксперта** в окне диалога “Эксперт по разработке форм”.
- Создавая многотабличную форму с помощью эксперта, при определении типа макета представления данных в многотабличной форме установите опцию **С подчиненной таблицей**.
- Для создания объекта интерфейса объект-таблица используется кнопка **Таблица** на панели элементов.
- При создании формы для связывания объекта-таблицы с псевдонимом используется свойство **Alias**.
- Для добавления в объект-таблицу вычисляемого поля используется кнопка **Добавить вычисляемое поле** окна диалога “Выбор полей”. Для открытия этого окна в окне свойств объекта-таблицы нажмите кнопку **Инструмент свойства Fields**.
- Для изменения текста заголовков и ширины столбцов объекта-таблицы нажмите кнопку **Свойства** окна диалога “Выбор полей”. В открывшемся окне диалога “Свойства поля” введите новое название в поле **Заголовок** и установите размер поля. Нажмите кнопку **ОК**.
- Чтобы скрыть находящиеся в объекте-таблице столбцы “Зап.” и “Уд.” откройте окно свойств объекта-таблицы. В окне свойств объекта-таблицы откройте список **Оформление** и задайте значение (.F.) для свойств **ShowDeleted** и **ShowRecNo**.
- Для сохранения целостности данных в многотабличной форме необходимо установить флажок **Целостность данных** в окне определения связи между таблицами при создании запроса.

Глава 14. Отчеты для связанных таблиц и группировка данных

В настоящей главе вы познакомитесь с созданием отчетов, в которых представлены данные из нескольких таблиц. Кроме того, мы обсудим вопросы, связанные с группировкой данных в отчетах. Эта техника может быть использована при подготовке как многотабличных, так и однотабличных отчетов.

Отчеты для многотабличных баз данных

В главе 9 вы научились создавать отчеты для отображения данных из одной таблицы. Теперь мы воспользуемся этими методами при подготовке отчетов для нескольких связанных таблиц. В качестве примера отобразим в отчете список клиентов и сведения о приобретенных ими товарах. Для того чтобы подготовить единый отчет, вам необходимо включить в него информацию из связанных таблиц CUSTOMER, ORDSALE и GOODS. Для решения поставленной задачи мы создадим отчет, основанный на таблице CUSTOMER, в который включим данные о клиентах и товарах из таблиц ORDSALE и GOODS, соответственно. На рис. 14.1 показан фрагмент отчета. Отметьте, что в отчете представлены не только идентификатор покупателя, но и его фамилия (таблица CUSTOMER), а также включено наименование товара (таблица GOODS).



Crystal Reports для dBASE

Файл Правка Вставка Формат Таблица Отчет Окно ?

Окно просмотра: d:\db5_exam\15_1.rpt

Продажи за март-июнь 1995 г.

Код покуп.	Фамилия	Код товара	Наименование	Дата продажи	Продано	Цена
1	Алексеев	2	Toshiba 1200	12.04.95	5	610,
2	Ткаченко	5	Borland C++	22.04.95	0	270,
2	Ткаченко	8	Энциклопедия по Г.	17.03.95	100	27,
7	Зайцева	4	Turbo Pascal 6.0	17.03.95	20	210,
10	Сухарев	1	ASTI 386	1.03.95	3	2 670,
10	Сухарев	4	Turbo Pascal 6.0	5.03.95	15	220,
12	Фролов	1	ASTI 386	16.03.95	2	2 650,
12	Фролов	6	Borland для Windows	21.03.95	0	820,

1 из 1 Отмена Поиск Закрыть Сегодня 11:01 Всего:18

Рис. 14.1. Отчет, содержащий данные о клиентах и приобретенных ими товарах.

Определение запроса для многотабличного отчета

На первом этапе создания многотабличного отчета вам необходимо определить источник данных, в качестве которого теперь будет использоваться не отдельная таблица, а выборка из нескольких связанных таблиц. Для определения выборки необходимо создать многотабличный запрос. Прежде чем приступить к созданию источника

данных в dBASE, определите главную и подчиненные таблицы, а также соотношение между ними. При этом необходимо принимать во внимание, что количество строк, напечатанных в отчете, будет определяться числом записей в выборке.

В рассматриваемом примере в качестве главной таблицы выберем CUSTOMER. Напомним, что таблицы CUSTOMER и ORDSALE связаны соотношением один-ко-многим, а ORDSALE и GOODS — соотношением один-к-одному.

Пример 14.1. Создание многотабличного отчета

Для создания многотабличного отчета в Crystal Reports вы должны прежде всего создать многотабличный запрос и использовать его для создания отчета. Давайте создадим отчет, фрагмент которого в окне просмотра представлен на рис. 14.1.

1. Выберите **Файл | Создать | Запрос** в меню главного окна dBASE.
2. В окне диалога “Выбор таблицы” выберите таблицу CUSTOMER, которая будет являться главной. На экране появится окно диалога конструктора запросов, содержащее бланк запроса со структурой выбранной таблицы.
3. Выберите поля “Код_покуп” и “Фамилия”, которые должны присутствовать в запросе из таблицы CUSTOMER.
4. Добавьте в запрос таблицу ORDSALE, для чего нажмите кнопку панели инструментов **Добавить таблицу**. Бланк запроса таблицы ORDSALE появится в окне конструктора запросов.
5. Выберите поля “Код_товара”, “Дата_прод”, “Продано” и “Цена”, которые должны присутствовать в запросе из таблицы ORDSALE.
6. Аналогичным образом добавьте в запрос таблицу GOODS и выберите поле “Наим”.
7. Для создания связи между таблицами CUSTOMER и ORDSALE установите указатель на значок таблицы CUSTOMER, нажмите кнопку мыши и, удерживая ее нажатой, переместите указатель к значку таблицы ORDSALE. На экране откроется окно диалога “Установление связи”, в котором определите связь между таблицами.
8. Аналогичным образом создайте связь между таблицами ORDSALE и GOODS. В этой связи главной будет являться таблица ORDSALE.
9. Поскольку нам нужен запрос, содержащий сведения о покупателях, покупавших товары с марта по июнь 1995 г., для поля “Дата_прод” таблицы ORDSALE необходимо ввести условие:

`>={01.03.95}, <{01.07.95}`

Окно конструктора запроса с подготовленным запросом представлено на рис. 14.2.

10. Сохраните запрос под именем SALES.QBE и закройте окно конструктора запросов.

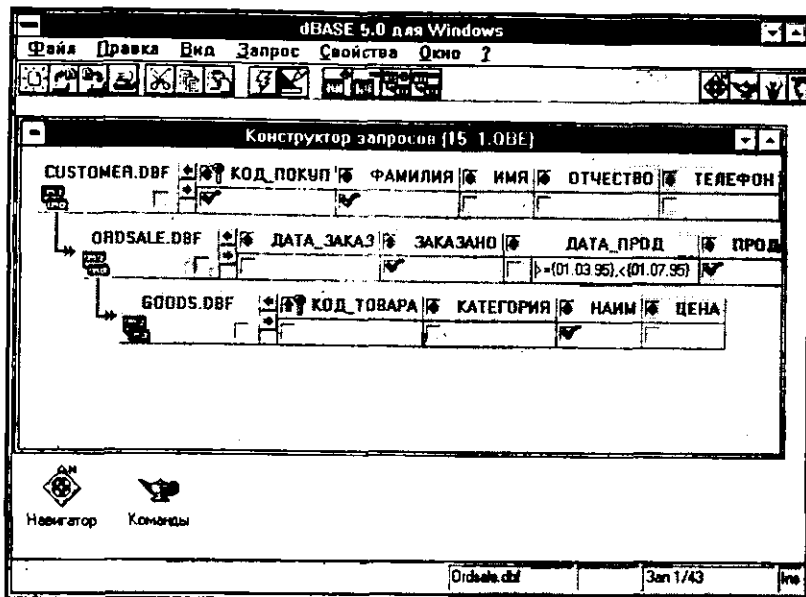


Рис. 14.2. Окно конструктора запросов с подготовленным запросом.

11. Теперь приступим непосредственно к созданию отчета, используя в качестве источника данных созданный запрос.
12. Выберите **Файл | Создать | Отчет** или нажмите мышью кнопку **Открыть** на панели инструментов, после чего выберите **Отчет**.
13. В открывшемся окне диалога "Выбор таблицы" выберите в качестве источника данных запрос **SALES.QBE** и нажмите **ОК**. В результате на экране появится окно конструктора отчетов Crystal Reports с открытым окном диалога "Вставка поля базы данных", которое содержит список полей, включенных нами в запрос (рис. 14.3).
14. Используя кнопку **Вставить**, перенесите все поля источника данных в отчет.
15. Для закрытия окна диалога нажмите кнопку **Готово**.
16. Для наглядности отчета измените надписи "Код_товара", "Код_покуп", "Дата_прод" и расположите их на двух строках.
17. Выворачивайте расположение полей и промежутки между ними.
18. Вставьте в отчет рамку и разделительные линии.
19. Поскольку код покупателя и код товара представляют целые числа, измените формат представления данных. Для этого выполните команду **Изменить формат** и укажите в окне диалога "Числовой формат", что дробная часть отсутствует.
20. В заключение измените начертание шрифтов всех заголовков с обычного на жирный. Окончательный формат отчета показан на рис. 14.4.

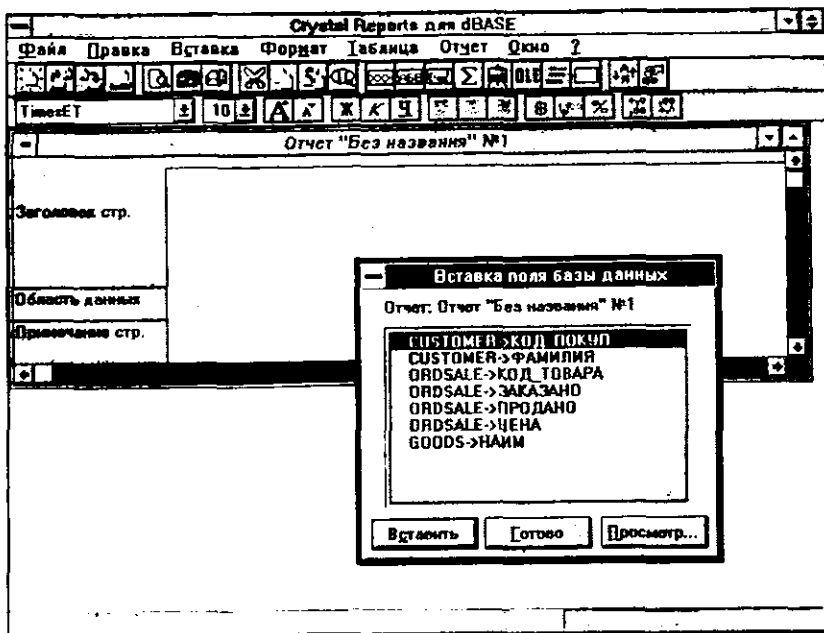


Рис. 14.3. Окно конструктора отчетов.

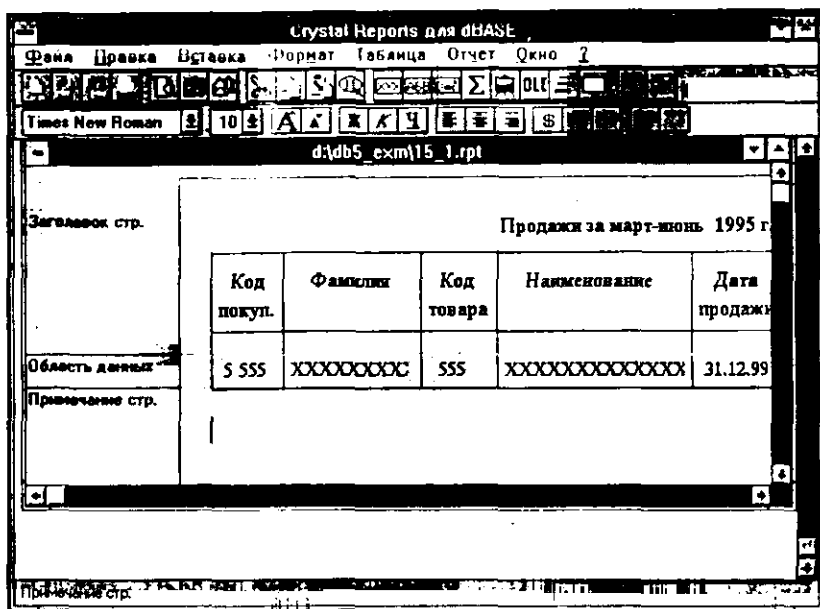


Рис. 14.4. Отчет, полученной из таблиц CUSTOMER, ORDSALE и GOODS.

Группировка данных в отчетах

Команда Раздел группы из меню Вставка в окне конструктора отчетов позволяет вам сгруппировать данные в логические блоки. Например, вы можете объединить в одну группу записи, которые имеют одинаковые значения в указанном поле.

Кроме того, вы можете вычислить и отобразить в отчете групповые значения по каждой группе, а также выполнить сортировку данных в пределах каждой группы.

Для демонстрации группировки данных в отчетах в качестве базового мы воспользуемся отчетом, разработанным в примере 14.1. Обратите внимание на отчет, представленный на рис. 14.1. Код покупателя и фамилия повторяются в нескольких строках. Такая ситуация достаточно типична, поскольку один покупатель может купить несколько разных товаров.

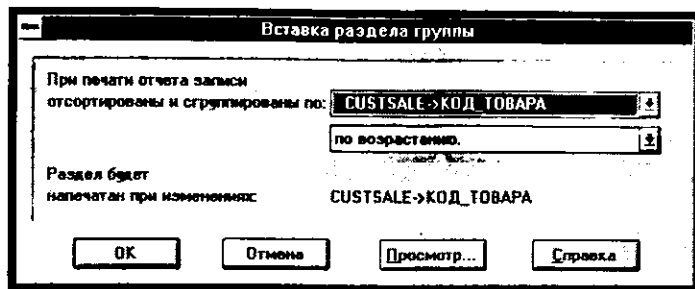


Рис. 14.5. Окно диалога “Вставка раздела группы”.

Определение группы

Для того чтобы определить группу, выберите опцию **Вставка | Раздел группы**. В результате dBASE откроет окно диалога “Вставка раздела группы” (рис. 14.5).

Данное окно диалога позволяет сгруппировать записи на основании значений полей таблицы или полей отчета. Для выбора поля группировки нажмите верхнюю кнопку раскрытия списка. В открывшемся списке находятся все поля источника данных, с которым вы работаете, и поля, помещенные в отчет. Выберите значение, по которому предполагаете сгруппировать данные в отчете.

Вторая кнопка раскрытия списка позволяет определить как будут сортироваться данные: по возрастанию или по убыванию. По умолчанию выбирается сортировка по возрастанию. Для изменения порядка сортировки нажмите кнопку раскрытия списка и выберите нужный порядок сортировки.

Если выбранное поле имеет тип даты или логический тип, то в окне диалога появляется дополнительный список, позволяющий выбрать специальные условия группировки:

- Для полей типа даты список содержит перечень периодов дат, обычно используемых для группировки данных. При выборе поля типа даты конструктор отчетов вначале сортирует данные по дате, а затем разбивает данные по группам в зависимости от выбранного условия.
- Для логических полей конструктор отчетов предоставляет в ваше распоряжение ряд условий, при выполнении которых будет производиться группировка.

При введении группы в отчет вы добавляете новую область, называемую областью группы. Область группы располагается вне области данных. Кроме того, область группы может располагаться внутри одной или нескольких других областей других групп. Взаимное расположение областей групп мы обсудим ниже при изучении вложенных групп.

Если вы используете группировку по полю, записи в отчете будут отсортированы таким образом, чтобы в каждую группу входили записи с одним и тем же значением в указанном поле. Все группы будут упорядочены в соответствии со значениями этого

поля. Таким образом данное средство может быть использовано для печати отсортированного отчета. При группировке по полю вы можете вычислить групповые значения по каждой группе и отобразить результаты в отчете.

Пример 14.2.

Давайте воспользуемся группировкой по полю для того, чтобы сгруппировать в отчете записи о всех покупках каждого клиента.

1. Добавим в запрос SALES.QBE, используемый в качестве источника данных отчета, поля "Имя" и "Отчество".
2. Откройте в окне конструктора отчет, созданный в предыдущем примере.
3. Выберите Вставка | Раздел группы.
4. В открывшемся окне диалога "Вставка раздела группы" нажмите верхнюю кнопку раскрытия списка и из списка полей источника данных, которые могут быть использованы для группировки данных, выберите поле "Код_покуп".
5. По умолчанию данные будут группироваться по возрастанию значения данного поля. В данном случае выберем предлагаемый порядок группировки. Для завершения формирования группы нажмите кнопку ОК.
6. В отчет добавилась область группы. Отметьте, что каждая запись в группе содержит один и тот же код покупателя и фамилию. Значительно лучше будет отобразить информацию о клиенте в заголовке группы. Перенесите в него поля, которые должны печататься один раз для каждой группы: "Код_покуп" и "Фамилия", а также добавьте поля "Имя", "Отчество", которые ранее отсутствовали в отчете.
7. Для того чтобы шапка таблицы, содержащая заголовки полей, печаталась для каждой группы, расположите ее также в области группы.

Окончательный вид отчета показан на рис. 14.6, а на рис. 14.7 — сам отчет в окне просмотра. Как вы можете видеть, записи в отчете упорядочены по возрастанию идентификаторов покупателей.

Пример 14.3.

Теперь внесем дальнейшие улучшения в формат отчета и отобразим групповые значения по каждой группе — общую стоимость товаров, приобретенных каждым из клиентов. Для вычисления итоговой стоимости вам необходимо вычислить сумму значений

Продано*Цена

по всем записям.

1. Откройте отчет, созданный в примере 14.2.
2. Для создания вычисляемого поля выберите команду Вставка | Поле выражения.

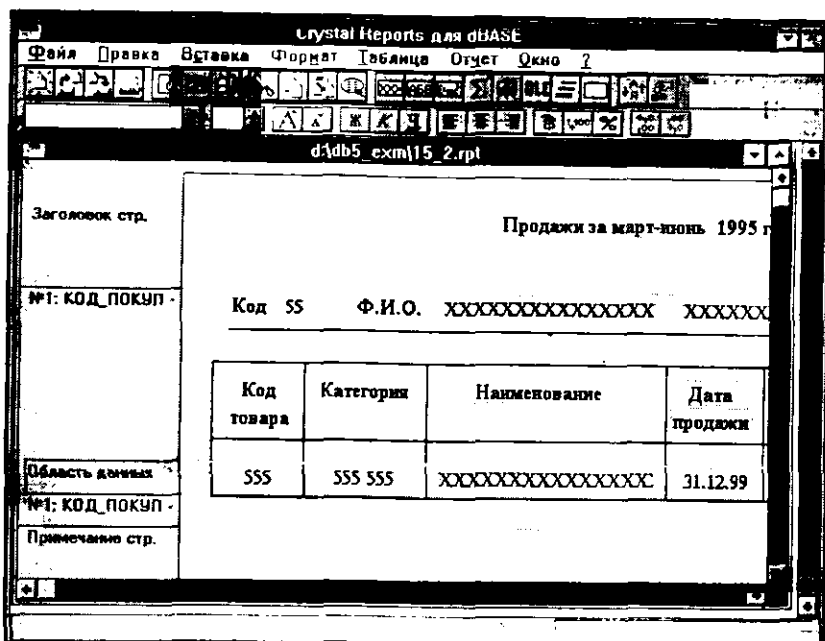


Рис. 14.6. Отчет с группой "Код_покуп".

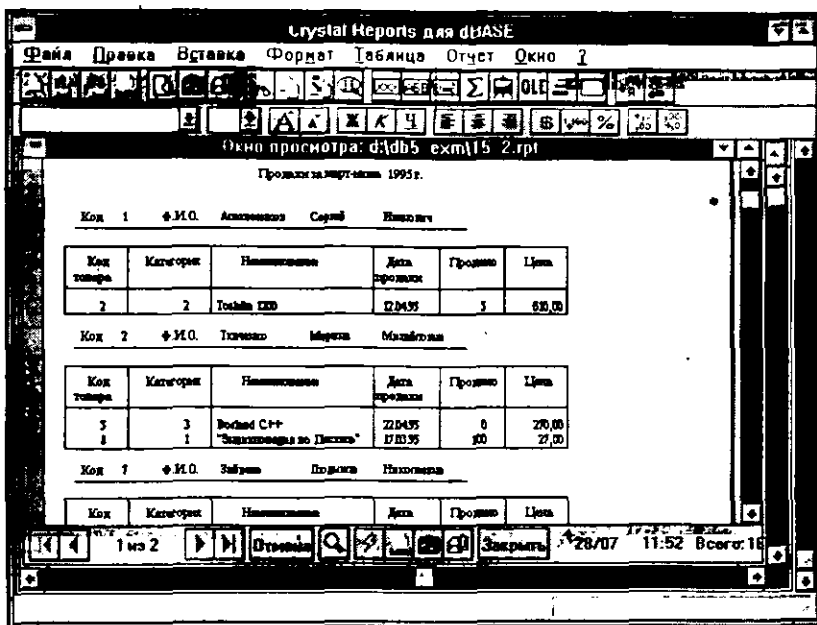


Рис. 14.7. Отчет в окне просмотра, сгруппированный по идентификаторам покупателей.

- В открывшемся окне диалога "Вставка выражения" в текстовое поле Имя выражения введите имя выражения, которое будет использоваться для ссылки на это выражение. Нажмите кнопку ОК.

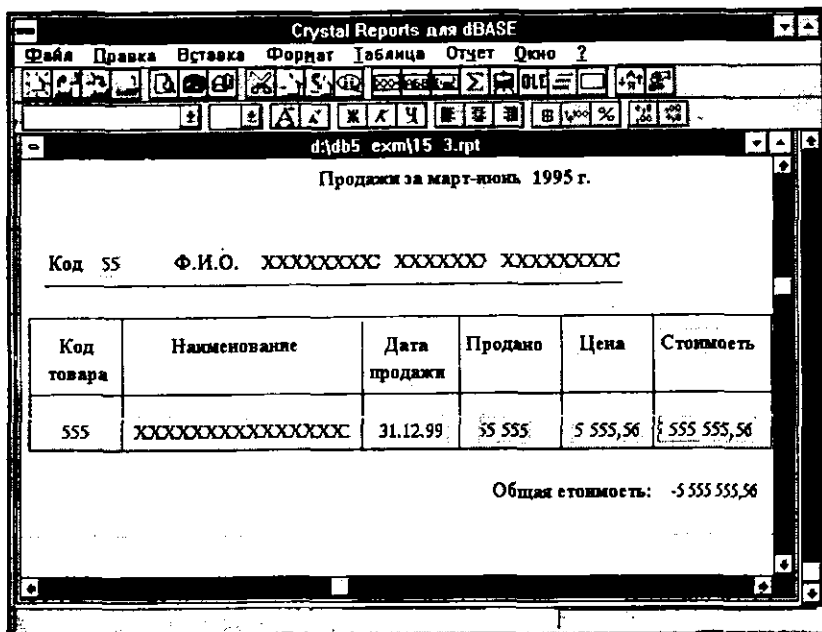


Рис. 14.8. Окончательный вид отчета с группой "Код_покуп".

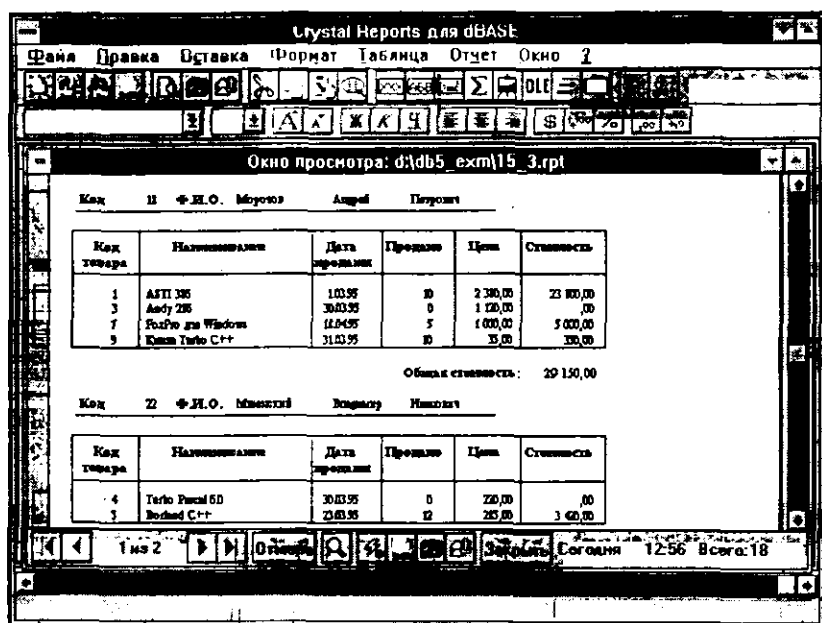


Рис. 14.9. Отчет, с итоговыми значениями, сгруппированный по идентификаторам покупателей.

- Для вычисления стоимости товаров, проданных в каждой партии, вам нужно ввести вычисляемое поле с выражением:

Продано*Цена

5. Сформируйте требуемое выражение для поля. Определение вычисляемого поля завершите нажатием ОК.
6. Создайте надпись для вычисляемого поля "Стоимость". Для этого воспользуйтесь командой меню Вставка | Надпись и расположите надпись над созданным полем.
7. Для вычисления общей стоимости товаров, приобретенных каждым из клиентов, выделите созданное вычисляемое поле "Стоимость" и выполните команду меню Вставка | Поле итога | Групповая операция. Из предложенного списка операций выберите групповой оператор Сумма. Поместите созданное поле в примечание группы.
8. Для созданного поля создайте надпись "Общая стоимость" и расположите ее слева от поля.

На рис. 14.8 и 14.9 показаны окончательный вид отчета и фрагмент отчета в окне просмотра, соответственно.

В предыдущем примере группировка данных осуществлялась в соответствии со значениями числового поля "Код_покуп". Вы можете сгруппировать данные по значениям поля любого типа.

Пример 14.4.

В качестве примера сгруппируем записи в отчете по значениям текстового поля "Фамилия". Для этого используйте отчет, разработанный в примере 14.3. Сохраните его под другим именем, воспользовавшись командой Файл | Сохранить как. Затем откройте окно свойств области группы и выберите команду Изменить группу. Откроется окно диалога "Изменение раздела группы" (рис. 14.10). Из списка полей таблицы вместо поля "Код_покуп" выберите поле "Фамилия" и нажмите кнопку ОК.

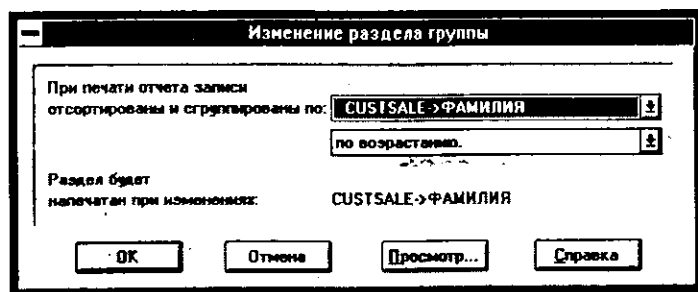


Рис. 14.10. Окно диалога "Изменение раздела группы".

На рис. 14.11 показано окно конструктора отчетов с отчетом, в котором для группировки использовалось текстовое поле "Фамилия". Фрагмент отчета показан на рис. 14.12.

Вы видите, что теперь записи о клиентах упорядочены в алфавитном порядке, а не по значениям их идентификаторов. Просмотрев отчет полностью, вы сможете убедиться, что по сравнению с предыдущим примером изменился только порядок следования групп записей. Порядок записей в каждой из групп и содержание каждой записи остались без изменения.

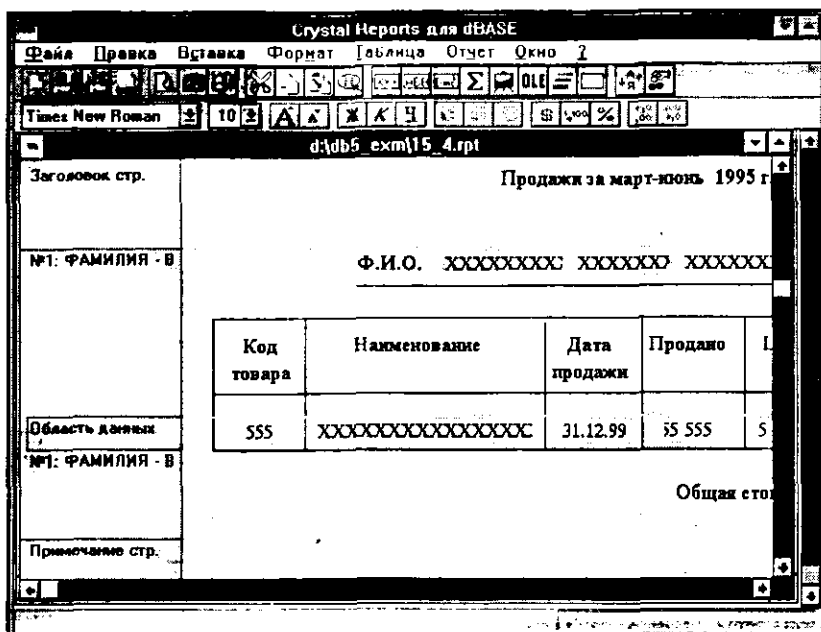


Рис. 14.11. Отчет с группой "Фамилия".

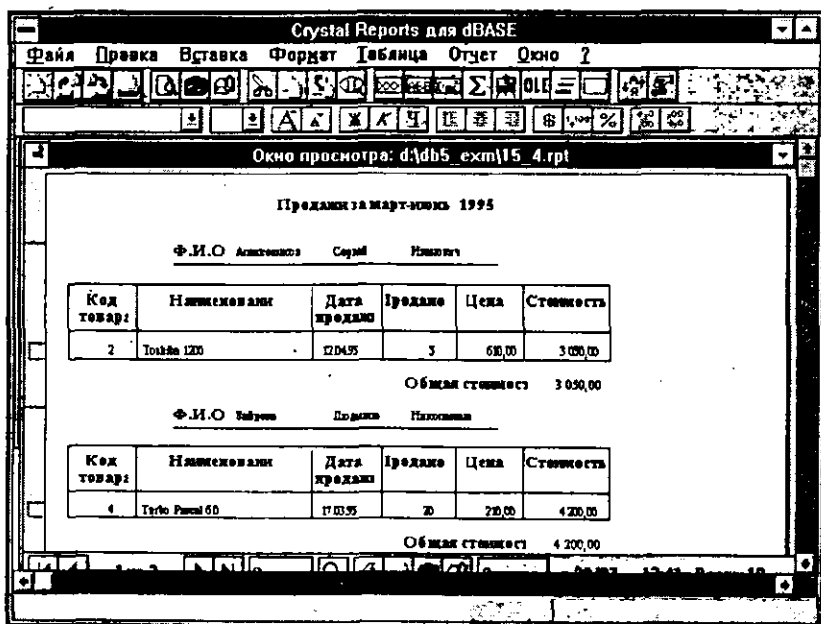


Рис. 14.12. Просмотр отчета, сгруппированного по фамилиям покупателей.

Группировка по диапазону значений для полей типа даты

При группировке по полям дат dBASE позволяет вам группировать записи в соответствии с диапазоном значений в указанном поле. При выборе для группировки

поля дат на экране появится окно диалога “Вставка раздела группы”, которое имеет вид, представленный на рис. 14.13.

В зависимости от выбранной вами опции dBASE будет группировать записи, даты в которых относятся к одному дню, неделе, двум неделям, месяцу, кварталу, полугодию или году. При группировке записи в каждой из групп будут упорядочены в соответствии с датой в указанном поле.

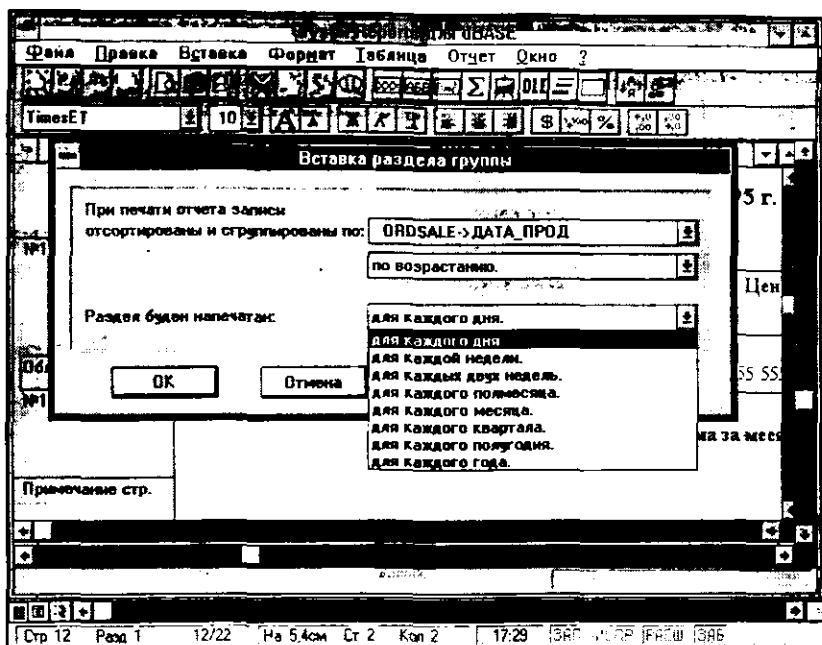


Рис. 14.13. Окно диалога “Вставка раздела группы” при группировке по диапазону дат.

Группировка по диапазону дат может оказаться очень полезной для отображения в отчете групповых значений за указанный период времени.

Пример 14.5.

В данном примере вам предстоит создать отчет, в котором объединены в группы все выполненные заказы по каждому месяцу, и отобразить в отчете суммарную стоимость месячных продаж.

1. Откройте отчет, разработанный в примере 14.4. Сохраните отчет под другим именем, воспользовавшись командой **Файл | Сохранить как**.
2. Откройте меню свойств области группы и выберите команду **Изменить группу**. В результате откроется окно диалога “Изменение раздела группы”. Из списка полей таблицы вместо поля “Фамилия” выберите поле “Дата_прод”.
3. В окно диалога “Изменение раздела группы” добавился список **Раздел будет напечатан** со значениями диапазона изменения данных (рис. 14.13). Нажмите кнопку раскрытия добавившегося списка и выберите значение интервала “месяц”. Нажмите кнопку **OK**.

4. В первом столбце удалите поле "Код_покуп" и расположите в нем поле "Фамилия", перенеся его из области группы.
5. Измените надпись "Общая стоимость" на "Сумма за месяц".

Теперь окно конструктора будет иметь вид, показанный на рис. 14.14. Фрагмент отчета в окне просмотра показан на рис. 14.15.

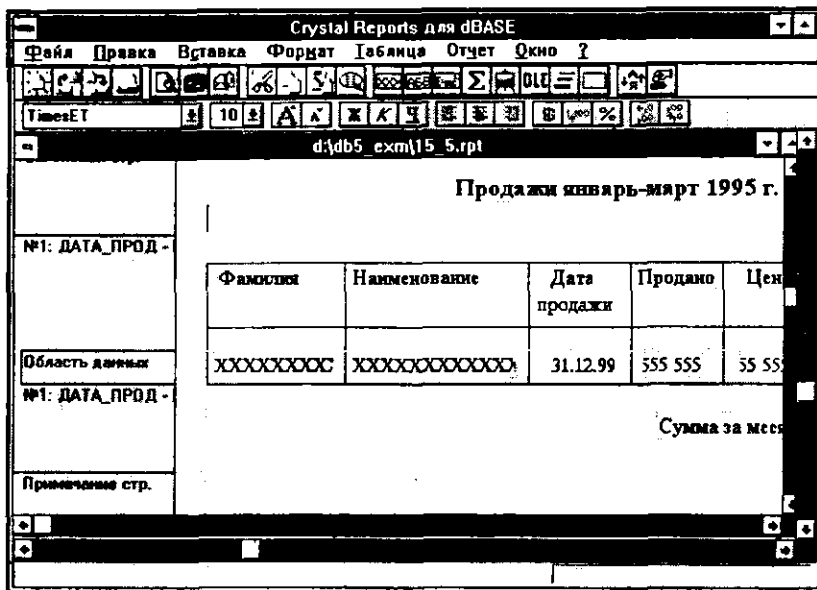


Рис. 14.14. Группировка записей по месяцам.

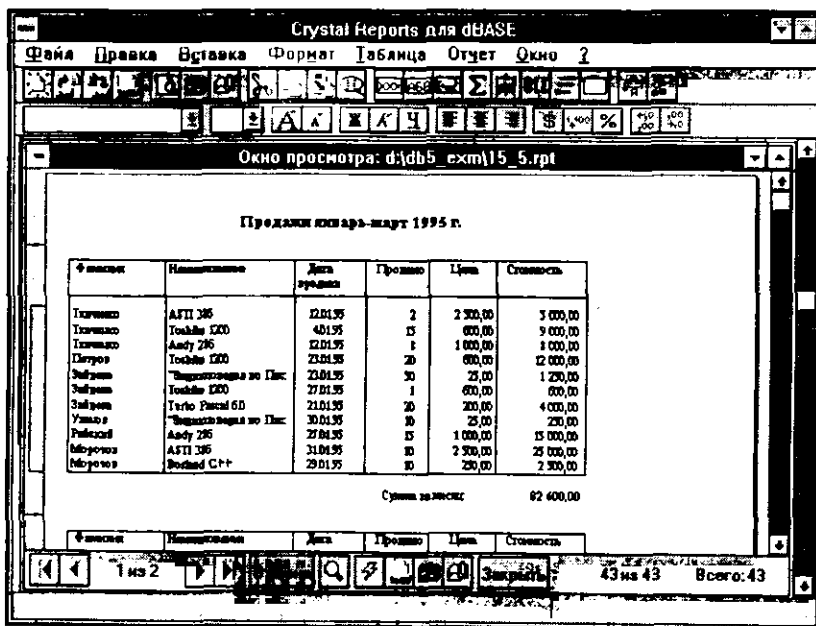


Рис. 14.15. Просмотр отчета, сгруппированного по месяцам.

Вложенные группы

Конструктор отчетов позволяет разместить в одном отчете несколько областей групп. При добавлении в отчет новой группы ее размещение осуществляется автоматически. Процедура создания вложенных групп полностью аналогична вашим действиям при определении первой группы.

При наличии в отчете нескольких групп, dBASE сортирует записи в пределах каждой из групп. При этом первыми будут отсортированы записи во внешней (первой) группе, затем записи во второй группе и т.д.

В качестве примера создадим отчет, в котором записи сгруппированы по фамилиям покупателей, а внутри каждой группы еще и по покупкам, сделанным клиентом в течение каждого месяца. Такой отчет позволит вам проанализировать сведения о товарах, покупаемых клиентом в различные месяцы.

Вы можете использовать групповые поля для отображения в отчете нескольких уровней групповых значений по всем или некоторым из вложенных групп.

Пример 14.6. Создание вложенных групп

Теперь вам предстоит создать отчет, в котором отображена стоимость покупок, сделанных клиентом в течение каждого месяца, а также суммарную стоимость всех покупок клиента за все месяцы.

1. В качестве исходного воспользуйтесь отчетом, показанным на рис. 14.12. Этот отчет уже содержит группу по полю "Код_покуп". Откройте данный отчет в окне конструктора отчетов.
2. Определите вторую группу, основанную на значениях поля "Дата_прод". Отметьте, что dBASE автоматически разместит эту группу внутри существующей группы.
3. В предыдущем отчете заголовок таблицы печатался для каждого покупателя. В данном отчете желательно отобразить заголовок для каждого месяца. Для этого переместите заголовок таблицы из заголовка группы "Код_покуп" в заголовок новой группы.
4. Исходный отчет уже содержит поле для отображения группового значения по вычисляемому полю "Стоимость", которое помещено в примечании группы поля "Код_покуп" после текста "Общая сумма". Теперь поместите аналогичное поле в примечании группы по дате продажи.

Окончательный вид отчета в конструкторе отчетов представлен на рис. 14.16. Если вы выведете отчет на экран, он будет иметь вид, показанный на рис. 14.17. Отметьте, что в отчете присутствуют уровни групповых значений: по месяцам и по каждому из покупателей.

Удаление групп из отчета

Для удаления группы из отчета выберите опцию **Удалить** из меню свойств группы.



Будучи удаленной, после сохранения отчета группа не может быть восстановлена. Если вам потребуется восстановить группу, вы должны будете создать ее заново. Поэтому соблюдайте осторожность при удалении групп из отчета.

Crystal Reports для dBASE - [d:\db5_exm\15_6.rpt]

Файл Правка Вставка Формат Таблица Отчет Окно ?

TimesET 1 10

№1: КОД ПОКУП.

Покупатель: XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX

№2: ДАТА ПРОД.

Наименование	Дата продажи	Продано	Цена	Стоим
XXXXXXXXXXXXXXXXXX	31.12.99	555 555	55 555,56	5 555

Область данных

№2: ДАТА ПРОД.

№1: КОД ПОКУП.

Примечание стр.

Сумма за месяц: -555

Общая сумма: -555

Примечание стр.

Рис. 14.16. Отчет, сгруппированный по покупателям и месяцам.

Crystal Reports для dBASE - [Окно просмотра: d:\db5_exm\15_6.rpt]

Файл Правка Вставка Формат Таблица Отчет Окно ?

Покупатель: Зайцева Дарья Иванова

Наименование	Дата продажи	Продано	Цена	Стоимость
Toshiba 1200	27.01.95	1	600,00	600,00
Turbo Pascal 6.0	21.01.95	20	200,00	4 000,00

Сумма за месяц: 4 600,00

Наименование	Дата продажи	Продано	Цена	Стоимость
ASTI 386	17.02.95	1	2 500,00	2 500,00
Andy 286	19.02.95	1	1 000,00	1 000,00

Сумма за месяц: 3 500,00

Общая сумма: 8 100,00

2 из 4 Отмена Закрыть 31/07 10:48 Ввод: 43

Рис. 14.17. Фрагмент отчета в окне просмотра, сгруппированного по покупателям и месяцам.

Переопределение групп

Опция Изменить группу из меню свойств группы позволяет вам изменить способ группировки. Вы можете изменить поле или диапазон, которые использовались при группировке.

В меню области группы отчета выберите **Изменить группу**. dBASE откроет уже знакомое вам окно диалога “Изменение раздела группы”, с помощью которого можно задать группировку по нужному полю и при необходимости диапазон изменения даты.

Перекрестные таблицы

Таблицы dBASE могут содержать значительный объем информации. При наличии в таблице нескольких тысяч или даже сотен записей анализ текстовых и числовых данных, содержащихся в одной или нескольких таблицах, на первый взгляд представляется трудно выполнимой задачей. Однако существуют два мощных средства, позволяющих достаточно просто представить большие объемы данных в удобном для дальнейшего анализа виде — диаграммы и перекрестные таблицы.

Диаграмма является одним из наиболее мощных и наглядных средств, предназначенных для отображения данных из таблиц в виде графиков. Представление информации в виде диаграмм наиболее удобно для анализа сравнительных характеристик, тенденций изменения данных и т.д.

При представлении больших объемов информации в графическом виде существует серьезная проблема. Если таблица не подготовлена к построению диаграммы специальным способом, то, как правило, данные в таблице представлены в виде нескольких составляющих, расположенных в большом количестве записей. Если вы попытаетесь построить диаграмму для такой таблицы обычным способом, то она будет иметь слишком сложный и загроможденный вид, из которого трудно будет извлечь полезную информацию. Однако dBASE позволяет преобразовать информацию к виду, наиболее приемлемому для построения диаграмм. Для этой цели вы можете воспользоваться перекрестными таблицами.

Информация в перекрестной таблице группируется в соответствии со значениями одного или нескольких полей и фактически приводится к виду, аналогичному электронной таблице. Если на диаграмме информация представляется в графическом виде, то в перекрестной таблице она отображается в виде чисел. В dBASE отсутствуют средства построения диаграмм, но в среде Windows имеется целый ряд электронных таблиц, которые вы можете использовать для построения диаграмм из подготовленных в dBASE данных. Построение диаграмм будет обсуждаться в главе 16, а сейчас обратимся к созданию перекрестных таблиц.

Создание перекрестных таблиц

Информация в перекрестной таблице представляется в виде, похожем на электронную таблицу. Каждому столбцу и каждой строке соответствует поле исходной таблицы, при этом эти поля должны быть различными. При создании перекрестной таблицы dBASE выполняет вычисления для каждой из пар строка/столбец исходной таблицы и помещает результат в перекрестную таблицу. Например, с помощью перекрестной таблицы вы можете определить сколько клиентов проживают в каждом городе или как распределяются проданные товары среди покупателей. Очевидно, что каждый покупатель может покупать один и тот же товар несколько раз, поэтому получить такие сведения, не прибегая к перекрестной таблице было бы затруднительно.

Как мы уже сказали, перекрестная таблица похожа на электронную таблицу, однако имеет перед последней одно явное преимущество. Вам не нужно вручную заполнять каждую ячейку перекрестной таблицы, предварительно определив ее значение. dBASE автоматически разбивает данные на группы в соответствии с одним или несколькими факторами и выполняет групповые вычисления над данными, включенными в одну группу. После чего информация, полученная в результате групповых вычислений, сортируется и заносится в перекрестную таблицу.

В dBASE для построения перекрестных таблиц используется конструктор отчетов Crystal Reports, который позволяет вам не только просмотреть и печатать построенную таблицу, но и сохранить ее в виде электронной таблицы.

При работе с dBASE вы можете использовать одномерные и многомерные перекрестные таблицы. Прежде чем приступить непосредственно к созданию перекрестной таблицы вы должны определиться какой из типов перекрестных таблиц в наибольшей степени подходит к выполнению текущей задачи.

Одномерные перекрестные таблицы

Одномерная перекрестная таблица представляет собой простейший тип перекрестной таблицы, с помощью которой вы можете проанализировать распределение данных одного типа по одному фактору.

Беларусь	Казахстан	Латвия	Россия	Украина
2	1	1	20	6

The status bar at the bottom shows navigation controls, '1 из 1', 'Отмена', search icons, 'Закрыть', and system information: 'Сегодня 21:38 Всего: 30 100%'.

Рис. 14.18. Горизонтальная одномерная перекрестная таблица.

Пример 14.7.

В качестве примера создадим перекрестную таблицу, отображающую распределение клиентов по странам (рис. 14.18). Исходная информация содержится в таблице CUSTOMER.

1. Выберите **Файл | Создать | Перекрестная таблица**. На экране появится окно диалога "Выбор таблицы", с помощью которого выберите таблицу CUSTOMER. В результате dBASE вызовет конструктор отчетов и откроет окно диалога "Перекрестная таблица" (рис. 14.19).
2. На этом этапе укажите какое поле следует использовать в качестве заголовков столбцов. Выберите поле, значения которого будут использованы в качестве заголовков столбцов, и нажмите кнопку **Добавить столбец**. В текущем примере ука-

жите "Страна". В области "Столбцы" описания перекрестной таблицы появится наименование поля CUSTOMER->Страна.

Поскольку мы создаем горизонтальную одномерную таблицу, вам не нужно определять поле, используемое в качестве заголовков строк.

3. Для того чтобы определить поле, по которому будет выполняться группировка значений, выберите поле "Код_покуп" и нажмите кнопку Поле итога.

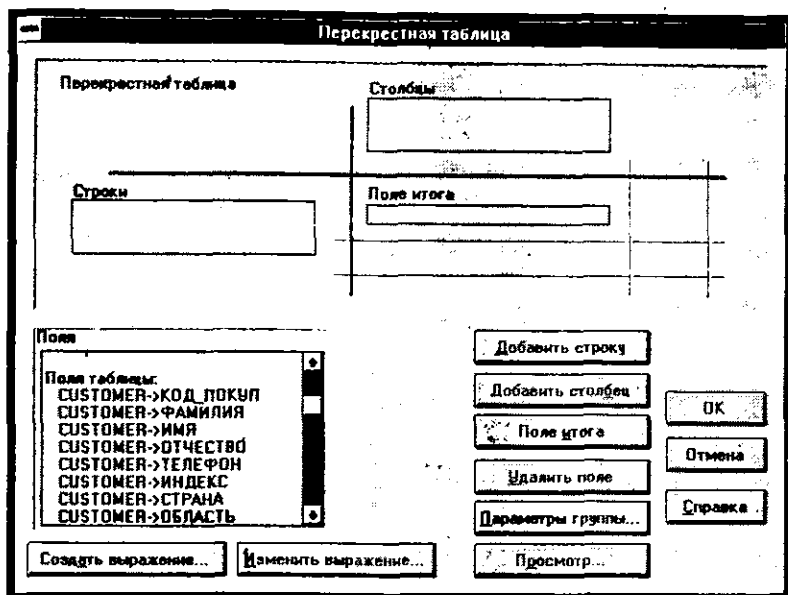


Рис. 14.19. Окно диалога "Перекрестная таблица".

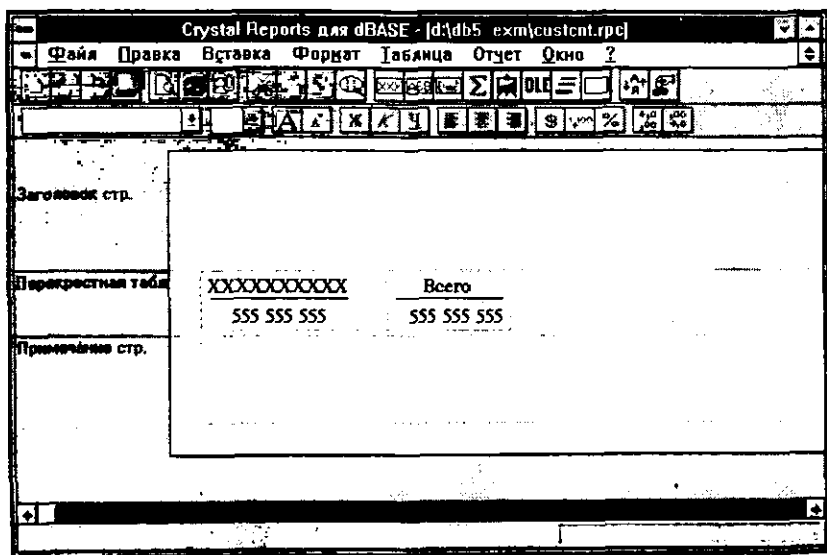


Рис. 14.20. Отчет с перекрестной таблицей.

4. Нажмите кнопку **ОК** и в окне конструктора отчетов появится отчет с перекрестной таблицей (рис. 14.20). Обратите внимание на то, что описание перекрестной таблицы приводится в отдельном разделе “Перекрестная табл” и кроме заданного нами итогового поля содержит поле, в котором выводится итоговое значение по всей строке перекрестной таблицы.
5. По умолчанию при выборе для группировки числового поля dBASE вычисляет сумму значений, соответствующих каждому значению строка/столбец. Об этом свидетельствует сообщение в строке состояния при выборе итогового поля.

Поскольку нам требуется определить число клиентов, проживающих в каждой стране, выберите итоговое поле и нажмите правую кнопку мыши. Далее из меню свойств выберите **Изменить итоговую операцию**. При этом откроется окно диалога “Поле итога” (рис. 14.21), в котором вы можете указать один из возможных видов итоговых операций. В данном случае выберите итоговую операцию **Подсчет**.

Более подробно каждая из итоговых операций будет рассмотрена далее в настоящем разделе.

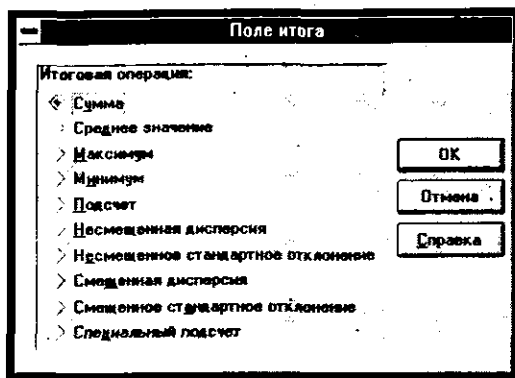


Рис. 14.21. Окно диалога “Поле итога”.

6. Аналогичным образом выберите групповую операцию **Подсчет** для поля, содержащего итоговое значение по строке.
7. Измените формат итоговых полей на целочисленный.
8. Для просмотра построенной перекрестной таблицы нажмите кнопку **Просмотр**. Результат просмотра приведен на рис. 14.18. Если вам потребуется напечатать перекрестную таблицу, находясь в окне конструктора отчета, выберите **Файл | Печать** или нажмите кнопку **Печать** на панели инструментов.
9. Сохраните созданную перекрестную таблицу, воспользовавшись командой **Файл | Сохранить как**.

В приведенном выше примере одномерная перекрестная таблица была расположена горизонтально. При наличии в результирующей таблице достаточно большого числа полей такой вид будет неудобным для просмотра. Для того чтобы определить вертикальную одномерную таблицу, вместо области “Столбцы” выберите область “Строки”, а из списка полей исходной таблицы — поле, которое будет использоваться в качестве заголовков строк.



Для преобразования горизонтальной перекрестной таблицы в вертикальную вы можете использовать команду **Инвертировать перекрестную таблицу** из меню свойств раздела **Перекрестная табл.**

Crystal Reports для dBASE - [Окно просмотра: d:\db5\exam\custent.rpt]

Файл Правка Вставка Формат Таблица Отчет Окно ?

Беларусь	2	
Казахстан	1	
Латвия	1	
Россия	20	
Украина	6	
Всего	30	

Тиз 1 [Назад] [Пагинация] [Отмена] [Поиск] [Справка] [Закреть] 10/08 21:38 Всего: 30 100

Рис. 14.22. Вертикальная перекрестная таблица.

Пример 14.8.

Продолжим работу над перекрестной таблицей, созданной в примере 14.7.

1. Выполните команду **Макет перекрестной таблицы** из меню свойств раздела описания перекрестной таблицы, и на экране появится окно диалога “Перекрестная таблица”. Выберите поле **CUSTOMER->Страна** в области “Столбцы” и нажмите кнопку **Удалить поле**.
2. Выберите поле **CUSTOMER->Страна** из списка допустимых полей и нажмите кнопку **Добавить строку**.
3. Нажмите кнопку **ОК**. Затем просмотрите созданную вертикальную перекрестную таблицу (рис. 14.22).



Для переноса полей между областями в окне диалога “Перекрестная таблица” вы можете использовать технику *перенести и оставить* (drag and drop). Для этого выберите поле, нажмите кнопку мыши и не отпуская кнопку мыши, перемещайте курсор мыши в другую область. После того как вы отпустите кнопку мыши поле будет перенесено из одной области в другую.

Типы групповых операций

dBASE при вычислении групповых значений позволяет использовать следующие типы группировки, которые вы можете выбрать в окне диалога “Поле итога”.

Операция	Назначение
Сумма	Итоговые значения равны сумме значений для каждой пары строка/столбец.
Подсчет	Итоговые значения равны количеству значений для каждой пары строка/столбец.
Минимум	Итоговые значения равны наименьшему из всех значений для каждой пары строка/столбец.
Максимум	Итоговые значения равны наибольшему из всех значений для каждой пары строка/столбец.
Среднее	Итоговые значения равны среднему значению для каждой пары строка/столбец.
Несмещенная дисперсия	Несмещенная дисперсия для каждой пары строка/столбец.
Несмещенное стандартное отклонение	Несмещенное стандартное отклонение для каждой пары строка/столбец.
Смещенная дисперсия	Смещенная дисперсия для каждой пары строка/столбец.
Смещенное стандартное отклонение	Смещенное стандартное отклонение для каждой пары строка/столбец.

При выборе типа группировки сначала данные будут отсортированы по выбранному полю, а затем будет выполняться группировка данных и вычисление итога по выбранному полю.

Двумерные перекрестные таблицы

Двумерная перекрестная таблица группирует информацию по двум факторам. Например, таблица SALECOMP содержит сведения о продажах компьютеров в 1995 г.

Ниже приведена структура таблицы SALECOMP:

Наименование поля	Тип поля
Код_покуп	Фиксированный
Продано	Фиксированный
Дата_прод	Дата
Цена	Фиксированный
Тип_комп	Текстовый

Данные в этой таблице были получены на основании информации, хранящейся в таблицах O_S_95 и GOODS при помощи многотабличного запроса. В этой таблице нас интересует, как распределились проданные компьютеры среди покупателей в 1994 году. Если указанная таблица имеет несколько сот записей, получение требуемой информации оказывается достаточно трудоемкой задачей. В dBASE с помощью групповых отчетов вы сможете получить требуемые данные, однако наиболее просто это

сделать, воспользовавшись двумерной перекрестной таблицей. На рис. 14.23 показана двумерная перекрестная таблица, которая отображает зависимость количества проданных компьютеров от модели компьютера и кода покупателя.

Числа в ячейках таблицы соответствуют количеству проданных компьютеров для каждой пары уникальных значений модель компьютера/код покупателя. Значения из поля "Тип_комп" можно расположить в верхней строке таблицы, а значения из поля "Код_покуп" — в левом столбце таблицы.

	CP_35 SX	CP_40 SX	CP_55 LS	CP_55 SX	Сумма
1	17	18	127	90	252
2	10	18	0	0	28
3	12	0	65	0	77
4	32	26	0	0	58
5	19	9	105	50	183
6	25	55	30	55	165
7	5	0	80	0	85
8	10	65	75	80	230
9	42	40	20	40	142
10	15	40	0	0	55
11	58	0	0	0	58
12	1	55	20	75	151
13	0	0	45	0	45
14	61	55	105	160	381
15	0	10	115	100	225
16	18	20	50	50	138
17	0	0	0	155	155
18	22	25	0	75	122

Рис. 14.23. Двумерная перекрестная таблица.

Пример 14.9. Создание двумерной перекрестной таблицы

Для создания двумерной перекрестной таблицы воспользуемся конструктором отчетов.

1. Откройте окно конструктора отчетов для таблицы SALECOMP.
2. В окне диалога "Перекрестная таблица" укажите:

"Тип_комп"	в качестве поля, значения которого будут отображаться в качестве наименований столбцов;
"Код_покуп"	в качестве поля, значения которого будут отображаться в качестве наименований строк;
"Продано"	в качестве поля, по которому будут группироваться значения.

Заполненное окно диалога "Перекрестная таблица" показано на рис. 14.24.

3. Выберите ОК для завершения определения таблицы. После возврата в окно конструктора сохраните перекрестную таблицу.

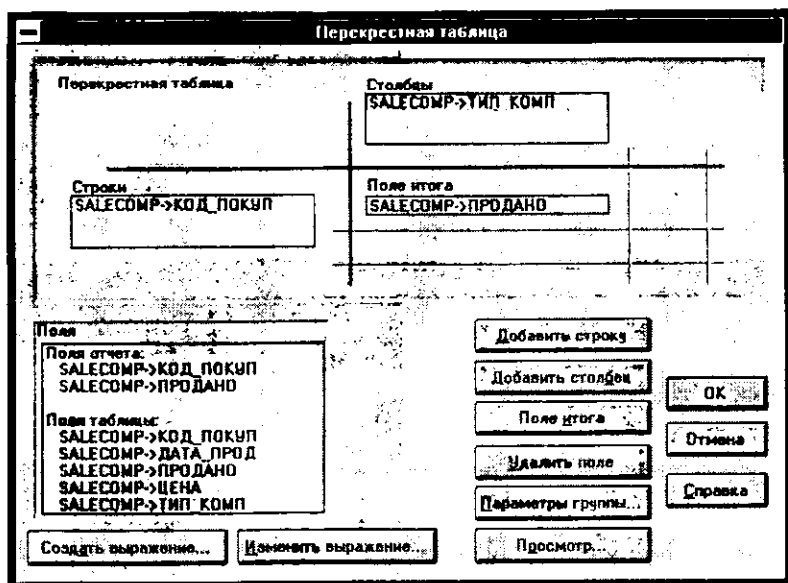


Рис. 14.24. Определение двумерной перекрестной таблицы.

4. Для просмотра перекрестной таблицы нажмите кнопку **Просмотр** на панели инструментов. На экране появится перекрестная таблица, которая будет иметь вид, показанный на рис. 14.23.

Группировка по полям дат при построении перекрестной таблицы имеет свои особенности, которые позволяют вам определить дополнительные параметры. Рассмотрим эти особенности на следующем примере, в котором продажи компьютеров будут сгруппированы не по их типам, а по датам продажи.

Пример 14.10. Группировка по датам

1. Откройте окно конструктора отчетов для таблицы SALECOMP.

2. В окне диалога “Перекрестная таблица” укажите:

- “Дата_прод” в качестве поля, значения которого будут отображаться в качестве наименований столбцов;
- “Код_покуп” в качестве поля, значения которого будут отображаться в качестве наименований строк;
- “Продано” в качестве поля, по которому будут группироваться значения.

3. Выберите группу “Дата_прод” и нажмите кнопку **Параметры продаж** для настройки параметров группировки по датам. На экране появится окно диалога “Параметры группы перекрестной таблицы” (рис. 14.25). В этом окне диалога вы можете выбрать один из интервалов дат, обычно используемых для группировки данных. В данном примере мы сгруппируем данные по месяцам. Кроме этого вы можете использовать список **Значение для столбца** для выбора одного из двух вариантов выводимых значений при группировке по периоду даты: первая дата пе-

риода или последняя дата периода. Выберите из этого списка второй вариант. Нажмите ОК для возврата в окно диалога “Перекрестная таблица”.

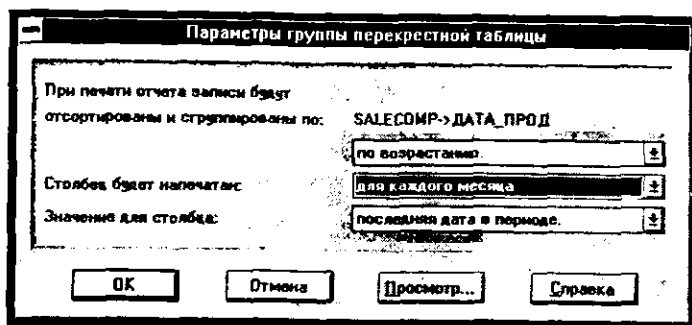


Рис. 14.25. Определение двумерной перекрестной таблицы.

4. В окне диалога “Перекрестная таблица” нажмите ОК для завершения определения таблицы. После возврата в окно конструктора сохраните перекрестную таблицу.
5. Для просмотра перекрестной таблицы нажмите кнопку Просмотр на панели инструментов. На экране появится перекрестная таблица, которая будет иметь вид, показанный на рис. 14.26.

Crystal Reports для dBASE - [Окно просмотра: d:\db5_exm\salecomd.rpt]

Файл Правка Вставка Формат Таблица Отчет Окно ?

TimesET 10

	31.01.94	28.02.94	31.03.94	30.04.94	31.05.94	30.06.94
1		90	30			
2						
3						
4						
5	10					
6		30	16			
7						
8						
9				32		
10				38		
11						
12		75			20	
13						
14			30	25	Н	66
15	40			25		60
16		50	50	20		
17			30	25		
18	15			20	25	
19						

1 из 1 Отчеты Закрыть Сегодня 08:45 Всего:96 100%

SALECOMP->DATA_ПРОД 1/2.85 П.5.19

Рис. 14.26. Группировка по датам.

Группировка по двум полям

При создании перекрестной таблицы вы можете группировать данные по нескольким полям исходной таблицы. Например, нас может интересовать не только

продажа компьютеров по покупателям, но и объемы продаж по месяцам. Для этого добавьте в область "Строки" еще одно поле "Дата_прод" (рис. 14.27) и определите для этого поля группировку по месяцам. Результат построения такой перекрестной таблицы вы видите на рис. 14.28.

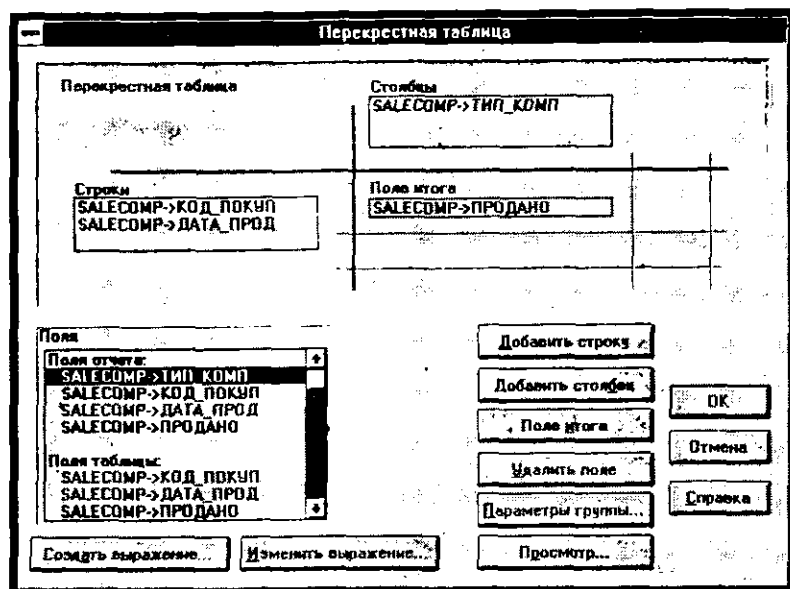


Рис. 14.27. Определение перекрестной таблицы с группировкой по покупателям и датам.

Crystal Reports для dBASE - [Окно просмотра: d:\db5_exm\salecomp.rpt]

Файл Правка Вставка Формат Таблица Отчет Окно ?

	CP_35_SX	CP_40_SX	CP_55_LS	CP_55_SX	Сумма
1 31.01.94	9	0	0	90	99
28.02.94	0	0	30	0	30
31.08.94	0	18	37	0	55
31.10.94	8	0	0	0	8
31.12.94	0	0	60	0	60
Сумма	17	18	127	90	252
2 31.08.94	0	18	0	0	18
30.09.94	10	0	0	0	10
Сумма	10	18	0	0	28
3 31.08.94	0	0	40	0	40
31.10.94	12	0	25	0	37
Сумма	12	0	65	0	77
4 28.02.94	7	0	0	0	7
31.07.94	0	26	0	0	26
30.09.94	25	0	0	0	25
Сумма	32	26	0	0	58
5 31.01.94	0	0	0	10	10
31.03.94	14	0	0	0	14
31.07.94	0	0	0	0	0

1 из 2 Отмена Закрыть Сегодня 21:01 Всего: 96 100%

Рис. 14.28. Группировка по покупателям и датам.

Использование групп отчета при создании перекрестной таблицы

Комбинирование групп отчета с перекрестными таблицами предоставляет в ваше распоряжение дополнительные возможности. Допустим, вам требуется просмотреть сгруппированные по месяцам продажи компьютеров покупателям. Для этого откройте отчет с перекрестной таблицей, созданный в примере 14.10. Далее добавьте в отчет раздел группы по полю “Дата_прод”, разместите в заголовке группы надпись с текстом Месяц и правее нее вычисляемое поле, которое содержит формулу, определяющую наименование месяца

```
CMONTH (ORDSALE->ДАТА_ПРОД)
```

На рис. 14.29 приведен законченный отчет в окне конструктора, а на рис. 14.30 — просмотр отчета. Обратите внимание на то, что в отличие от перекрестной таблицы, приведенной на рис. 14.28, данный отчет содержит не одну перекрестную таблицу, а по одной перекрестной таблице на каждую группу отчета.

Создание перекрестных таблиц с использованием запроса

dBASE позволяет создать перекрестную таблицу, в которой будут отображены данные сразу из нескольких таблиц. Ваши действия по созданию перекрестной таблицы на основании нескольких таблиц аналогичны созданию перекрестной таблицы по одной таблице. Единственное отличие состоит в том, что вам необходимо предварительно создать запрос, который будет использоваться в качестве источника данных.

Пример 14.11. Построение перекрестной таблицы по нескольким таблицам

1. Создайте запрос из таблиц SALECOMP и CUSTOMER, который содержит все поля из таблицы SALECOMP за исключением идентификатора покупателя, а также вычисляемое поле “ФИО” из таблицы CUSTOMER, объединяющее фамилию и инициалы покупателя.
2. Откройте окно конструктора отчетов для запроса, созданного в предыдущем шаге.
3. В окне диалога “Перекрестная таблица” укажите:

“SALECOMP->Дата_прод” в качестве поля, значения которого будут отображаться в качестве наименований столбцов;

“CUSTOMER->ФИО” в качестве поля, значения которого будут отображаться в качестве наименований строк;

“SALECOMP->Продано” в качестве поля, по которому будут группироваться значения.

4. В окне диалога “Перекрестная таблица” нажмите ОК для завершения определения таблицы. После возврата в окно конструктора сохраните перекрестную таблицу.

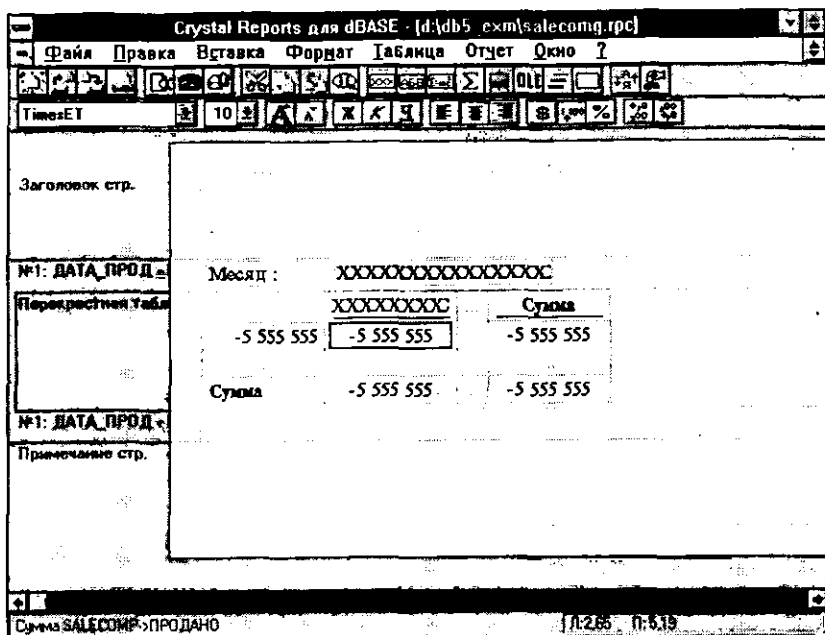


Рис. 14.29. Определение группы отчета.

Crystal Reports для dBASE - [Окно просмотра: d:\db5\exm\salecomp.rpt]

Файл Правка Вставка Формат Таблица Отчет Окно ?

TimesET 10

Месяц :	Январь				
	CP 35 SX	CP 40 SX	CP 55 LS	CP 55 SX	Сумма
1	9			90	99
5				10	10
12	1				1
15		10	30		40
18		15			15
20			20		20
Сумма	10	25	50	100	185
Месяц :	Февраль				
	CP 35 SX	CP 40 SX	CP 55 LS	CP 55 SX	Сумма
1			30		30
4	7				7
6			30		30
12				75	75
14	5				5
16				50	50
20		16			16

1 из 3

Отмена

Закреть

Сегодня 10:12 Всего: 96 100%

Сумма SALECOMP>ПРОДАНО

п.255, п.519

Рис. 14.30. Использование группы отчета.

- Для просмотра перекрестной таблицы нажмите кнопку **Просмотр** на панели инструментов. На экране появится перекрестная таблица, которая будет иметь вид, показанный на рис. 14.31. Отметьте, что использование фамилии покупателя вместо его идентификатора делает перекрестную таблицу более наглядной.

Crystal Reports для dBASE - [Окно просмотра: Отчет "Без названия" №1]

Файл Правка Вставка Формат Таблица Отчет Окно ?

	CP_35 SX	CP_40 SX	CP_55 LS	CP_55 SX
Александров А И	0	0	0	90
Алексеев С И	17	18	127	90
Антонов М Ю	0	0	0	155
Гершман А В	19	20	50	15
Загребский А А	10	65	75	80
Зайцева Д И	58	0	0	0
Зайцева Л Н	5	0	80	0
Ильин Ф П	19	9	105	50
Казанский С И	0	31	20	70
Морозов А М	0	10	115	100
Морозов А П	22	35	0	75
Морозов Ю А	12	0	65	0
Мишенский В И	0	20	0	0
Петров М Н	57	81	30	55
Райский М С	18	20	50	50
Свешников С И	0	0	45	0
Семесов Ф П	42	40	20	40
Сухарев А А	15	40	0	0
Томашев М М	10	18	0	0

1 из 1 Отмена Закрыть Сегодня 21:32 Всего: 96 100%

Рис. 14.31. Перекрестная таблица, созданная на основе запроса.

Настройка элементов перекрестной таблицы

Перекрестная таблица представляет собой сложный объект, который состоит из итоговых полей, области заголовков строк, области заголовков столбцов и поясняющих надписей. Вы можете изменить свойства элементов перекрестной таблицы точно также, как настраивали свойства остальных объектов dBASE. Для изменения свойств объекта укажите на него и нажмите правую кнопку мыши. В результате на экране появится меню свойств объекта, в котором для итоговых полей вы можете изменить вид группировки данных, а также формат, шрифт и цвет объекта.

Меню свойств полей, используемых для формирования строк и столбцов позволяет изменить свойства перекрестной таблицы. При изменении вида группировки данных по полю вам может потребоваться изменить надпись, связанную с этим полем. Для этого вы можете воспользоваться пунктом **Редактировать надпись** меню свойств надписи.

Экспорт перекрестной таблицы

Созданную с помощью dBASE перекрестную таблицу вы можете в дальнейшем использовать в программах управления электронными таблицами. Для этого вам необходимо экспортировать построенную перекрестную таблицу в формат той электронной таблицы, в которой предполагается осуществлять дальнейшую обработку.

Для экспорта перекрестной таблицы нажмите кнопку **Экспорт** на панели управления или выполните команду **Файл | Печать | Файл**, в результате чего откроется окно диалога "Экспорт" (рис. 14.32). Далее укажите формат данных и имя файла.

Конструктор отчетов Crystal Reports позволяет экспортировать данные в формат популярных электронных таблиц Excel, Lotus 1-2-3, Quattro Pro. Мы воспользуемся экспортом таблиц при построении диаграмм в главе 16.

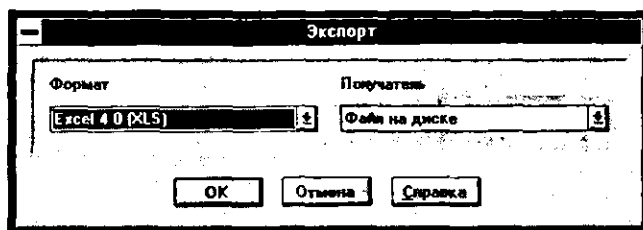


Рис. 14.32. Окно диалога “Экспорт”.

Заключение

- Конструктор отчетов позволяет вам создать отчеты, в которых представлены данные из нескольких таблиц. При этом вам необходимо воспользоваться многотабличным запросом.
- Подобно полям главной таблицы, в отчете могут быть использованы поля подчиненных таблиц.
- Команда **Раздел группы** из меню **Вставка** в окне конструктора отчетов позволяет сгруппировать данные в логические блоки.
- Для изменения порядка сортировки записей выберите опцию **Отчет | Порядок сортировки записей**.
- Для удаления группы из отчета выберите опцию **Удалить** из меню свойств группы.
- Для изменения способа группировки воспользуйтесь опцией **Изменить группу** из меню свойств группы.
- Конструктор отчетов позволяет создавать перекрестные таблицы по одной или нескольким таблицам.
- Для создания перекрестной таблицы выберите **Файл | Создать | Перекрестная таблица**.
- Для определения параметров перекрестной таблицы используйте окно диалога “Перекрестная таблица”.
- Для экспорта перекрестной таблицы нажмите кнопку **Экспорт** на панели управления или выполните команду **Файл | Печать | Файл**.

Глава 15. Система меню приложения

Хорошо написанное законченное приложение может использоваться пользователем любой квалификации. Обычно законченное приложение имеет свое собственное меню, которое заменяет основное меню dBASE и содержит команды, предназначенные для выполнения конкретных задач.

Перед тем, как приступить непосредственно к созданию приложения, вы можете создать все требуемые объекты (таблицы, экранные формы, отчеты, запросы). Затем отдельные объекты могут быть объединены посредством меню.

В данной главе мы рассмотрим использование конструктора меню для создания меню приложения.

Планирование приложения

Первым этапом создания любого приложения является определение требований, предъявляемых к законченному приложению. Прежде всего вы должны определить состав информации, которая будет содержаться в создаваемой базе данных.

После того, как вы определили данные, которые будут храниться в вашей базе данных, создайте все входящие в нее таблицы. На этом этапе вы определяете отношения между таблицами, структуру каждой таблицы и совпадающие поля для связывания таблиц.

Состав информации невозможно определить без учета конкретных задач, для решения которых предназначается создаваемое приложение. Поэтому одновременно с составом информации вы должны определить те средства, которые получит в свое распоряжение пользователь при работе с вашим приложением.

В качестве примера рассмотрим "Систему учета заказов и продаж", которая должна предоставлять пользователю следующие возможности:

- Вводить, сохранять и печатать сведения о покупателях
- Вводить и сохранять данные о товарах
- Печатать список товаров выбранной категории
- Вводить и сохранять данные о заказах покупателей и отпущенных им товарах
- Печатать сведения о товарах, проданных определенным покупателям
- Печатать список невыполненных заказов
- Определять статистические данные о сделанных заказах и проданных товарах за определенный период времени
- Отображать итоговые данные по месячному объему продаж и динамику продаж в графическом виде

Приложение должно содержать эффективную справочную систему, которая не требует специального руководства. В среде Windows предпочтительнее всего создавать справочную систему в принятом в Windows стандарте, так как в этом случае пользователь сможет получить любую информацию в знакомой ему среде.

После того, как вы спроектировали таблицы, входящие в состав базы данных, и определили основные требования, предъявляемые к приложению, вы можете приступить к созданию структуры меню приложения или, как его часто называют, дерева меню. Дерево меню можно организовывать на основе функций, выполняемых приложением (например, Продажа, Поступление товара) или же на основе таблиц, исполь-

зуемых в приложении (Покупатели, Товар). Оба варианта широко используются на практике, выбор одного из них или использование альтернативного принципа построения дерева меню определяется конкретными требованиями.

Прежде, чем начинать описывать структуру меню с помощью конструктора меню, нарисуйте его эскиз на бумаге, посоветуйтесь с пользователями приложения по поводу его структуры. Пока меню нарисовано только на бумаге, вы можете его легко изменить, добавить новые пункты и удалить лишние. На рис. 15.1 приведен эскиз приложения "Система учета заказов и продаж", пункты которого раскрываются ниже:

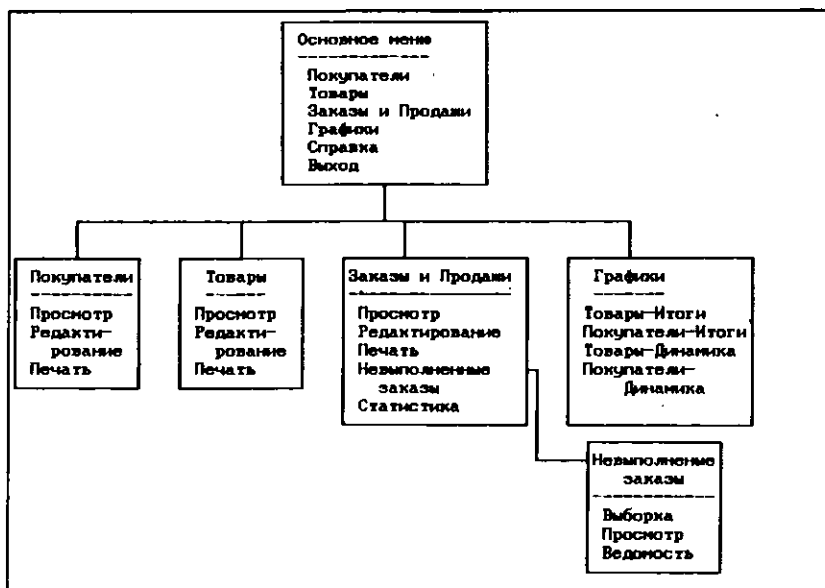


Рис. 15.1. Эскиз приложения "Система учета заказов и продаж".

Основное меню

Команда	Назначение
Покупатели	Просмотр, редактирование и печать сведений о покупателях;
Товары	Просмотр, редактирование и печать информации о товарах;
Заказы и Продажи	Просмотр, редактирование и печать данных о заказах и продажах;
Графики	Отображение итоговой информации в графическом виде;
Справка	Отображение информации о приложении;
Выход	Выход из приложения.

Подменю Покупатели

Команда	Назначение
Просмотр	Просмотр информации о покупателях;
Редактирование	Редактирование данных о покупателях;
Печать	Печать информации о покупателях.

Подменю Товары

Команда	Назначение
Просмотр	Просмотр информации о товарах;
Редактирование	Редактирование данных о товарах;
Печать	Печать информации о товарах.

Подменю Заказы и Продажи

Команда	Назначение
Просмотр	Просмотр информации о заказах и продажах за определенный период времени;
Редактирование	Редактирование данных о заказах и продажах;
Печать	Печать информации о заказах и продажах за определенный период времени;
Невыполненные заказы	Формирование, просмотр и печать заказов, невыполненных к определенной дате;
Статистика	Вычисление статистических данных.

Подменю Невыполненные заказы

Команда	Назначение
Выборка	Выбор данных о невыполненных заказах;
Просмотр	Просмотр информации о невыполненных заказах;
Ведомость	Печать информации о невыполненных заказах.

Подменю Графики

Команда	Назначение
Товары-Итоги	Отображает общее количество продаж по каждой категории товара;
Покупатели-Итоги	Отображает количество товаров, проданных покупателям;
Товары-Динамика	Отображает динамику продаж указанных товаров;
Покупатели-Динамика	Отображает динамику продаж определенным покупателем.

В настоящей главе мы рассмотрим создание системы меню на примере приложения "Система учета заказов и продаж".

Создание меню

В окне конструктора меню создаются следующие типы меню:

➤ Главное меню

- Ниспадающее меню, открывающееся под главным меню и содержащее список подчиненных пунктов
- Подменю, открывающееся справа от ниспадающего меню и содержащее список подпунктов

Главное меню

Горизонтальное меню, расположенное в верхней части экрана, называется *главным меню*. Примером главного меню является основное меню dBASE. У каждой формы может быть одно главное меню. Для создания строки меню выполните следующие шаги:

1. Откройте окно конструктора меню.
2. Опишите пункты главного меню.
3. Отобразите строку меню на экране.
4. Определите действия, выполняемые при выборе пункта меню.

Остановимся подробнее на каждом из приведенных этапов создания меню.

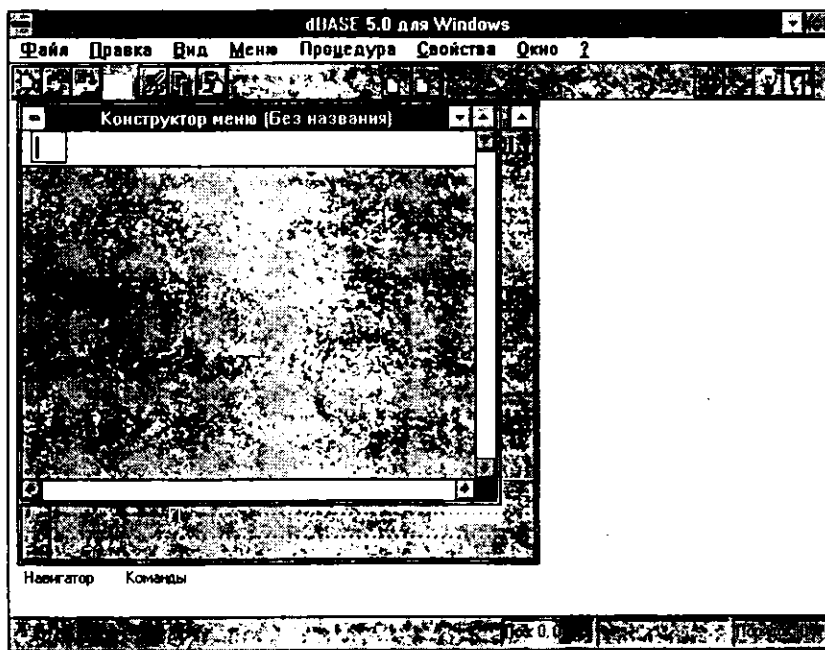


Рис. 15.2. Окно конструктора меню.

Окно конструктора меню

Для открытия окна конструктора меню выполните одно из следующих действий:

- Находясь в окне конструктора форм выберите команду из основного меню конструктора форм **Макет | Операции | Конструктор меню**

- Введите CREATE MENU в окне команд

На экране появится окно конструктора меню, а в основном меню dBASE появится новый пункт Меню (рис. 15.2).

Описание меню состоит из двух частей. В первой части вы описываете вид меню. Во второй части описываете действия, выполняемые при выборе пункта меню. Если вы опишете только первую часть, меню будет отображаться на экране, но при выборе пунктов меню никакие действия выполняться не будут.

Пример 15.1. Определение пунктов главного меню

В данном примере определим тексты пунктов главного меню приложения "Система учета заказов и продаж".

1. Откройте новое окно конструктора меню.
2. В поле расположения курсора введите текст первого пункта основного меню.
3. Для создания нового пункта меню правее существующего, нажмите клавиши → или Tab.
4. Для создания нового пункта меню левее существующего, выберите команду Меню | Вставить пункт меню или нажмите на панели инструментов кнопку Вставить.



На рис. 15.3 представлено главное меню.

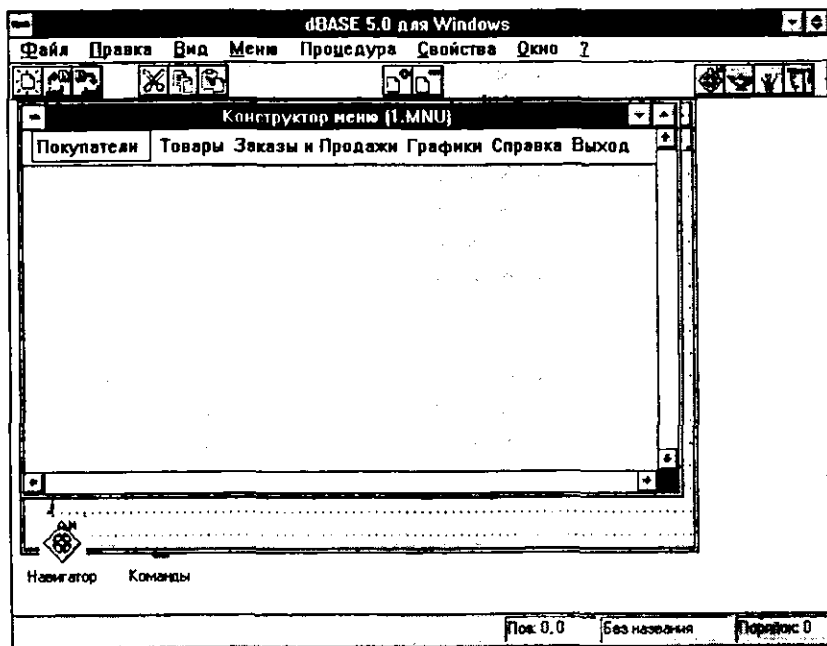


Рис. 15.3. Главное меню приложения.

Сохранение меню

Для сохранения созданного меню выполните следующие действия.

1. Выполните команду меню **Файл | Сохранить как**, в результате чего откроется окно диалога "Сохранить как".
2. Выберите каталог, в котором предполагаете сохранить файл, а в текстовом поле **Имя файла** введите имя сохраняемого меню с расширением .MNU.
3. Нажмите кнопку **ОК**.

В качестве примера сохраните созданное в примере 15.1 меню под именем **ORDSALE.MNU**.

Определение назначенных клавиш пункта меню

Выбор пункта меню осуществляется при помощи клавиш-стрелок или мыши. Дополнительно вы можете определить символ из имени пункта меню, активизирующий этот пункт при нажатии комбинации **Alt** и указанного символа, называемого *назначенной клавишей*. Для предоставления пользователю возможности ускоренного выбора пункта меню включите в его имя перед активизирующим символом "&". Обычно в качестве назначенной клавиши используется первый символ имени пункта, однако допустимо использование любого символа из текста пункта. Указанные вами символы отображаются в строке меню с подчеркиванием.

Применение назначенных клавиш для пунктов меню показано на рис. 15.4.

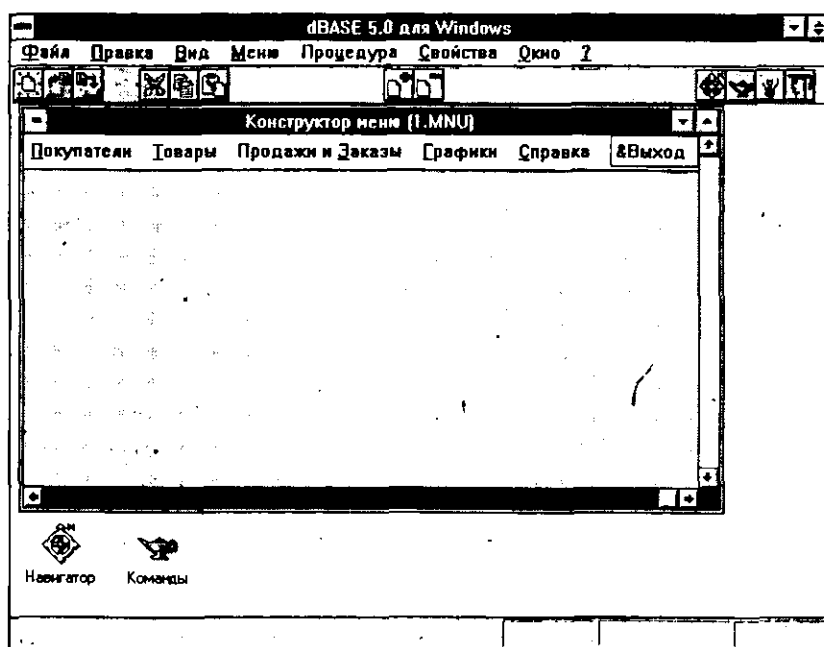


Рис. 15.4. Просмотр строки меню с назначенными клавишами.

Определение клавиш быстрого доступа

Назначенные клавиши предназначены для ускоренного выбора пункта Меню с помощью символов текста пункта меню. В dBASE существует возможность определения для каждого пункта меню *клавиши быстрого доступа*. В качестве такой клавиши могут использоваться символы, функциональные клавиши, а также комбинации клавиш. При нажатии на эту клавишу активизируется связанный с ней пункт меню. Отличие между назначенными клавишами и клавишами быстрого доступа состоит в том, что с помощью клавиш быстрого доступа команда выбирается без предварительного отображения меню команд.

Для назначения клавиши быстрого доступа для пункта меню нажмите правую кнопку мыши и выберите команду всплывающего меню **Окно свойств** данного пункта меню. На экране откроется окно свойств пункта меню. В этом окне укажите комбинацию клавиш в поле ввода значения свойства **Shortcut** (рис. 15.5). Например, в качестве клавиши быстрого доступа задано CTRL+X.

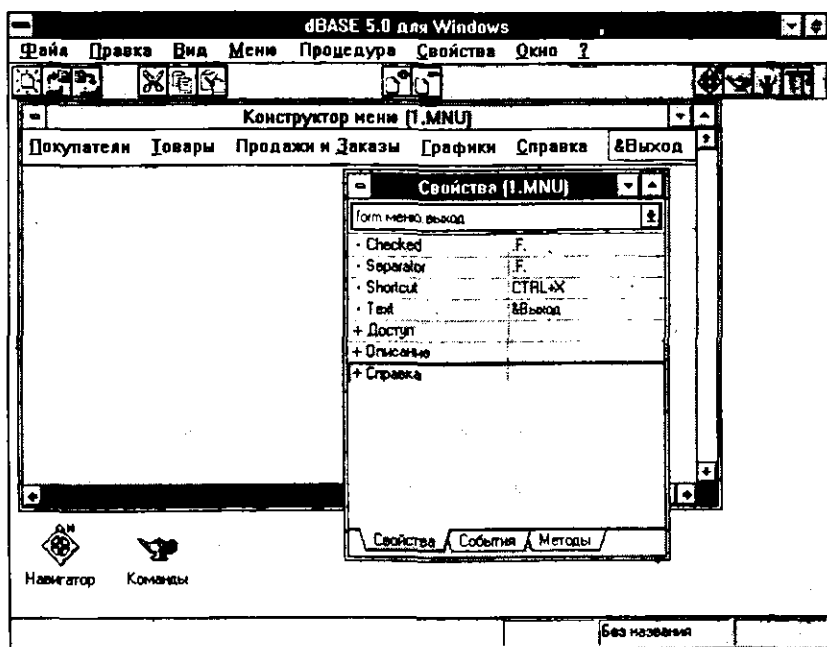


Рис. 15.5. Окно свойств пункта меню.



Конструктор меню автоматически выравнивает клавиши быстрого доступа по правому краю имени пункта меню.

Определение сообщения для пункта меню

Для каждого пункта меню вы можете определить сообщение, которое будет выводиться в строке состояния при установке курсора на этот пункт меню. Для определения сообщения используется свойство **StatusMessage** пункта меню. Вы можете использовать в качестве сообщения не только строку текста, но и произвольное символьное выражение.

Пример 15.2. Определение сообщений

В данном примере определим сообщения для пунктов главного меню.

1. Откройте файл ORDSALE.MNX в окне конструктора меню.
2. Установите курсор на пункт главного меню **Покупатели**.
3. Нажмите правый курсор мыши и выберите пункт всплывающего меню **Окно свойств**.
4. В окне свойств в поле ввода значения свойства **StatusMessage** введите строку текста **Ввод и вывод данных о покупателях**.
5. Аналогичным образом определите сообщения для остальных пунктов меню. На рис. 15.6 представлен результат отображения сообщения в строке состояния.

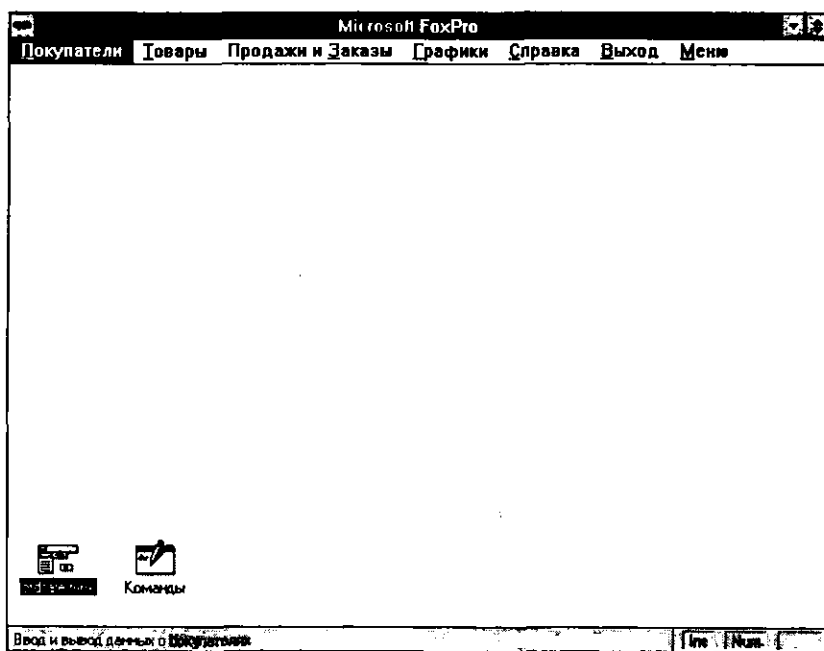


Рис. 15.6. Отображение сообщения пункта меню.

Создание ниспадающего меню

Выбор каждой команда меню нашего приложения, кроме пунктов **Справка** и **Выход**, должен открывать ниспадающее меню. Для его создания установите курсор на пункт главного меню и нажмите клавишу ↓. Под выбранным меню открывается новый пункт (рис. 15.7).

Пример 15.3. Определение пунктов ниспадающего меню

В данном примере определим пункты ниспадающего меню **Покупатели**.

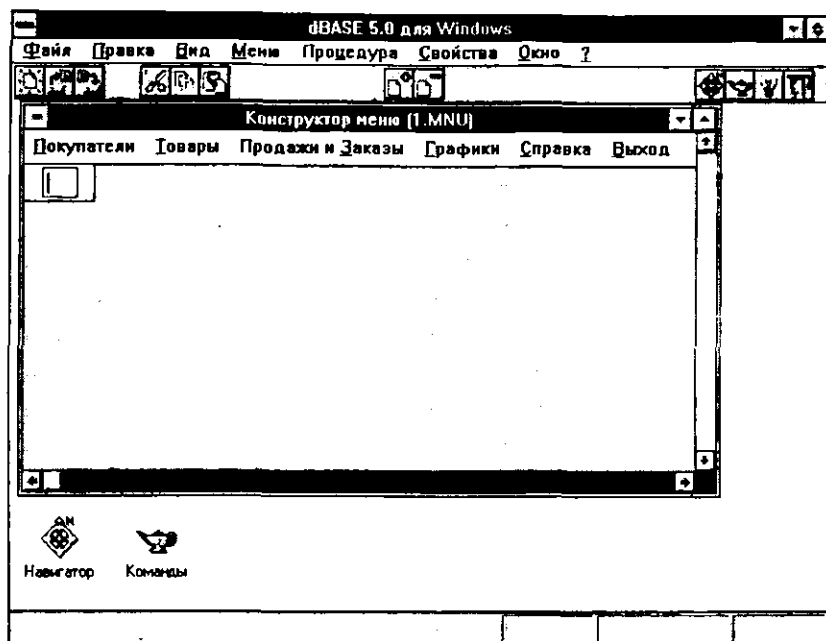


Рис. 15.7. Создание ниспадающего меню "Покупатели".

1. Откройте файл ORDSALE.MNX в окне конструктора меню.
2. Установите курсор на пункт главного меню Покупатели и нажмите клавишу ↓. Ниже выбранного пункта меню открывается новый пункт ниспадающего меню (рис. 15.7).

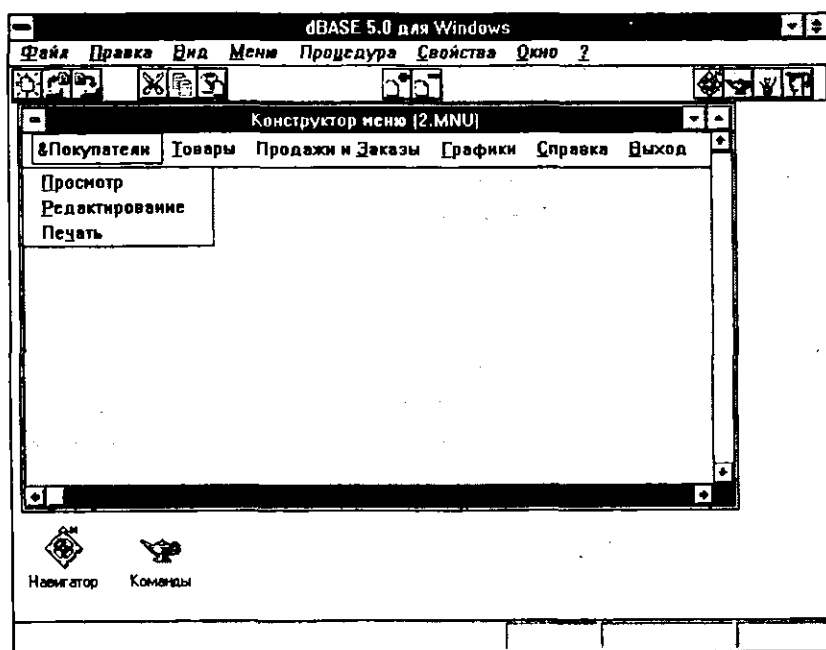


Рис. 15.8. Пример ниспадающего меню Покупатели.

3. Введите текст Просмотр.
4. Нажмите клавишу *Enter* и аналогичным образом определите остальные пункты ниспадающего меню.
5. Определите назначенные клавиши для пунктов ниспадающего меню.

На рис. 15.8 показан пример ниспадающего меню для пункта главного меню Покупатели.

Создание разделителей элементов меню

Элементы созданного нами ниспадающего меню можно разделить линиями, что позволяет улучшить внешний вид меню, объединить в группы схожие по смыслу команды.

Для создания разделительных линий можно воспользоваться любым из следующих способов.

- Установите курсор на пункт ниспадающего меню, выше которого хотите создать разделительную линию, и выполните команду Меню | Вставить разделитель.
- Создайте новый пункт ниспадающего меню на месте предполагаемой разделительной линии. Откройте окно свойств и установите значение свойства *Separator* равным (.T.).

Пример 9.4. Создание разделителя

В данном примере создадим разделитель в подменю “Покупатели”, который отделяет пункт меню Печать от двух предыдущих элементов меню.

1. Откройте файл ORDSALE.MNX и установите курсор на пункт ниспадающего меню Печать главного меню Покупатели.
2. Выполните команду Меню | Вставить разделитель.

Ниспадающее меню с разделительной линией представлено на рис. 15.9.

Применение меток для переключения пунктов меню

Метки пунктов меню используются для выбора параметров, свойств или элементов из предложенного списка. Иначе их можно назвать выключателями, так как пользователи осуществляет выбор одного из двух состояний: “включено” или “выключено”. По умолчанию пункты меню выключены и свойство *Checked*, определяющее признак метки, имеет значение (.F.).

Метки удобны для выбора режима работы приложения. Пункты меню с метками аналогичны объектам интерфейса “флажок” в форме.

Метки также позволяют выбирать один пункт меню из группы однородных пунктов. Такая метка отмечает только один пункт группы: при установке новой метки предыдущая автоматически снимается. В этом случае пункты меню с метками аналогичны группе переключателей формы. На рис. 15.10 показан пример с выбранным (отмеченным) пунктом меню.

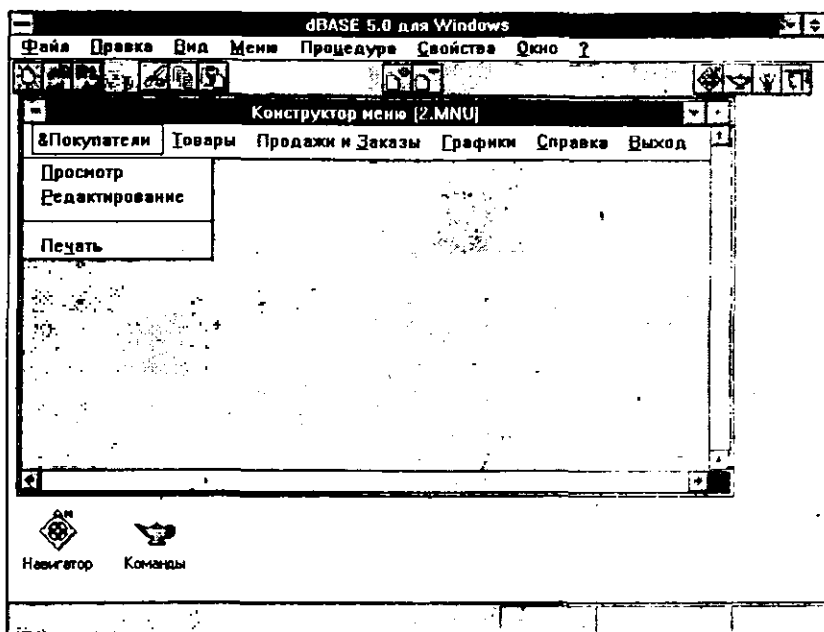


Рис. 15.9. Создание разделителя в ниспадающем меню Покупатели.

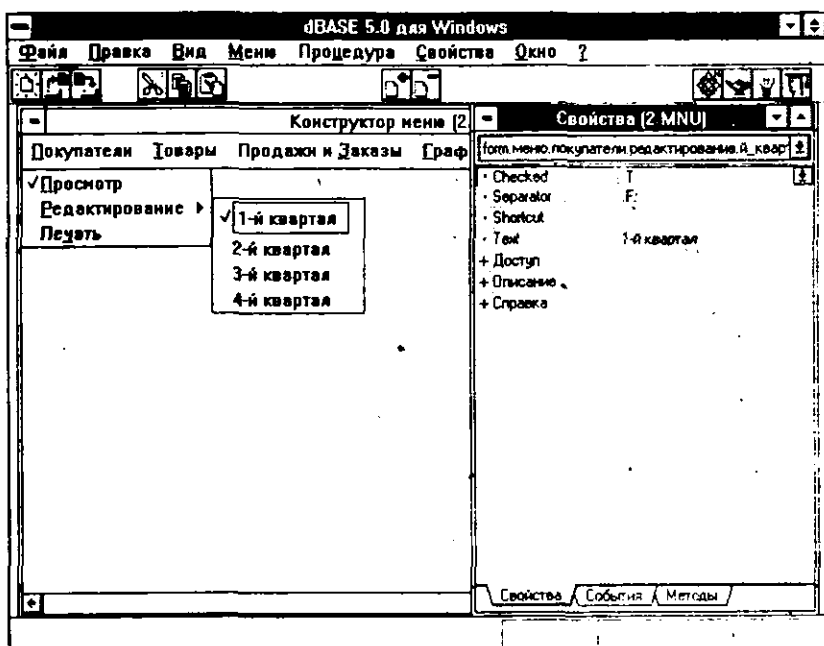


Рис. 15.10. Использование меток для переключения пунктов меню.



При работе с метками пунктов меню в приложении вы можете воспользоваться языком dBASE, который позволяет анализировать свойства пунктов меню.

Определение действий для пунктов меню

Каждый пункт меню содержит событие OnClick (рис. 15.11), которое наступает при нажатии мышью этого пункта меню или при нажатии клавиши *Enter* при выбранном пункте меню.

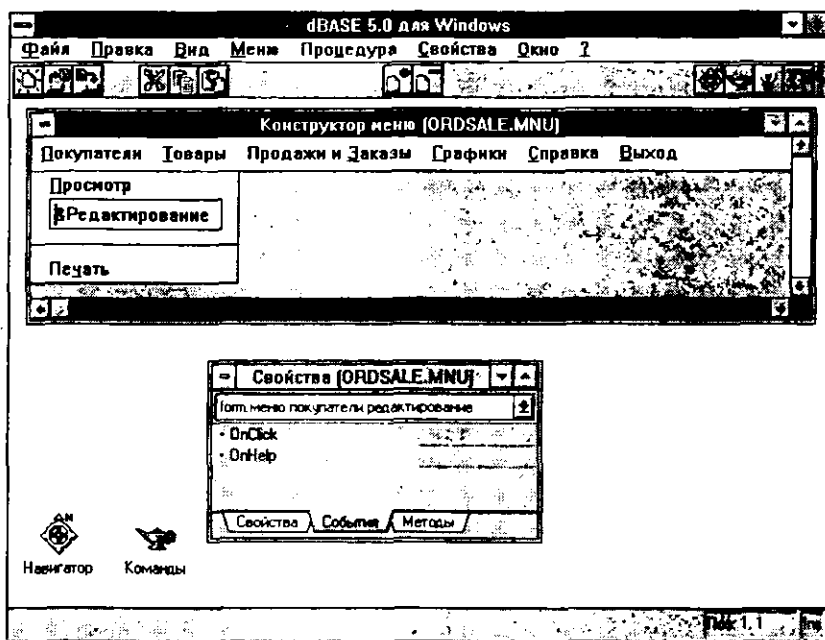


Рис. 15.11. События пункта меню.

Если связать это событие с текстом программы, то программа будет выполняться в момент наступления события. Например, можно написать текст программы, обеспечивающий вызов формы, печать отчета, выполнение запроса. Такая программа называется программой обработки события или обработчиком событий. В dBASE предусмотрено два способа написания и связывания обработчиков событий с событиями:

- Можно написать небольшую программу, называемую макроопределением, и связать ее с событием OnClick текущего объекта.
- Допускается также написать процедуру и связать ее с событием. В отличие от макроопределения процедуру можно связывать с несколькими событиями и с несколькими объектами.

В рамках данной книги мы не ставим цель изучение языка dBASE, но рассмотрим несколько команд, необходимых для вызова ранее созданных форм, отчетов и запросов. Прежде всего отметим, что форма является программой dBASE, поэтому для ее вызова необходимо выполнить команду DO с указанием имени выполняемой программы. Имя программы совпадает с именем файла формы. Например, для вызова формы просмотра списка покупателей необходимо выполнить команду

```
DO CUSTOMER.WFM
```

Для печати отчета с предварительным просмотром его на экране выполните команду `REPORT FORM CUSTOMER.RPT`

```
REPORT FORM CUSTOMER.RPT
```

При вызове запроса на экран вы должны предоставить пользователю возможность изменять его и вводить новые условия выбора. Поэтому необходимо вызвать запрос в режиме конструирования. Для этого используйте команду `MODIFY QUERY` с указанием имени запроса. Например, для вызова запроса о продажах товаров необходимо выполнить команду

```
MODIFY QUERY ORDSALE.QBE
```

Определение процедур

Рассмотрим создание процедуры и связывание ее с событием `OnClick` пункта меню. Для этого выполните следующую последовательность действий:

1. В конструкторе меню выберите пункт создаваемого меню.
2. Откройте окна свойств объекта и выберите карточку "События" (рис. 15.12).

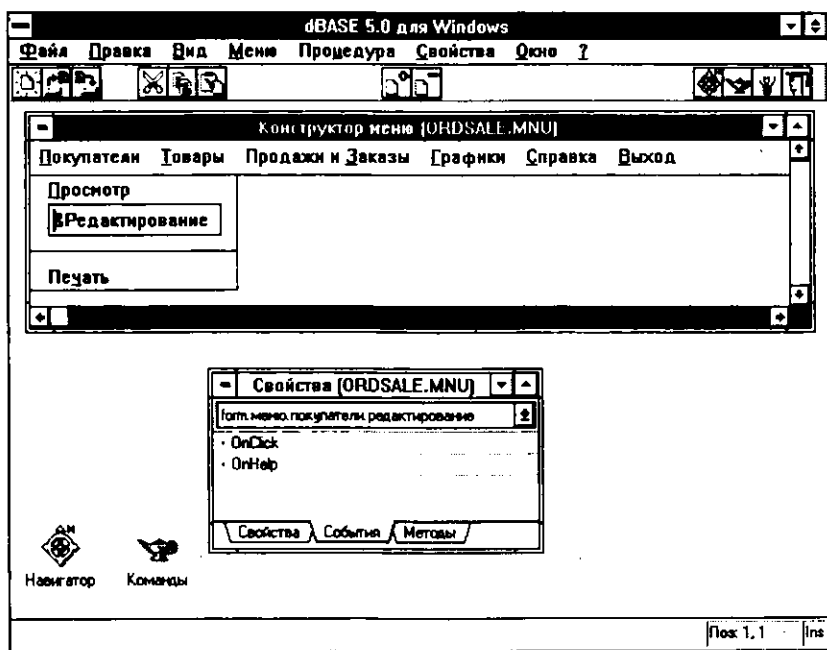


Рис. 15.12. Окно редактора процедур.

3. Выберите событие `OnClick` и нажмите кнопку **Инструмент** справа от поля ввода. На экране открывается окно редактора процедур и создается процедура с уникальным именем процедуры, состоящее из имени объекта и события.
4. Создайте тело процедуры, расположенное за инструкцией `PROCEDURE`. Тело процедуры может содержать множество команд dBASE. Мы разместим в процедуре

только команды вызова форм, запросов и печати отчетов. Например, для вызова формы напечатайте DO CUSTOMER.WFM (рис. 15.13).

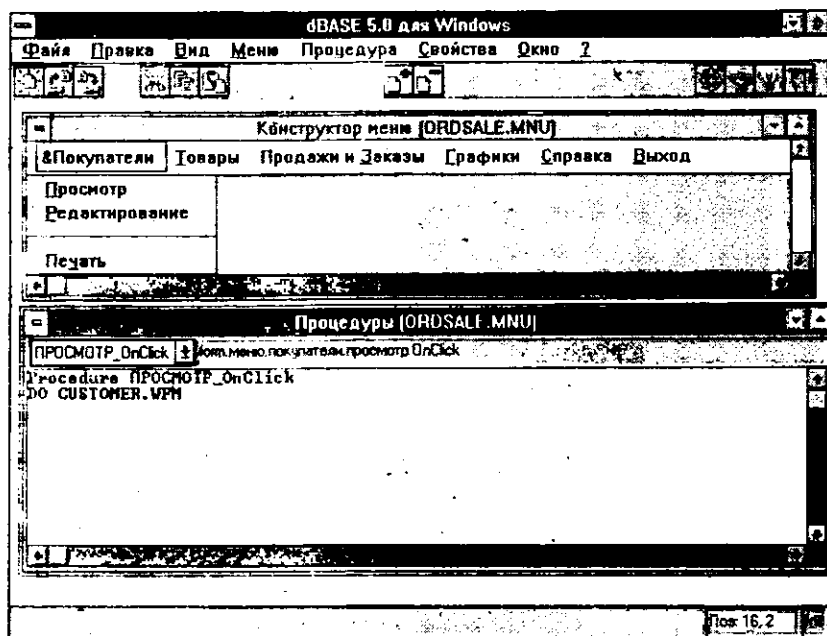


Рис. 15.13. Команда вызова формы.

Теперь данная процедура связана с событием, выбранным на шаге 1, поэтому при выборе пункта меню эта процедура будет выполнена. Событие и процедура вместе становятся одним из методов и появляются на карточке "Методы" окна свойств объекта.

Определение макроопределения

Макроопределения предназначены для хранения небольших безымянных групп команд или выражений. Макроопределения вводятся непосредственно в поле ввода выбранного события.

Для того чтобы создать макроопределение и связать его с событием, выполните следующую последовательность действий:

1. В конструкторе меню выберите пункт создаваемого меню.
2. Откройте окна свойств объекта и выберите карточку "События".
3. Выберите событие **OnClick** и перейдите в поле ввода.
4. Введите в поле макроопределение.

При вводе макроопределения необходимо учитывать синтаксис макроопределения, самый простой вариант которого имеет следующий вид:

```
{;< команда >[;< команда >...]}
```

Обратите внимание на то, что каждое макроопределение заключается в фигурные скобки { }, а перед каждой командой ставится знак точка с запятой (;). Например, для вывода отчета в режиме просмотра можно использовать макроопределение, приведенное на рис. 15.14.

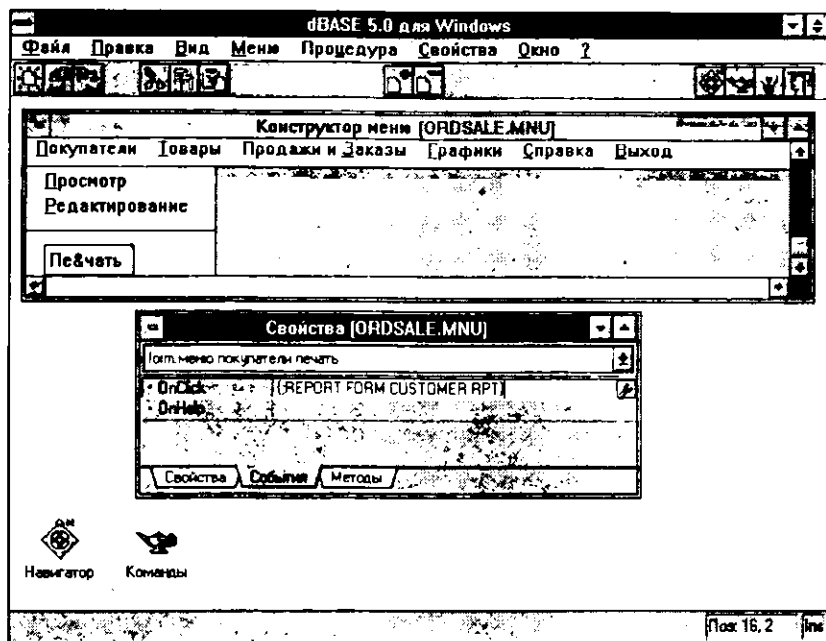


Рис. 15.14. Формирование макроопределения.

Блокирование команд меню

Иногда возникает необходимость сделать какой-либо пункт меню, недоступным для пользователя. Например, если приложением пользуются несколько пользователей, вы можете создать два или более вариантов меню. Каждый из них будут содержать одни и те же пункты меню, но доступ к ним будет различным. Для определения доступа к пункту меню используется свойство **Enabled**. Значение (.T.) этого свойства разрешает доступ к данному пункту меню, а значение (.F.) — запрещает доступ.

Закключение

- Первым этапом создания меню является определение требований, предъявляемых к законченному приложению, и построение эскиза меню.
- Для создания меню используется конструктор меню, который вызывается командой **Макет | Операции | Конструктор меню** из основного меню конструктора форм.
- В окне конструктора меню вы можете создать следующие типы меню: Главное меню, Ниспадающее меню и Подменю.
- Для создания нового пункта меню главного меню правее существующего, нажмите клавиши **→** или **Tab**.

- Для создания нового пункта меню главного меню левее существующего, выберите команду **Меню | Вставить пункт меню** или нажмите на панели инструментов кнопку **Вставить**.
- Для создания ниспадающего меню установите курсор на пункт главного меню и нажмите клавишу ↓.
- Элементы ниспадающего меню можно разделить линиями, что позволяет улучшить внешний вид меню, объединить в группы схожие по смыслу команды.
- Вы можете определить символ из имени пункта меню, активизирующий этот пункт при нажатии комбинации *Alt* и указанного символа, называемого *назначенной клавишей*.
- В dBASE существует возможность определения для каждого пункта меню *клавиши быстрого доступа*. В качестве такой клавиши могут использоваться символы, функциональные клавиши, а также комбинации клавиш.
- Для определения меток пунктов меню используйте свойство **Checked**.
- Для определения действий для пунктов меню используйте событие **OnClick**.
- Для определения доступа к пункту меню используйте свойство **Enabled**.
- Для сохранения созданного меню выполните команду меню **Файл | Сохранить как**.

Глава 16. Обмен данными с другими приложениями Windows

До настоящего момента мы использовали dBASE в качестве источника информации и инструмента для ее обработки. Кроме того, dBASE поддерживает два механизма обмена с другими приложениями Windows: динамический обмен данными (Dynamic Data Exchange — DDE) и механизм работы с составными объектами (Object Linking and Embedding — OLE).

В главе 11 вы познакомились со средствами экспорта и импорта данных, которые позволяют вам осуществить обмен данными с рядом приложений. При изменении данных в исходном приложении вам придется каждый раз повторно осуществлять операции экспорта или импорта. Динамический обмен данными и механизм работы с составными объектами позволяют осуществить непосредственную связь между dBASE и требуемым приложением. В этом случае при изменении данных в приложении, данные в таблицах dBASE будут обновляться автоматически. Например, вы можете использовать в своих объектах данные из Excel или Quattro Pro, а также работать с объектами, созданными с использованием различных программ подготовки текстов. При этом dBASE будет отслеживать изменения, внесенные во внешние приложения.

Приложение, которое является источником при обмене данных называется *сервером*. Приложение, выполняющее роль приемника данных, называется *клиентом* (или — в OLE 2.0 — *контейнером*). dBASE может выступать в роли как сервера, так и клиента при динамическом обмене данными, а также контейнера OLE 2.0. Для того чтобы эффективно использовать средства DDE и OLE, вы должны иметь представление о работе как с сервером, так и клиентом.

Динамический обмен данными в dBASE выходит за рамки настоящей книги, поскольку этот механизм поддерживается только на уровне языка dBASE.

Создавая связи с другими приложениями, вы выходите за пределы dBASE. При этом вы не сможете обеспечить поддержку связи между приложениями. Например, при переименовании или перемещении файла внешнего приложения, связь будет разорвана. Разрыв связи может быть обусловлен простым перемещением элемента, по которому осуществлялась связь, в пределах электронной таблицы.

В данной главе мы также рассмотрим построение диаграмм. Для этого будем использовать перекрестные таблицы, созданные с помощью конструктора отчетов, и электронную таблицу в качестве средства отображения диаграмм.

Работа с составными объектами (OLE)

Механизм работы с составными объектами (OLE) позволяет хранить в таблицах dBASE сами объекты или связь с объектами, которые созданы с помощью приложения, выступающего в качестве OLE-сервера. При использовании OLE для размещения данных в dBASE, вы можете открыть исходное OLE-приложение не покидая dBASE, и внести в них требуемые изменения. При таком подходе для редактирования OLE-значений могут быть использованы требуемые текстовые или графические процессоры.

В качестве OLE-объектов вы можете использовать графику или текст. При этом должен возникнуть вопрос: зачем использовать еще один объект, когда в вашем распоряжении имеются текстовые и графические поля и соответствующие им объекты при разработке конструкторских объектов? Дело в том, что OLE-объект сохраняет связь со своим исходным приложением, что позволяет вам использовать OLE-сервер для внесения изменений в объект. Поэтому, если вы предполагаете, что содержимое

объекта может достаточно часто изменяться, целесообразно использовать OLE-объект. Например, ваша таблица содержит сведения о товарах, внешний вид которых достаточно часто претерпевает изменения. Использование OLE упростит процедуру корректировки данных.

Встраивание OLE-объектов

Встраивание OLE-объекта может осуществляться в поле OLE в окне таблицы или окне формы. Встроенный объект хранится в OLE-поле таблицы dBASE и OLE-сервер используется только для его редактирования. При этом вы можете использовать существующий объект или создать новый объект в процессе встраивания. Для этих целей используются команда Правка | Вставить объект или буфер обмена.

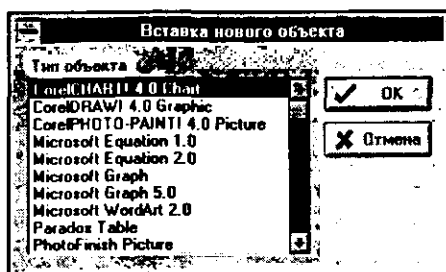


Рис. 16.1. Окно диалога "Вставка нового объекта".

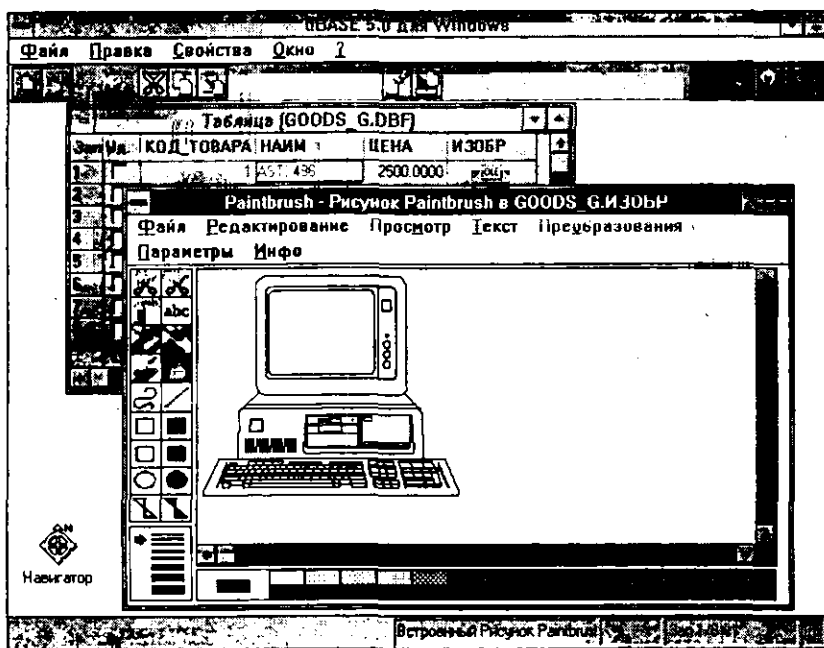


Рис. 16.2. Создание изображения компьютера.

Пример 16.1. Встраивание нового объекта OLE

Встраивание нового объекта OLE предполагает вызов OLE-сервера из dBASE. В этом случае вам необходимо выполнить приведенную ниже последовательность действий.

1. Откройте таблицу, содержащую OLE-поле, в которой вы предполагаете разместить новый OLE-объект (GOODS_G).
2. Перейдите в OLE-поле и нажмите дважды кнопку мыши или клавишу *F9*.
3. Выберите **Правка | Вставить объект**, в результате чего будет открыто окно диалога "Вставка нового объекта" (рис. 16.1).
4. Список **Тип объекта** содержит список всех OLE-серверов, установленных в вашей системе. В качестве примера выберите **Рисунок Paintbrush**.
5. Нажмите **ОК**. dBASE откроет указанное вами приложение.
6. Создайте изображение компьютера, которое будет использовано в качестве нового OLE-значения (рис. 16.2).

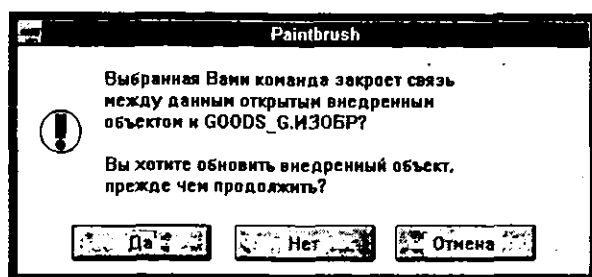


Рис. 16.3. Запрос подтверждения обновления объекта.

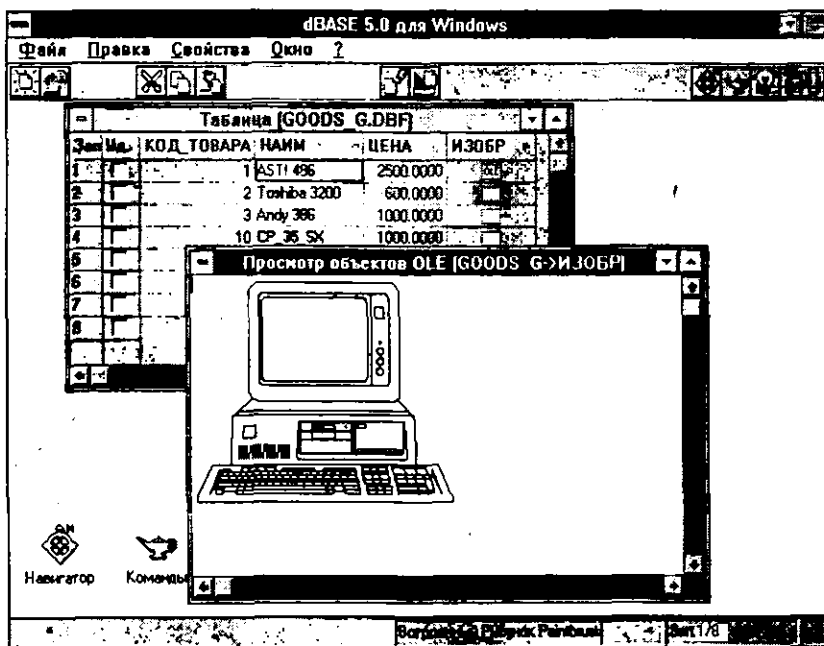


Рис. 16.4. Встраивание в поле таблицы нового OLE-объекта.

7. Закройте сервер OLE. Вам будет предложено обновить объект (рис. 16.3), после чего созданный вами объект будет размещен в OLE-поле в окне таблицы (рис. 16.4).
8. Если вам потребуется внести изменения во встроенный OLE-объект, откройте таблицу в режиме редактирования и дважды нажмите мышью требуемое значение. В результате будет открыто приложение, в окне которого содержится OLE-объект поля таблицы.

Пример 16.2. Встраивание существующего OLE-объекта

Для встраивания в dBASE существующего OLE-объекта необходимо выполнить следующие действия:

1. Откройте на OLE-сервере (Microsoft Paintbrush) объект (файл COMP3.PCX), который вы предполагаете разместить в поле таблицы (рис. 16.5).

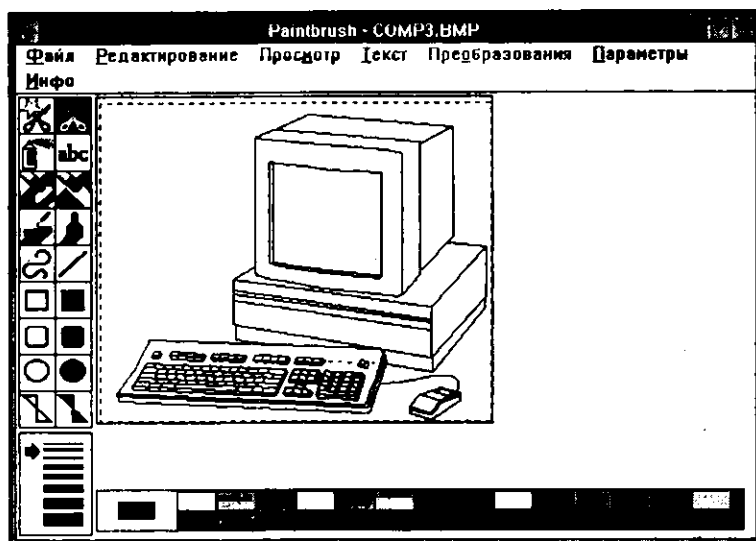


Рис. 16.5. Копирование OLE-объекта в буфер обмена.

2. Скопируйте содержимое файла в буфер обмена.
3. Нажмите *Ctrl+Esc* и с помощью окна "Список Задач" откройте dBASE.
4. Откройте таблицу GOODS_G в режиме редактирования и укажите на поле "Изобр" второй записи. Дважды нажмите кнопку мыши или нажмите клавишу *F9*.
5. Выберите Правка | Вставить, в результате чего в окне просмотра OLE-поля появится изображение компьютера, которое было скопировано в буфер обмена Windows (рис. 16.6).
6. Для того чтобы внести изменения в размещенное OLE-значение с использованием OLE-сервера, откройте окно просмотра OLE-объекта и дважды нажмите кнопку мыши.

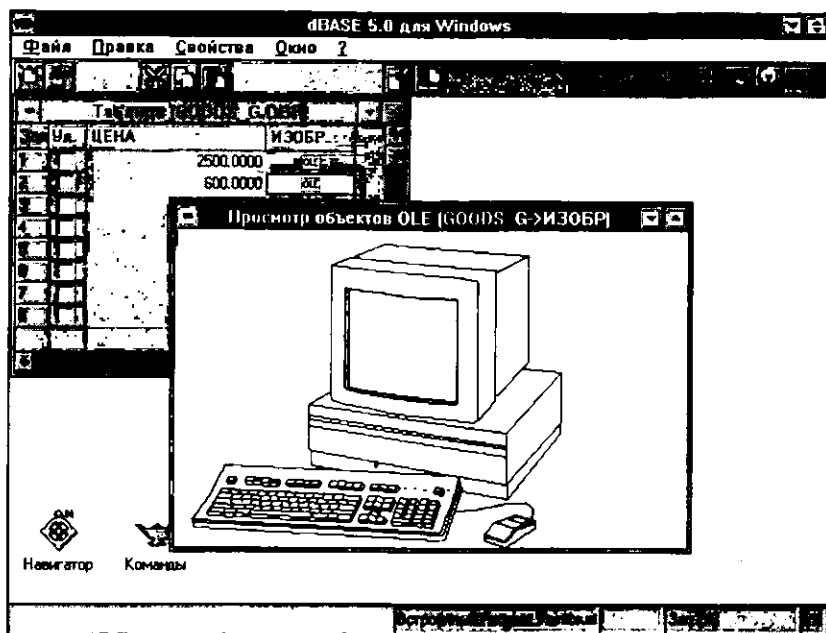


Рис. 16.6. Встраивание в поле таблицы существующего OLE-объекта.

Связывание OLE-объектов

При связывании OLE-объекта с OLE-полем таблицы исходный OLE-объект хранится в своем файле, а в OLE-поле таблицы хранится только связь с OLE-объектом. В таком случае этот объект может редактироваться отдельно от dBASE. При открытии OLE-поля в окне просмотра вы увидите последнюю редакцию OLE-объекта.

Пример 16.3. Связывание OLE-объекта

Для связывания в dBASE существующего OLE-объекта необходимо выполнить следующие действия:

1. Откройте на OLE-сервере (Microsoft Paintbrush) объект (файл COMP2.PCX), который вы предполагаете разместить в поле таблицы.
2. Скопируйте содержимое файла в буфер обмена.
3. Нажмите *Ctrl+Esc* и с помощью окна "Список Задач" откройте dBASE.
4. Откройте таблицу GOODS_G в режиме редактирования и укажите на поле "Изобр" третьей записи. Дважды нажмите кнопку мыши или нажмите клавишу *F9*.
5. Выберите **Правка | Вставить связь**, в результате чего в окне просмотра OLE-поля появится изображение компьютера, которое было скопировано в буфер обмена Windows (рис. 16.7). В строке состояния будет выведено сообщение о том, что в поле находится связанный объект.

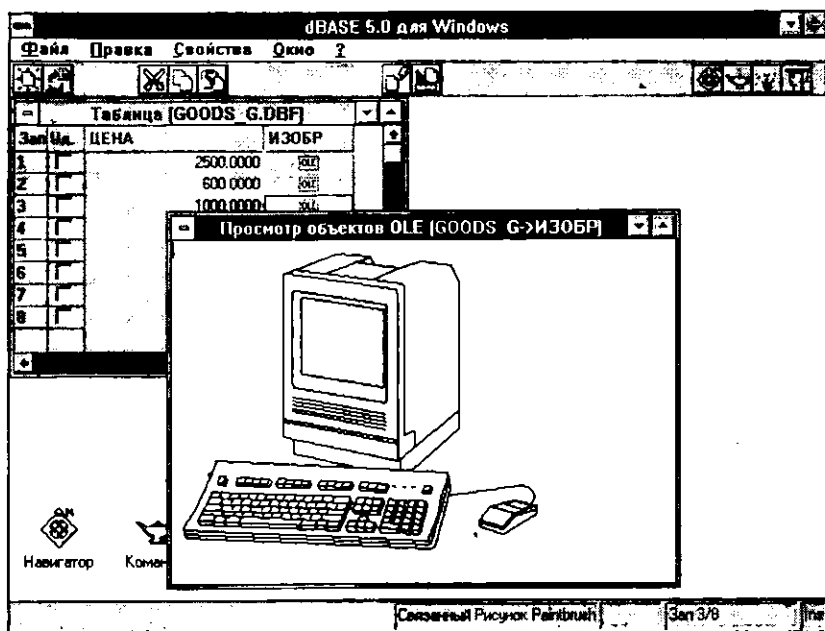


Рис. 16.7. Связывание OLE-объекта.

6. Для того чтобы внести изменения в размещенное OLE-значение с использованием OLE-сервера, откройте окно просмотра OLE-объекта и дважды нажмите кнопку мыши.

Если OLE-значение было размещено со связью, окно диалога “Связи” позволяет изменить или удалить установленную связь (рис. 16.8). Для открытия этого окна откройте требуемую таблицу, перейдите в нужное поле и выберите Правка | Связи.

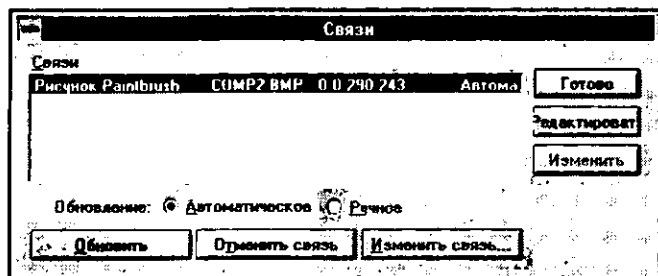


Рис. 16.8. Окно диалога “Связи”.

Окно диалога “Связи” предоставляет в ваше распоряжение следующие средства:

Элементы окна диалога	Назначение
Связи	Отображает список связей с указанием пути и имени файла источника, типа объекта и режима обновления связи.
Обновить	Обновляет выделенную связь.

Элементы окна диалога	Назначение
Обновление	Определяет режим обновления связи. При выборе Автоматическое обновление связанных объектов будет осуществляться автоматически. В случае выбора Ручное для обновления связанных объектов необходимо нажать кнопку Обновить или выбрать команду Обновить в меню объекта.
Редактировать	Открывает исходный файл для выделенной связи.
Изменить связь	Заменяет исходный файл для выделенной связи, используя окно диалога "Изменение связи".
Отменить связь	Разрывает указанную связь. В результате объект становится статическим, не допускающим изменения или обновления.

Удаление OLE-объекта

Для удаления OLE-объекта из OLE-поля вы можете выполнить команду Правка | Удалить или нажать кнопку Удалить на панели инструментов.

Диаграммы

В качестве средства визуального анализа вы можете использовать графическую интерпретацию табличной информации в виде диаграмм. Такой подход позволяет обнаружить скрытые закономерности в данных и сделать выводы, мимо которых вы могли бы пройти, не прибегни вовремя к помощи диаграмм.

Для представления одной и той же информации вы можете использовать диаграммы различных типов. При этом большое значение имеет правильный выбор типа диаграммы. Как и всякое средство статистического анализа, оно может привести к ложным выводам в случае неправильного выбора типа диаграммы.

dBASE не содержит средств для построения диаграмм, но средства обмена между приложениями Windows позволяют вам использовать данные dBASE при построении диаграмм в электронных таблицах. Отметим при этом, что ряд электронных таблиц могут открывать таблицы dBASE и использовать данные из них, не прибегая к средствам экспорта и импорта.

Для знакомства с диаграммами прибегнем к помощи электронной таблицы Microsoft Excel 5.0. В настоящем разделе вы узнаете:

- Как создать диаграмму в электронной таблице
- Как выбрать требуемый тип диаграммы и изменить используемые по умолчанию параметры диаграмм
- Как напечатать диаграмму и т.д.

Исходные данные для построения диаграмм вы можете подготовить двумя способами:

1. Создать перекрестную таблицу с помощью конструктора отчетов и экспортировать ее в формат электронной таблицы.
2. Создать запрос, содержащий данные для построения диаграммы и сохранить результат выборки в новой таблице dBASE.

В настоящей главе для создания диаграмм воспользуемся перекрестной таблицей SALEYEAR (рис. 16.9), которая создана с помощью конструктора отчетов и экспортирована в формат электронной таблицы Microsoft Excel, а также таблицей SALEOCT (рис. 16.10) в формате dBASE, созданной с помощью запроса.

Crystal Reports для dBASE - [Окно просмотра: Отчет "Без названия" №1]

Файл Правка Вставка Формат Таблица Отчет Окно ?

TimesET 10

	CP_35 SX	CP_40 SX	CP_55 LS	CP_55 SX	Сумма
31.01.94	10	25	50	100	185
28.02.94	12	36	60	125	233
31.03.94	30	30	100	60	220
30.04.94	70	40	50	50	210
31.05.94	30	35	100	40	205
30.06.94	40	80	95	100	315
31.07.94	60	56	80	100	296
31.08.94	19	36	77	95	227
30.09.94	35	19	105	160	319
31.10.94	20	50	50	100	220
30.11.94	10	65	40	115	230
31.12.94	30	45	100	60	235
Сумма	366	517	907	1 105	2 895

Таблица

Сегодня: 13:48 Всего: 96 100%

Рис. 16.9. Перекрестная таблица SALEYEAR.

dBASE 5.0 для Windows

Файл Правка Вид Таблица Свойства Окно ?

Таблица [SALEOCT.DBF]

Заг.№	ИДЕНТ	ТИП	КОМП	КОЛИЧЕСТВО
1		1	CP_35 SX	20
2		2	CP_40 SX	50
3		3	CP_55 LS	50
4		4	CP_55 SX	100

Навигатор Команды

3 из 4

Рис. 16.10. Таблица SALEOCT.

Таблица SALEOCT содержит данные о количестве компьютеров четырех моделей, проданных в октябре, а SALEYEAR — стоимость тех же моделей компьютеров, проданных в каждом месяце. Эти таблицы получены из таблицы SALECOMP, в которой содержится информация о продажах компьютеров.

В данном случае мы преднамеренно в качестве примера использовали таблицы, которые не нужно приводить к виду электронной таблицы, прежде, чем построить по ним диаграмму. Как правило, диаграмма может быть построена по достаточно сложной таблице с помощью средств, с которыми вы познакомились при создании перекрестных таблиц. Однако, в ряде специальных случаев этих средств оказывается недостаточно, и тогда вам необходимо самостоятельно подготовить таблицу к виду, приемлемому для построения диаграммы.

Пример 16.4. Простая диаграмма

В данном примере построим простую диаграмму, используя данные таблицы SALEOCT. Для этого выполните следующие действия:

1. Вызовите электронную таблицу Microsoft Excel 5.0, выполните команду **Файл | Открыть** и выберите таблицу SALEOCT.DBF. Данные из таблицы SALEOCT будут помещены в электронную таблицу (рис. 16.11).

A1	B1	C1	D1	E1	F1	G1
1	ИДЕНТ	ТИП КОМП	КОЛИЧЕСТВО	20	50	50
2	ИДЕНТ	1 CP 35 SX	20	50	50	100
3	ИДЕНТ	2 CP 40 SX	50	50	100	100
4	ИДЕНТ	3 CP 55 LS	50	100	100	100
5	ИДЕНТ	4 CP 55 SX	100	100	100	100

Рис. 16.11. Таблица SALEOCT в электронной таблице.

2. В Microsoft Excel 5.0 удобным средством для построения диаграммы является мастер диаграмм, который позволяет вам выбрать тип диаграммы, а также определить форматы данных и другие параметры. Прежде чем вызвать мастер диаграмм выделите область данных, используемую при построении диаграммы.
3. Вызовите мастер диаграмм нажатием одноименной кнопки на панели управления. Курсор мыши принимает форму перекрестья, предлагая определить вам расположение и размеры диаграммы. Установите курсор в левом верхнем углу

предполагаемого места расположения диаграммы, нажмите левую кнопку мыши и, не отпуская ее, выделите пунктирным прямоугольником размер диаграммы. После этого отпустите кнопку мыши и после появления окна мастера по выбору типа диаграммы, выберите Гистограмму (рис. 16.12). На следующем шаге выберите формат диаграммы.

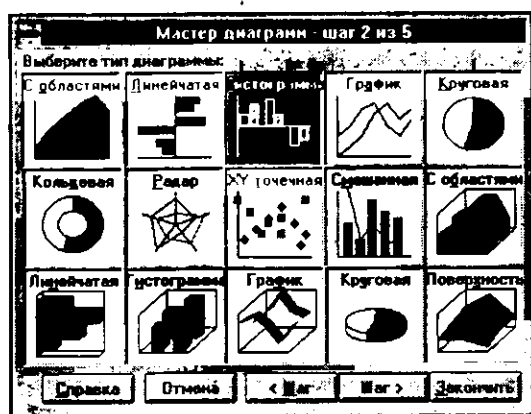


Рис. 16.12. Выбор типа диаграммы.

4. На последнем шаге построения диаграммы вы определяете заголовки диаграммы (рис. 16.13).

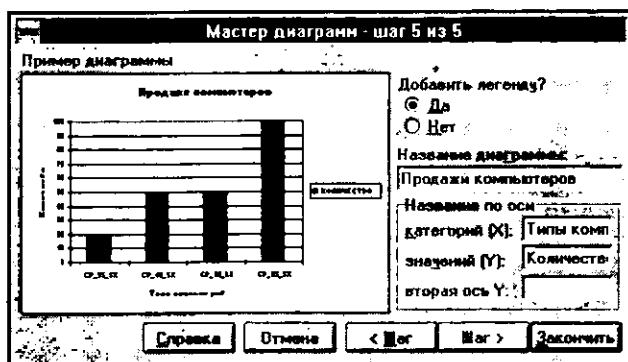


Рис. 16.13. Определение заголовков диаграммы.

5. Завершите работу по созданию диаграммы нажатием кнопки **Закончить**. Excel разместит созданную диаграмму в указанном вами месте (рис. 16.14). При необходимости вы можете совершенствовать созданную с помощью мастера диаграмм (изменить тип диаграммы, стандартные цвета, параметры сетки, шрифты и т.д.).

Диаграмма для таблицы SALEOCT, показанная на рис. 16.14, наглядно отображает соотношение между количеством компьютеров различных моделей, проданных в октябре.

Для одномерных таблиц, какой является SALEOCT вы можете построить круговую диаграмму. Круговая диаграмма для таблицы SALEOCT показана на рис. 16.15. Каждому элементу данных на диаграмме соответствует сектор круга, а суммарному

значению — весь круг. Такое представление данных позволяет наглядно оценить относительный вклад каждой из величин в итоговое значение.

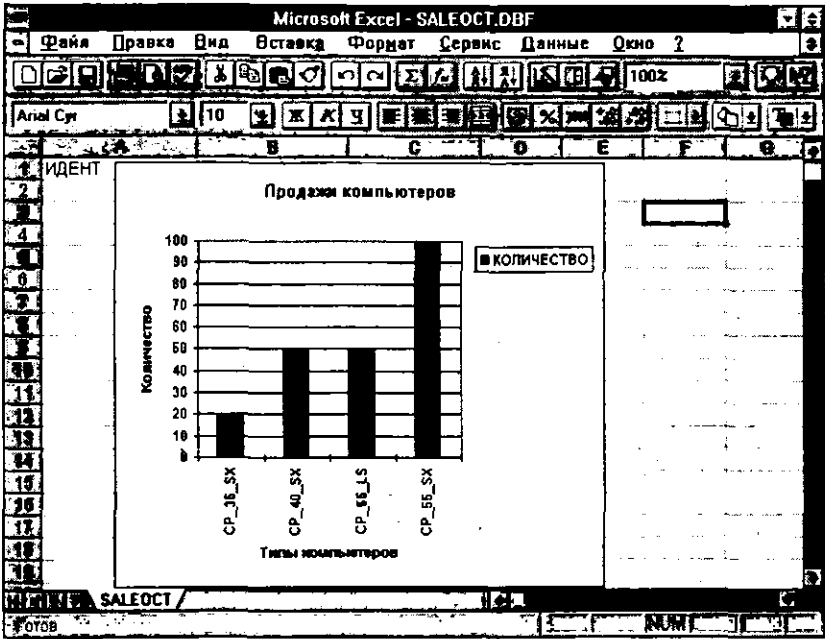


Рис. 16.14. Диаграмма для таблицы SALEOCT.

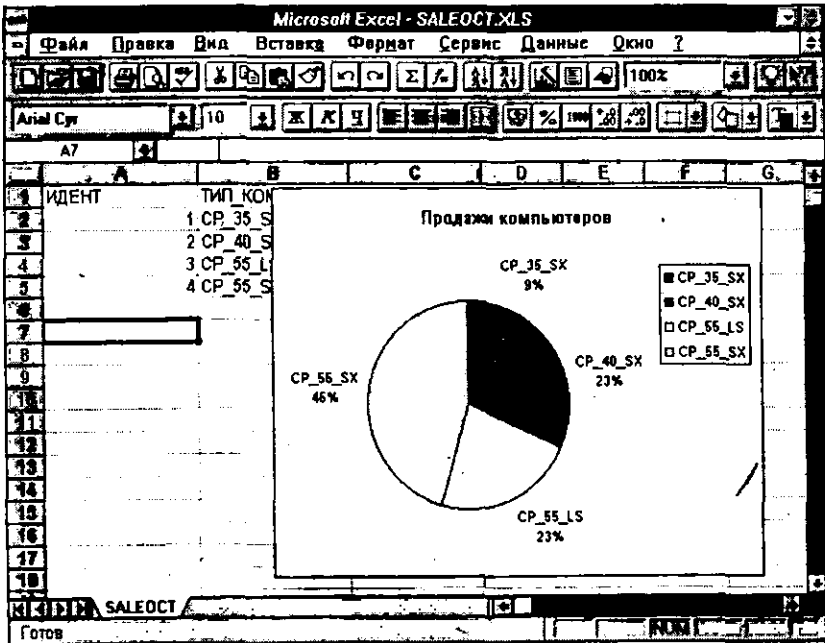


Рис. 16.15. Круговая диаграмма для таблицы SALEOCT.

Пример 16.5.

Таблица SALEOCT содержит всего одно числовое поле, поэтому диаграмма, показанная на рис. 16.14, представляет собой простейшую диаграмму. Для построения более сложной диаграммы обратимся к перекрестной таблице SALEYEAR (рис. 16.9).

1. Вызовите электронную таблицу Microsoft Excel 5.0, выполните команду **Файл | Открыть** и выберите электронную таблицу SALYEAR.XLS, которая получена из перекрестной таблицы SALYEAR.RPC.
2. При построении перекрестной таблицы в качестве наименований строк использовались последние даты месяца. В электронной таблице вы можете заменить их на краткие наименования месяцев простым изменением формата отображения данных.
3. Таблица SALEYEAR содержит четыре столбца, данные из которых могут быть отображены на диаграмме. Для построения диаграммы выберите все строки и столбцы таблицы за исключением итоговых по строке и столбцам. При построении диаграммы в качестве типа диаграммы выберите объемную гистограмму. В этом случае диаграмма будет иметь вид, показанный на рис. 16.16.

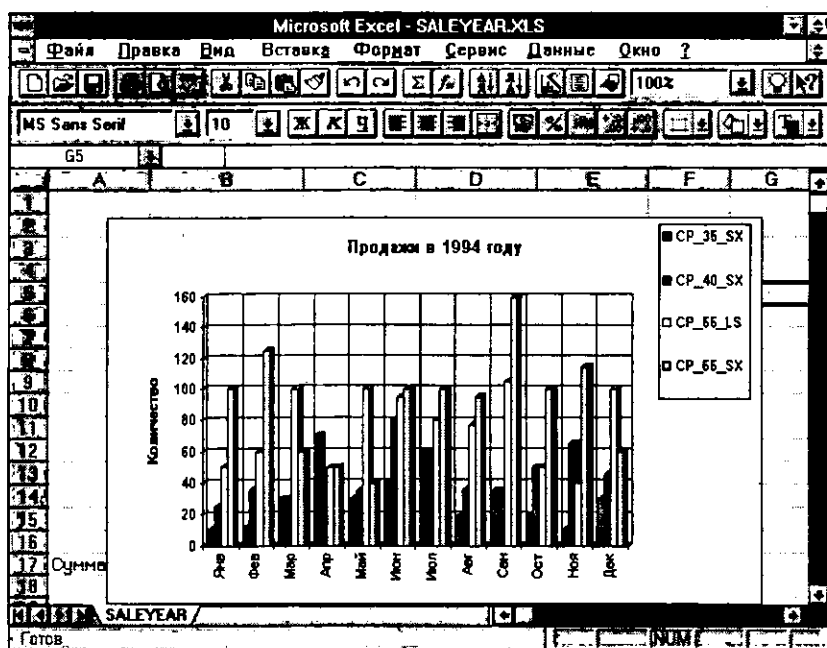


Рис. 16.16. Диаграмма для таблицы SALEYEAR.

Электронные таблицы содержат множество разнообразных типов диаграмм, позволяют вам настраивать формат отображения данных, заголовок диаграммы, надписи осей, параметры сетки, шрифт и формат выводимых значений и т.п. Для получения более подробной информации о создании диаграмм в электронных таблицах обратитесь к руководству пользователя той электронной таблицы, с которой вы будете работать.

Заключение

- dBASE поддерживает два механизма обмена с другими приложениями Windows: динамический обмен данными (Dynamic Data Exchange — DDE) и механизм работы с составными объектами (Object Linking and Embedding — OLE).
- Приложение, которое является источником при обмене данных называется *сервером*. Приложение, выполняющее роль приемника данных называется *клиентом* (*контейнером* — в OLE 2.0).
- При встраивании OLE-объекта вы можете создать новый объект или указать в качестве объекта содержимое файла. Для встраивания OLE-объекта выберите **Правка | Вставить объект** или **Правка | Вставить**.
- Для связывания OLE-объекта выберите **Правка | Вставить связь**.
- Для построения диаграммы используйте электронную таблицу. Предварительно подготовьте в dBASE данные для построения диаграммы в формате таблицы dBASE или в формате той электронной таблицы, которую вы собираетесь использовать.

Глава 17. Создание справочной системы

Для успешной эксплуатации разработанное вами приложение должно быть дружелюбным к пользователю. При возникновении затруднений пользователь должен быстро и просто получить справку о возможных действиях. Одним из средств обеспечения дружелюбности является наличие в приложении справочной системы, содержащей информацию о приложении, описание его основных функций и инструкцию по работе с приложением.

dBASE предоставляет вам различные средства для создания справочной системы. Для отображения справочной информации может использоваться строка состояния. Кроме того, вы можете создать справочную систему в стандарте Windows и использовать ее в приложении. При создании справочной системы приложения могут использоваться оба перечисленных средства, каждое в соответствующем месте.

Использование строки состояния

Для определения справочной информации для объектов интерфейса служит категория Справка свойств объекта. В этой категории вы можете определить сообщение, которое будет выводиться в строке состояния при установке фокуса на этом объекте, а также связать с объектом интерфейса определенную тему из заданного файла справочной системы в стандарте Windows.

В данном разделе определим сообщение, связанное с объектом интерфейса. Для этого перейдите в поле ввода значения свойства `StatusMessage` и введите поясняющий текст. На рис. 17.1 показано определение сообщения для поля "Постоянный" формы ввода списка клиентов, а на рис. 17.2 отображение в строке состояния сообщения, поясняющего назначение поля.

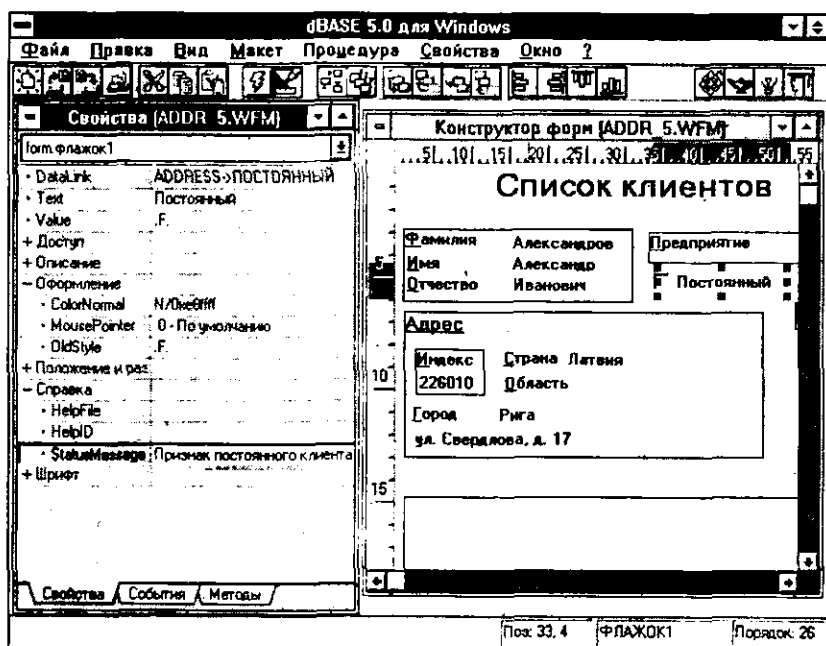


Рис. 17.1. Определение сообщения для поля.

дBASE 5.0 для Windows

Файл Правка Вид Форма Свойства Окно ?

Форма

Список клиентов ◀ Назад ▶ Далее ✓ Закрыть

Фамилия: Алексеев Предприятие: АО Интерсемс

Имя: Сергей

Отчество: Иванович ☒ Постоянный

Адрес: Индекс: 630050 Страна: Россия Телефон: 3832243124

Область: Номер: 3832243124

Город: Новосибирск

ул. Ильича, д. 34

Осуществляет поставку и гарантийное обслуживание IBM PC и совместимых с ними компьютеров.
В настоящее время расширяет перечень предлагаемых услуг и товаров: активно выходит на рынок локальных и глобальных сетей.

Признак постоянного клиента Address.db Зан 1/30

Рис. 17.2. Вывод сообщения в строке состояния.

Для пояснения назначения объектов могут использоваться заголовки исходной таблицы, но для начинающих пользователей будет полезным более развернутое описание назначения поля.

Создание и использование справочной системы в стандарте Windows

Справочная система в стандарте Windows

Далее мы кратко остановимся на основных принципах построения справочной системы в стандарте Windows.

Основой справочной системы являются темы, содержащие пояснительный текст. Содержание справочной системы включает список тем, доступных в системе. Каждая тема может иметь заголовок, уникальный идентификатор и ключи для поиска тем. Дополнительно каждой теме можно поставить в соответствие уникальный индекс темы, который должен быть целым числом. Для организации контекстного вызова темы из справочной системы вы можете использовать числовые значения индексов или же значения ключа. Использование идентификаторов тем для контекстного вывода справочной информации на экран не допускается.

В справочной системе вы можете осуществить поиск темы по значению ключа. Для этого выберите опцию Поиск и на экране появится окно диалога "Поиск", в котором ключи тем отображаются в разделе "Ключи", а заголовки тем — в разделе "Темы". Каждая тема может иметь более одного ключа поиска. В этом случае эта тема будет выбрана в окно "Темы" при вводе для поиска любого из ключей. Кроме того, несколько тем могут иметь одинаковые ключи. В этом случае при вводе ключа в окне "Темы" будут отображаться заголовки всех тем, ключ которых совпадает с указанным значением. Из этого списка вы можете выбрать интересующую вас тему и просмотреть ее содержимое.

Организация перекрестных ссылок позволяет связать между собой отдельные темы. Текст, используемый для организации перекрестной ссылки, выделен зеленым цветом и подчеркиванием.

Как правило, в описании системы используются термины, специфичные для конкретной системы. Например, в системах бухгалтерского учета такими терминами будут проводка, дебет, кредит. В качестве термина может рассматриваться любая фраза из текста темы. Справочная система Windows позволяет дать каждому термину приложения краткое определение. Термины, имеющие определения, на экране выделены зеленым цветом и пунктирным подчеркиванием. Если установить курсор мыши на термин, для которого определено краткое описание, при нажатии на кнопку мыши это описание появится на экране в рамке поверх текста темы.

Создание справочной системы включает следующие этапы:

1. Создается описание справочной системы в любом текстовом редакторе, поддерживающим формат RTF. Структура документа должна соответствовать требованиям, предъявляемым компилятором справочной системы (Help Compiler).
2. Создается текстовый файл с расширением .HPJ, который содержит параметры для компиляции справочной системы, в том числе имя файла с описанием справочной системы и заголовок справочной системы.
3. Создается стандартный файл справочной системы путем компиляции ее текстового описания. Для этого используется Help Compiler, входящий в состав многих продуктов фирм Borland (например, BORLAND C++) и Microsoft. В BORLAND C++ для запуска компилятора используется командный файл HC.BAT.
4. В приложении определяются свойства HelpFile и HelpID объектов, предназначенные для вызова справочной системы. Вызов справочной системы осуществляется при нажатии клавиши F1. Свойство HelpFile содержит имя файла справочной системы, а свойство HelpID — конкретную тему справочной системы.

В рамках данной главы мы рассмотрим только наиболее часто используемые средства при создании справочной системы. Подробную информацию вы можете найти в руководстве по применению Help Compiler.

Создание тем и ключей для их поиска

В предыдущем разделе вы познакомились с принципами построения справочной системы в стандарте Windows. Давайте рассмотрим непосредственно процесс подготовки справочной системы, который включает в себя создание описания в текстовом редакторе (например, Word для Windows) с последующей перекодировкой в формат RTF.

Для начала создадим простую справочную систему, состоящую из нескольких тем, для каждой из которых определим заголовки, идентификаторы и ключи для поиска.

Прежде всего создайте в редакторе Word для Windows новый файл. После этого введите тексты тем справочной системы. В тексте вы можете форматировать отдельные части темы, а также вставлять в текст рисунки. Каждая тема справочной системы должна находиться на отдельной странице. Поэтому для разделения тем вставьте между ними признак начала новой страницы путем нажатия комбинации клавиш **Ctrl+Enter** (рис. 17.3).

Теперь вы можете определить для этих тем заголовки, идентификаторы и ключи поиска. Для этого необходимо пользоваться сносками. Указатели сноски должны

располагаться в начале темы, а сам текст заголовка, идентификатора и ключа поиска — в области сноски.

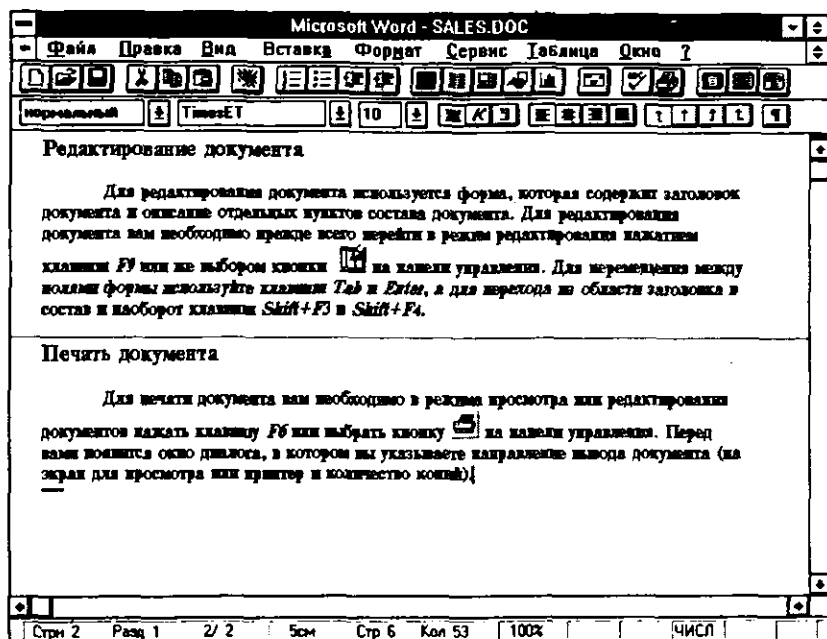


Рис. 17.3. Ввод текстов тем с рисунками.

Прежде всего определим заголовок темы. Для этого установите курсор в начало темы и выполните команду Вставка | Сноска. В окне диалога для определения сноски выберите опцию **Нестандартный маркер**, а в качестве маркера сноски используйте символ \$. Затем введите текст сноски, который будет являться текстом заголовка темы (рис. 17.4).



Между маркером сноски и текстом заголовка должен быть только один пробел. Заголовок не должен содержать более 128 символов и не может быть форматированным.

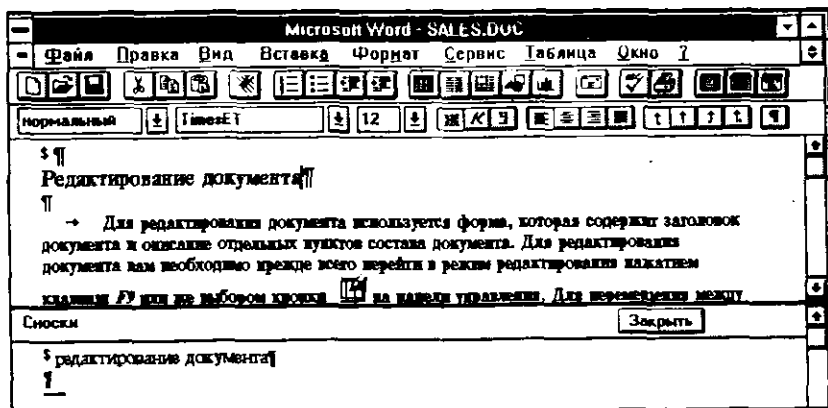


Рис. 17.4. Определение заголовка темы.

Идентификатор темы определяется аналогично заголовку темы. В качестве маркера сноски используйте символ #. Идентификатор может содержать латинские буквы, цифры и символы точка (.) и подчеркивание (_). Длина идентификатора может быть до 255 символов, но в практике ее уменьшают до разумных пределов. Имя идентификатора должно быть уникальным, поскольку оно используется для организации перекрестных ссылок.

При определении ключей в качестве маркера сноски используйте латинский символ "к". В тексте сноски вы можете определить несколько ключей поиска, разделив их знаком (;). Общая длина строки с ключами поиска должна быть не более 255 символов. Ключи могут содержать любые символы ANSI, включая пробелы и знаки пунктуации (за исключением точки с запятой). При поиске тем по ключам регистр введенных букв не учитывается.

На рис. 17.5 показан вид справки при вызове справочной системы и на рис. 17.6 — окно диалога "Поиск" справочной системы.

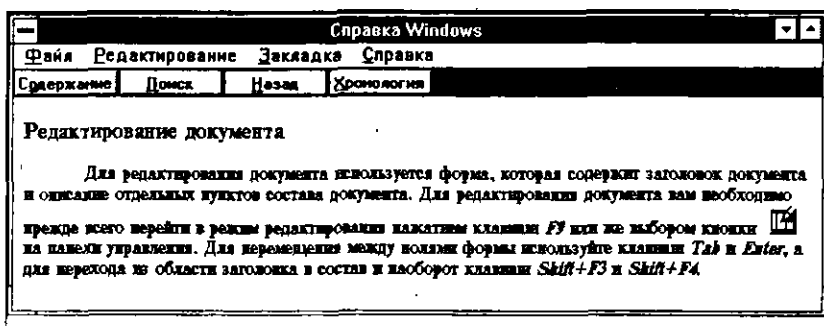


Рис. 17.5. Справочная система.

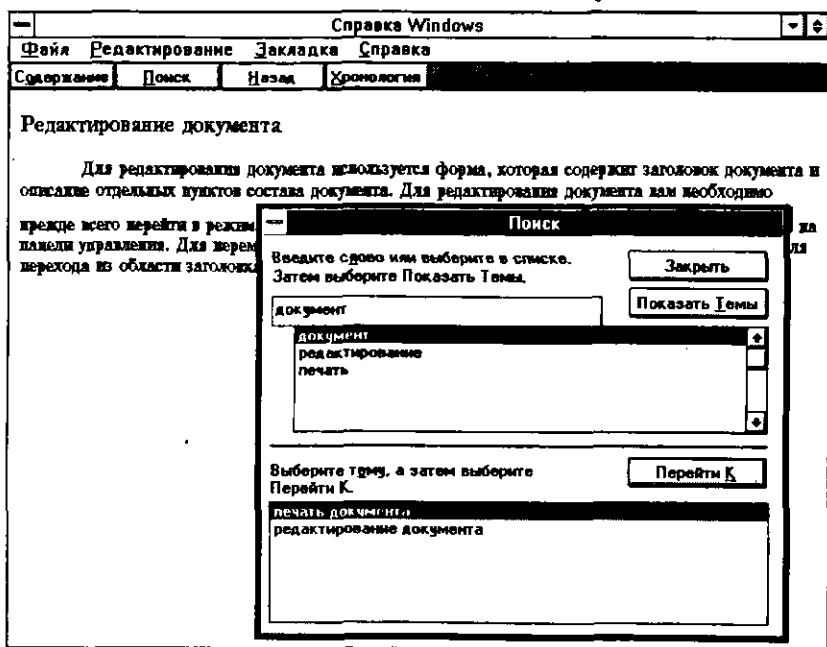


Рис. 17.6. Поиск темы по ключам справочной системы.

Далее вам необходимо откомпилировать текстовое описание справочной системы.

Трансляция справочной системы

Транслятор может обрабатывать описание только в формате RTF, поэтому прежде всего перекодировать созданное описание из формата Word для Windows в формат RTF. Затем создайте файл проекта справочной системы, содержащий параметры ее компиляции. Файл проекта может содержать до шести разделов, которые выполняют следующие функции:

Раздел	Функция
[Files]	Определяет список файлов с описанием тем.
[Options]	Определяет заголовок справочной системы и некоторые другие параметры трансляции.
[BuildTags]	Определяет список тем, включаемых в справочную систему.
[Alias]	Определяет псевдонимы тем.
[Map]	Назначает число идентификатору темы.
[Bitmaps]	Определяет список графических файлов, включаемых в справочную систему.

Раздел [Files]

В этом разделе перечисляются файлы в формате RTF, включаемые в состав справочной системы. Например:

```
[Files]
SALES.RTF      ; инструкция пользователя приложением SALES
COMMON.RTF    ; инструкция по общим правилам ввода данных
```

Раздел [Options]

Раздел [Options] содержит описание параметров трансляции. Рассмотрим основные из них.

Параметр TITLE определяет заголовок окна справочной системы. Заголовок должен содержать не более 32 символов.

Синтаксис определения заголовка:

TITLE=имя заголовка

Параметр COMPRESS используется для сжатия справочного файла, создаваемого компилятором. Параметр может принимать одно из двух значений TRUE или FALSE.

Параметр INDEX определяет идентификатор содержания справочной системы. По умолчанию при вызове справочной системы в качестве содержания используется первая по порядку тема. Если вам необходимо назначить в качестве содержания какую-либо другую тему, присвойте параметру INDEX идентификатор требуемой темы.

Например:

INDEX=основное меню

Параметр WARNING определяет количество сообщений об ошибках и предупреждениях, выводимых при компиляции.

Синтаксис определения параметра:

WARNING=уровень

где **уровень** может принимать одно из следующих значений:

Уровень	Выводимая информация
1	Только наиболее серьезные ошибки.
2	Средний уровень предупреждений.
3	Все предупреждения.

Если в описании тем справочной системы использовались шрифты, которые отсутствуют на компьютере, где будет выполняться приложение, используйте параметр FORCEFONT. Этот параметр указывает компилятору преобразовать все используемые в описании шрифты к одному типу. Тип шрифта присваивается параметру FORCEFONT. Наименование шрифта должно совпадать с его именем, которое показывается в окне диалога программы Панель Управления Windows. Пробелы в наименовании типа шрифта не допускаются.

Раздел [Map]

Раздел [Map] является обязательным при организации контекстной ссылки в справочной системе. Он содержит описание связи между идентификаторами тем и соответствующими им целыми значениями индекса тем. Если вы собираетесь использовать в приложении контекстный вызов справочной системы, вам необходимо описать в этом разделе все идентификаторы, по которым будет осуществляться контекстный вызов. Пример использования раздела [MAP] будет приведен ниже при рассмотрении процесса организации контекстного вызова.

Пример трансляции справочной системы

В качестве примера файла проекта приведем его содержимое для трансляции созданного выше текстового описания справочной системы:

```
[FILES]
sales.rtf; файл с описанием тем справочной системы
TITLE=Справочная система
WARNING=2
```

Организация перекрестных ссылок

Перекрестные ссылки являются наиболее мощным средством справочной системы. Используя их, вы можете из текущей темы перейти в связанные с ней или близкие ей темы.

В предыдущем разделе мы создали две темы и для обеих определили идентификаторы. Теперь создадим еще одну тему, которая будет содержать общее описание работы с документами, в которой кроме этого будет приведена ссылка на связанные с ней темы. Текст темы приведен на рис. 17.7.

Следующим шагом создадим ссылку между фразой "Редактирование документа" этой темы и темой с идентификатором редактирование документа. Для этого

установите для фразы “Редактирование документа” формат Двойное подчеркивание. Далее установите курсор на следующую позицию за выделенной фразой, определите формат Скрытый и напечатайте идентификатор темы, с которой устанавливается связь (в данном случае: редактирование документа). Идентификатор должен располагаться сразу же за выделенной фразой. Пробелы или любые другие символы между ними не допускаются.

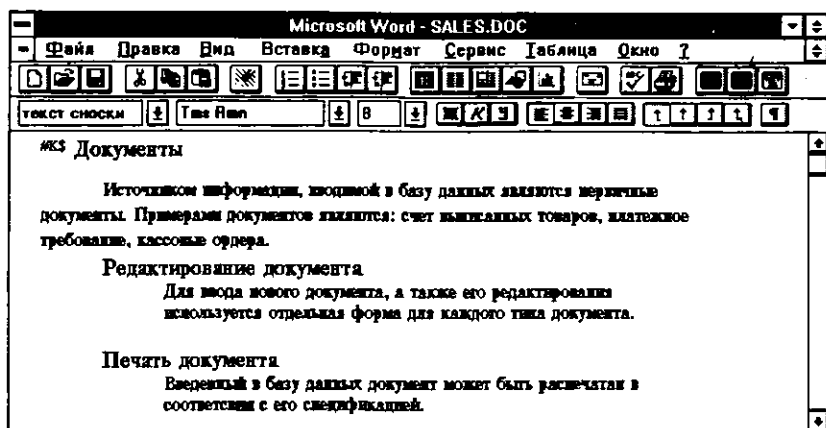


Рис. 17.7. Содержание справочной системы.



Обычно скрытый текст не отображается на экране, что довольно неудобно при создании описания справочной системы. Для отображения скрытого текста установите в окне “Опции” (команда Сервис | Опции) для категории **Просмотр** режим отображения скрытого текста.

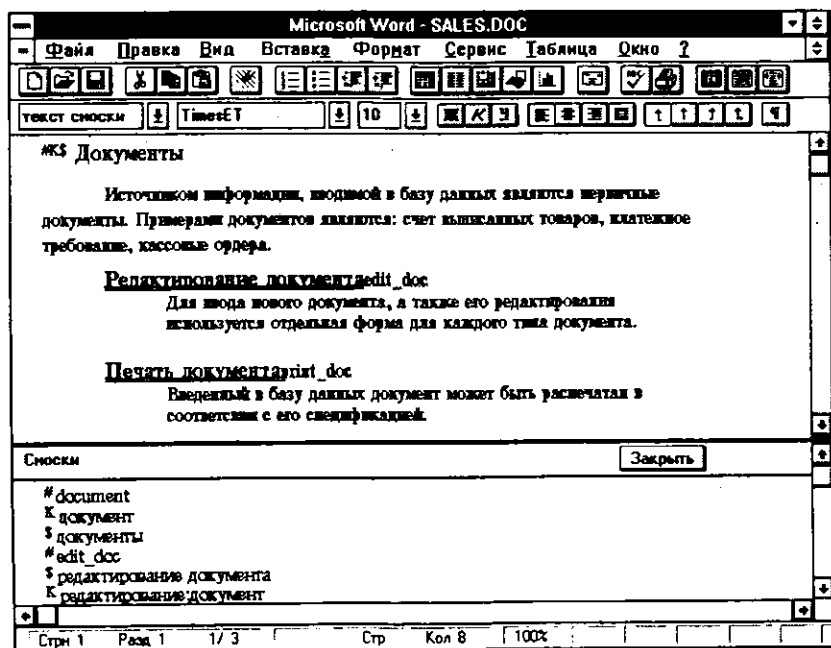


Рис. 17.8. Описание перекрестных ссылок.

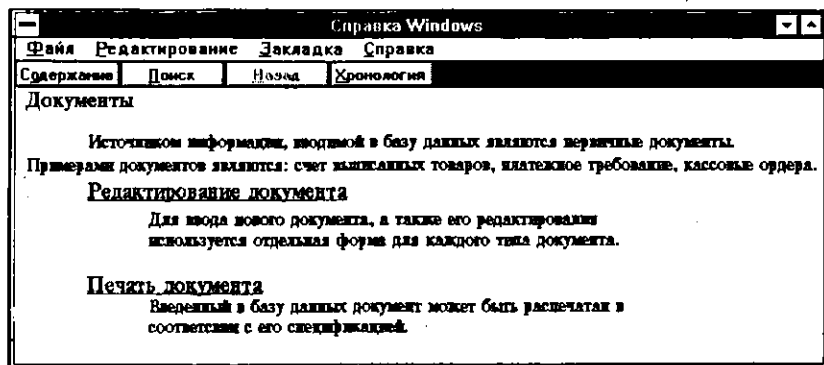


Рис. 17.9. Перекрестные ссылки.

Далее установите связь фразы “Печать документа” с темой **печать документа**. На рис. 17.8 приведен окончательный текст темы, а на рис. 17.9 — его вид при вызове справочной системы. Несмотря на то, что в тексте фраза была установлена в формате с двойным подчеркиванием, в справочной системе она появляется с обычным подчеркиванием. При позиционировании курсора мыши на выделенной фразе меняется вид курсора, а при последующем нажатии на кнопку мыши осуществляется переход в связанную с этой фразой тему.

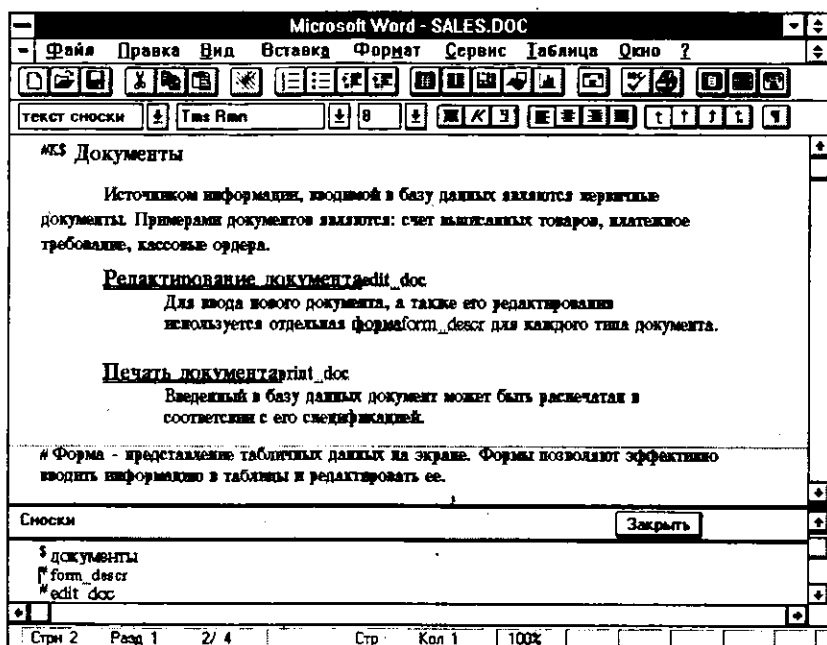


Рис. 17.10. Описание пояснения термина “форма”.

Создание пояснений к терминам

В справочной системе в дополнение к перекрестным ссылкам вы можете использовать удобное средство быстрого отображения пояснения к отдельным терминам. Пояснение создается также, как отдельная тема, и для нее определяется идентифика-

тор темы. После этого термин или любая фраза, требующая кратких пояснений, выделяется одинарным подчеркиванием и сразу за термином в скрытом формате вводится идентификатор поясняющей темы (рис. 17.10).

При вызове справочной системы термин выделяется цветом и пунктирным подчеркиванием. При нажатии кнопкой мыши на термин, имеющий пояснение, открывается окно, в котором отображается пояснение к тексту (рис. 17.11). При нажатии кнопки мыши вне окна с текстом пояснения, это окно закрывается и вы опять возвращаетесь в то же окно, откуда вызвали окно пояснения. В этом состоит основная разница между использованием пояснений и перекрестных ссылок.

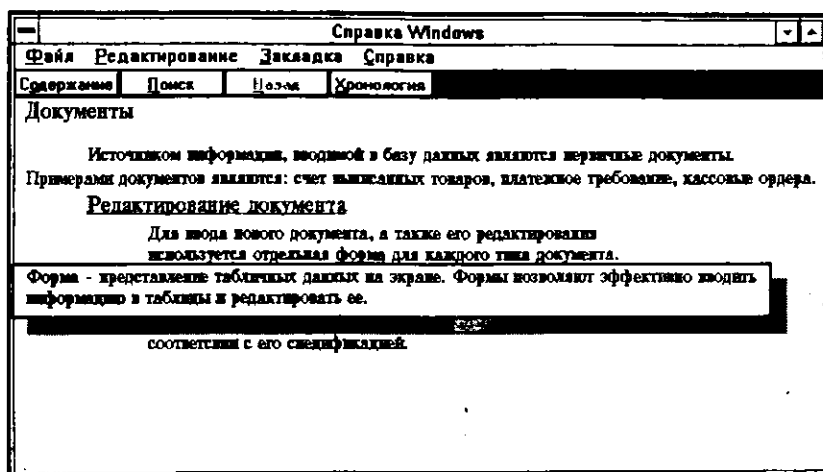


Рис. 17.11. Отображение пояснения к термину "форма".

Глава 18. Новые возможности dBASE для Windows версии 5.5

Версия dBASE для Windows 5.0 явилась новым шагом в развитии dBASE. В ней были реализованы все последние достижения современных СУБД, включая визуальное объектно-ориентированное программирование, запросы по образцу, использование экспертов и инструкторов при создании форм.

Совсем скоро, в начале осени 1995 г., в продаже появится новый продукт компании Borland Int — dBASE для Windows версия 5.5. Надеемся, что к тому времени, когда вы, уважаемый читатель, будете держать в руках нашу книгу, вам будет доступна новая версия, которая объединяет интуитивный и динамичный язык 4-го уровня с расширенным рядом инструментальных средств для ускоренной разработки приложений. Средства разработки dBASE 5.5 смогут удовлетворить как опытных программистов, так и пользователей, которым приходится прибегать к программированию время от времени. Кроме этого dBASE будет полезен профессиональным разработчикам приложений клиент/сервер.

Совершенно новым для dBASE является возможность создания выполняемых программ, т.е. программ с расширением .EXE. Данное средство позволяет создавать законченные приложения, которые не будут требовать наличия у конечного пользователя системы dBASE и имеет неоспоримые преимущества при тиражировании разработанного продукта.

В настоящей главе нам хотелось бы поделиться своими впечатлениями о бета-версии нового dBASE для Windows.

Новые эксперты

Первое, что обращает внимание при работе с новой версией dBASE 5.5, это наличие большого количества разнообразных экспертов, которые помогают вам при создании таблиц, форм, отчетов, этикеток. Использование экспертов позволяет вам создавать законченные формы и другие объекты, позволяя в дальнейшем модифицировать их с учетом конкретных требований. Кратко остановимся на каждом из этих экспертов.

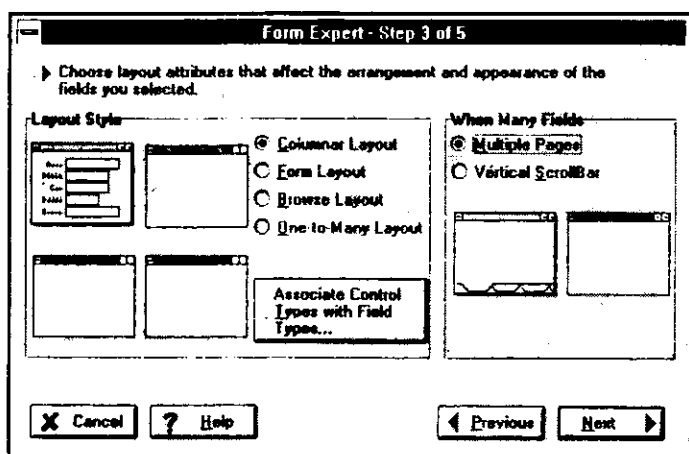


Рис. 18.1. Выбор вида формы и расположения страниц.

Эксперт по созданию формы

Новый эксперт по созданию форм стал полностью настраиваемым. Теперь после стандартных шагов по выбору таблицы и ее полей, отображаемых в форме, в дополнение к выбору вида формы, вы можете указать режим отображения страниц и связать элементы управления с типами полей (рис. 18.1) Новым является возможность создавать формы, в которых страницы отображаются в виде карточек с корешками. Это очень наглядный способ отображения многостраничных данных. Для перехода на страницу достаточно выбрать соответствующий ей корешок.

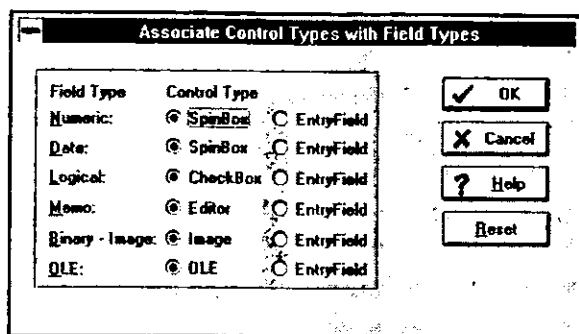


Рис. 18.2. Связывание типов полей с типами управляющих объектов.

Для каждого типа поля формы вы можете указать возможность непосредственного ввода данных или же ввод данных с помощью связанных с этим типом управляющих объектов. Например, для ввода логических значений могут использоваться флажки, а для ввода значений в числовые поля — счетчики (рис. 18.2).

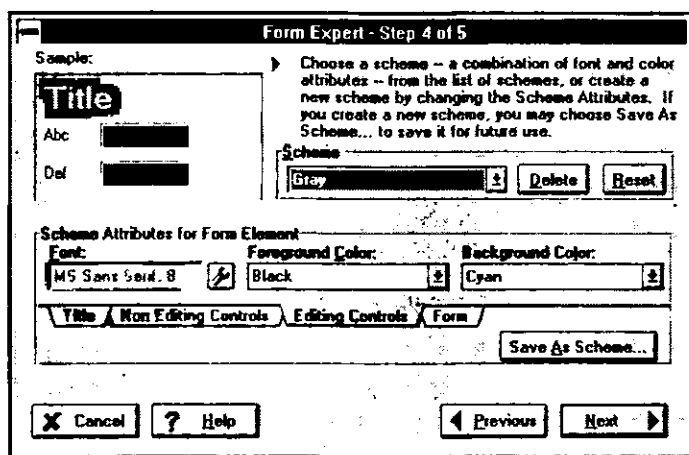


Рис. 18.3. Окно диалога для определения схемы отображения данных.

Для формы вы можете задать схему отображения данных, которая включает совокупность цветов и шрифтов, определенных для различных групп объектов интерфейса. Окно диалога (рис. 18.3), используемое для задания схемы, содержит отдельные карточки для каждой группы объектов. В dBASE входит большое количество созданных ранее схем, которыми вы можете воспользоваться. Кроме того, вы можете изменить ряд параметров выбранной схемы и сохранить ее под новым именем.

Использование схем является одним из элементов стандартизации программных средств в вашей организации и сэкономит немало времени при создании формы.

После завершения создания формы с помощью эксперта вы можете запустить ее на выполнение или же продолжить работу над ней с помощью конструктора форм.

Эксперт по созданию таблицы

Эксперт по созданию таблицы окажет существенную помощь при создании таблиц, предлагая на ваш выбор большой набор шаблонов таблиц. Эксперт содержит два шага создания таблицы. На первом вы выбираете подходящий вам шаблон таблицы и поля из выбранного шаблона (рис. 18.4). На втором шаге выбирается тип таблицы и задается ее имя.

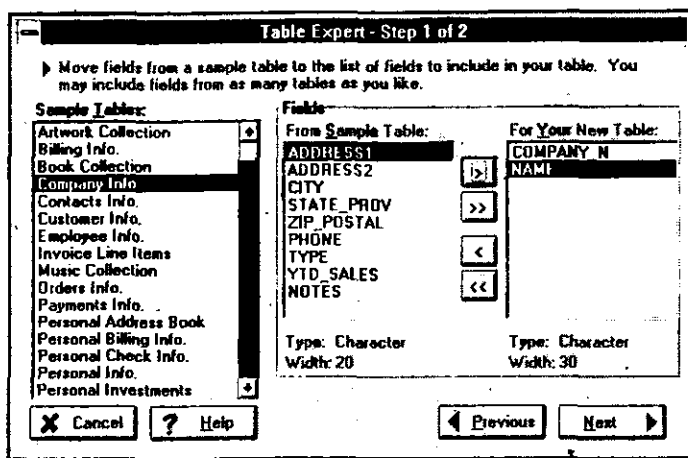


Рис. 18.4. Выбор шаблона таблицы и полей из этого шаблона.

Эксперт по созданию отчетов

Создание отчета с помощью эксперта состоит из четырех шагов:

- Шаг 1. Запрашивает у пользователя имя источника данных для создания отчета. В качестве источника данных может использоваться таблица или созданный ранее запрос.
- Шаг 2. Выбираются поля из источника данных, определенного на первом шаге.
- Шаг 3. Выбирается стиль отображения данных в отчете из списка заданных стилей:
 - Columnar (в одну колонку)
 - Tabular (табличный)
 - Group (группировка)
 - Group with Summary (группировка с итогами)

Кроме этого на данном шаге вы можете определить заголовок отчета и установить признак отображения номеров страниц и даты в отчете(рис. 18.5).

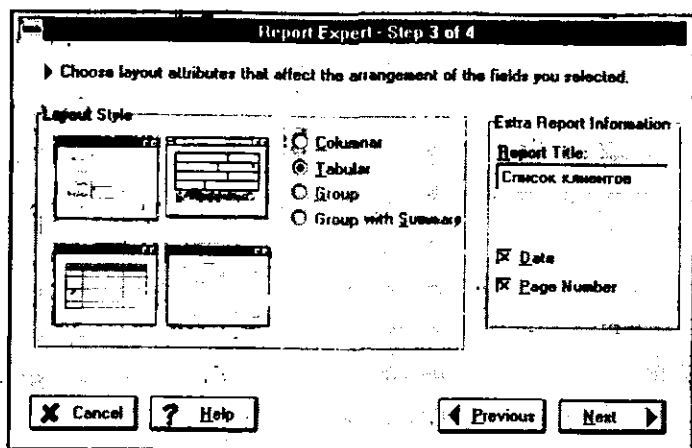


Рис. 18.5. Создание отчета с помощью эксперта.

Шаг 4. На последнем шаге вы можете сохранить созданный вами отчет и запустить его на выполнение или же продолжить дальнейшее улучшение в конструкторе отчетов.

Эксперт по созданию этикеток

Аналогично эксперту по созданию отчета, данный эксперт содержит четыре шага создания этикеток. На первом шаге у пользователя запрашивается имя таблицы или запроса. На втором этапе формируется сама этикетка. При формировании этикетки могут использоваться элементы, которые приведены в окне эксперта:

- Поля из таблиц, выбранных на первом шаге
- Специфические объекты текста: запятые, точки, дефисы, двоеточия и пробелы
- Текст
- Вычисляемые поля

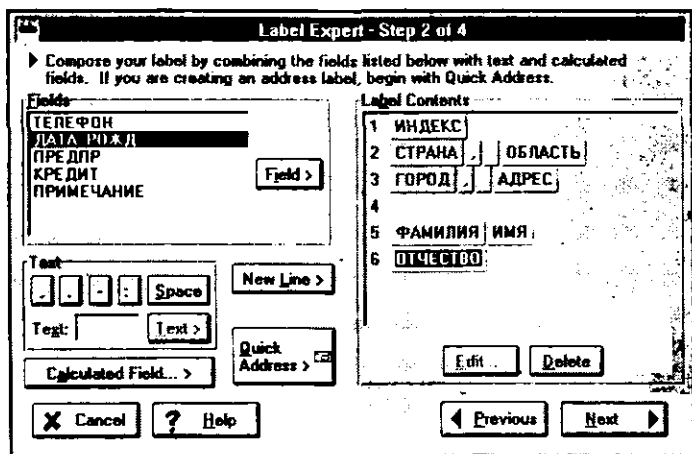


Рис. 18.6. Формирование этикетки с помощью эксперта.

Как только вы выберете эти элементы, они сразу же добавятся к текущей выделенной строке содержания этикетки. Кнопка **New Line** добавляет новую строку к содержанию этикетки и перемещает на нее фокус. Кнопка **Delete** удаляет элемент, на котором находился фокус, а если в строке находился только один элемент, удаляет также и строку. На рис. 18.6 приведен вариант заполнения этикетки на данном шаге ее создания.

Для быстрого создания этикетки вы можете прибегнуть к помощи кнопки **Quick Address**. При этом используются поля из выбранных вами таблиц и предполагается, что наименования полей соответствуют их смыслу. Эксперт позволяет интерпретировать в качестве полей отчета имена полей таблицы, которые подходят по смыслу. Например, для фамилии может выбираться любое из полей источника данных, имеющее имя **FNAME**, **F_NAME**, **FIRST_NAME**, **NAMEFIRST**.

На третьем шаге выбирается формат этикетки из большого списка предложенных экспертом. Для каждого формата заранее определены расположение на странице, расстояния от границ листа и расстояния между этикетками.

После выбора формата этикетки сохраните ее, а затем или запустите на выполнение или же продолжите дальнейшее улучшение в конструкторе отчетов.

Новые классы объектов

Для расширения возможностей dBASE в версии 5.5 введены новые классы объектов, а для имеющихся классов определены дополнительные свойства и методы.

Класс Popup

Объект класса **Popup** является всплывающим меню, которое может выводиться в любом месте формы. Этот объект имеет два параметра: ссылку на связанную с ним форму и имя объекта. Для создания объекта класса **Popup** используется конструктор меню, который вызывается командой **File | New | Popup**. Созданный с помощью конструктора меню файл сохраняется с расширением **.POP**.

Для отображения всплывающего меню на экране используется метод **Open()** объекта типа **Popup**. Почти во всех программах, работающих в среде **Windows**, для активизации всплывающего меню используется правая кнопка мыши. Для реализации этого подхода вы должны связать событие **OnRightMouseDown** формы с вызовом метода **Open()** соответствующего объекта класса **Popup**. Такой подход позволяет вызывать контекстные всплывающие меню.

Класс Shape

Объект класса **Shape** является цветовой областью внутри формы. Для этого объекта не определено свойство **Border**, но определена пара цветов: цвет фона и цвет переднего плана. Свойство **PenStyle** определяет стиль линии, ограничивающей выделенную область и может принимать следующие значения:

- 0 Непрерывная линия
- 1 Штриховая линия
- 2 Пунктирная линия
- 3 Штрих-пунктирная линия
- 4 Штрих-пунктирно-пунктирная линия

Кроме того, вы можете задать толщину линии и стиль объекта, который может принимать приведенные ниже значения.

- 0 Закругленный прямоугольник
- 1 Прямоугольник
- 2 Эллипс
- 3 Круг
- 4 Закругленный квадрат
- 5 Квадрат

Данное средство может быть использовано для выделения в форме группы объектов. На рис. 18.7 показано объединение группы однородных объектов с помощью цветовой области.

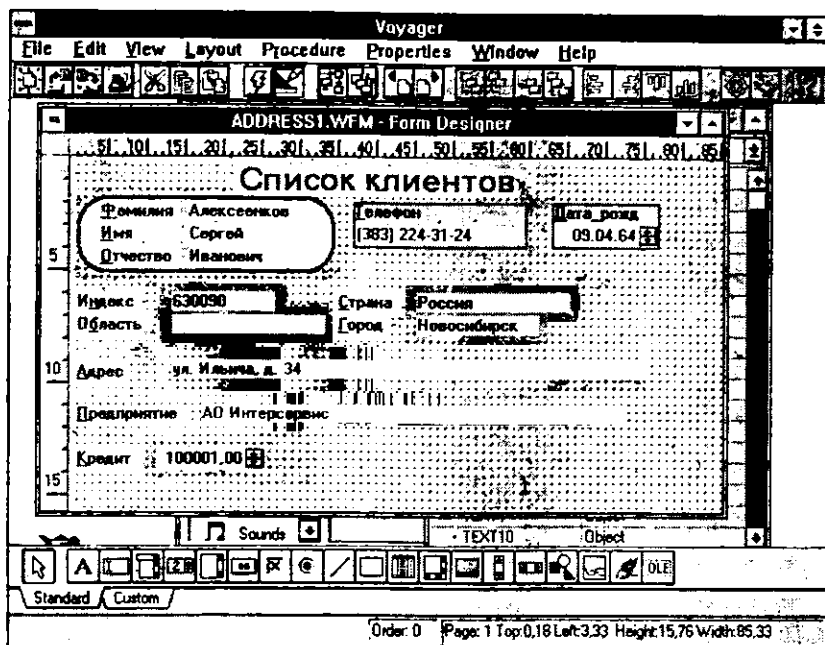


Рис. 18.7. Выделение области с помощью объекта класса Shape.

Класс PaintBox

Этот класс представляет собой прямоугольную область, которая имеет все стандартные свойства, такие как высота, ширина и т.п. Данный класс обрабатывает все стандартные события мыши, а также два новых события **OnPaint** and **OnFormSize**, которые позволяют вам изменять вид объекта при наступлении этих событий.

Класс TabBox

Класс **TabBox** предназначен для управления переходом между страницами формы с использованием корешков аналогично тому, как это делается в меню свойств объекта и окнах диалога установки параметров dBASE. Такой метод управления в последнее время широко используется во многих приложениях, работающих в среде Windows и практически является стандартом для перехода между страницами в многостраничных формах. В качестве примера в созданную ранее форму просмотра сведений добавим новую страницу, которая будет содержать дополнительные сведения о клиентах. Для удобства перехода между страницами создадим корешки этих страниц, каждому из которых присвоим соответствующей заголовок (рис. 18.8).

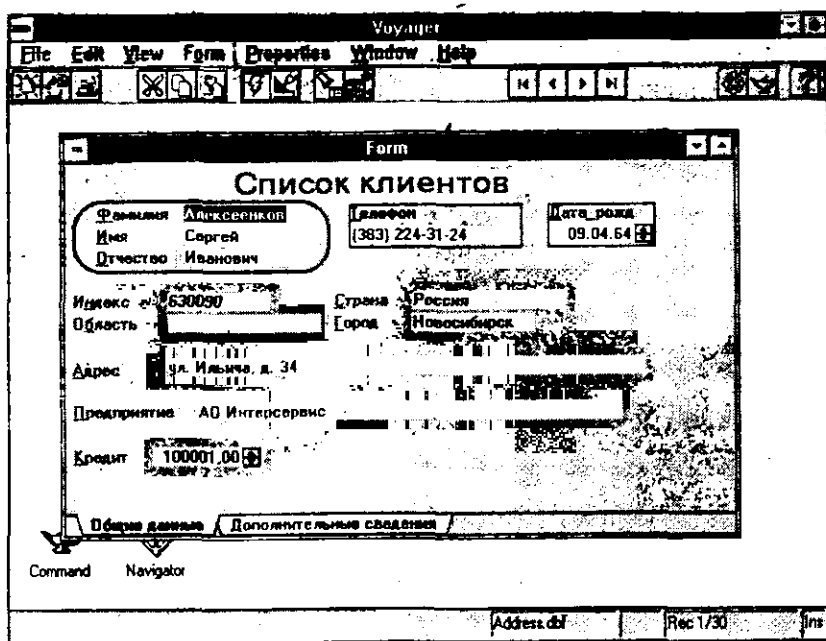


Рис. 18.8. Использование корешков в многостраничных формах.

Класс MenuBar

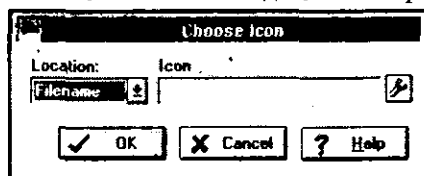
Данный класс определяет меню верхнего уровня для формы. По умолчанию конструктор меню при создании меню создает объект этого класса. MenuBars является особенно полезным, если вы собираетесь использовать ниспадающее меню Edit для выполнения операций с временным буфером данных.

Новые свойства и события объектов

В новой версии появилось значительное количество новых свойств, которые расширяют возможности существующих объектов. В данном разделе мы рассмотрим часть этих свойств, которые нам кажутся наиболее интересными.

Свойство Icon

Это свойство задает пиктограмму, которая будет отображаться на экране при минимизации формы. Для установки значения этого свойства выберите свойство Icon в меню свойств объекта, в результате откроется окно диалога выбора пиктограммы. Вы можете выбрать файл с расширением .ICO или пиктограмму из списка ресурсов динамической библиотеки DLL.



Свойства Hint и Showhint

В dBASE и большинстве других программ, работающих в среде Windows, для каждой кнопки панели инструментов в маленьком окне можно получить краткое описание данной кнопки, состоящее из одного или нескольких слов. Для этого достаточно установить курсор мыши на эту кнопку и не перемещать ее в течение секунды

или более. Теперь в dBASE можно определить аналогичное описание для любого объекта интерфейса формы. Для этого перейдите в меню свойств объекта и задайте текст или выражение для свойства **Hint** данного объекта. С данным свойством связано и другое свойство **Showhint**, которое определяет будет ли отображаться описание, заданное в свойстве **Hint**. Таким образом, вы можете настроить вашу систему, которая будет отображать краткое описание только для начинающих пользователей.

Свойство DesignView

Это свойство определяет источник данных (таблица или запрос), который будет использоваться при создании формы. Вы можете использовать данное свойство для облегчения создания и связывания формы, в которой не определено свойство **View**.

Событие Onchar

Данное событие связано с объектом, на котором находится фокус, и наступает при вводе печатного символа или комбинации клавиш. Событию передаются скан-код нажатой клавиши и количество повторных нажатий.

События Onkeydown и Onkeyup

Событие **Onkeydown** наступает при нажатии на любую клавишу, а событие **Onkeyup** при отжатии нажатой клавиши. Событиям передаются скан-код нажатой клавиши и количество повторных нажатий.

Конструктор форм

В dBASE 5.5 конструктор форм значительно расширил свои возможности. Прежде всего значительно улучшен эксперт по созданию формы. Появились новые классы базовых объектов. Изменилась к лучшему и среда конструктора форм.

Аналогично эксперту вы теперь можете связать с определенными типами данных типы объектов интерфейса. Дополнительно вы можете определить унифицированный размер для группы объектов. Для этого достаточно выделить объекты и выполнить команду меню **Layout | Size**.

Схема отображения данных

При создании формы с помощью эксперта вы могли задать схему отображения данных. Аналогичное средство имеется и в конструкторе форм. Для создания схемы или ее изменения выполните команду **Layout | Set Scheme** в окне конструктора форм, и на экране появится окно диалога для определения схемы. Пользователи, знакомые с окном диалога "Цвета" Панели Управления Windows, а также с аналогичным окном диалога в Windows 95, заметят, что настройка цветов осуществляется стандартным для Windows способом, наглядно отображающим все внесенные вами изменения.

Данное окно диалога отличается от аналогичного окна диалога эксперта только заголовком и отсутствием кнопок перехода к следующему шагу создания формы. Если вы выберете кнопку **Apply Now**, эффект от установки сразу же отразится в форме.

Использование специальных объектов интерфейса

При создании форм вы можете использовать минимально необходимый набор стандартных объектов интерфейса. Кроме того, вы можете использовать целый ряд специальных объектов, которые содержат не только его отображение, но и все связан-

ные с ним действия. В качестве специальных объектов могут использоваться объекты, созданные вами в среде dBASE (с расширением .CC) или объекты Visual Basic (с расширением .VBX).

Применение объектов интерфейса типа VBX, значительно расширяет возможности создаваемых форм. Теперь вы можете использовать тысячи различных объектов интерфейса, создаваемых третьими фирмами.

Если в предыдущей версии вы могли добавить новые объекты интерфейса, используя основное меню dBASE, то их удаление требовало коррекции файла DBASEWIN.INI. В новой версии вы можете выполнить команду **File | Setup Custom Controls**, которая открывает окно диалога (рис. 18.9), позволяющее добавить или удалить специальные объекты интерфейса.

В стандартную поставку входят три файла, содержащие специальные объекты интерфейса, которые можно включить в панель инструментов:

Файл	Объекты
BUTTONS.CC	Содержит множество кнопок для управления приложением. Эти кнопки позволяют вам передвигаться по записям в форме, а также выполнять ряд стандартных действий, используемых в dBASE.
PASSWORD.CC	Этот файл содержит объект интерфейса для ввода пароля. Объект интерфейса выглядит как обычное поле ввода, но при вводе символов они будут отображаться в виде звездочек. Введенное вами значение будет присвоено свойству EnteredPassword. Вы можете анализировать это свойство для обработки неправильного значения пароля.
TABBOX.CC	Этот файл содержит два объекта интерфейса, которые являются подклассами класса Tabbox. Первый из них, PageTabbox, при открытии отображает корешки каждой страницы формы. При этом каждый корешок содержит текст "Page <номер страницы>". Вторым объектом является SearchTabbox, который позволяет осуществлять поиск по буквам алфавита, отображаемым в этом объекте интерфейса.

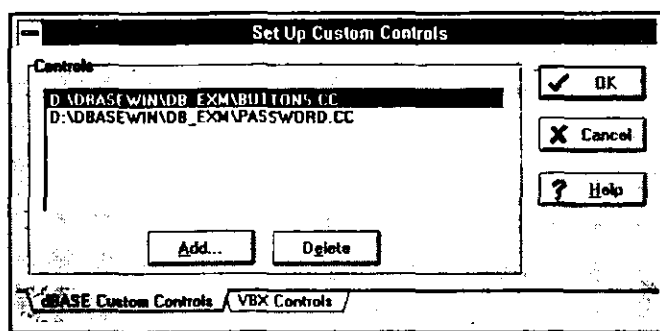


Рис. 18.9. Окно диалога для управления специальными объектами.

Кроме того, в версии 5.5 появилось новое средство, которое позволяет вам сохранить объекты интерфейса, созданные в форме, в качестве специальных объектов интерфейса.

Предположим, что вы создали форму и считаете, что в создаваемом вами приложении ряд объектов интерфейса будут использоваться довольно часто. В этом случае вы выделяете их, используя средства dBASE, а затем выполняете команду **File | Save Custom**. На экране появляется окно диалога "Save Custom" (рис. 18.10), которое позволяет вам сохранить описание выбранных объектов интерфейса в качестве специальных объектов (по умолчанию с расширением .CC).

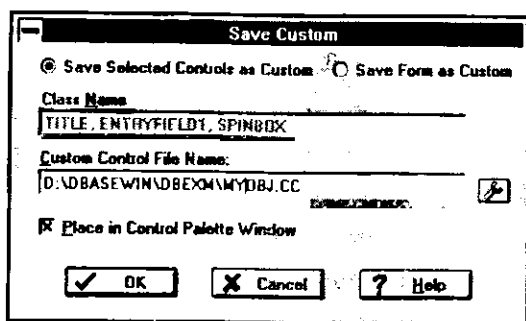


Рис. 18.10. Окно диалога для сохранения выбранных объектов интерфейса в качестве специальных.

Установите переключатель **Save Selected Controls as Custom**. Поле **Class Name** содержит список выделенных вами объектов интерфейса. Этот список недоступен для редактирования. Вы можете сохранить специальные объекты интерфейса в новом файле или добавить их в уже существующий файл. Имя файла указывается в поле **Custom Control File Name**. Если вы хотите немедленно отобразить выбранные вами объекты в карточке "Custom" панели элементов, установите флажок **Place in Control Palette Window**.

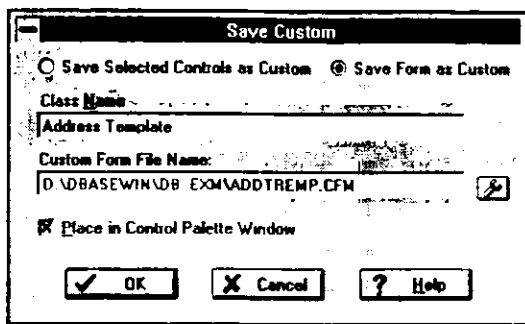


Рис. 18.11. Окно диалога для сохранения формы в виде базового класса.

Использование базовых форм

Совершенно новым средством является возможность сохранения созданной вами формы в виде базового класса. Это позволяет вам создать форму, которая имеет определенные атрибуты, такие как объекты интерфейса, цвета и другие, сохранить ее в виде базового класса, а затем использовать при создании целого ряда новых форм. Например, вы можете создать форму в качестве шаблона для отображения списка клиентов, который включает фамилию, имя и отчество клиентов, а также кнопки поиска и перемещения по записям. Вы можете сохранить созданный вами шаблон в виде базового класса. Выполните для этого команду **File | Save Custom**. На экране по-

является окно диалога "Save Custom" (рис. 18.11). Установите переключатель **Save Form as Custom**. В поле **Class Name** введите имя создаваемого базового класса, а в поле **Custom Form File Name** введите имя файла, которое будет содержать описание этого класса. По умолчанию создается файл с расширением **.CFM**. Возможность создания базовых классов иллюстрирует одно из фундаментальных преимуществ объектно-ориентированного подхода.

Теперь вы можете использовать созданный вами шаблон при разработке новых форм. Для этого выполните команду **File | Set Base Form Class** и в окне диалога "Set Base Form Class" выберите файл, содержащий требуемый класс, а затем из списка классов, хранящихся в данном файле, — требуемый класс (рис. 18.12).

После установки базового класса при создании новой формы, в нее автоматически добавятся все элементы формы, определенные в базовом классе. Вы можете добавить в форму дополнительные объекты интерфейса, расширяющие возможности базового класса.

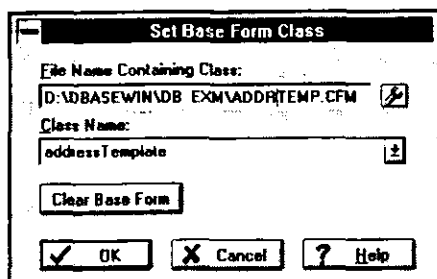


Рис. 18.12. Установка базового класса.

Новые возможности панели инструментов

Панель инструментов содержит ряд свойств, которые настраиваются в окне диалога "Control Palette Properties" (рис. 18.13). Вы можете установить режим отображения кнопок (только пиктограммы, только текст или текст и пиктограмма вместе), а также положение панели на экране (в верхней или нижней части экрана или же в произвольном месте экрана). Возможность настройки панелей является обычной для всех систем, работающих в среде Windows и приятно, что эта возможность поддерживается в новой версии.

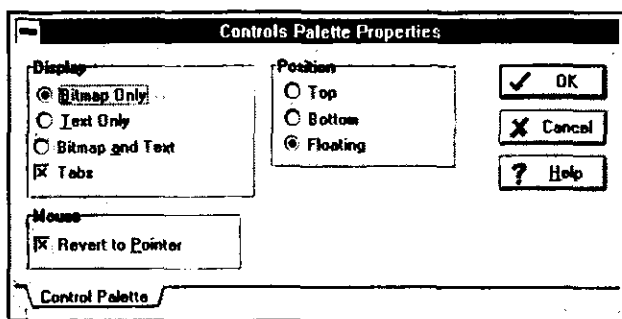


Рис. 18.13. Настройка свойств панели инструментов.

Конфигурирование панели инструментов с помощью таблицы регистрации классов

Специальные объекты интерфейса, входящие в состав панели инструментов отображаются в отдельной карточке с именем Custom. Для настройки панели инструментов вы можете использовать таблицу регистрации классов (CREG0009.DBF). Данная таблица для каждого из используемых в панели инструментов специальных объектов интерфейса определяет пиктограмму, которая будет отображаться на панели. Кроме этого вы можете объединить объекты в категории, каждая из которых будет отображаться на отдельной карточке. Внутри каждой карточки объекты могут объединяться в группы, которые разделяются увеличенным пустым пространством.

Панель полей

В новой версии появилась новая панель — панель полей. Эта панель содержит список всех полей источника данных формы (рис. 18.14). Аналогично панели инструментов, вы можете установить режим отображения и расположение панели полей. При размещении в форме объекта из панели полей автоматически устанавливается значение свойства DataLink размещаемого объекта.

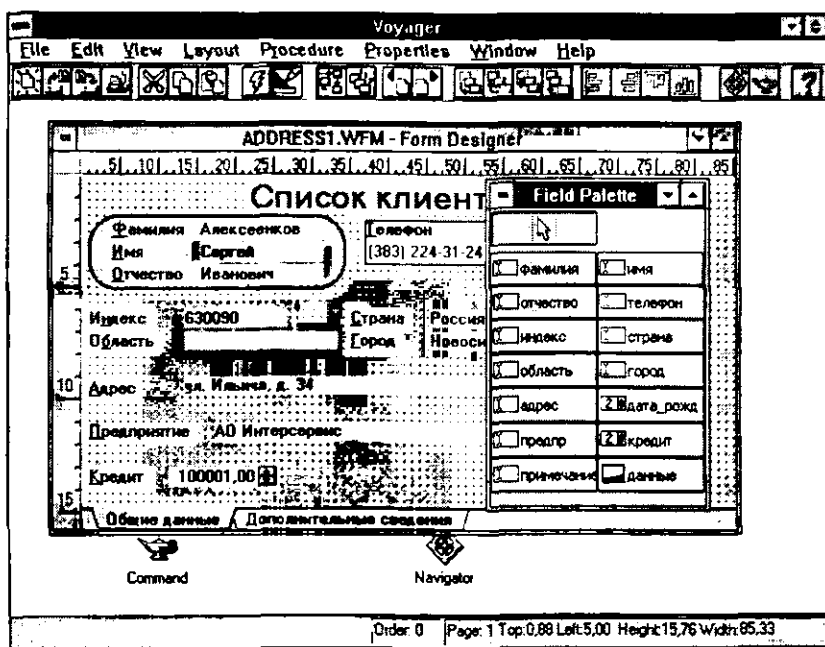


Рис. 18.14. Панель полей.

Целостность данных

В dBASE 5.5 появилась возможность определения целостности данных для таблиц Paradox и других баз данных, которые поддерживают целостность данных. Абсолютное большинство СУБД, с которыми dBASE может осуществить связь используя SQL-Link или интерфейс ODBC (например, ORACLE, INFORMIX, SYBASE), поддерживают целостность данных с помощью средств SQL. Для таблиц dBASE целостность данных не поддерживается. В dBASE целостность данных поддерживается на

уровне базы данных, в отличие от Paradox и PowerBuilder, которые поддерживают целостность данных на уровне таблиц.

Для определения целостности данных выполните команду **File | Database | Define Referential Integrity** и на экране появится окно диалога "Define Referential Integrity", которое содержит список всех установленных на данный момент условий выполнения целостности данных (рис. 18.15).

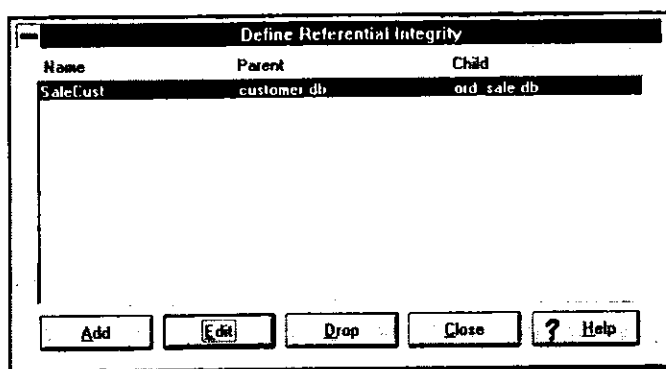


Рис. 18.15. Окно диалога "Define Referential Integrity".

Вы можете редактировать установленные в текущий момент условия целостности данных или же добавить новое условие. В любом случае на экране появляется окно диалога "Add Referential Integrity Rule" (рис. 18.16).

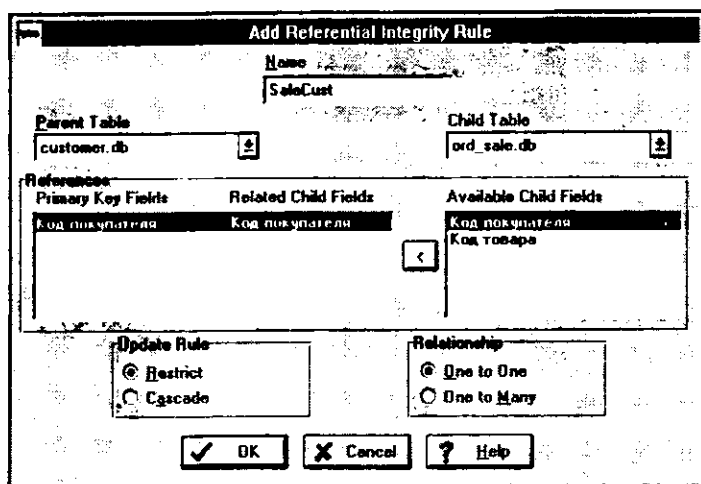


Рис. 18.16. Установка условия целостности данных.

Прежде всего вам необходимо выбирать первичную таблицу и таблицу, связанную с ней. С одной и той же первичной таблицей могут быть связаны несколько других таблиц. В то же время каждая таблица может быть связана с несколькими первичными таблицами. Первичная таблица должна иметь первичный ключ по связываемым полям. После выбора таблиц в связываемой таблице выбираются поля, которые соответствуют полям первичного ключа первичной таблицы.

Далее вы выбираете отношение между связываемыми таблицами один-ко-многим или один-к-одному, а также правила удаления и модификации данных: "Restrict"

или "Cascade". Если вы выберете "Restrict", то будет запрещен ввод, удаление и модификация значений ключевых полей, если на это значение имеется ссылка из какой-либо таблицы. Если же вы выберете "Cascade", то при удалении записи из первичной таблицы будут удалены все соответствующие записи из связанных таблиц.



В терминах SQL определение целостности данных осуществляется созданием внешних ключей.

Встроенный SQL

Теперь вы можете использовать SQL-операторы непосредственно в тексте программы dBASE. Необходимость в выполнении установки SET SQL ON или создании отдельного PRS-файла отсутствует. Встроенный SQL можно применить к любому источнику данных, начиная от локальных до удаленных, связь с которыми осуществляется с помощью ODBC или SQL-Link. Результатом выполнения SQL оператора SELECT является рабочая область, с которой вы можете выполнять все стандартные действия dBASE. В отличие от традиционных средств клиент/сервер, которые создают временные таблицы для хранения результатов выборки, dBASE создает таблицу, позволяя вам не только читать данные из нее, но и модифицировать их.

В операторах SQL вы можете использовать переменные из ваших программ. Это позволяет вам динамически определять условие выборки данных. Например, вы можете вводить в форме наименование города, а затем осуществить выборку всех клиентов из этого города. Перед переменной, используемой в встроенном SQL, необходимо напечатать двоеточие. Пример SQL выражения, использующего переменную приведен ниже:

```
SELECT * FROM customer WHERE ГОРОД = :getCity
// Предполагается, что значение переменной getCity
// определено в форме или программе
```

Ниже приведен список новых команд, поддерживающих встроенный SQL:

```
ALTER TABLE
COMMIT*
CREATE INDEX
CREATE TABLE
CREATE VIEW*
DELETE
DROP INDEX
DROP TABLE
DROP VIEW*
GRANT*
INSERT
REVOKE*
ROLLBACK*
SELECT
SET TRANSACTION*
UPDATE
```

Команды, отмеченные звездочкой (*), не могут применяться к локальным таблицам.

Защита данных и ограничение доступа

В dBASE 5.5. появились совершенно новые средства защиты доступа к данным с помощью пароля. Теперь определено понятие Администратора, который может определять группы пользователей, входящих в эту группу пользователей, а также пароли для доступа к отдельным таблицам и ограничивает доступ к полям защищаемой таблицы.

Для определения параметров защиты таблиц выполните команду **File | Database Administration** и в открывшемся окне диалога нажмите кнопку **Table Security**. На экране появится окно диалога “Administrator Password” (рис. 18.17), в котором вы вводите пароль администратора. Если пароль вводится в первый раз, dBASE предложит подтвердить введенный пароль.

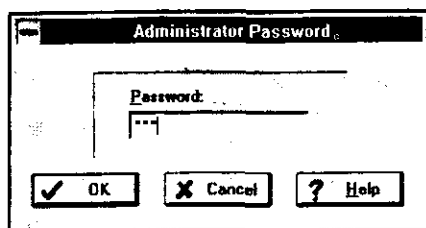


Рис. 18.17. Ввод пароля администратора.

Далее открывается окно диалога “Security Administration”, которое содержит три карточки “Users”, “Tables” и “Login”. В карточке “Users”, которая содержит список всех определенных пользователей (рис. 18.18), вы определяете группы пользователей и непосредственно пользователей этих групп. Для добавления нового пользователя выберите кнопку **Add**, а для изменения параметров пользователя — кнопку **Edit**. Если вы хотите удалить пользователя нажмите кнопку **Delete**.

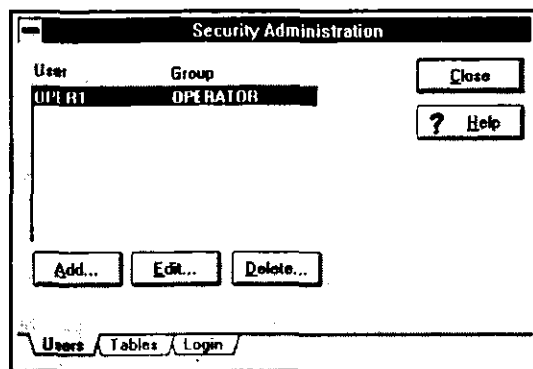


Рис. 18.18. Определение групп и пользователей, входящих в группы.

При добавлении нового пользователя на экране появляется новое окно диалога, “Add User” (рис. 18.19), которое содержит поля для ввода имени группы и пользователя, пароля, определенного для данного пользователя, уровня доступа, а также полного имени пользователя.

При редактировании пользователя вы можете изменить определенный для него пароль, уровень доступа и полное имя пользователя.

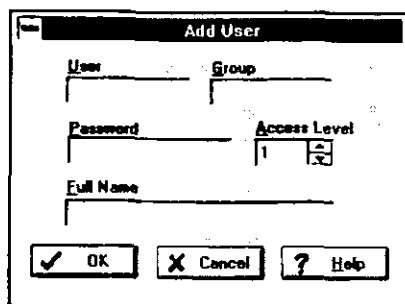


Рис. 18.19. Добавление нового пользователя.

Для определения пароля на таблицу и уровня доступа к полям таблицы используется карточка "Tables", в которой вы можете выбрать таблицу и нажать кнопку Edit. При этом на экране появляется окно диалога "Edit Table Privileges" (рис. 18.20). В этом окне диалога вы определяете необходимый уровень доступа для чтения, редактирования данных, а также добавления и удаления записей в таблице. Дополнительно вы можете определить для любого поля таблицы права для каждого из восьми уровней доступа.

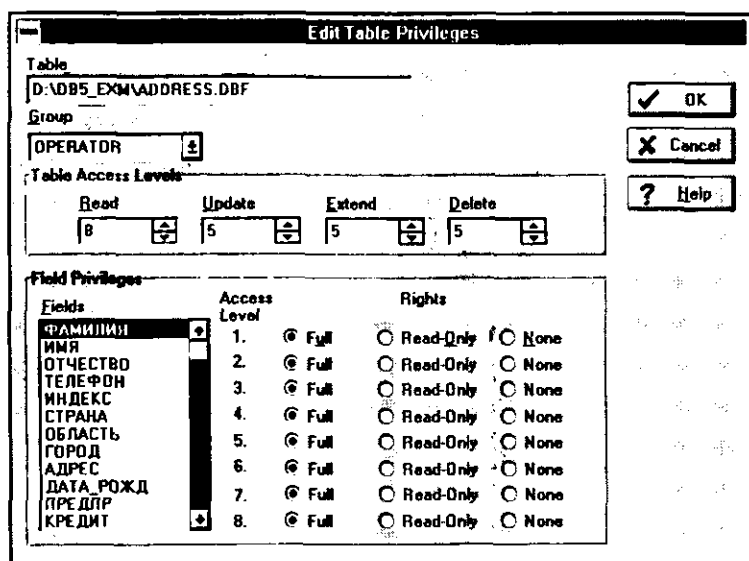


Рис. 18.20. Определение пароля таблицы и ограничений на поля.

И, наконец, последняя карточка "Login" определяет в какой момент будет запрашиваться пароль пользователя. Вы можете установить запрос пароля при запуске dBASE или же при открытии защищенной таблицы.

При открытии защищенных таблиц предварительно открывается окно диалога "Login", в котором вы должны ввести наименование группы, имя пользователя и пароль (рис. 18.21).

Для управления защитой таблиц введены следующие новые команды и функции:

Logout	Protect	Set Encryption
Access ()	User ()	

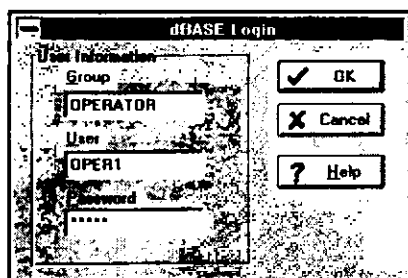


Рис. 18.21. Окно диалога "Login".

NULL значение

В версии dBASE 5.5 полностью поддерживается тип данных NULL. NULL в реляционных базах данных используется для индикации отсутствия значения, не смешивая его со значением 0 для числовых полей или пустой строкой для символьных полей. В dBASE теперь имеется системная константа NULL, которую вы можете присвоить переменным и полям таблицы, а также использовать в операциях сравнения. Если в какое-либо поле не введены данные, считается что значение этого поля равно NULL.

Расширенные средства работы с удаленными базами данных

Современные средства разработки приложений, как правило, включают возможность доступа к удаленным базам данных в режиме клиент/сервер. В dBASE и Paradox для этих целей служат SQL-Link и ODBC-драйверы. В новой версии добавлены средства, расширяющие возможности работы с удаленными базами данных.

Уровни изоляции транзакции

Функция BEGINTRANS() содержит параметр, задающий уровень изоляции транзакции. Синтаксис функции теперь имеет следующий вид:

```
BEGINTRANS( [ имяБазыДанных [ , уровеньИзоляции ] ] )
```

Параметр уровень изоляции может принимать одно из следующих числовых значений:

Значение	Назначение
0	Использует уровень изоляции принятый по умолчанию на сервере базы данных.
1	Dirty Read (Чтение данных, для которых обработка транзакции не завершена).
2	Read Committed (Чтение данных, для которых обработка транзакции завершена).
3	Repeatable Read (При чтении занятых данных, ожидает их освобождения и осуществляет повторное чтение).

Хранимые процедуры

Большинство серверов баз данных поддерживают хранимые процедуры, которые являются принадлежностью не отдельного приложения, а сервера базы данных и могут использоваться любым приложением, работающим с этим сервером.

Для выполнения хранимой процедуры вы должны объявить ее прототип в программе, а затем выполнять как обычную функцию dBASE.

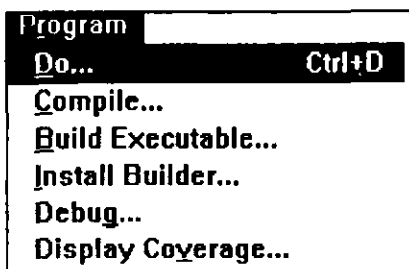
Для облегчения работы с хранимыми процедурами команда OPEN DATABASE имеет дополнительный параметр AUTOEXTERN, который автоматически объявляет все хранимые процедуры сервера базы данных. Поэтому при использовании этого параметра все хранимые процедуры становятся доступными сразу же после открытия базы данных.

Новые типы графических форматов

В версии dBASE 5.0 поддерживались только два типа графических изображений: BMP и PCX. Теперь количество поддерживаемых графических форматов увеличено (ICO, TIFF, WMF, EPS) и вам нет необходимости преобразовывать изображения из этих форматов к формату BMP или PCX перед использованием их в dBASE.

Компилятор dBASE

Компилятор dBASE является совершенно новым средством и позволяет разработчикам программного обеспечения создавать законченные выполняемые программы (с расширением .EXE), функционирующие в среде Windows. Эти программы не требуют для выполнения наличия dBASE и могут распространяться разработчиками отдельно от dBASE.



Компилятор dBASE поставляется отдельно, но при его установке на компьютере он встраивается в среду разработки dBASE и управление компилятором происходит уже из dBASE. Меню Program после установки компилятора содержит новые пункты меню:

Для успешного выполнения откомпилированного приложения требуется наличие следующих дополнительных файлов:

- DBWC.EXE, содержащий базовые программы выполнения откомпилированных приложений
- DBWC0009.DLL, содержащий ресурсы
- Файлы IDAPI
- Приложение, в котором запускаются отчеты или почтовые этикетки требует наличия файла CRPE.DLL.

Прежде чем создать выполняемый модуль, вы должны создать все формы, отчеты, меню и программы, входящие в состав приложения, и отладить их в среде dBASE. После этого вы можете выполнить компиляцию входящих в приложение файлов с помощью команды Program | Compile. При этом открывается окно диалога "Compile dBASE Source Files" (рис. 18.22).

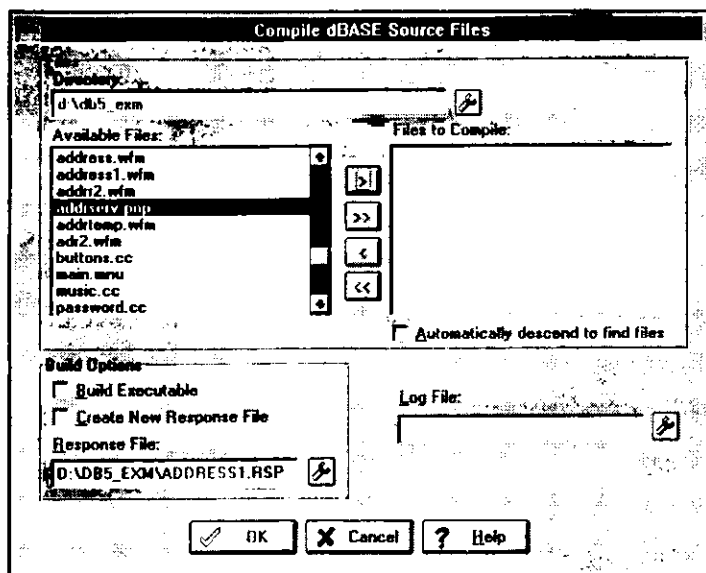


Рис. 18.22. Окно диалога "Compile dBASE Source Files".

Окно диалога "Compile dBASE Source Files" в области "Available Files" содержит список всех исходных файлов указанного каталога. Выберите из них файлы, которые вы собираетесь компилировать, и перенесите их в область "Files to Compile". Вы можете указать имя файла в который будет записан выбранный для компиляции список файлов. Этот файл можно использовать при последующих компиляциях выбранных файлов. Кроме того, в этом окне вы можете установить флажок **Build Executable**. При установке флажка **Build Executable** после успешной компиляции исходных файлов dBASE откроет окно диалога создания выполняемого модуля (рис. 18.23).

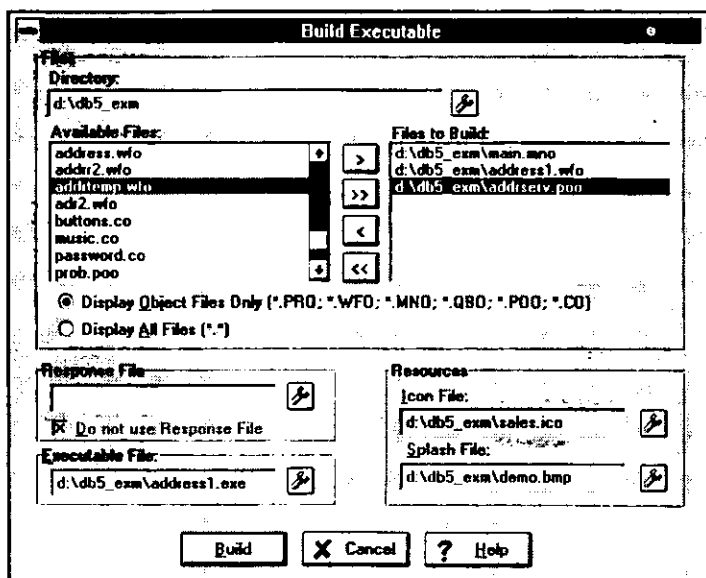


Рис. 18.23. Окно диалога "Build Executable".

Окно диалога "Build Executable", аналогично окну диалога компиляции, содержит список файлов, которые используются при создании выполняемого приложения. Для выполняемого приложения вы можете задать пиктограмму, которая будет отображаться на экране при минимизации приложения. Имя файла, содержащего пиктограмму указывается в поле **Icon File**. Кроме пиктограммы можно задать изображение, которое будет выводиться на экран при загрузке приложения. Для этого в поле **Splash File** укажите имя файла с изображением.

Создание файлов установки

В состав компилятора входит средство упаковки откомпилированного приложения и дополнительных файлов, необходимых для выполнения этого приложения, в компактный формат, который может использоваться конечными пользователями для установки приложения с дискет. Перед тем как перейти к упаковке приложения вы должны убедиться в том, что:

- На вашем компьютере имеются все необходимые системные файлы (Borland Database Engine, языковые драйверы и т.п.).
- Вы создали выполняемый вариант вашего приложения.
- Все файлы, входящие в состав приложения, за исключением системных файлов, скопированы в определенный каталог.

После подготовки файлов для упаковки выполните команду **Program | Install Builder**, в результате чего на экране появится окно диалога "Install Builder" (рис. 18.24).

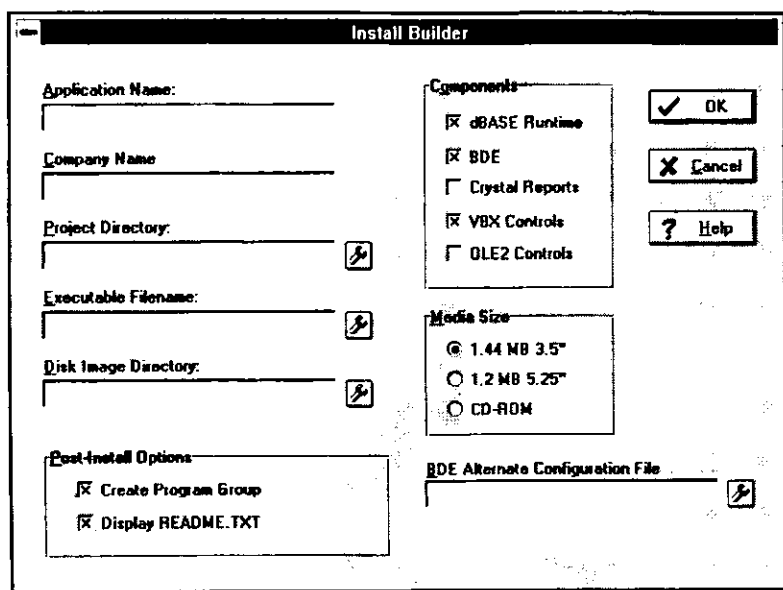


Рис. 18.24. Окно диалога "Install Builder".

Окно диалога "Install Builder" содержит поля для описания параметров упаковки приложения. В поле **Application Name** для создаваемого приложения вы можете указать имя приложения, которое будет отображаться на экране в процессе установки, а также в заголовке создаваемой группы программ. Поле **Company Name** содержит на-

именование фирмы, разработавшей приложение. Имя выполняемого файла с расширением .EXE указывается в поле Executable FileName. Все файлы, используемые в приложении должны находиться в каталоге, указанном в поле Project Directory.

Дополнительно вы можете указать наименование каталога, в котором будут располагаться подготовленные для установки каталоги. Эти каталоги будут называться DISK1, DISK2 и т.д. Объем каталогов определяется носителями информации, на которых вы собираетесь распространять приложение. Выбор носителя информации осуществляется с помощью переключателя Media Size. В качестве возможных вариантов можно выбрать не только дискеты, но и лазерный компакт диск.

В дополнение к файлам приложения вы можете включить системные файлы, необходимые для работы вашего приложения. Перечень этих файлов был приведен ранее при рассмотрении вопроса о создании выполняемого файла приложения.

В процессе упаковки dBASE сжимает заданные файлы и создает файлы с расширением .PAK. В состав упакованных файлов включается программа установки приложения, которая и запускается конечным пользователем при установке приложения на его компьютере.

Поддержка Windows 95

Вы можете запускать dBASE для Windows версии 5.5 как в среде Windows 3.1, так и в среде Windows 95. При работе в среде Windows 95 вам доступны новые возможности этой операционной системы:

- Длинные имена файлов и универсальные соглашения по именам файлов
- Новый стиль пользовательского интерфейса
- Интеграция с Explorer

Все новые функциональные возможности dBASE 5.5 не требуют 32-разрядных библиотек и дополнительной памяти. Поэтому вы можете использовать эту версию на компьютере с памятью 8 МБ в среде Windows 3.1.

Таким образом dBASE 5.5 является прекрасным продуктом этапа перехода от Windows 3.1 к Windows 95, позволяя вам выполнять созданные приложения на обеих платформах.

ПРИЛОЖЕНИЯ

Приложение I. Установка и настройка dBASE

Требования к компьютеру

Средство	Пояснения
Процессор	80386 или выше.
Оперативная память	6 Мбайт. Быстродействие возрастает с увеличением памяти, поэтому для нормальной работы желательно не менее 8 Мбайт оперативной памяти.
Жесткий диск	Для установки и успешной работы dBASE требует 24 Мбайт свободного пространства для полной установки и 10 Мбайт для минимальной.
Видеомонитор	VGA или SVGA.
Microsoft Windows	Версия 3.1 или более поздняя.
Среда русификации	Компания Borland гарантирует корректную работу dBASE для Windows только на русской версии Windows 3.1. Если вы работаете на англоязычной копии Windows, русифицированный при помощи CyrWin, ParaWin или любого другого руссификатора, вам следует позаботиться о том, чтобы в шрифтах Arial и MS Sans Serif были русские буквы. В противном случае вас ждут неприятности, которые никакого отношения к dBASE не имеют.
Мышь	Наличие мыши не является строго обязательным, однако, доступ к некоторым элементам интерфейса можно получить только с помощью мыши.
Сеть	Наличие сети не является обязательным. dBASE может работать в сетях Netware 3.11, 3.12, 4.01, Personal Netware (Novell DOS 7), Banyan Vines 5.5, IBM LAN Server 3.0 *, MS LAN Manager 2.2, Lantastic 6.0, Windows для рабочих групп 3.11, Windows NT 3.5.

Установка dBASE

Создание резервной копии

Перед тем как приступить к установке dBASE, сделайте полную резервную копию дискет дистрибутива и используйте эту копию в качестве рабочей. Исходные дискеты и их рабочую копию следует хранить в надежном месте.



Копию дискет дистрибутива dBASE можно получить с помощью команды DOS DISKCOPY или Диспетчера Файлов Windows.

Программа установки INSTALL

Запуск программы

Для того, чтобы запустить программу установки dBASE INSTALL.EXE, выполните приведенные ниже действия.

1. Запустите Windows.
2. Вставьте первую системную дискету в дисковод A.
3. Выберите команду **Файл | Выполнить** в главном меню Диспетчера Программ.
4. В строку ввода **Командная строка** введите `a:install`, затем выберите **ОК**. На экране появится окно диалога, с помощью которого вам будет предложено указать режим установки (рис. I.1).



Если каталог, в котором находятся системные файлы Windows, включен в список путей DOS с помощью команды `PATH` и Windows еще не запущена, то можно запустить сразу Windows и `INSTALL`, введя `a:install` в ответ на приглашение DOS.

5. Вы можете выбрать любой из режимов. В качестве примера выберите **Выборочная**, что позволит вам установить только необходимые вам компоненты. Нажмите **Далее**.

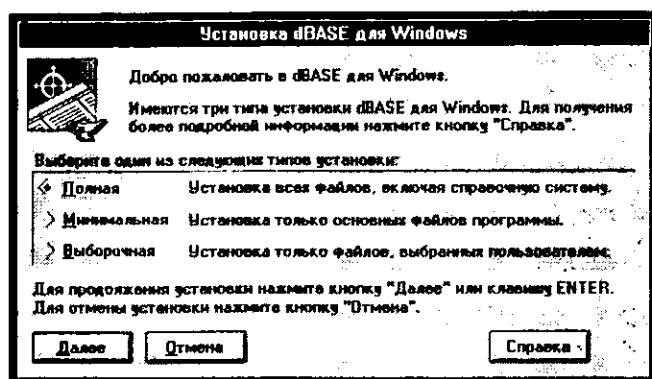


Рис. I.1. Выбор режима установки.

6. Далее вам будет предложено указать имя пользователя и организации, а также каталог для размещения dBASE. Введите требуемые параметры, после чего на экране появится окно диалога установки параметров dBASE для Windows 5.0, с помощью которого вы можете выбрать требуемые компоненты (рис. I.2). Нажмите кнопку **Далее**.
7. В заключение вам будет предложено указать каталоги для программ IDAPI и файлов настроек IDAPI (рис. I.3).



Если на вашем компьютере установлены программные продукты, использующие IDAPI, установите файлы IDAPI dBASE в каталог уже содержащий файлы IDAPI.

8. После нажатия **Далее** dBASE проверяет наличие свободного места на жестком диске и при его наличии начинает процесс инсталляции.

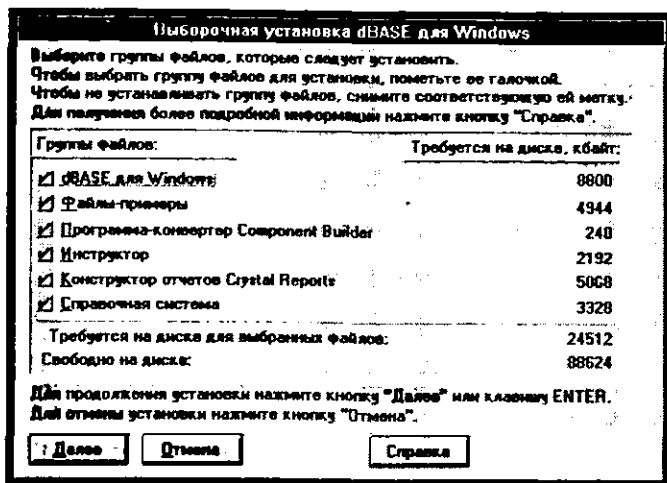


Рис. I.2. Выбор параметров установки.

При установке dBASE для Windows, на экране появляются рекламные щиты, которые содержат различную полезную информацию (рис I.4).

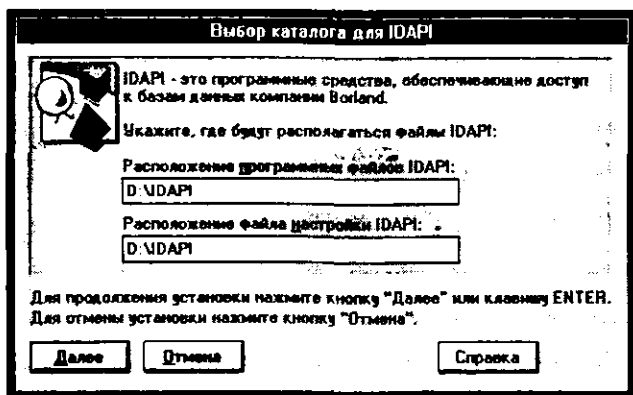


Рис. I.3. Изменение каталогов IDAPI.

Время от времени на экране будет появляться окно диалога, предлагающее вам сменить дискету (рис. I.5). В этом случае вставьте в дисковод нужный диск и выберите OK.

Если в окне диалога программы установки было указано, что следует создать группу dBASE, то INSTALL создаст группу dBASE для Windows и поместит в нее пиктограммы dBASE. В противном случае пиктограммы dBASE будут скопированы в текущую группу.

Установка отдельных компонент

Если спустя некоторое время вам потребуется установить какие-либо файлы, не установленные в процессе инсталляции dBASE для Windows, вновь запустите INSTALL и укажите группы файлов, которые следует установить.



Рис. I.4. Информация при установке dBASE для Windows.

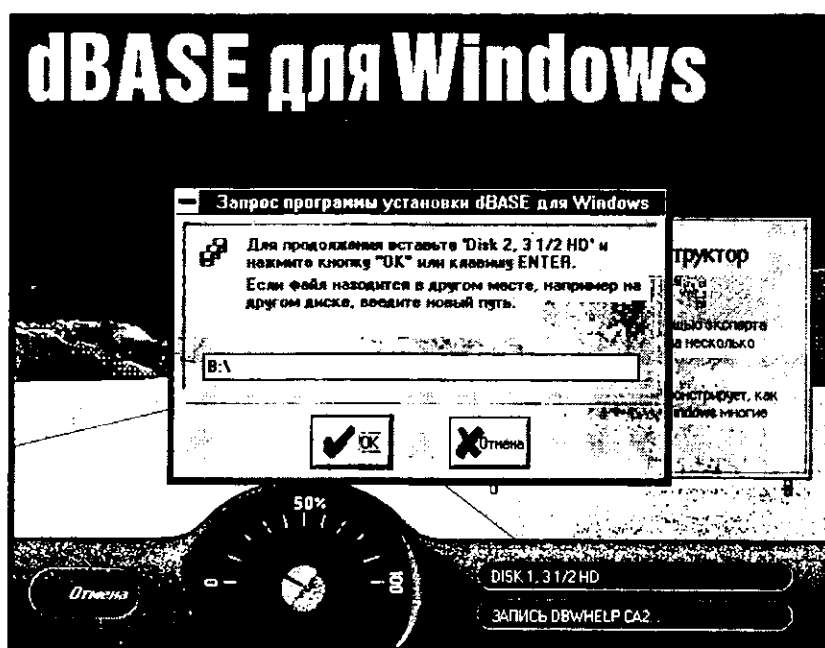


Рис. I.5. Окно диалога "Другая дискета".

Использование утилиты SHARE

Для обеспечения целостности ваших данных перед запуском Windows запустите утилиту DOS SHARE. Наличие SHARE является обязательным, если вы собираетесь

- Использовать файлы dBASE совместно с другими приложениями (например, ObjectVision, dBASE, Quattro Pro, SideKick и приложениями, разработанными с помощью предыдущей версии dBASE для Windows).
- Работать в dBASE совместно с другими приложениями, использующими IDAPI, и осуществлять доступ в одним и тем же таблицам на локальном диске.
- Работать с dBASE в сетях. Для того, чтобы узнать, обязательно ли наличие SHARE для успешной работы с dBASE в конкретной сети, обратитесь к администратору сети.

Для того, чтобы загрузить SHARE автоматически, выполните следующие действия:

- Если на вашем компьютере есть файл AUTOEXEC.BAT, введите C:\DOS\SHARE перед командой WIN, которая запускает Windows.
- Если у вас нет файла AUTOEXEC.BAT, то создайте текстовый файл, введите в него C:\DOS\SHARE и сохраните его под именем C:\AUTOEXEC.BAT. Затем перезагрузите свою рабочую станцию.



Если системные файлы DOS на вашем компьютере хранятся в каталоге отличном от C:\DOS, соответственно измените приведенные выше инструкции.

При запуске программы SHARE.EXE (из AUTOEXEC.BAT или из CONFIG.SYS) следует указать следующие параметры:

```
SHARE /F:4096 /L:40
```

Параметр /F указывает размер области на диске, в которой хранится информация о совместном доступе к данным. Значение по умолчанию — 2048. Параметр /L устанавливает число файлов, к которым можно осуществить доступ одновременно.

Изменение файлов CONFIG.SYS и AUTOEXEC.BAT

Перед запуском dBASE проверьте значения некоторых параметров в файлах CONFIG.SYS и AUTOEXEC.BAT

Файл CONFIG.SYS должен содержать следующие строки:

```
FILES=60
BUFFERS=40
```

Если вы внесли изменения в файлы CONFIG.SYS и AUTOEXEC.BAT, то обязательно выйдите из Windows и перезагрузите вашу рабочую станцию для того, чтобы новые значения параметров стали текущими.

При использовании на вашем компьютере SMARTDRV.EXE (или другой программы кэширования диска), можете уменьшить значение параметра BUFFERS в файле CONFIG.SYS до 10.

Настройка dBASE

Программа настройки параметров

Для изменения параметров используется программа настройки параметров IDAPI, которая запускается выбором пиктограммы из группы dBASE для Windows. После запуска программы на экране появляется окно диалога "Программа настройки IDAPI" (рис. 1.6). Для настройки параметров программа настройки IDAPI использует шесть карточек, наименования которых отображаются на корешках, расположенных в нижней части окна диалога. На рис. 1.6 показана карточка "Драйверы", в которой приведены параметры, используемые при создании, сортировке и обработке таблиц.

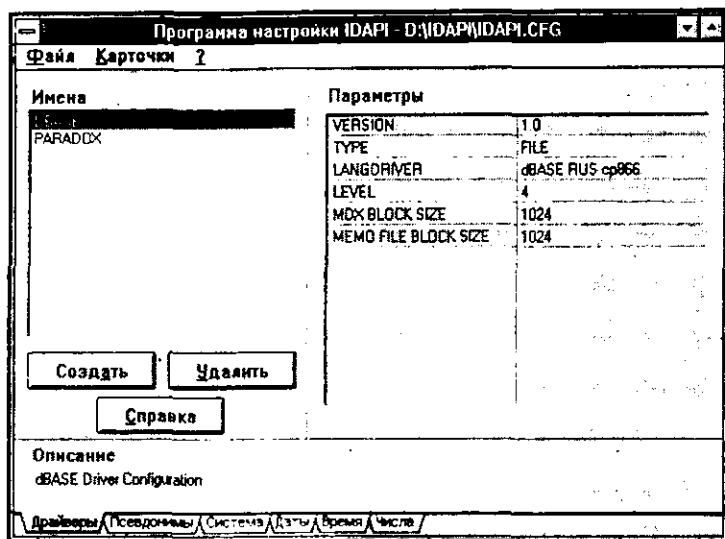


Рис. 1.6. Окно диалога "Программа настройки IDAPI".

В списке **Имена** отображаются типы драйверов, установленных на вашем компьютере. При инсталляции устанавливаются драйверы dBASE и Paradox. Кроме того, вы можете установить драйверы, используемые для доступа к SQL-серверам. Для этой цели предназначены кнопки **Создать** и **Удалить**, которые позволяют добавить в список доступных драйверов драйвер ODBC и удалить его.

В списке **Параметры** приведены параметры, соответствующие выбранному драйверу, и их текущие значения. Описание текущего параметра отображается в разделе "Описание".

Для перехода к следующей карточке вам достаточно нажать мышью соответствующий корешок.



Для получения информации о значениях конкретных параметров воспользуйтесь кнопкой **Справка** или клавишей **F1**.

После завершения настройки параметров системы закройте окно диалога "Программа настройки IDAPI". Вам будет предложено обновить файл IDAPI.CFG. Для того чтобы вновь заданные параметры стали текущими вам необходимо выйти из dBASE и запустить его снова.

Приложение II. Использование графического интерфейса

dBASE для Windows является полностью графической СУБД, которая запускается из Microsoft Windows. Графический интерфейс во многих случаях намного проще в использовании, чем алфавитно-цифровой, и предоставляет пользователю расширенные графические возможности при отображении данных на экране и печати графических данных. dBASE для Windows в полной мере использует графический интерфейс Windows.

Использование мыши

Большую часть операций в dBASE можно выполнить с помощью клавиатуры, однако использование мыши во многих случаях облегчит вам работу. С помощью мыши вы можете выполнить множество разнообразных действий: перемещение по окну, установка ширины столбцов, перемещение окна диалога и т.д. Некоторые действия (например, выбор кнопок панели инструментов) осуществляется только с использованием мыши.

Управление заданиями

Windows является многозадачной системой, поэтому на одном компьютере могут одновременно выполняться несколько задач. Вы можете переключаться из одной задачи в другую с помощью Диспетчера Программ. Для этого выполните команду **Переключиться в** в системном меню dBASE, и на экране появится "Список задач" (рис. II.1), которые выполняются в данный момент. Например, вы можете использовать "Список задач" для перехода в текстовый процессор или Диспетчер Печати. Для выбора "Списка задач" кроме системного меню вы можете использовать комбинацию клавиш *Ctrl+Esc*.

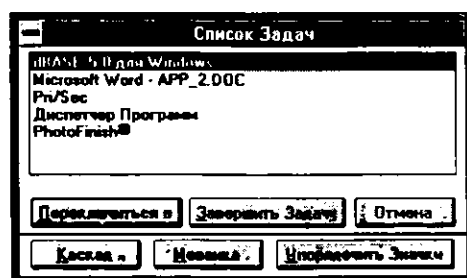


Рис. II.1. Окно диалога "Список задач".

Для перехода между задачами вы дополнительно можете использовать комбинацию клавиш *Alt+Tab*. При каждом нажатии этой комбинации клавиш активизируется следующая задача из списка выполняющихся в данный момент приложений.

Управление окнами

Главное окно dBASE

При запуске dBASE на экране появляется главное окно dBASE (рис. II.2). Все окна, открываемые в dBASE, являются дочерними по отношению к главному окну

dBASE. Это означает, что каждое окно является самостоятельным во время работы dBASE, но закрывается при выходе из dBASE.

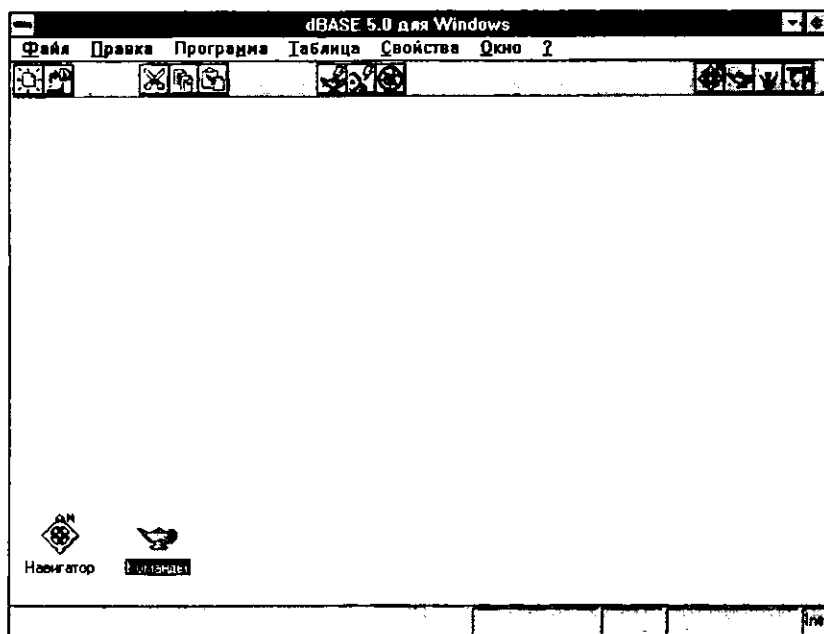


Рис. II.2. Главное окно dBASE.

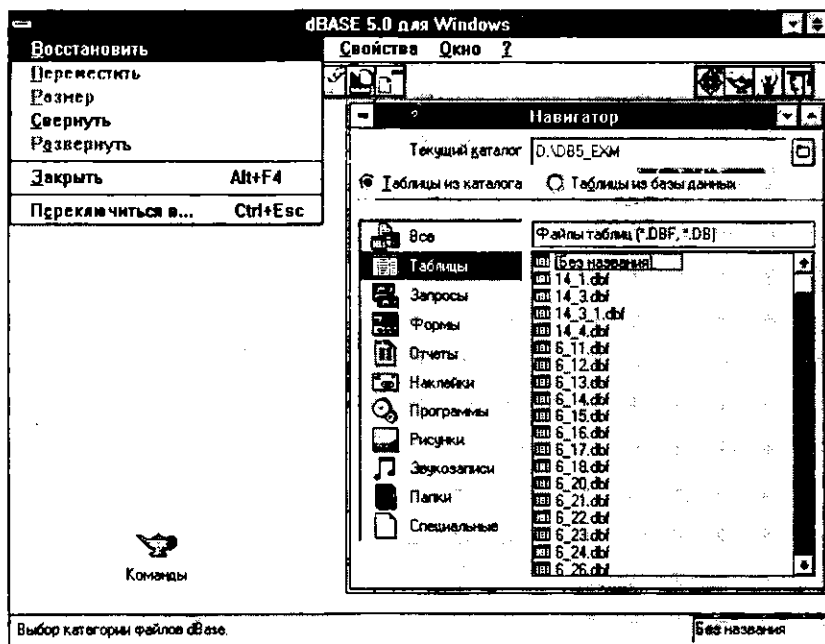


Рис. II.3. Системное меню.

Тип открываемого окна зависит от типа отображаемого в нем объекта, например, формы появляются в окне формы, а бланки запроса — в окне запроса. Окну каждого типа соответствует свой собственный набор команд и функций, применимых только к объекту данного типа. Все команды и функции главного окна dBASE относятся и ко всем дочерним окнам.

Системное меню

Каждое окно имеет системное меню (рис. II.3), которое обеспечивает вас средствами для управления данным окном: изменения размера окна, перемещения и закрытия. Главное окно dBASE, в отличие от дочерних окон, дополнительно содержит команду для переключения в другие действующие в данный момент приложения.

Для вызова системного меню выберите кнопку системного меню в левом верхнем углу главного окна dBASE или нажмите комбинацию клавиш *Alt+Spacebar*.

Переход между окнами

Для перехода между окнами вы можете воспользоваться любым из перечисленных ниже способов:

- Выберите команду **Окно** из основного меню dBASE, затем выберите нужное окно из списка.
- Нажмите мышью нужное окно.
- Нажмите клавишу *Ctrl+F6*.
- Выберите команду **Следующий** системного меню текущего окна.

Расположение окон на экране

Окна в рабочей области могут располагаться произвольно. Если вам потребуется расположить их в определенном порядке, выполните команду **Окно | В стопку. Окно | Сверху вниз** или **Окно | Слева направо** из основного меню dBASE. Результат выполнения этих команд показан на рис. II.4, II.5 и II.6 соответственно.

Изменение размеров и перемещение окон

Для изменения размеров окна установите курсор мыши на любую границу окна, нажмите кнопку мыши и продолжайте удерживать ее во время перемещения курсора. После того, как окно примет требуемые размеры, отпустите кнопку мыши.

Для перемещения окна поместите курсор в любое место верхней рамки окна, за исключением переключателей размера и кнопки закрытия окна. Затем нажмите кнопку мыши и продолжайте удерживать ее во время перемещения курсора. После того как окно окажется в требуемом месте, отпустите кнопку мыши.



Перемещение окна вверх ограничивается панелью инструментов dBASE, а вниз — строкой состояния.

Если вы предпочитаете использовать для перемещения и изменения размеров активного окна системное меню, выполните команды **Размер** или **Переместить**.

Если вам потребуется распахнуть окно (т.е. изменить размеры окна от текущих до максимальных, вы можете воспользоваться одним из описанных выше способов.

Однако, более эффективно вы можете распахнуть окно с помощью следующих действий:

- Нажмите мышью кнопку распахивания окна в правом верхнем углу экрана.
- Выполните команду Развернуть системного меню окна.

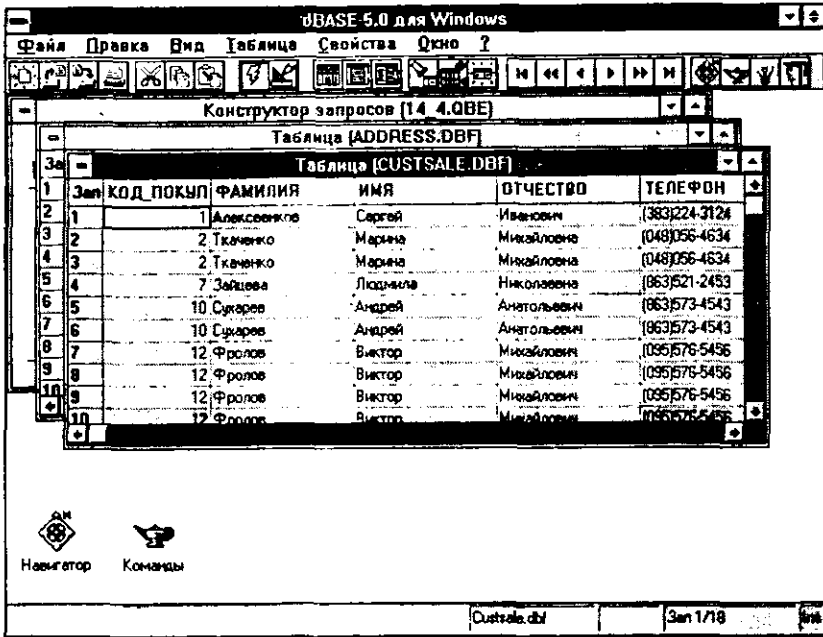


Рис. II.4. Результат выполнения команды Окно | В стопку.

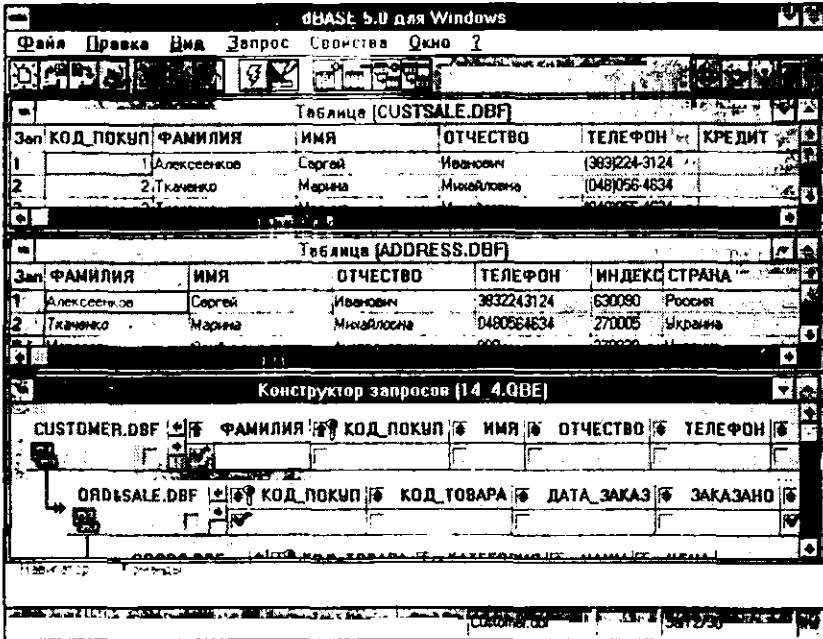


Рис. II.5. Результат выполнения команды Окно | Сверху вниз.

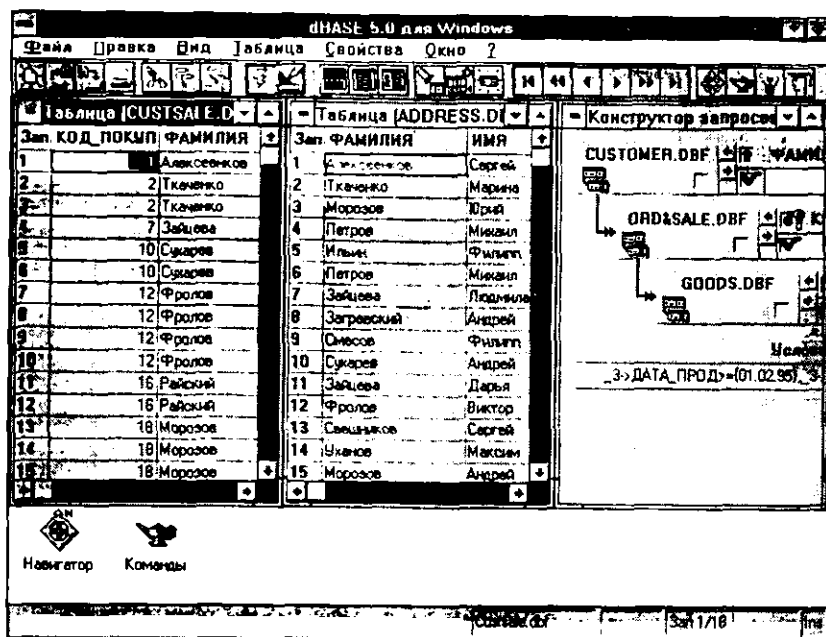


Рис. II.6. Результат выполнения команды Окно | Слева направо.

Для изменения окна до размеров пиктограммы используйте один из ниже описанных методов:

- Нажмите мышью кнопку свертывания окна в правом верхнем углу экрана.
- Выполните команду Свернуть системного меню окна.

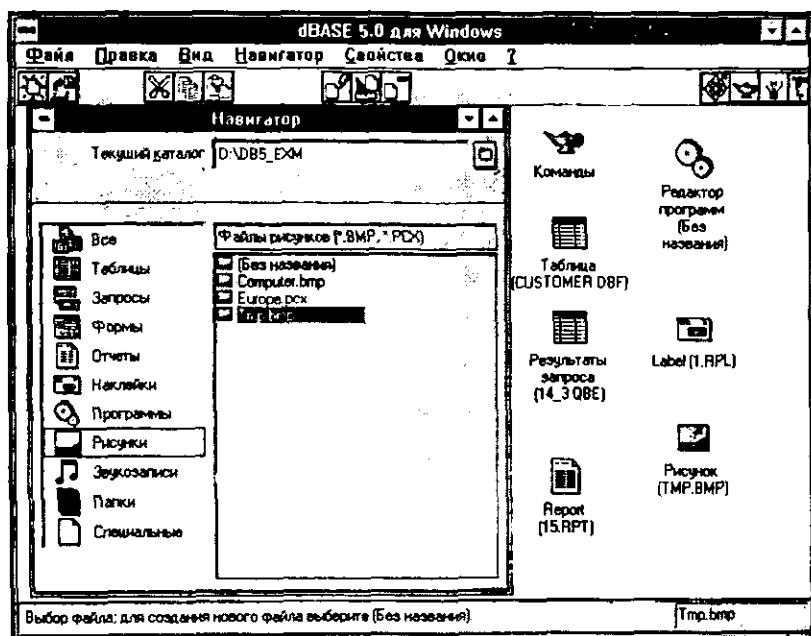


Рис. II.7. Пиктограммы объектов dBASE.



Каждый тип объектов dBASE имеет свою пиктограмму, подписью которой является заголовок окна (рис. П.7).

Прокрутка информации через окна

При работе с большими объемами данных на экране помещается только часть необходимой информации даже в тех случаях, когда вы распахнули окно. В этих случаях для отображения в окне требуемой информации вы можете прокрутить ее через окно в горизонтальном и вертикальном направлениях.

Прокрутка осуществляется с помощью *вертикальной и горизонтальной полос прокрутки*. Каждая из полос содержит *кнопки прокрутки* (стрелки) и *движок прокрутки* (рис. П.8). Однократное нажатие мышью кнопки прокрутки сдвигает изображение в окне на один элемент. Например, при просмотре таблицы каждое нажатие мышью левой кнопки прокрутки будет сдвигать изображение таблицы на одно поле влево.

Движок прокрутки показывает положение информации, отображаемой в данный момент в текущем окне, относительно всего содержимого окна. При перемещении с помощью мыши движка прокрутки одновременно будет происходить перемещение информации в окне.

Зап	КОД_ПОКУП	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ТЕЛЕФОН	ИНДЕКС	СТРАНА
1		Алексеев	Сергей	Иванович	(383)224-3124	630090	Россия
2		Ткаченко	Марина	Михайловна	(048)056-4634	270005	Украина
3		Морозов	Юрий	Анатолеевич	(048)034-5874	270029	Украина
4		Петров	Михаил	Николаевич	(816)025-6128	173024	Россия
5		Ильин	Филипп	Петрович	(816)025-6156	173003	Россия
6		Петров	Михаил	Николаевич	(816)017-7124	173003	Россия
7		Зайцева	Людмила	Николаевна	(863)521-2453	346415	Россия
8		Загребский	Андрей	Анатолеевич	(831)256-4216	603006	Россия
9		Смессе	Филипп	Петрович	(883)245-3123	630056	Россия
10		Сухарев	Андрей	Анатолеевич	(963)573-4543	357736	Россия
11		Зайцева	Дарья	Ивановна	(257)212-2453	142490	Россия
12		Фролов	Виктор	Михайлович	(095)576-5456	140160	Россия
13		Свешников	Сергей	Иванович	(095)453-5985	142432	Россия
14		Уланов	Максим	Сергеевич	(812)623-4456	184230	Россия
15		Морозов	Андрей	Михайлович	(343)232-4542	620151	Россия
16		Райский	Максим	Сергеевич	(846)253-4542	443115	Россия
17		Антонов	Максим	Юрьевич	(095)532-1532	125167	Россия
18		Морозов	Андрей	Петрович	(095)131-7714	117334	Россия
19		Алексеев	Александр	Иванович	(013)225-6541	226010	Литва

Рис. П.8. Прокрутка изображения в окне.

Закрытие окон

Для закрытия окна вы можете выбрать команду **Закрыть** из системного меню, нажать клавишу **Alt+F4** или нажать мышью кнопку закрытия окна.

Для закрытия всех окон выполните команду **Окно | Закрыть все окна** из главного меню dBASE.

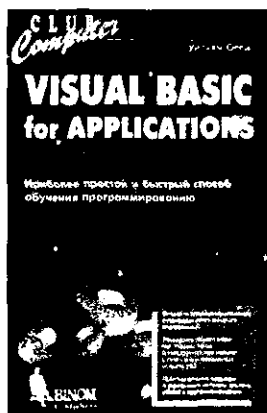
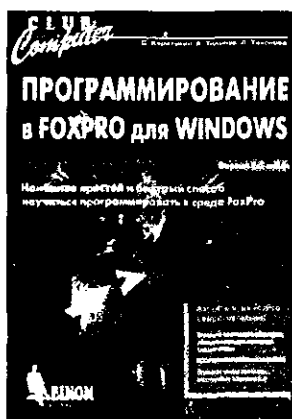
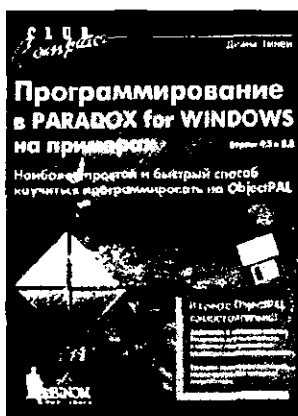


ИЗДАТЕЛЬСТВО

БИНОМ

Серия "...на примерах"

Вышли из печати



Готовятся к выходу

- Работа в Windows 95 на примерах
- Работа в Paradox для Windows 5.0 на примерах
- Работа в Excel для Windows на примерах
- Работа в Microsoft Word для Windows на примерах



ИЗДАТЕЛЬСТВО
БИНОМ

103473 МОСКВА-473 а/я 133.

Факс (095) 211-1114

Телефоны

(095) 211-1690

(095) 211-4344

Приглашаем к сотрудничеству авторов
и научных редакторов в области
электроники и программирования

Принимаются заказы на размещение
рекламы в наших изданиях

Каратыгин С.А., Тихонов А.Ф.
РАБОТА В «BASE ДЛЯ WINDOWS НА ПРИМЕРАХ

Подписано в печать 23.08.95. Формат 70х108 1/16. Печ. л. 29.
Бумага газетная. Печать офсетная.
Тираж 15000 экз. Заказ 1064

Издательство «БИНОМ», 1995 г.
Москва, ул. Новослободская, д. 50/1, стр. 1а.
Лицензия на издательскую деятельность № 063367 от 20 мая 1994 г.

Отпечатано с готовых диапозитивов в полиграфической фирме
«Красный пролетарий»
103473, Москва, Краснопролетарская, 16

