Тим АНДЕРСОН

VISUAL BASIC
ҚАДАМ БА ҚАДАМ

531
51-65

ЎЗБЕКИСТОН РЕСПУБЛИКАСИ ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ

ЎРТА МАХСУС, КАСБ-ҲУНАР ТАЪЛИМИ МАРКАЗИ

ЎРТА МАХСУС, КАСБ-ҲУНАР ТАЪЛИМИНИ РИВОЖЛАНТИРИШ ИНСТИТУТИ

Тим АНДЕРСОН

VISUAL BASIC ҚАДАМ БА ҚАДАМ

ТОШКЕНТ "ЎЗБЕКИСТОН" 200[illegible]

BUXORO MUHANDISLIK-TEXNOLOGIYA INSTITUTI
ARM
REESTR № 26683

32.973
Л 65

ISBN 5-640-03134-4

А $\frac{2404040000-46}{M351(04)2002}$ 2002

© Original Copyright.
Computer Step, 1997.
© Издание на русском языке,
ЗАО «Издательство БИНОМ», 1998
© Ўзбек тилига таржима,
«ЎЗБЕКИСТОН» нашриёти, 2002.

Мундарижа

Биринчи боб. Биринчи қадамлар

Кириш 8
Visual Basic ни ўрнатиш 9
Сизнинг биринчи илова — дастурингиз 12
Асбоблар панели билан танишамиз 14
Хусусиятлар ойнаси билан танишамиз 16
Visual Basic шакллари 17
Шаклнинг асосий хоссалари 18
Visual Basic буйруқ тугмалари 18
Visual Basic ҳодисалари билан танишамиз 19
Сизнинг биринчи код сатрингиз 21
Коддаги хосса қийматларини ўзгартириш 22
Visual Basic арифмометри 23
Хатоликлар 24
Маълумотлар қандай олинади 25
Лойиҳа ойнаси билан танишамиз 26
Яратилган дастурни сақлаш 28
Мавжуд лойиҳани юклаш 29

Иккинчи боб. Бошқариш элементлари

Бошқариш элементлари ва услублари 32
Буйруқ тугмачаси (CommandButton) 33
Ёзув (Label) 34
Матн ойнаси (TextBox) 36
График ойна (PictureBox) 38
Байроқча (ChekBox) 39
Улагич (OptionButton) 41
Рамка (Frame) 42
Шакллар ва чизиқлар (Line, Shape) 44
Расм (Image) 45
Рўйхат ойнаси (ListBox) 46
Тўр (FlexGrid) 49
Умумий диалог ойнаси (Common Dialog) 52
Комбинацияланган ойна (ComboBox) 55

БИБЛИОТЕКА
Инв. № 97228

Таймер (Timer) 57
OLE бошқариш элементи 58
Бошқариш элементларини улаш тартиби 60
Бошқа стандарт бошқариш элементлари 62

Учинчи боб. Basic асослари

Бошидан бошлаймиз 65
Ўзгарувчилар ва кўриниш соҳаси 68
Маълумотлар турлари билан танишамиз 69
Маълумотлар турини ўрганамиз 70
Калькулятор-дастур 72
If... Then... Else 73
For... Next 74
Do... Loop 75
Case... Else 77
With... End with 78
Массивлар 79
Фойдаланувчи аниқлайдиган маълумот турлари 81
Ёрдамчи дастурлар тузиш 82
Қисм дастурлар параметрлари 84
Функцияларни яратиш 85
MsgBox ва InputBox дан фойдаланиш 86
PrintForm ёрдамида босиб чиқариш 87
Printer объекти ёрдамида босиб чиқариш 89

Тўртинчи боб. Visual Basicнинг асбоблар воситалари

Меню яратиш 92
Менюни созлаш 93
Қалқиб чиқувчи меню яратиш 94
Менюни ишга тушириш 96
Дастурни бажариш вақтида менюни ўзгартириш 98
Код муҳаррири 99
Созлагич ҳақида 101
Созлаш ҳақида батафсилроқ 102
Узиш нуқталаридан фойдаланиш 103
Хатоларни кузатиш 105
ActiveX бошқариш элементлари 106
Кенгайтмалардан фойдаланиш 108

Мурожаатларни бошқариш ойнаси 109
Visual Basic параметрларини созлаш 111
Лойиҳа хусусиятларини ўрнатиш 113
Шакллар андазалари 114
Лойиҳа андазалари 116
Мустақил ишловчи (автоном) дастур яратиш 117
Windowsнинг бош менюсига ёрлиқ қўшиш 118

Бешинчи боб. Маълумотлар базасини (маконини) яратиш

Маълумотлар базасига кириш 121
Visual Data Manager дан қандай фойдаланилади 122
Жадвални ишлаб чиқиш 124
Маълумотларни киритиш 126
Маълумотлар базасининг жадваллари ҳақида батафсилроқ билим 128
Data бошқариш элементи билан танишамиз 130
Маълумотлар билан ишлаш учун шакллар яратиш 131
Маълумотлар шаклини такомиллаштириш 132
Ёзувлар қўшиш 134
Ёзувларни ўчириш 135
Data бошқариш элементи хатоликларининг олдини олиш 137
Ёзувларни берилган тартибда кўриб чиқиш 138
Table ва Dynaset туридаги ёзувлар тўплами 140
Ёзувларни қидириш 141
Ёзувларни тўрда кўриб чиқиш 145
Crystal Reports билан танишамиз 147
Маълумотлар базаси билан ишлашдаги кейинги қадам 151

Олтинчи боб. Office да Visual Basic

Ёрдамчи (илова) дастурлар учун Visual Basic версияси 154
VBA ни ўрнатиш 155
VBA дастурлаш воситалари 156
Excel да макрос ёзиш 158
Макроснинг тузилмаси 159
Макросни ишга туширишнинг учта усули 160
Автоматик тарзда бажарилувчи макрослар 163

Кириш

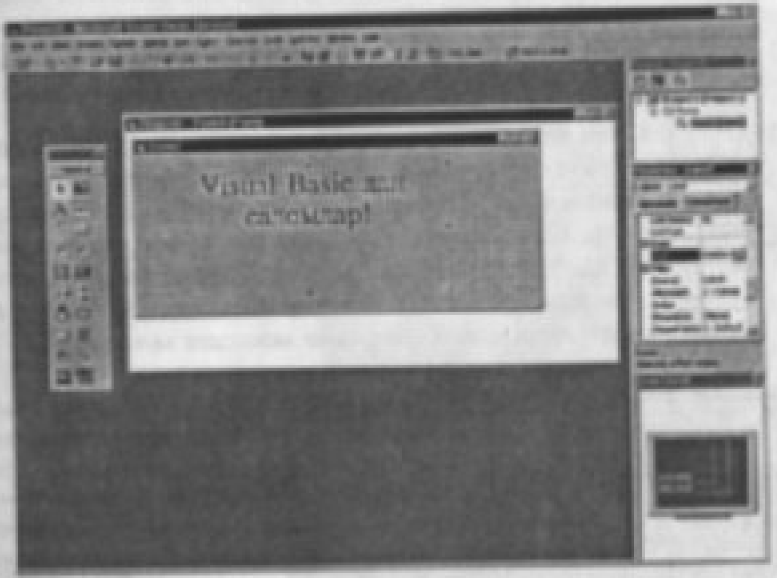
Visual Basic — Windows асосида ишлайдиган дастурлар тузиш учун қулай восита хисобланади. Дастурларни мустақил тузишдан мақсад компьютерга мутлақ хокимлик қилиш, яъни иш давомида юзага келадиган муаммоларни тезрок хал этиш имконини яратишдир. Компьютер дастурлари сермеҳнат ишларни автоматлаштиради, хатоларни камайтиради ва меҳнат унумдорлигини оширади. Бундан ташқари, дастурлар тузиш жуда хам мароқли машғулотдир!

Basic (БЕЙСИК) сўзи Beginner's All-purpose Symbolic Instruction Code дан ҳосил бўлган (Бошловчилар учун умум йўналишдаги дастурлаш тили). Бу тилнинг мазкур дастур версиясидаги Visual сўзи дастур тузиш жараёнида кўплаб операциялар визуал воситалар ёрдамида, яъни бевосита кодларни ёзмасдан бажарилишини англатади.

Дастурлар тузиш сермашаққат жараён, лекин Visual Basic бу ишни сезиларли даражада соддалаштиради ва сизга уни осонроқ бажаришга имкон беради. Visual Basic нинг яна бир ютуғи шундаки, у Microsoft Office пакети дастури ва Internet ресурслари билан ишлайди. Visual Basic бир неча версияда ишлаб чиқарилади

- Learning Edition (Ўқув тахрири). Бу версия бошловчилар учун жуда мос бўлиб, Windows да ишловчи дастурлар тузиш учун зарур бўлган барча элементларни ўзида жамлаган.
- Professional Edition — (Профессионал — касбий тахрири). Компилятор дастурлари ишини тезлаштирувчи ActiveX қўшимча компонентларини ва маълумотлар базаларини бошқаришнинг кенгайтирилган воситаларини ўз ичига олади. Бу версия жиддий иловаларни ишлаб чиқариш учун мўлжалланган
- Enterprise Edition (Корхоналар учун тахрири) кўпчилик фойдаланувчили маълумотлар базаларига эга иловаларни ишлаб чиқишга имкон беради ва иловаларни дастурчилар гурухлари хамкорликда ишлаб чиқиш воситаларини ўз ичига олади.
- Visual Basic for Applications (Иловалар учун Visual Basic). Visual Basic нинг мазкур версияси Microsoft Office дастури таркибига киради. Агар сизда Office нинг тўлиқ версияси ёки бу пакетнинг исталган компоненти бўлса, у холда сизда Visual Basic нинг иловалар учун версияси хам бор бўлади.

Сиз иш жойингиз (компьютерингиз) да ўрнатилган Visual Basic версиясини аниқлашингиз мумкин, бунинг учун дастурни ишга туширишда пайдо бўладиган экрандаги ёрлиққа эътибор бериш зарур.



Visual Basic — бу хар бир киши учун дастурлашдир.

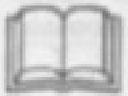
Visual Basic ни ўрнатиш

Visual Basic ни ўрнатиш учун SETUP.EXE дастурини ишга тушириш зарур. Агар сиз Visual Basic версиясига CD-ROM олган бўлсангиз, у холда ўрнатиш дастури CD-ROM

ни диск юритувчига қўйишингиз билан автоматик ишга тушади.

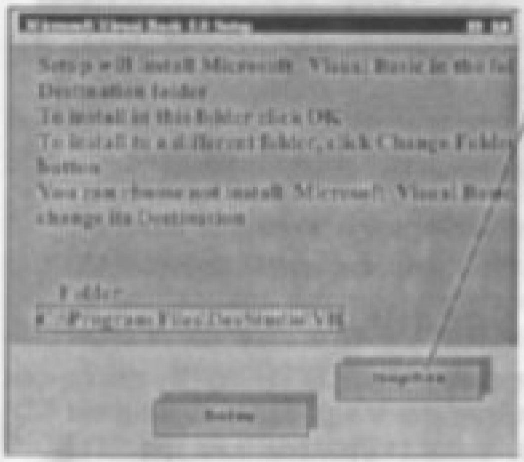
Setup дастурини ишга туширишдан олдин сизнинг тизимингиз қуйидаги кўрсаткичларга — спецификацияларга жавоб бериш-бермаслигини текшириб кўринг:

- компьютер 486 русумли процессор ёки Pentium асосида; операцион тизими Windows 95 ёки Windows NT;
- камида 16 Мбайт тезкор хотира — RAM;
- Қаттиқ дискда камида 35 Мбайт бўш жой.



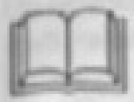
Қаттиқ дискда 35 Мбайт бўш жой энг кам талаб қилинадиган ҳажм ҳисобланади. Бунга эга бўлган ҳолда сиз баъзи қўшимча имкониятларни ўрната олмайсиз. Яхшиси, 100 Мбайтдан ёки ундан каттароқ қийматдан бошлаган маъқул. Агар қаттиқ дискда жой етишмаса, у ҳолда Setup дастури бу ҳақда сизни огоҳлантиради.

Катта ҳажмдаги дастурларни ўрнатиш жараёнида қаттиқ дискдаги ҳамма бўш жойни сарфлаш мумкин. Windowsнинг иш жараёнида вақтинчалик ахборотларни (файлларни) сақлаш учун қаттиқ дискда маълум миқдорда бўш жой керак. Сизга шахсий ҳужжатларингиз учун ҳам бўш жой керак. Бу мақсадда 30 Мбайт ва ундан кўпроқ жой қолдириш мақсадга мувофиқдир.



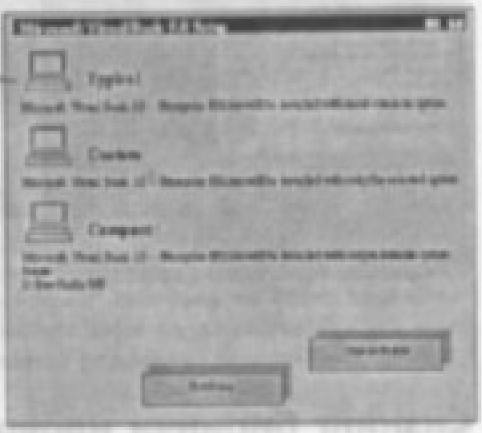
1. Visual Basic ни ўрнатиш папкасини мустақил танламоқчи бўлсангиз, танлаш учун Change Folder папкани ўзгартириш тугмасини босинг. Одатда бу қуйидаги ҳолатда талаб қилиниши мумкин: берилган дискка қараганда кўпроқ бўш жой бўлган мантиқий диск мавжуд бўлганда.

Шундан сўнг Setup дастури сиздан ўрнатиш вариантини сўрайди. Агар сиз Custom (Махсус) вариантини танламаган бўлсангиз, дастурни ўрнатиш автоматик тарзда давом этади.



Агар Visual Basic нинг қайси компонентлари ўзингизга зарур эканлигини билмасангиз, унда Typical (Оддий) вариантини танлаш мақсадга мувофиқдир. Иш тажрибангиз ортиши билан сиз Setup дастурини яна ишга тушириб, ўзингизга зарур компонентни ўрнатиш ёки чиқариб ташлаш имконига эга бўласиз.

2. Ўрнатиш вариантини танлаш учун тугмани босинг

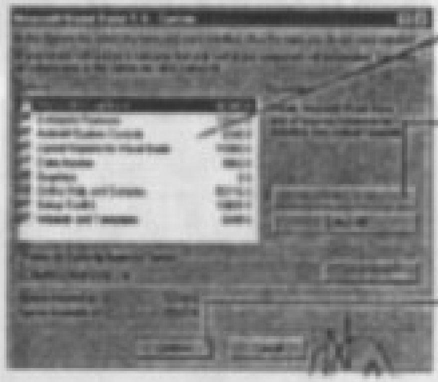


- Typical (оддий) вариантини танланг, бунда Setup дастури ўрнатиш учун зарур компонентларини ўзи танлайди. Агар Visual Basic билан энди танишаётган бўлсангиз, айнан шу вариантни танланг.

- Ўрнатилиши зарур бўлган компонентларнинг аниқ термасини (наборини) кўрсатмоқчи бўлсангиз, Custom (Махсус) вариантини танланг.

- Агар қаттиқ дискингизда бўш жой ҳажми кам бўлса, Compact (Ихчам) вариантини танланг.

Агар Visual Basic ни ўрнатиш жараёнида Custom (Махсус) вариантини танлаган бўлсангиз, кейинги диалог сизга зарур компонентларни шахсан ўрнатиш имконини беради. Visual Basic нинг ишлаши учун рўйхатдаги фақат биринчи компонент зарурдир. Рўйхатда қатор компонентларнинг мавжудлиги сиз танлаб олган Visual Basic версиясига боғлиқ.



3. Бу рўйхатдаги ўрнатилишини истаган компонентларни сичқончани босиб белгиланг.

4. Айни пайтда ажратилган компонент учун керакли ташкил этувчиларни (агар улар бўлса) танлаш учун Change Option (Параметрларни ўзгартириш) тугмасини босинг.

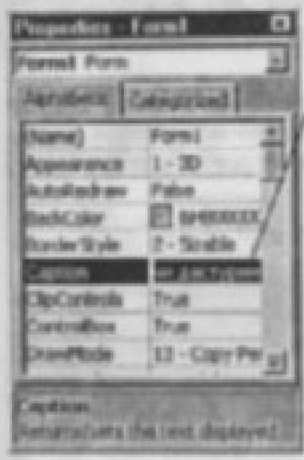
5. Ўрнатиш жараёнини охирига етказиш учун Continue (Давом эттириш) тугмасини босинг.

Сизнинг биринчи илова — дастурингиз

Visual Basic билан ишлашга дарҳол киришиб кетиш учун қуйида учта оддий қадам билан иловани яратиш кетма-кетлиги келтирилган.

Visual Basic нинг сиз яратадиган шаклни (формани) ишлашга мажбур қиладиган ҳаракатга келтирувчи механизми ҳодисалар, хоссалар ва услублардир. Мазкур бобдаги мисолларни кетма-кет бажариб борсангиз, уларнинг қандай ишлатилишини билиб оласиз.

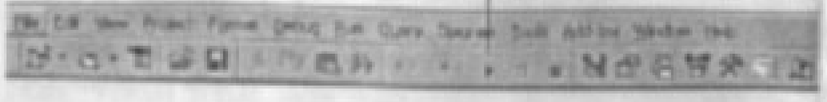
1. Visual Basic ни ишга туширинг ва очилаётган диалогда "Standard EXE" белгиси (Стандарт EXE-файл) танланганини текширинг. Сўнг Open (Очилсин) тугмасини босинг.



2. Энди Properties (Хоссалар) номли кичик ойнани топинг ва Caption (Сарлавҳа) хоссасидан ўнгда жойлашган майдонда сичқончани босинг. Бу ерда "Менинг биринчи дастурим" деб ёзинг ва Return клавишини босинг. Шакл сарлавҳаси сатридаги матн қандай ўзгаришини кузатинг.

Агар хоссалар ойнасини ёки Visual Basic нинг бирор бошқа асосий ойнасини топа олмасангиз, уни View (Кўриниш) менюсидан танлаб олинг. Шундан сўнг танланган ойна экранда кўринади.

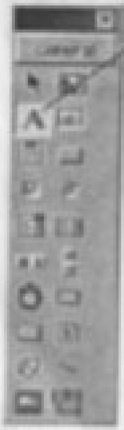
3. Татбиқни ишга туширищ учун ушбу стрелкани босинг.



Сизнинг биринчи илова дастурингиз фақат шакл (ойна) ҳосил қилади. Сиз бу шаклнинг ўлчамларини, рангини ўзгартиришингиз, уни суришингиз ёки кичиклаштиришингиз, ойнани ёпиб, ундан чиқишингиз мумкин. Аслида мазкур илова-дастур ҳеч қандай иш бажармаса-да, ҳозиргина сиз Windows учун дастур тузишга муваффақ бўлдингиз!

Асбоблар панели билан танишамиз

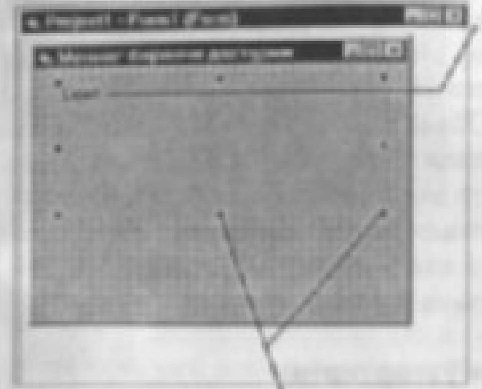
Visual Basicнинг асбоблар панели сизнинг дастурларингиз учун қурилиш блокларидан иборатдир. Олдинги бўлимда яратилган шаклдан фойдаланиб, қуйидаги амалларни ишда синаб кўриш учун уларни бажаринг.



1. Асбоблар панелидаги "A" тугмачасини босинг. Мазкур белги шаклга ёзув (Label) қўйиш учун ишлатилади. Босилгандан сўнг белгининг тасвири ўзгариб, бу унинг босилганлигини кўрсатади.

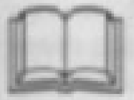
Сиз Visual Basic билан ишлаётганда кўп вақт асбоблар панелида стрелка-кўрсаткич белгиси танланади. Бошқа асбоб белгилари сиз шаклга фақат янги объектни жойлаштиришингиз керак бўлган жойлардагина танлаб олинади.

Ўлчамларни ўзгартириш маркёрлари

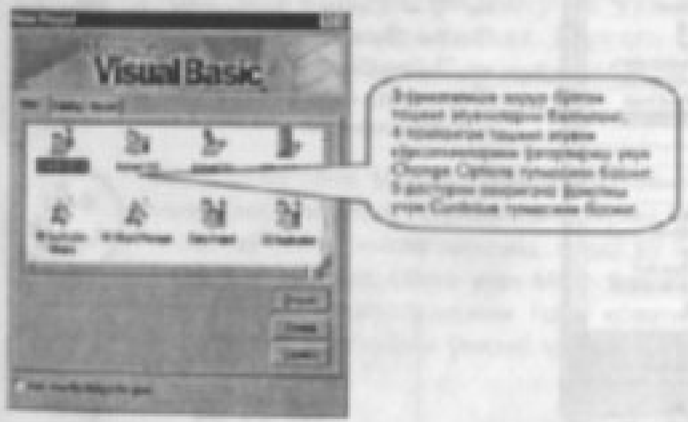


Ўлчамларни ўзгартириш маркёрлари

2. Курсорни шакл ичига жойлаш ва сичқоннинг чап тугмасини ёзувнинг чап юқори бурчаги шаклда туриши керак бўлган жойда босинг. Тугмани босилган ҳолда тутиб туришда давом этиб, ёзувни чизиш учун курсорни ўнгга ва пастга суринг. Шундан сўнг тугмани қўйиб юборинг. Агар сиз ёзув жойи ва ўлчамларини ноқулай жойлаштириб қўйган бўлсангиз, ёзувни саккизта маркёрдан бири ёрдамида унинг жойи, ҳажмини ўзгартиришингиз мумкин. Ёзувни шаклга жойлаштириш учун ёзув жойлаштирилиши зарур бўлган жойга сичқонча кўрсаткичи ёрдамида жойлаштирилади. Ёзувни жойини кенгайтириш учун сичқонча тугмасини босилган ҳолатда паст-ўнг томонга сурилади. Кейин тугмача қўйиб юборилади.



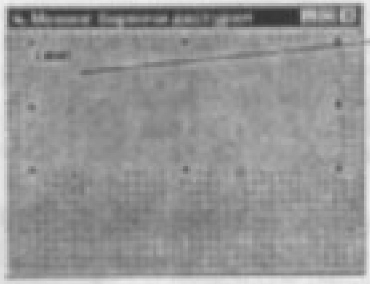
Асбоблар панелининг мазмуни Visual Basic нинг ўрнатилган версиясига ҳамда унинг қандай ўрнатилганлигига боғлиқ. Агар сиз асбоблар панелида сичқонча ўнг тугмасини боссангиз, компонентларни қўшиш ёки йўқотишингизга имкон берувчи менюни кўрасиз.



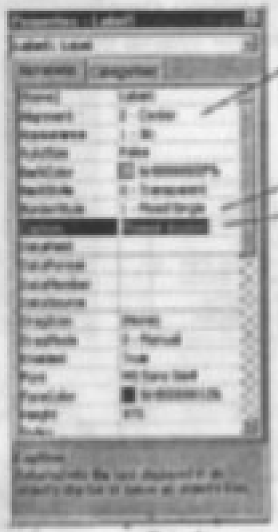
Хоссалар ойнаси билан танишамиз

Visual Basicда хоссалар ойнаси (Properties) — бу атрибутлар рўйхатидир. Ҳар бир атрибут ёки "хосса" айни пайтда шаклда танланган объектга (ёзув, кнопка, матн ойнаси ва ҳ.к.) таъсир кўрсатади. Объект хоссаларининг қийматини ўзгартириб, сиз унинг ташқи кўриниши ва ҳолатини бошқарасиз. Тузилажак дастур элементлари, ташкил этувчилари: ёзув, тугмача, матн ойнаси ва ҳоказоларнинг хоссалари акс этади. Элементлардан қайси бири танланганига қараб хоссалар ойнасидаги манзара ўзгариб туради.

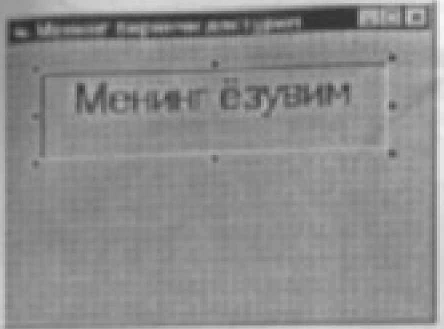
Ёзувнинг хоссаларини ўзгартириш



1. Шаклга ёзувни жойлаштиринг (олдинги бўлимдаги шаклдан фойдаланиш мумкин). Шаклда ёзув танлаб олинганига ишонч ҳосил қилинг.



2. Энди баъзи қийматларни ўзгартириш учун хоссалар ойнасининг ўнг томонини босинг. Alignment (Текислаштириш) хоссаси қийматини "2 — Center" (Марказ)га ўзгартиринг. "Border Style"(Шакл тури) хоссаси қийматини "1-FixedSingle"га ўзгартиринг. "Caption" (Сарлавҳа) кўрсаткичига "Менинг ёзувим" матнини киритинг.



Сизнинг қарорларингизга жавобан ёзув қандай ўзгаришига эътибор беринг.

Visual Basicдаги ҳар бир объектнинг ўз номи бор (Name хоссаси). Хоссалар ойнасининг юқори қисмида ойна (карнай) пайтда танланган объектнинг номи кўрсатилган.

Visual Basic шакллари

Visual Basic иловаларининг кўпчилиги экран шакли (формаси) асосида яратилади. Шакл — сиз "расм" чизишингиз мумкин бўлган оқ матодир. Кўпчилик ҳолларда илова бир нечта шаклга эга бўлади, лекин Visual Basic иш жараёнида уларни яшириш ва кўрсатишга имкон беради. Асосий шаклни ёпиб, сиз иловадан чиқасиз.

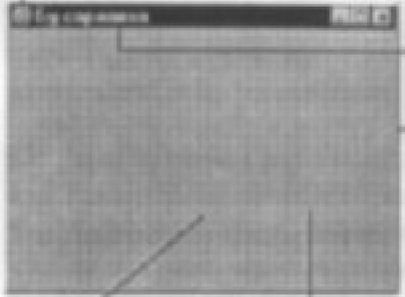
Visual Basic нинг барча объектлари сингари шакллар ҳам хоссаларга эга. Шаклни танлаш учун уни объектлардан ҳоли бўлган бирор жойда босинг. Шундан сўнг шаклнинг хоссалари хоссалар ойнасида пайдо бўлади.

Шакл — бу ойна. Шунинг учун мазкур ойна бошқа ойналар каби Minimize (Минималлаштириш, ўраб қўйиш), Maximize (Максималлаштириш, очиш) ва Close (ёпиш) тугмачаларига эга. Шакл учун MinButton ва MaxButton хоссаларининг қийматларини False ҳолатига ўтказиб, дастурда бу икки тугмани ўчириб қўйиш мумкин.

TEXNOLOGIYA INSTITUTI
ARM
96684

Шаклнинг асосий хоссалари

Icon (белги): кичик расм бўлиб, шаклнинг юқориги чап томонида ва шунингдек унинг ўралган ҳолида пайдо бўлади.



Caption (сарлавҳа) — шаклнинг сарлавҳаси сатрида пайдо бўладиган матн.

Border Style (шакл тури) қуйидагиларни аниқлайди: шакл четларининг кўриниши, шаклнинг ўлчамларини, сичқонча билан ўзгартира оласизми? Шаклнинг ўнг юқори бурчагида қандай тугмалар пайдо бўлади. Мазкур мисолда хосса Fixed Single да тасвирланган.

Background (фон): шакл фони рангини белгилайди.

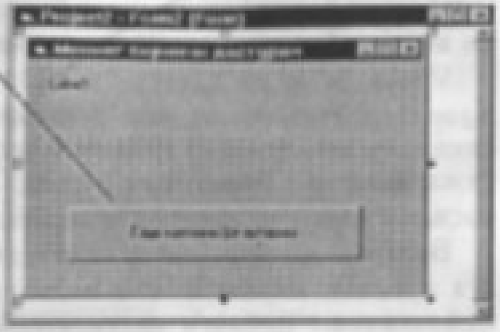
StartUpPosition (бошланғич ҳолат): шаклнинг Windows иш столида бошланғич ҳолатини белгилайди.

Visual Basic буйруқ тугмалари

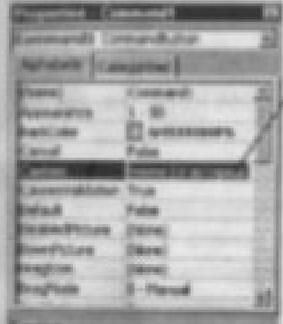
Буйруқ тугмаси (CommandButton) — Visual Basic нинг энг фойдали объектларидан биридир. Тугмалар фойдаланувчига "Иловаларни" бошқариш имконини беради. Visual Basic воситаларидан фойдаланиб, сиз фойдаланувчи қайси тугмани босганини аниқлайсиз ва унга мос амални бажарасиз.

Тугмани шаклга жойлаштириш

1. Асбоблар панелида тугма белгисини босинг, бунда у босилган ҳолатда қолсин.



2. Тугмачани шаклга жойлаштириш учун шаклнинг бирор жойида босинг ва тугма чегарасини пастга чўзинг. Ўлчамни ўзгартириш маркёрларидан фойдаланиб, тугма ўлчамларини ўзгартиришингиз мумкин. Тугманинг ўрнини ўзгартириш учун уни босинг ва сичқончанинг чап тугмасини бостан ҳолатда тугмани шаклнинг керакли жойига суринг.



3. Хоссалар ойнасига (Caption хоссаси) "Ёзув матнини ўзгартириш" тугмасининг сарлавҳасини киритинг.

Сарлавҳани босиш жараёнида & белги махсус қийматга эга ва акс эттирилмайди. ALT тугмаси ва тагига чизилган белги ёрдамида сиз объектни сичқонсиз танлаш имконига эга бўласиз. Бу клавишларнинг эквивалент комбинацияси дейилади.

Visual Basic ҳодисалари билан танишамиз

Windows дастурий таъминотининг катта қисмининг иши асосан фойдаланувчи ҳаракатларига жавоб тарзида бўлади. Масалан, матн процессорида сиз клавишни боссангиз, ҳодиса

(клавишни босиш) ҳужжатида белги пайдо бўлади. Жавобни келтириб чиқаради. (Ҳужжат кигиланади).

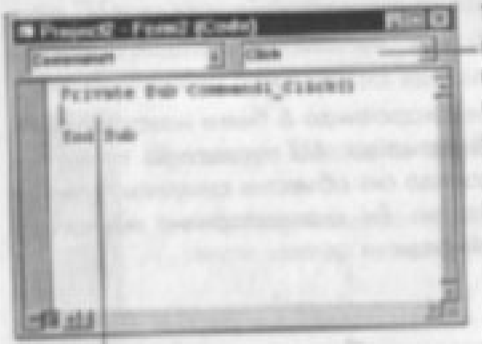
Visual Basic да дастур тузишда бу каби ҳаракатларга жавоблар жуда муҳим роль ўйнайди. Дастур яратиш жараёнида сиз маълум бир ҳодисага нисбатан жавоб сифатида бажариладиган йўриқнома ёзасиз. Сичқонча тугмасининг босилиши ҳодисага мисол бўла олади.

Буларнинг барчасини амалда текшириб кўриш учун шаклга тугмани киритинг ёки олдинги бўлимдаги мисолдан фойдаланинг. Кейинги қадамда сичқонча тугмасини икки марта босинг. Visual Basic код муҳаррирининг маълум кетма-кетликда ёзилган ушбу матнини очади:

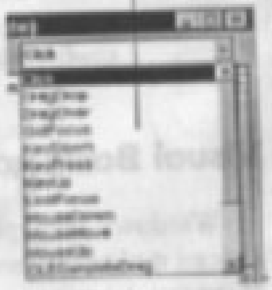
```
Private Sub Command_Click ( )
End Sub
```

Бу матн сатрлари орасига ўз кодингизни киритишингиз мумкин бўлиб, у сичқонча тугмасини босилганда бажарилади.

Код муҳарририни очиш учун сичқонча тугмасини икки марта босиш шарт эмас, бунинг ўрнига объектни сичқончанинг ўнг тугмаси ёрдамида кўрсатиш ва менюдан View Code (кодни кўриш) сатрини танлаш кифоя.



Тугмалар бошқа ҳодисаларга ҳам жавоб бериши мумкин. Қуйидаги рўйхатда ҳодисаларга нисбатан бажариладиган жавобларни кўришингиз мумкин.

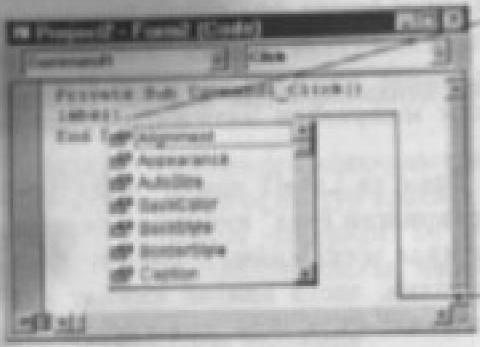


Сичқонча тугмасини икки марта босиб, код муҳарририни очишингиз ва дарҳол ўз кодингизни ёзишни бошлашингиз мумкин.

Сизнинг биринчи код сатрингиз

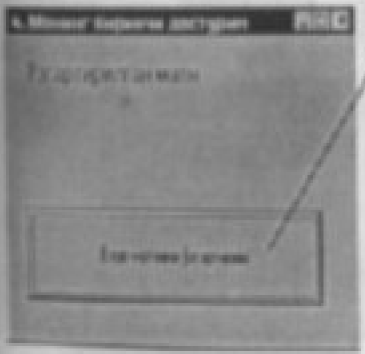
Мазкур бўлимдаги мисолни бажариш учун сизга тугмали ва ёзувли шакл зарур.

1. Click босиш ҳодисаси учун сичқонча тугмасини икки марта босиб, код муҳарририни очинг.



2. Label1 ёзувини ёзинг. (Нуқтани ёддан чиқарманг). Код ойнасида рўйхат пайдо бўлади, с белгисини ёзинг, рўйхатда "Caption" сўзи (Сарлавҳа) ажратилади, пробел ёки Tab тугмасини босинг ва "Caption" ёзуви матнга автоматик ёзилади.
3. "Ўзгартирилган матн" деб ёзиб, код сатрини охирига етказинг.

Код муҳаррирининг юқорида келтирилган имконияти AutoList Members (компонент рўйхати) деб аталади. Бу рўйхат қулай бўлишига қарамай, у баъзи кишиларнинг ғашига тегади. Агар бу нарса сизга ҳам тааллуқли бўлса, мазкур рўйхатни менюнинг Tools (Сервис) сатри Options (Параметрлар) диалогидан топиб, ўчириб қўйишингиз мумкин.



4. Visual Basic нинг асбоблар чизмасидаги Start тугмасини босиб, дастурни ишга туширинг. Шаклдаги тугмани боссангиз, ёзувнинг мазмуни ўзгаради.

(клавишни босиш) хужжатида белги пайдо бўлади. Жавобни келтириб чиқаради. (Хужжат янгиланади).

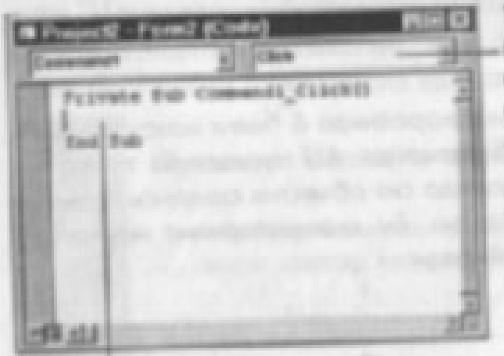
Visual Basic да дастур тузишда бу каби харакатларга жавоблар жуда мухим роль ўйнайди. Дастур яратиш жараёнида сиз маълум бир ходисага нисбатан жавоб сифатида бажариладиган йўриқнома ёзасиз. Сичқонча тугмасининг босилиши ходисага мисол бўла олади.

Буларнинг барчасини амалда текшириб кўриш учун шаклга тугмани киритинг ёки олдинги бўлимдаги мисолдан фойдаланинг. Кейинги қадамда сичқонча тугмасини икки марта босинг. Visual Basic код мухаррирининг маълум кетма-кетликда ёзилган ушбу матнини очади:

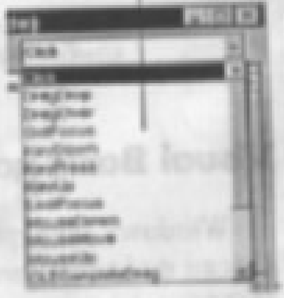
```
Private Sub Command_Click ()
End Sub
```

Бу матн сатрлари орасига ўз кодингизни киритишингиз мумкин бўлиб, у сичқонча тугмасини босилганда бажарилади.

Код мухарририни очиш учун сичқонча тугмасини икки марта босиш шарт эмас, бунинг ўрнига объектни сичқончанинг ўнг тугмаси ёрдамида кўрсатиш ва менюдан View Code (кодни кўриш) сатрини танлаш кифоя.



Тугмалар бошқа ходисаларга хам жавоб бериши мумкин. Қуйидаги рўйхатда ходисаларга нисбатан бажариладиган жавобларни кўришингиз мумкин.

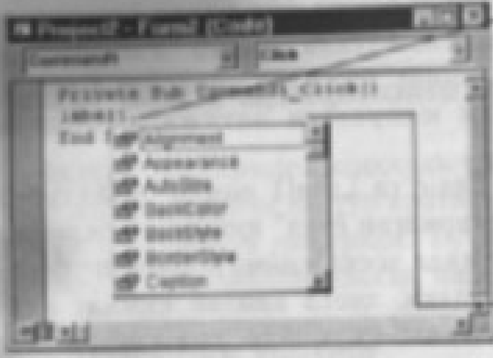


Сичқонча тугмасини икки марта босиб, код мухарририни очишингиз ва дархол ўз кодингизни ёзишни бошлашингиз мумкин.

Сизнинг биринчи код сатрингиз

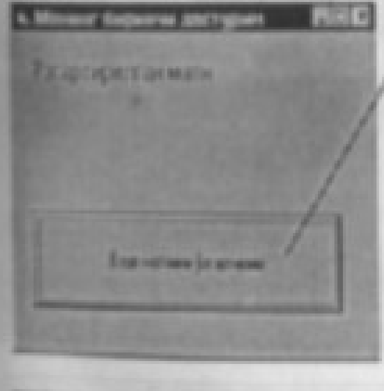
Мазкур бўлимдаги мисолни бажариш учун сизга тугмали ва ёзувли шакл зарур.

1. Click босиш ходисаси учун сичқонча тугмасини икки марта босиб, код мухарририни очинг.



2. Label1 ёзувини ёзинг. (Нуқтани ёддан чиқарманг). Код ойнасида рўйхат пайдо бўлсада с белгисини ёзинг, рўйхатда "Caption" сўзи (Сарлавҳа) ажратилади, пробел ёки Tab тугмасини босинг ва "Caption" ёзуви матнга автоматик ёзилади.
3. "Ўзгартирилган матн" деб ёзиб, код сатрини охирига етказинг.

Код мухаррирининг юқорида келтирилган имконияти AutoList Members (компонент рўйхати) деб аталади. Бу рўйхат қулай бўлишига қарамай, у баъзи кишиларнинг ғашига тегади. Агар бу нарса сизга хам тааллуқли бўлса, мазкур рўйхатни менюнинг Tools (Сервис) сатри Options (Параметрлар) диалогидан топиб, ўчириб қўйишингиз мумкин.



4. Visual Basic нинг асбоблар чизмасидаги Start тугмасини босиб, дастурни ишга туширинг. Шаклдаги тугмани боссангиз, ёзувнинг мазмуни ўзгаради.

Коддаги хосса қийматларини ўзгартириш

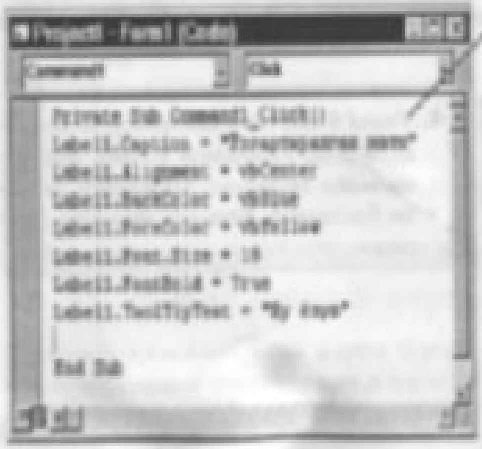
Сичқонча тугмасини босилганда жавоб тарихасида бажарилувчи кодни яна бир марта батафсилроқ кўриб чиқайлик.

Label1 Caption = "Ўзгартирилган матн"

Сиз бу шакл сарлавхасини бошка йўл билан ўзгартиришингиз мумкин эди. Агар сиз ёзувни танлаб "Ўзгартирилган матн" сарлавҳасини, унинг хоссалар ойнасида Caption сатрига ёзганингизда хам юқоридаги натижага эришган бўлардингиз.

Сиз ёзган код Visual Basic га Label1 номли объектнинг Caption хоссасига "Ўзгартирилган ёзув" қийматини беришга кўрсатма беради. Код матнида хосса қийматларининг берилишининг афзаллиги шундаки, хосса қандай қиймат олишини олдиндан билишни талаб қилмайди, сиз уларни дастурни бажариш вақтида аниқлашингиз мумкин. Кейинги бўлимдаги мисолда бу қандай ишлашини кўрамиз.

Бу ёндошувнинг самарали эканлигига ишонч ҳосил қилиш учун ёзувнинг бошқа хоссаларига кодда турлича қийматлар бериб кўринг.



Visual Basic объекти хоссаси кодда қандай ўрнатилишини кўриш учун бу мисолни теринг ва бажаринг

Мазкур бўлимда келтирилган мисол ёрдамида сиз Alignment (Жойлаштириш), BackColor (фон ранги), Fore Color (Олд томон ранги), Font (Шрифт) каби хоссаларни ўрнатишни билиб олдингиз. Visual Basic да хар бир объект жуда кўп хоссаларга эга, уларни эслаб қолиш одатда қийинчилик туғдиради, шуни осонлаштириш учун эса тегишли номли константаларни, масалан, VbCentre ва vbBlue ни ишлатиш мумкин. Visual Basic Маълумотномасидан (Справка) мавжуд константалар рўйхатини олиш мумкин.

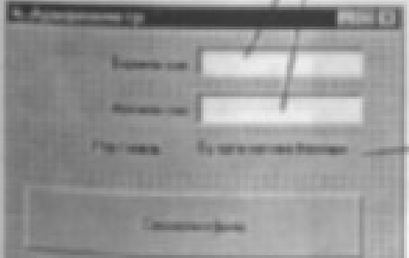
Visual Basic арифмометри

Мазкур бўлимнинг қизиқарли томони шундаки, сиз аниқ вазифани бажарувчи дастур яратасиз: у иккита сонни қўшади ва йиғиндисини чиқаради. Бу мисолда янги объект — TextBox (Матн ойнаси) билан ишлаймиз.

Арифмометрни яратиш

Агар кодни ёзиш жараёнида объектга мурожаат қилмоқчи бўлсангиз, у ҳолда объектга маънoли ном беринг. Мисол учун lb Result ёзуви lb — Label ёзув номи, Result (Натижа) каби иккита ёзувдан ташкил топган бўлиб, шу ёзувдан фойдаланаётганлигимни билдириб туради.

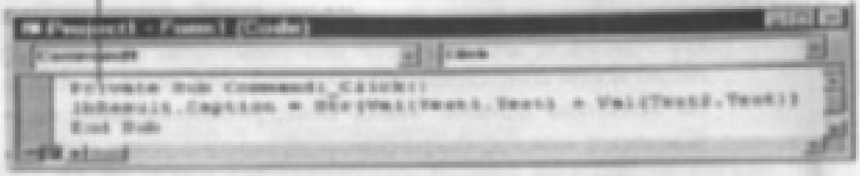
1. Бу белги матн ойнаси белгиси, шаклга бундай иккита элемент жойлаштиринг, TEXT (Матн) хоссасини бўш қолдиринг.



2. Шаклга тугма ва тўртта ёзувни расмда кўрсатилганидек жойлаштиринг.

3. Бу ёзувнинг хоссалар ойнасидаги Name (Ном) сатрига lbResult сўзини ёзинг.

4. Click (тугмасини босилиши) "Сонларни қўшиш" ҳодисасига жавоб тариқасида коднинг фақат бир сатри бажарилади:
LBResult.Caption= Str (Val(Text1.Text)+Val(Text2.Text)

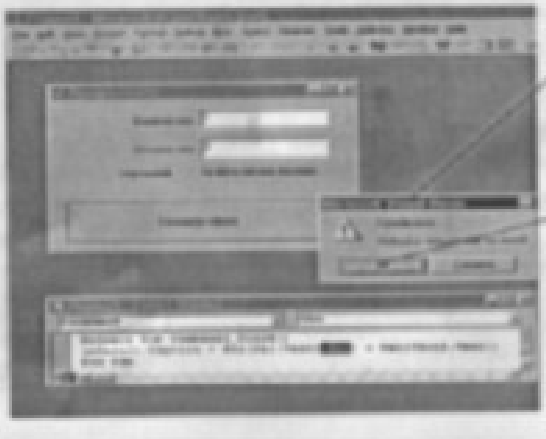


5. Энди дастурнинг ишлашини текшириб кўринг: Start тугмасини босинг. Ҳар бир матн ойнасида сонни ёзинг ва қўшиш натижасини акс эттириш учун "Сонларни қўшилсин" тугмасини босинг.

Visual Basic да кодни ёзиш жараёнида кодлари ёзувларни бош ҳарф ёки кичик ҳарфда ёзиш ихтиёрийдир, чунки Visual Basic код регистрига эътиборсиздир.

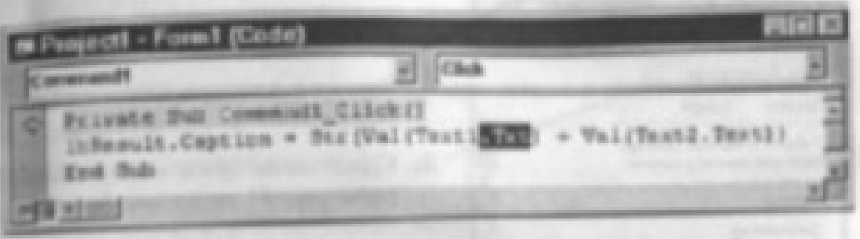
Хатоликлар

Биз ҳаммамиз хатоликка йўл қўйимиз, адашамиз. Агар сиз коди хатолик билан ёзилган дастурни ишга туширсангиз нима бўларкин? Масалан, Text1. Text ўрнига Text. 1Txt ёзувини ёзиб кўринг ва дастурни бажаринг. Кўпчилик ҳолларда Visual Basic ишни тўхтатади ва хатолик ҳақида ахборот беради, Visual Basic муҳаррири ойнасида хатоликка йўл қўйилган сатр коди кўринади.



1. Visual Basic ишни тўхтатади ва хатолик ҳақида хабар беради

2. "OK" тугмасини босинг, Visual Basic коднинг хатоликка йўл қўйилган қисмини кўрсатади.



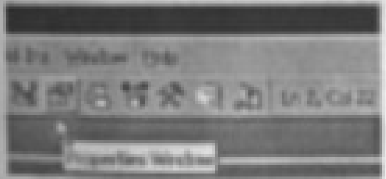
Экранда хатоликка йўл қўйилгонлиги ҳақида хабар пайдо бўлганда Help (Маълумот)тугмасини ёки F1 тугмасини босиб йўл қўйилган хато ҳақида тўлиқроқ ахборот ва уни тузатиш ҳақида маслаҳат олиш мумкин.

Сиз хатоликларни қамраб олувчи код ёзишингиз мумкин, бу дастурнингизни ишончлироқ ишлайдиган қилади. Сиз хатоликларни қамраб олиш билан 7-боб, "Тажрибали дастурчилар учун маслаҳатлар" қисмида танишасиз.

Маълумотлар қандай олинади

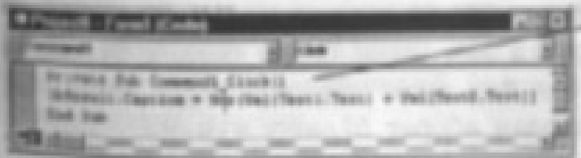
Visual Basic дастурлашни имкони борича соддалаштирса ҳам, лекин баъзи операциялар мураккаблигича қолди ва сиз ҳали ҳам катта ҳажмли ахборотни эслаб қолишингиз зарур. Бундай ҳолларда Visual Basic маълумотномасидан фойдаланишни билиб олиш муҳимдир.

Асбоблар ҳақида маълумот

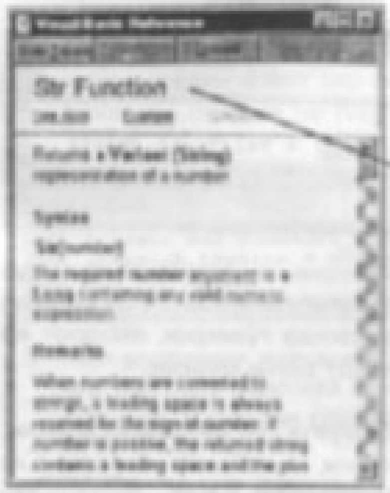


Асбоблар панелида истолган асбоб ҳақида ахборот олиш учун сичқонча кўрсаткичини шу асбобнинг нишони устида бир зум тўхтатинг.

Help (Маълумотнома) менюси

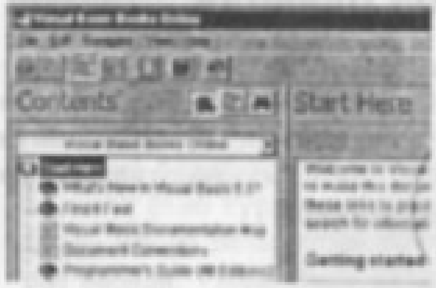


1. Қайси сўз ҳақида маълумотнома олмоқчи бўлсангиз, шу сўз устига курсорни олиб боринг.



2. Кўрсатилган сўзга маълумотнома олиш учун F1 тугмасини босинг.

Books Online



Help (Маълумотнома) менюсида Books Online (интерактив китоблар) сатрини танлаб, Visual Basic бўйича қўлланманинг интерактив версиясини очинг.

Visual Basic жуда машҳур бўлгани учун у ҳақда маълумотнома манбалари, шу жумладан китоблар, журналлар ва Internet да дискуссия гуруҳлари мавжуд.

Лойиҳа ойнаси билан танишамиз

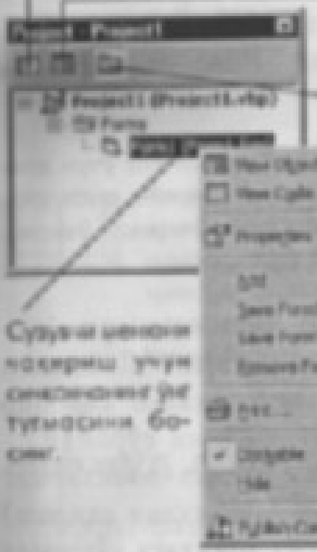
Visual Basic дастури бир неча шакллардан иборат бўлиши мумкин. У бирорта ҳам шакл билан боғлиқ бўлмаган "соф" код модули ёки бир неча шакллар билан боғланган мураккаб элементлардан иборат бўлиши мумкин.

Катта лойиҳанинг барча компонентларини бошқариш мураккаб бўлиши мумкин. "Лойиҳа ойнаси" (Project Window) бу муаммони ечишга ёрдам беради, яъни у лойиҳанинг барча компонентлари рўйхатини дарахтсимон структура кўринишида акс эттиради.

Танланган объект код муҳарририни ишга тушириш учун шу ерда тугмани босинг.

Объектнинг ўзини, масалан, шаклни кўриш учун шу ерда тугмани босинг.

Объектни узлуксиз рўйхат ёки папкаларга ажратилган рўйхат кўринишида кўриш учун шу ерда тугмани босинг.



Сузувчи менюни чақириш учун сичқончанинг ўнг тугмасини босинг.

Агар Visual Basicнинг сиз лойиҳа ойнасини, хоссалар ойнасини ёки исталган бошқа ойнани кўролмаётган бўлсангиз, уни экранга чиқариш учун View (Кўриниш) менюсидан фойдаланинг.

Лойиҳа ойнасидан қачон фойдаланиш керак?

Лойиҳа ойнаси лойиҳа дастури ишга туширилгач, одатда ҳар доим ҳам кўринавермайдиган шаклни, код модулини ёки бошқа объектларни экранга чиқаришингиз учун зарур. Лойиҳа ойнасидан фойдаланиб, айни жорий пайтда ишлаётган объектнигина қараб чиқиб, ўз иш столингизда тартиб ўрнатишингиз мумкин.

Агар сиз катта, бир нечта элементдан иборат лойиҳа яратаётган бўлсангиз, лойиҳа ойнаси ёрдамида навбатдаги шакл ёки код модули устида ишлашга ўтишингиз мумкин.

Яратилган дастурни сақлаш

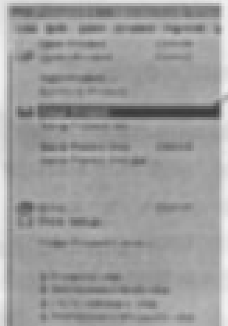
Visual Basic да яратилган дастур дискда бир нечта файлларда сақланади. Мисол учун, энг содда стандарт лойиҳа .VBP кенгайтмали файл ва шаклнинг .FRM кенгайтмали файлидан иборат.

Visual Basic да яратиладиган ҳар бир лойиҳа учун доимо янги папка (каталог) яратинг. Бу қоидага риоя қилсангиз, лойиҳадан нусха олиш, уни кўчириш, ўчириш каби ишларни бажариш қулай бўлади.

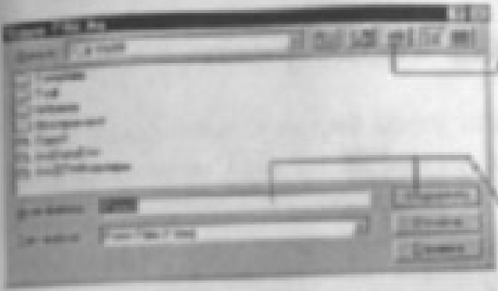
Visual Basic да яратилган лойиҳани биринчи марта сақлаётган бўлсангиз, File менюсидан Save Project (Лойиҳани сақлаш) сатрини танлайсиз, Visual Basic сиздан ҳар бир сақланаётган объект номи ва дискдаги жойини сўрайди.

Лойиҳани кейинги сақлашларда эса дискда лойиҳанинг файлларини янгилаш учун Save Project (Лойиҳани сақлаш) сатрини танлашнинг ўзи кифоя. Агар сиз лойиҳага қўшимча объектлар, шакллар ёки модуллар қўшган бўлсангиз, у ҳолда янги ном бериш ва сақланиш жойини кўрсатишингиз зарур.

Лойиҳани биринчи марта сақлаш



1. File (Файл) менюсидан Save Project (Лойиҳани сақлаш) сатрини танлаб олинг.



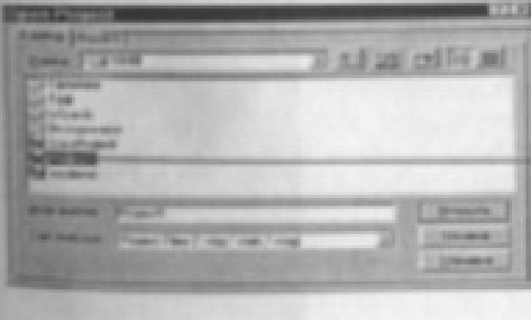
2. Янги каталог (папка) қосил қилиш учун бу ерда тугмани босинг. Папка номини киритинг. Папкани очиш учун унинг устига сичқонча кўрсаткичини қўйиб, икки марта босинг.
3. Лойиҳанинг биринчи файли номини киритинг ва файлни сақлаб қолиш учун Save (Сақлаш) тугмасини босинг.
4. Юқоридаги кадрни лойиҳанинг ҳар бир файли учун такрорланг.

Агар сиз сақланган лойиҳа билан ишлаганда File менюсидан Save As (Қандай сақлаш) сатрини танласангиз, у холдан сиз лойиҳанинг асосий файлига янги ном ва жой берида сиз лойиҳанинг асосий файлига янги ном ва жой беришингиз мумкин. Лекин лойиҳанинг бошқа файллари жойида қолади. Қолган файлларнинг ҳам жойи ва номини ўзгартириш учун юқорида кўрсатилган кадрни қайтаринг.

Мавжуд лойиҳани юклаш

Visual Basic да мавжуд лойиҳани очиш учун қуйидаги амаллар кетма-кетлигини бажаринг:

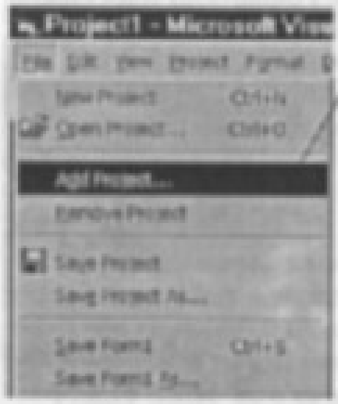
1. File (Файл) менюсидан Open Project (Лойиҳани очиш) сатрини танланг.



2. Охирги пайтда ишлаган лойиҳаларнинг рўйхатини очиш учун Recent (Янгилар) тугмасини босинг.
3. Лойиҳанинг .VBP кенгайтмали файлини танланг, уни тугмани босиб, ажратиб олинг ва Open (Очиш) тугмасини босинг.

Бир нечта лойиҳани бир вақтда юклаш

Visual Basic бир вақтнинг ўзида бир нечта лойиҳани очиш имконини беради. Бунинг учун File (Файл) менюсида Add Project (Лойиҳа қўшиш) сатридан фойдаланинг.



Жорий юкланган лойиҳани ёпмасдан, янги лойиҳани очиш учун Add Project (Лойиҳа қўшиш) сатрини танланг.

Бир вақтда юкланган бир нечта лойиҳа сизни адаштириши мумкин. Шунинг учун иложи борича бир нечта лойиҳани бирданига юкламаслик керак. Жуда мураккаб дастурларни, масалан, ActiveX элементларидан фойдаланиб яратиш бундан мустасно.

ИККИНЧИ БОБ

Бошқариш элементлари

Visual Basic кўп сондаги тайёр объектларга эга бўлиб, бу объектлардан сиз ўзингиз тузадиган дастурингизда фойдаланишингиз мумкин.

Мазкур бобда Visual Basic асбоблар панелининг барча асосий объектлари муҳокама қилинади.

Бошқариш элементлари ва услублари

Visual Basic асбоблар панелида жойлашган объектларни одатда бошқариш элементлари деб аталади. Агар сиз Windows кобиғида ишловчи бошқа дастурлар билан ишлаган бўлсангиз, Visual Basic бошқариш элементларининг кўпчилиги сизга яхши таниш.

1-бобда биз бошқариш элементлари ўзига хос хусусиятларга эгалиги ва бу хусусиятлар бошқариш элементларининг ташқи кўриниши ва ишлашини белгилашини кўрган эдик. Бошқариш элементлари ҳодисаларга эга бўлиб, улар натижасида жавоб ишга туширилади.

Аниқ ишларни кўриб чиқишдан олдин Visual Basic бошқариш элементлари хусусиятларидан яна бири — услублар моҳияти билан яқинлаб танишиш зарур.

Услуб нима?

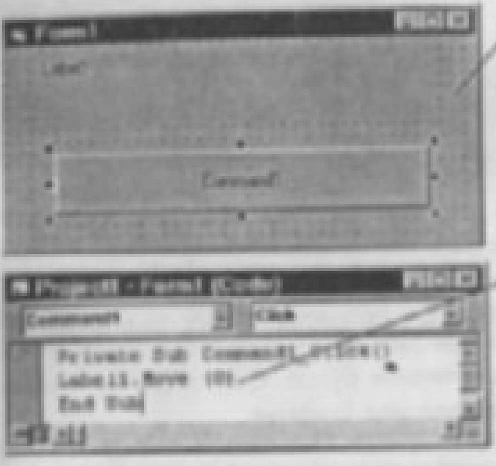
Хусусият — бу объектда бор нарса, услуб — бу объект бажарадиган иш.

Мисол кўрамиз: автомобилда ранг, модель, ишлаб чиқарилган йил ва тезлик каби хусусиятлар бор. Энди автомобилнинг жойидан қўзғалиб, тезлик олиши ва тўхташини олайлик. Булар автомобиль бажарадиган иш. Агар автомобилни Visual Basic объекти деб тасаввур қилсак, у ҳолда Start (Жойидан қўзғалиш) ва Stop (Тўхташ) унинг иккита услуби ҳисобланади. Кўп ҳолларда услублар учун қўшимча маълумотлар зарур бўлади, бу маълумотлар параметрлар (кўрсаткич қийматлар) деб аталади. Код ёзиш жараёнида параметрлар услуб номидан кейин қавс ичида ёзилади ва ҳаракатнинг қай тарзда бажарилишини белгилайди. Масалан, автомобиль тез ёки секин тўхташи мумкин.

Агар Visual Basic маълумотлар тизимини исталган аниқ бошқариш элементи учун очсангиз, у ҳолда унинг ушбу хоссаларини кўриш мумкин: Properties (Хусусиятлар), Events (Ҳодисалар) ва Methods (Услуб) лар. Юқоридагиларнинг ҳаммаси биргаликда танланган бошқариш элементи нима қила олиши мумкинлиги ҳақида тўлиқ ахборот беради.

Мисол

Ёзувлар Move (Ўрин алмаштириш) услубиятига эга бўлиб, у сизга уни шакл ичида позициялашга имкон беради



1. Шаклга ёзувни ва тугмани жойлаштиринг.

2. Тугманинг Click (Босиш) ҳодисасининг кодида: Label1.Move(0) ёзувини ёзинг. Дастурни ишга туширинг ва тугмани босинг.

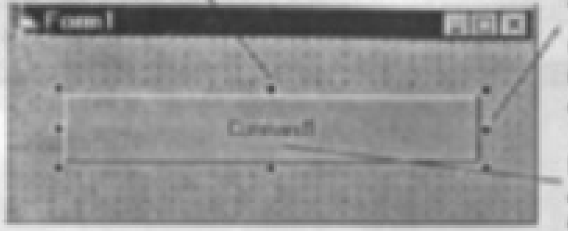
Буйруқ тугмаси (CommandButton)

Агар 1-бобдаги барча мисолларни бажарган бўлсангиз, у ҳолда энди буйруқ тугмалари сизга яхши таниш. Дастур тузиш жараёнида фойдаланувчига маълум бир оддий операцияни бажариш, маълумотнома олиш, танлашни тасдиқлаш ёки бекор қилиш имкониятини яратиш учун тугмалардан фойдаланиш зарур.

Шаклда тугмаларнинг ортиқча кўп бўлиши фойдаланувчини чалғитади. Бунинг ўрнига дастур лойиҳасига қўшимча шакллар ёки диалог ойналарини қўйишга ҳаракат қилиб кўриш мумкин.

Одатда, сиз дастур тузиш жараёнида тугмаларга хусусиятлар берасиз ва ҳар бир тугманинг Click (Босиш) ҳодисасига код ёзасиз.

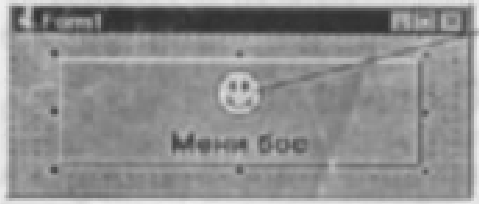
Тугма ўлчамларини ўзгартириш учун ўлчовларни ўзгартириш маркерларидан фойдаланинг. Caption (Сарлавҳа) хусусияти тугмада қандай ёзув пайдо бўлишини белгилайди.



Тугма учун клавишлар эквивалент комбинациясини яратиш учун & белгисидан фойдаланинг.

Font (Шрифт) хусусияти сарлавҳа шрифтини ўзгартиради.

Агар сиз, фойдаланувчига шаклдаги тугмани Return (Enter) клавиш ёрдамида босишга имконият яратмоқчи бўлсангиз Default (Ўрнатилган) хусусиятини True ҳолатига ўтказинг. Фойдаланувчи Escape клавиши ёрдамида шаклдаги тугмани ишлатмоқчи бўлса, Cancel (Бекор қилиш) хусусиятини True ҳолатига ўтказинг.



Шаклга ўрнатилган тугманинг кўриниши чиройли бўлиши учун Style (Стиль) хусусиятини Graphical (График) ҳолатига ўтказинг. Сўнгра Picture (Расм) хусусиятини ўрнатиб, зарур белги ёки расм танланади. Мажбур ҳолатда тугмани босилган ҳолати учун бир расм, босилмаган ҳолати учун эса бошқа расм қўйиш мумкин.

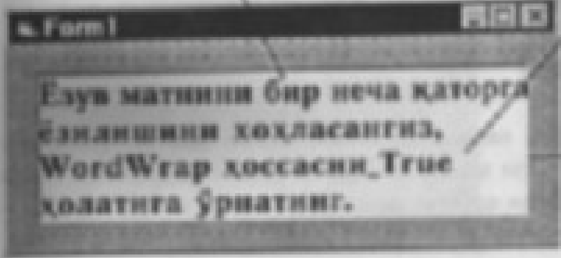
Тугманинг бемаврид босилишининг олдини олиш учун Enabled (Уланган) ва Disabled (Ўчирилган) хусусиятларини одатда кодда кўрсатилади.

Ёзув (Label)

1-бобда сиз Visual Basic объектларидан яна бири "Ёзув" билан танишгансиз. Ёзувлар — фойдаланувчи ўзгартириши, таҳрир этиши керак бўлмаган матнларни акс эттириш

учун ишлатилади. Ёзувнинг матнини сиз кодда ўзгартиришингиз мумкин, бунинг учун Caption (Сарлавҳа) хусусиятини ўрнатиш зарур.

Бир неча сатрли қатордан ёзувни ёзиш учун Word Wrap (Сўзларни автоматик кўчириш) хусусиятини True ҳолатига ўтказинг.

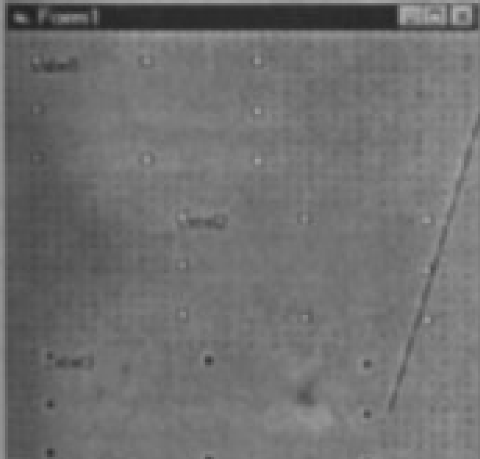


Оқ ёки рангли мувит (фон) шаклга чирой беради.

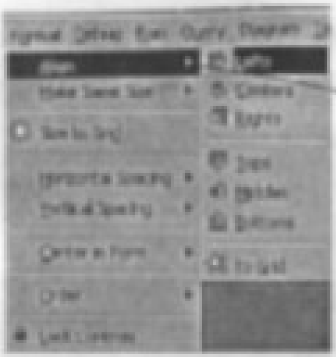
Ойнанинг чегараларини асосий қилиш учун BorderStyle хусусиятини Fixed ҳолатига ўтказинг.

BackStyle (мувит услуби) хусусиятини Transparent (Шаффоф) ҳолатига ўтказсангиз, шакл мувити шаффоф рангли бўлади. Сўнгра сиз бу шаклни бошқа шакл ёки расм устига қўйиб, қизиқарли натижага эга бўлишингиз мумкин.

Ёзувни қандай текислаш мумкин?



1. Шаклга бир нечта ёзув жойлаштиринг. Бу шаклда жойлаштирилган ёзувларни асосий ёзувга нисбатан жойлаштириш учун сурилиши зарур бўлганларини Shift тугмаси ва кўрсаткич ёрдамида ажратинг. Энг охирида ажратилган ёзув асосий ёзувга айланиб қолади ва унинг ўлчовларини кўрсатиб турувчи маркёрлар қора рангга киради. Қолган ёзувлар эса шу ёзувга кўра текисланади.



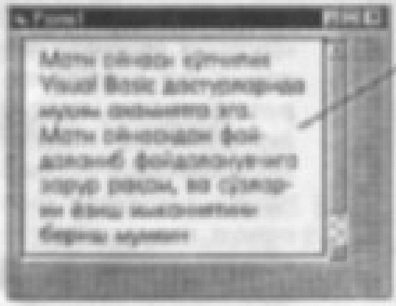
2. Format (Шакл) менюсидан Align (Текислаш) сатрини танланг. Ёзувларни асосий ёзувнинг чап томонига кўра текислаш учун очилган менюдан Left (чап томон) сатрини танланг.

Format (Шакл) менюсида Make Same Size (Ўлчамларни текислаш) сатри ҳам бор. Менюнинг мазкур сатри Align (Ростлаш) сатри каби ишлайди. Бу иккала буйруқ ёрдамида чиройли ёзувларни тез ва осон ҳосил қилиш мумкин.

Матн ойнаси (TextBox)

Visual Basic дастурларини яратишда матн ойналари жуда муҳим аҳамиятга эга. Ёзув ва матн ойнасининг асосий фарқи шундаки, дастурнинг ишлаш жараёнида ҳам матн ойнаси жойлаштирилган жойга исталган матнни ёзиш мумкин. Матн ойнаси ёзувдан кўра каттароқ ҳажмдаги матнни акс эттириши мумкин бўлиб, бунда фойдаланувчи матнни ойна орқали суриб (айлантириб) ўқиши мумкин. 1-бобда "Арифмометр" дастурини тузишда биз матн ойнаси элементидан фойдаланганмиз.

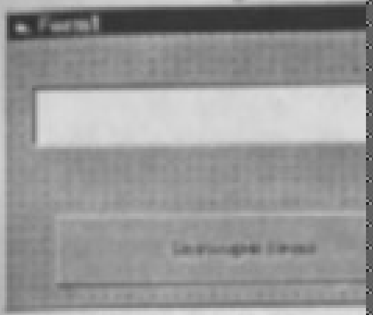
Айлантириладиган матн ойнасини яратиш



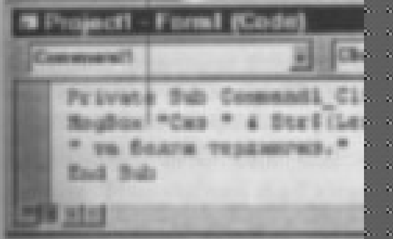
Айлантириладиган матн ойнасини яратиш учун унинг MultiLine (Кўп сатрли) хусусиятини True қийматига ўзгартиринг, ScrollBars (Айлантиргич полосаси) хусусиятини "2-Vertical" (Вертикал) қийматига ўзгартиринг.

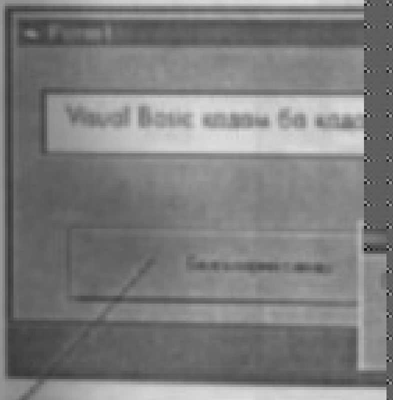
Белгиларни санаш да

Қуйидаги мисол матн
ёрдамида қандай кириш



2. Тугмачанинг Click (Босиш) ҳо
MSGBOX ("Сиз" &Str$(Len (Tex

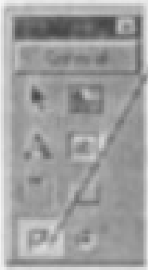




3. Дастурни ишга туширинг ва
тугмани босинг.

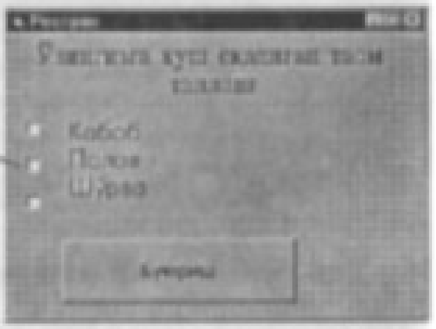
тини берадиган менюлар яратиш имконини беради. Кейинги мисолда, ресторанда хўранда учун танлаш имконини берадиган меню мисолида байроқчалардан фойдаланишни ўрганамиз.

"Кобоб" дастури

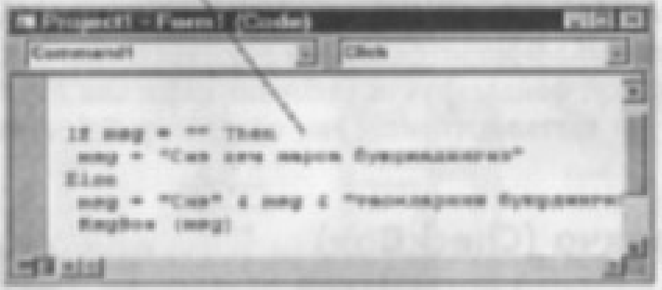


1. Олдинги мисолларда кўрсатилган тартибда шакл яратинг. Шаклга учта байроқча элементини жойлаштиринг.

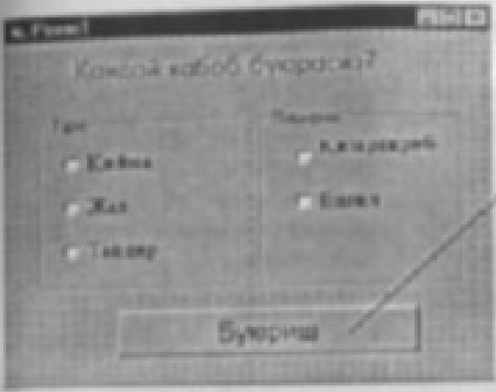
2. Шаклга ёзув жойлаштиринг ва ҳар бир байроқча номини киритинг.



1. Ҳар бир байроқча учун Value (Қиймат) хусусияти белгиланган ёки белгиланмаганлигини текширувчи код ёзишингиз мумкин, яъни агар Value хусусияти бирор байроқчада vbChecked га тенг бўлса, у ҳолда бу байроқча белгиланган.



Мазкур бўлимда кўриб чиқилган мисолда Visual Basic кодини тўғри ёки нотўғрилигини кузатиш ва дастур ишини текшириш имконига эга эмасмиз. Бунинг учун Visual Basic коди ҳақида батафсил ахборотга эга бўлиш зарур бўлиб, у билан навбатдаги бобда шуғулланамиз. Ҳозирча байроқчани билиб олганимиз етарли.

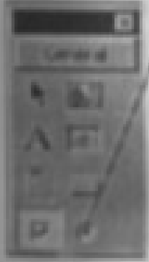


4. Кабобни еб, роҳатланинг!

Улагич (OptionButton)

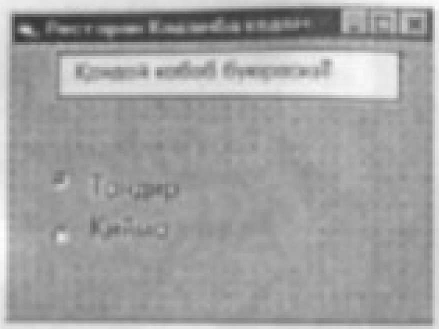
Улагичлар байроқчаларга ўхшаш хусусиятга эга бўлиб, муҳим фарқи шундаки, бир гуруҳда фақат битта улагичдан фойдаланиш мумкин. Бир улагичнинг ишга тушиши, автоматик тарзда бошқаларининг узилишига асос бўлади. Мисол учун кабоб пиширилган ёки қизартириб пиширилган бўлиши мумкин, лекин бир вақтнинг ўзида иккови ҳам бажариш мумкин эмас.

Кабоб тайёрлаш усули

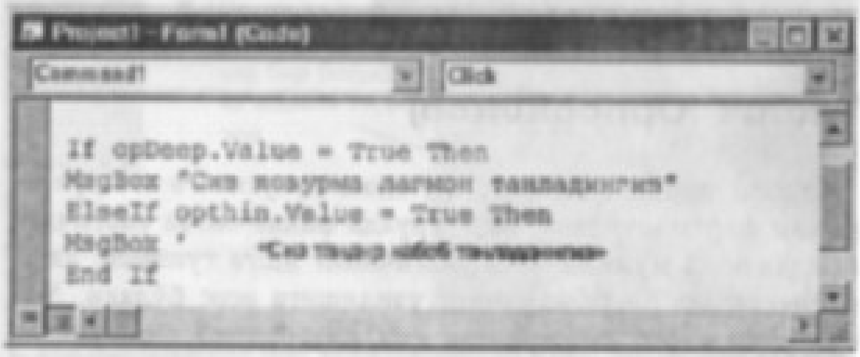


1. Асбоблар панелидан улагич нишонини танланг ва шаклга иккита улагични жойлаштиринг.

Шаклга улагичлар гуруҳи жойлаштирилганда улардан бирининг Value (Қиймат) хусусиятини True ҳолатига ўтказиб қўйиш зарур. Акс ҳолда дастур ишга туширилганда улагичлардан бирортаси ҳам ишга тушмайди.



2. Расмда кўрсатилганидек, шаклга 2 та улагич жойлаштиринг. Уларга мос равишда OptDeep (Яхшилаб қовурилган) ва OptThin (Енгил қовурилган) деб ном беринг. Улагичлардан бирини танланг ва унинг Value (Қиймат) хусусиятини True ҳолатига ўтказинг.



3. Улагичларнинг Value (Қиймат) хусусиятини текширувчи код матнини ёзинг. Агар у True га тенг бўлса, улагич ишга туширилган.

Сиз бу дастурни олдинги бўлимда тузилган дастур билан бирлаштиришингиз мумкин бўлиб, бу билан сиз фойдаланувчига кабоб таркиби ва турини битта шаклда танлаш имконини берган бўласиз.

Рамка (Frame)

"Рамка" сўзи безак элементи деган маънони беради ва ўз номи билан шаклнинг ташқи кўринишини безаш учун ишлатилади. Рамканинг яна ҳам муҳимроқ вазифаси шундаки, у бошқа объектлар учун контейнер бўлади. Бу айниқса биргаликда ишлайдиган объектларни, масалан, улагичларни гуруҳлаш учун фойдалидир.

"Кабоб" дастурида рамкалар.

1. Асбоблар панелида рамка белгисини танланг ва шаклга 2 та рамка жойлаштиринг.

2. Ҳар бир рамкани гуруҳнинг вазифаси тушунарли бўладиган қилиб номланг.

Берилган бошқариш элементи рамкада жойлашганлигини текшириш учун, уни кўчириш лозим. Рамкада гуруҳланган бошқариш элементлари у билан бирга кўчади.

Агар иккита объект бир хил ўлчамли шаклда текисланиши талаб этилса, shift клавишини босиш билан бирга босишни бажариб, иккала объектни танланг. Сўнгра Format менюсида Align ва Make Same Size сатрларини танлаб, иккала объектни текисланг.

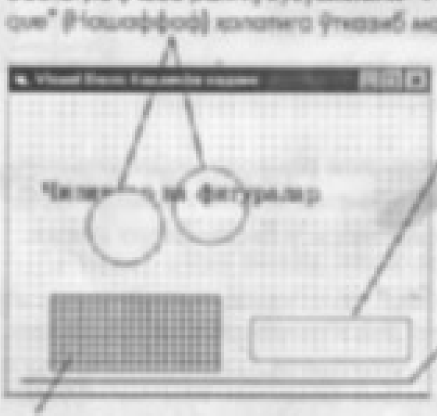
Шакллар ва чизиқлар (Shape, Line)

Геометрик шакллар ва чизиқларнинг хусусиятларини тушуниб олиш ўқувчи учун мураккаб бўлмаса керак. Дастур тузиш жараёнида улардан фойдаланиб шаклга безак ва аниклик киритинг. Шу билан бирга дастурни ишлатиш жараёнида улар учун код ёзиб, махсус эффектлар ҳосил қилиш мумкин.

Геометрик шакллар ва чизиқлардан фойдаланиш

Геометрик шакл ва чизиқларни чизиш учун мазкур бўлимда кўрсатилган объектлардан фойдаланиш шарт эмас. Visual Basic нинг шакл ва график ойналари яна бир хусусиятга эга, яъни бевосита код ичида расм, геометрик шакл ва чизиқлар чизиш мумкин. FillStyle (Бўяш услуби) хусусияти график ва диаграммаларни ҳар хил услубда бўяш имконини беради.

BackStyle (Асос ранги) хусусиятини "1-Transparent" (Шаффоф) ёки "1-Opaque" (Ношаффоф) ҳолатига ўтказиб махсус эффектлар ҳосил қилиш мумкин.



Shape (Шакл) хусусияти олтита геометрик шаклдан биттасини танлаш имконини беради. Булар бурчаклари юмалоқланган тўғри тўртбурчак ҳам киради.

Тўғри чизиқ — Visual Basic нинг энг оддий объектларидан биридир.

FillStyle (Бўяш) хусусияти объектни турли шакллардаги штрих чизиқлар билан тўлдириш имкон беради, бу графиклар ва диаграммалар учун жуда мосдир.

Агар сиз чизаётган расмда бир-бирини тўсувчи бир нечта объект (расм)лар бўлса, Format (Шакл) менюсидан Order (Тартиб) сатрини танланг ва керакли объектни олдинга ёки орқага кўчириб жойлаштиринг.

Расм (Image)

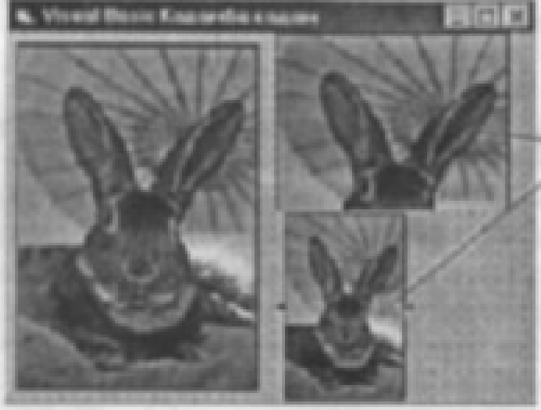
Дастур яратиш жараёнида расм ва график объектлардан фойдаланишда кўпинча бундай савол туғилади — қайси ҳолларда график ойнадан, қайси ҳолларда расмдан фойдаланиш зарур? График ойна расм элементида мавжуд бўлган барча хусусиятларга эга бўлиши билан бирга бошқа қўшимча хусусиятларга ҳам эга. Ўз навбатида бу имкониятлар маълум миқдорда хотира ҳажмидан жой олади. Шунинг учун дастур тузиш жараёнида имкони борича соддароқ ҳисобланган, хотирадан камроқ жой эгалловчи — расм элементидан фойдаланиш зарур.

Расм ўлчамларини ўзгартириш

1. Бу мисолни бажариш учун, шаклга учта бошқариш элементини жойлаштиринг, Picture (Расм) хусусиятидан фойдаланиб учала элементга бир хил расмни юкланг.

Учта бир хил расмни шаклга жойлаштиришнинг яна бир йўли қуйидагича: бошқариш элементини жойлаштиринг ва унга тасвирни юкланг, кейин шу элементни танлаб, Copy (Нусха) олинг, кейин Paste (Қўйиш) буйруғини беринг. Кўрсатилган амалларни бажарсангиз, "Хотите ли вы создать массив элементов управления?" ("Бошқариш элементлари массивини яратишни хоҳлайсизми?") саволи чиқади, мазкур ҳолатда No (Йўқ) жавобини танланг.

2. Юклаш чоғида бошқариш элементи ўз ўлчамларини автоматик ўзгартиради. Расм юклангандан кейин бошқариш элементларининг керакли ўлчамларини ўлчамлар маркёрлари ёрдамида ўзгартирилади. Қолган иккита элементдаги расмларни бошланғич (биринчи) расмдан кичикроқ қилиб жойлаштиришингиз мумкин.



3. Элементлардан бирининг Stretch (Чўзиш) хусусиятини True қийматига ўтказинг. Элементлардаги расм шаклга мосланади, ўлчамлари катталиги эса кесилган ҳолатда бўлади.

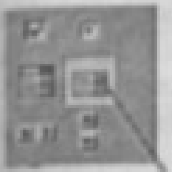
Бошқариш элементи — расм нима иш бажаради?

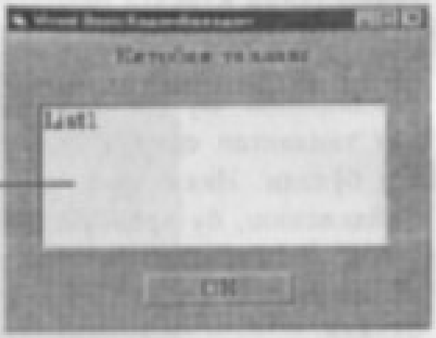
Бошқариш элементи — расм ҳам, график ойна сингари расм юклаш учун жуда яхши восита ҳисобланади. Лекин сиз расм элементидан бошқа бошқарув элементлари учун контейнер сифатида фойдалана олмайсиз, унда график ёки матн кодини ёза олмайсиз.

Рўйхат ойнаси (List Box)

Рўйхат ойнаси — бу Visual Basic бошқариш элементларининг ичида энг фойдалиларидан бири ҳисобланади. Фойдаланувчига танлаш имкониятини яратмоқчи бўлсангиз, қуйидагини ҳисобга олинг — рўйхат ойнаси байроқчалар ёки улагичлар сатрига қараганда бир қатор устунликларга эга, чунки унда бирдан то бир неча мингтагача элементларни жойлаштириш мумкин. Манзиллар китоби ёки бизнес-ёзувлар каби маълумотлар базаларини тузишда айни шу рўйхат ойнасидан фойдаланилади.

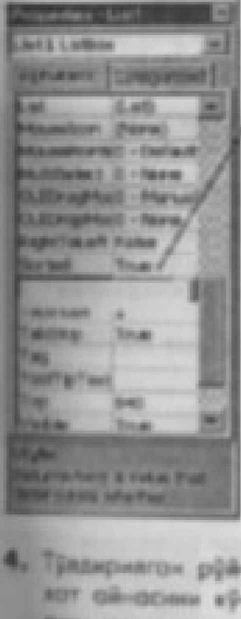
Рўйхат ойнасига сатрлар қўшиш





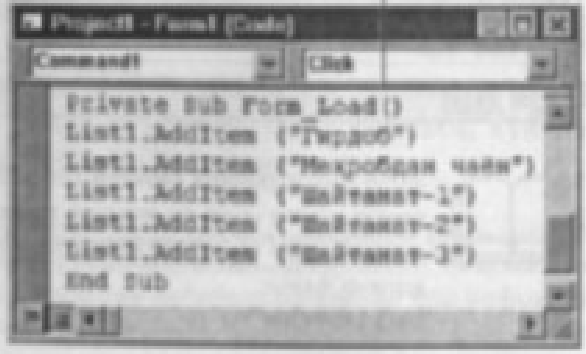
1. Бу мисолда шаклга рўйхат ойнаси, ёзув ва тугма жойлаштирилган.

Бу бўлимдаги мисолда Load (Юклаш) ҳодисасидан биринчи марта фойдаланамиз, унинг қулайлик томони шундаки, ёзилган код фойдаланувчи шаклни кўриб улгургунигача қадар бажарилади.



2. Sorted (Сараланган) хусусиятини True қийматига ўтказилса, рўйхат ойнаси элементлари алифбо тартибида сараланади.

3. Load (Юклаш) хусусияти кодини очиш учун ойнанинг тугмасини шакл устига келтириб икки марта босинг. Рўйхат ойнасига бир неча сатр қўшиш учун AddItem (қўшимча қўшиш) усулидан фойдаланинг. Мисол учун List1. AddItem ("Тирдоб")

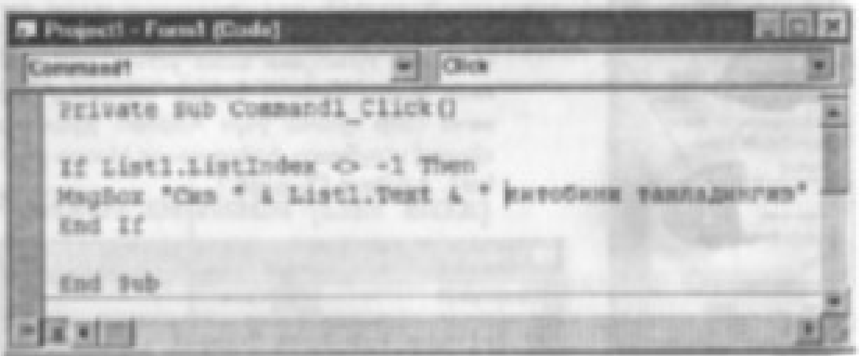


4. Тузилаётган рўйхат ойнасини кўриш учун дастурни ишга туширинг.

Рўйхат ойнасини расмда келтирилган тартибда тўлдирилса, тайёр рўйхат ойнасини хосил қиласиз. Фойдаланувчи рўйхат ойнасида келтирилган рўйхатдан қайси сатрни танлаганини кўрсата олмасангиз, рўйхат ойнасидан фойда йўқ хисоб. Бу вазифани иккита усулдан фойдаланиб бажариш мумкин: Биринчиси — ListIndex (Рўйхат индекси) дан фойдаланиш. Бу холатда рўйхатда фойдаланувчи томонидан танланган сатр рўйхат ойнасида ўз тартиб рақамига эга бўлади. Иккинчи усул — Text (Матн) хусусиятидан фойдаланиш, бу ерда у танланган сатр матнини ўз ичига олади. Агар рўйхат ойнасида сатрлардан бирортаси ҳам танланмаган бўлса, у холда ListIndex хусусияти -1қийматга эга бўлади.

Рўйхат ойнасида танланган сатр қийматини (матни)ни қайтариш

1. *Дастлаб сизнинг кодингиз ListIndex хусусияти қийматини текшириб кўриши зарур. Агар ListIndex –1 га тенг бўлса, рўйхатнинг бирорта ҳам элементи танланмаган. Агар ListIndex хусусиятининг қиймати –1 дан фарқ қилса, Text хусусиятидан фойдаланиб, танланган сатр мазмунини кўрсатинг.*



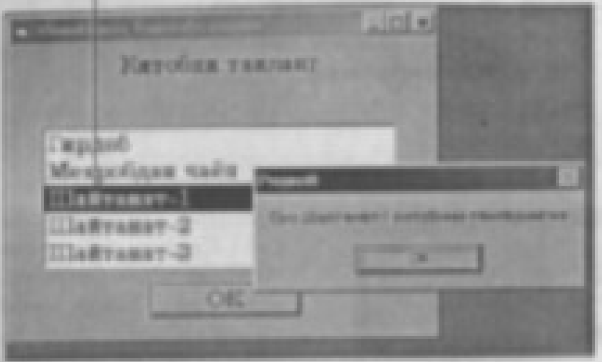

```
Private Sub Command1_Click()

If List1.ListIndex <> -1 Then
MsgBox "Сиз " & List1.Text & " китобини танладингиз"
End If

End Sub
```

Агар сизга рўйхат ойнаси сатрларидан бири шакл биринчи марта ишга туширилганида танланган бўлишини хоҳласангиз, Load ҳодисаси кодида ListIndex хусусиятини қуйидагича ёзинг:
List1.ListIndex =0

2. Энди дастурни ишга туширинг. Рўйхат ойнасидан ҳеч нарса танланмаган бўлсада, сиз OK тугмасини боссангиз ҳеч қандай ҳодиса юз бермайди. Агар сиз рўйхатдан китоб номини топиб OK тугмасини боссангиз, танланган китоб ахборотлар панелида пайдо бўлади.



Тўр (FlexGrid)

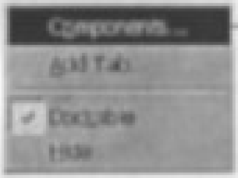
Тўр — рўйхат ойнасидан кейинги қадамдир. Тўр оддий рўйхат бўлмасдан, балки уяларга (ячейка) эга бўлган рўйхатдир. Хар бир уяда маълум матн ёки расм хамда матн ва расм биргаликда бўлиши мумкин. Тўр ёрдамчи элементи сифатида айлантиргичларга эга бўлгани учун, кўриш майдонига сиғадиганидан кўпроқ уяларни жойлаштириш имконига эга бўласиз.

FlexGrid ни ўрнатиш

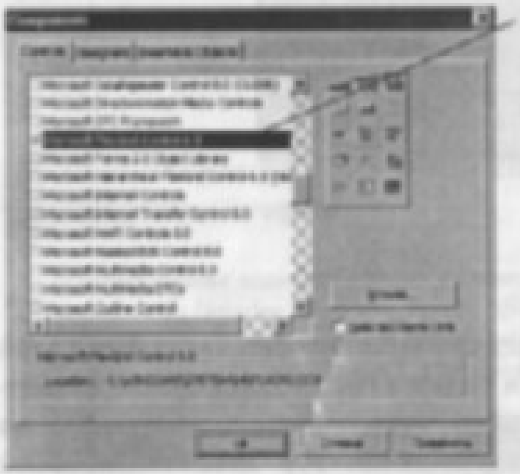
FlexdGrid — бу ActiveX бошқариш элементидир. Агар сизнинг асбоблар панелингизда FlexGrid бўлмаса, қуйидаги амалларни бажариб, уни ўрнатинг.

1. FlexGrid сариқ бланкси ана шундай кўринишга эга. Агар у асбоблар панелида кўринмаса, сичқончанинг ўнг тугмасини асбоблар панелининг бўш жойида босинг.

Агар сиз асбоблар панелидаги ҳар бир белги номини аниқлашни билмасангиз, унда сичқон курсорини унда тутиб туринг. Бир оз вақтдан сўнг бошқариш элементи номи изоҳи билан пайдо бўлади.



2. Сузувчи менюдан Components (Компонентлар) — сатрини танланг. Components диалог панели пайдо бўлади.



3. Рўйхатда Microsoft FlexGrid Controls 5,0 сатрини топинг, уни белгилаш учун унинг ёнида катак ойнада босинг. Агар бу рўйхатда FlexGrid 5,0 бўлишини топа олмасангиз, Browse (Топиш) тугмасини босинг ва Windows System каталогидан MSFLXGRD OCX файлини топинг. Агар сиз бу файлни топа олмасангиз, FlexGrid элементи сизнинг тизимингизда ўрнатилмаган. Visual Basic 5,0 ни такроран ўрнатинг.

4. OK тугмасини босинг, асбоблар панелида FlexGrid белгиси пайдо бўлади.

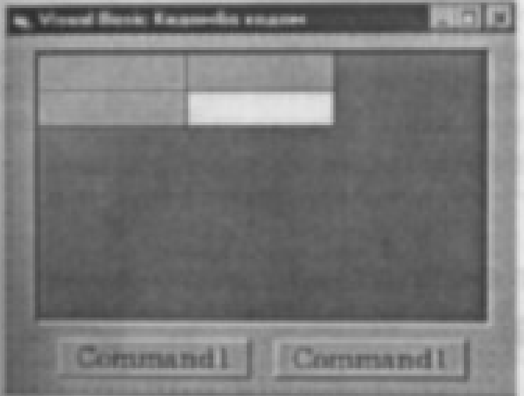
Active X бошқариш элементларининг барчаси FlexGrid элементи каби ўрнатилади. Сиз бу бўлимдан ActiveXнинг бошқа бошқаришнинг элементларини ўрнатиш бўйича қўлланма сифатида фойдаланишингиз мумкин.

FlexGrid — бошқариш элементи кўп вазифали элемент бўлиб, унинг имкониятлари сизни адаштириши мумкин. Биринчидан, FlexGrid ўлчамларини ўрнатишни ўзлаштириб олишингиз керак, улар Cols ва Rows (Устунлар сони ва қаторлар сони) хусусиятлари билан ўрнатилади. Иккинчидан, сиз уларнинг қийматларини бериш ва уларни

қайтаришни билишингиз зарур. Бунинг учун сиз Text ва CellPicture хусусиятларидан фойдаланинг. Дастур иши давомида FlexGrid жорий уяга эга бўлиб Cols ва Rows хусусиятлари уя кўрсаткичини белгиласа, Text ва CellPicture хусусиятлари унинг мазмунини бериш ёки қайтариш имконини беради.

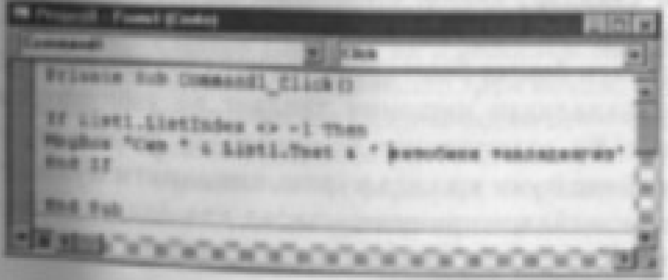
Бу бўлимда FlexGrid элементининг кўпгина хусусиятлари келтирилмади, FlexGrid элементининг барча хусусиятлари ҳақида тўлиқ маълумот олиш учун FlexGridни танланг ва F1 тугмасини босинг.

Мисол:

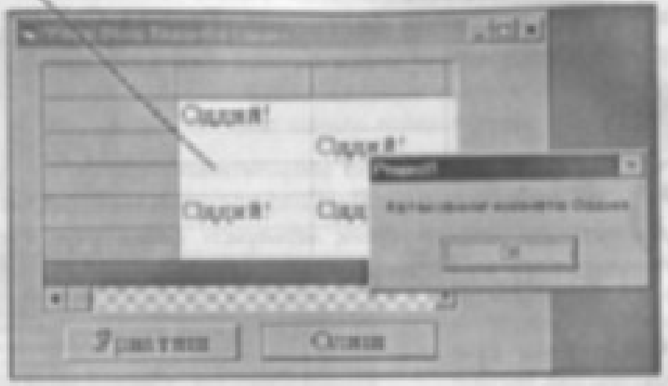


1. Шаклга FlexGrid бошқариш элементини ва иккита тугмани жойлаштиринг. FlexGridни танланг. Rows ҳамда Cols хусусиятлари учун 6 қийматларини ўрнатинг. Кейин FixedCols ва FixedRows хусусиятларини 0 га ўрнатинг. Тугмаларга "Ўрнатиш" ва "Натижа" деб ном беринг. Name хусусиятини мос равишда cmdSet ва cmdget ҳолатига ўтказинг.

2. "Ўрнатиш" тугмасининг Click (Босиш) ҳодисаси учун код FlexGrid объектининг Text (Матн) хусусияти учун оддий қийматни ўрнатади. "Натижа" тугмаси коди панелга Text (Матн) хусусиятининг қийматини чиқаради.



3. Дастурни ишга туширинг. Тўрнинг бирор уясида босинг ва "Ўрнатиш" тугмасини босинг, кейин уяда янги қиймат пайдо бўлиши учун "Натижа" тугмасини босинг.

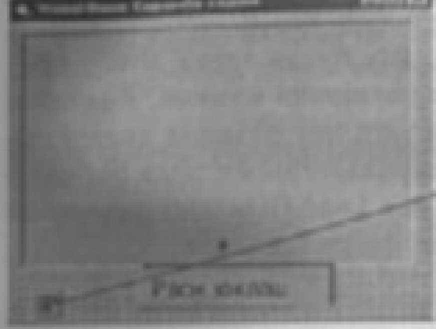


Агар фойдаланувчи бир нечта уя танласа, Cols ва Rows хусусиятлари кўрсаткичи танланган биринчи уяга, Colsel ва Rowsel хусусиятлари эса охирги танланган ячейкага мурожаат қилади.

Умумий диалог ойнаси (Common Dialog)

Windowsда ишловчи дастурларнинг кўпчилигида файлларни дискда сақлаш ёки дискдан юклаш лозим бўлади. Бунинг натижасида дискка жуда катта ҳажмдаги код ёзилиши зарур бўлар эди, лекин бахтимизга Visual Basic Windowsнинг стандарт диалог ойнасидан осон фойдаланиш имкониятини беради. Умуман, бошқариш элементи — умумий диалог ойнаси бошқа бошқариш элементларига ўхшаш: сиз асбоблар панелидан нишонни танланг ва уни шаклга жойлаштиринг. Бу элементнинг бошқалардан фарқи шундаки, иш жараёнида уни шаклда кўриш имконияти бўлмайди, уни шаклга жойлаштиришдан мақсад код ёзиш жараёнида унга мурожат қилиш имконини яратишдир.

Расмларни кўриш дастури



1. Шаклга график ойна, тугмача ва умумий диалог ойнаси — бошқариш элементини жойлаштиринг. Агар асбоблар панелида умумий диалог ойнаси нишони бўлмаса, сичқонча ёрдамида Components [компонентлар] сатрини танлаб, унга ActiveX Microsoft Common Dialog Control бошқариш элементини қўшинг.

2. Load [Юклаш] шакл ҳодисаси учун кодни очинг ва кодга қуйидаги сатрни қўшинг:
Common Dialog1. Filter =
"Pictures" (*.Bmp; .Ico, . Wmf) .*Bmp; *. Ico; *Wmf"
Кейин Click [Босиш] ҳодисаси учун "Расмни юклаш" деб ёзинг. Навбатдаги код:
CommonDialog1.ShowOpen
Set Picture1.Picture = LoadPicture (CommonDialog1.filename)

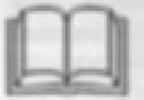
ActiveX элементларини ўрнатиш ҳақида 4-бобдаги "ActiveX бошқариш элементлари" бўлимига қаранг.

Энди сиз дастурни ишга туширишингиз ва унинг ишини текшириб кўришингиз мумкин. "Расмни юклаш" тугмасини босиб, умумий диалогда расмли бирорта файлни юкланг.

Одатда сизга код хусусиятларини ёзишнинг иккита усулидан бирини танлашга тўғри келади: Биринчиси хусусиятлар ойнасида кўрсаткичларни белгилаш, иккинчиси Load (юклаш) ҳодисаси учун код ёзиш. Қайси бирини афзал кўришингизнинг фарқи йўқ, лекин хусусиятларнинг анча узун қийматларини хусусиятлар ойнасида киритгандан ячейкага ёзгандан Load хусусияти кодида ёзган осонроқ.

Умумий диалог ойнаси қой тарзда ишлайди?

Умумий диалог ойнаси бошқариш элементи файлни юклаш, сақлаш, шрифт, ранг танлаш ёки босмага чиқариш ишларини бажариши мумкин. Шу билан бирга, у Windows сўров маълумотномасини ҳам чақириши мумкин. Код ёзиш жараёнида қандай ёзсангиз, шунга мос равишда диалог ойнаси пайдо бўлади. Бу ерда кўриладиган мисолда расмларни кўриш учун Open (Очиш) ёки Load (Юклаш) диалогини чақиришда ShowOpen усулдан фойдаланилади.



ShowOpen усули қой тарзда ишлашини билиб олсангиз, бошқа диалоглар, масалан, Save (Сақлаш) ёки Font (Шрифт) диалог ойналарининг ишини ҳам билиб оласиз.

Диалог ойнаси очилганда Filter (Фильтр) хусусиятида кенгайтмалари кўрсатилган файллargina чиқарилади. Фильтрнинг ижобий томони шундаки, у дастур оча олмайдиган файлларни кўрсатмайди, бу билан ўз навбатида хатоликнинг олди олинади. Фойдаланувчи Open (Очиш) тугмачасини босганда, диалог ойнаси йўқолади, танланган файл номи эса умумий диалог ойнасининг File name (Файл номи) хусусиятига жойлашади. Диалог ойнаси сарлавҳаси DialogTitle (диалог сарлавҳаси хусусияти орқали берилади.

Агар фойдаланувчи Cancel (Бекор қилиш) тугмачасини босса, File name хусусияти бўш ("") қийматга эга бўлади.

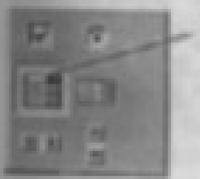
Filter (фильтр) хусусияти бу рўйхатда кўриниши мумкин бўлган кенгайтмаларни белгилайди. Бу хусусият иккита ёки ундан ортиқ қисмлардан иборат бўлиб, улар "|" белгиси билан ажратилади. Биринчи қисм матн бўлиб, у File of type (Файллар тили) ойнасига чиқарилади. Иккинчи қисм файлларнинг рухсат этиладиган кенгайтмаларини санаб ўтади ва улар бир-биридан "нуқтали-вергул" белгиси билан ажратилади.

Расмда кўрсатилган ҳолатда дастур ишчи ҳолда. Агар фойдаланувчи расм бўлмаган файлни танласа дастур автомат тарзда хато юз берганлиги ҳақида хабар беради ва ишни тўхтатади. Бундай хатоларни бартараф қилиш ҳақида маълумот берилган *7-бобга* қаранг.

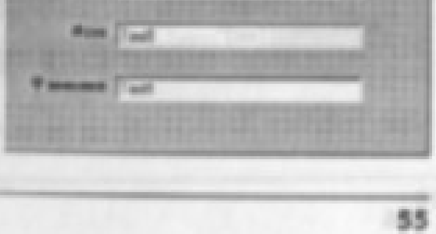
Комбинацияланган ойна (ComboBox)

Комбинацияланган ойна — бу рўйхат ойнаси билан бириктирилган бир сатрли матн ойнасидир. Бу ойнанинг оддий матн ойнасидан афзаллиги шундаки, унга энг кўп ишлатиладиган вариантларни олдиндан киритиб қўйиш мумкин, бунда фойдаланувчи уларни териб ўтиришига ҳожат қолмайди. Агар комбинацияланган ойнани рўйхат ойнаси билан солиштирадиган бўлсак, у ҳолда бу ойна иккинчисига нисбатан камроқ жой эгаллайди. Бундан ташқари комбинацияланган ойна фойдаланувчига ўзининг рўйхатга киритилмаган вариантини ёзиш имконини беради. Бу бўлимда шахснинг тўлиқ исмини унинг маслаги, унвони билан ёзиш мисоли келтирилган. Одатда анкеталарда жаноб, хоним ва шу каби умумий маълумотлар ёзилади. Лекин имкониятлар чегараси бу масалада анча кенг бўлиб, Малика ёки Президент каби кам учрайдиган вариантлар ҳам мавжуд. Бундай ҳолларда комбинацияланган ойна энг қулай ечим ҳисобланади.

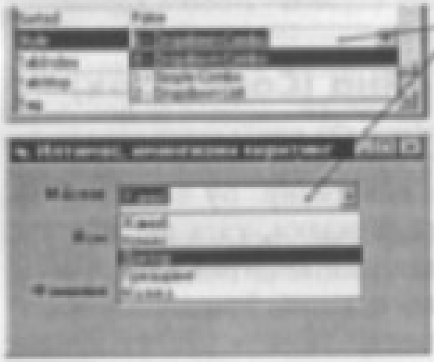
Мисол:



1. Шаклга комбинацияланган ойнани жойлаштиринг. Шаклда кўрсатилган бошқа ёзувлар ва матн ойналари дастурнинг ишига халал бермайди.

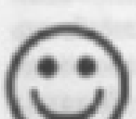


2. Шакл учун Load (Юклаш) ходисаси кодини очинг ва комбинациялашган ойнани тўлдириш кодини ёзинг; Масалан,
Combo1.AddItem ("Жаноб")
Combo1.AddItem ("Хоним")

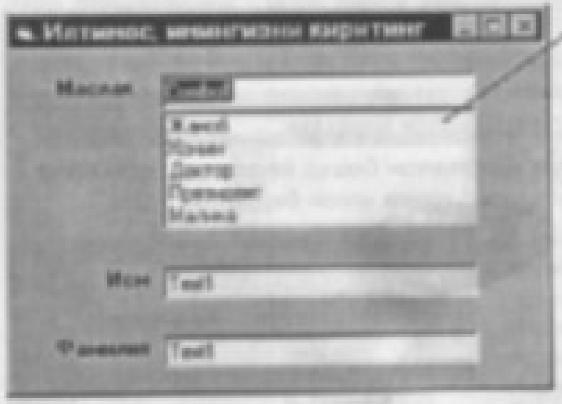


3. Style (Услуб) хусусиятига хар хил қийматлар бериб, уларнинг хар бирида дастур ишини текширинг. Одатда ўрнатилган қиймат сифатида фойдаланиш учун DropDownCombo (Пастга очилувчи рўйхат) қийматидан фойдаланилади. Лекин бу кўрсатилган рўйхатга киритилмаган вариантларни ёзиш имконини бермайди.

Фойдаланувчи танлаган қиймат комбинациялашган ойнанинг Text (Матн) хусусиятига жойлаштирилади.



Агар фойдаланувчи рўйхатга киритилмаган қийматни танлай олмаслигини истасангиз, рўйхат учун DropDown услубини танланг.



4. Simple Combo (комбинациялашган оддий рўйхат) услубидан фойдаланиш учун комбинациялашган ойна баландлигини ошириш зарур.

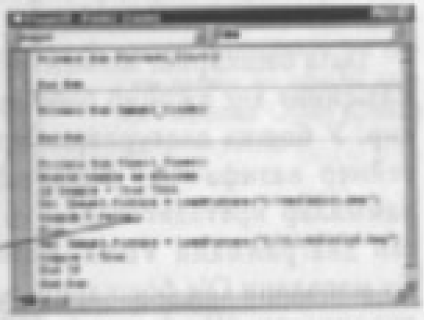
Таймер (Timer)

Таймер — бу умумий диалог ойнасига ўхшаш кўринмайдиган бошқариш элементидир. Таймернинг асосий вазифаси сиз тайинлаган вақт оралиқларида ходисани бошқаришдир.

Капалакни учишга мажбур қилинг!

Қуйидаги мисолда Visual Basic билан бирга қўйиладиган (бериладиган) BFLY1.BMP ва BFLY2.BMP график файллардан фойдаланилади. Агар сиз дастур мисолларини ўрнатган бўлсангиз, юқорида қайд қилинган график файллар сизнинг Visual Basic каталогингизнинг samples /pguide/vcr папкасига жойлаштирилган.

1. Шаклга таймер ва бошқарувчи элемент — расмни жойлаштиринг. Таймернинг Interval (Оралиқ) хусусиятига 200 (миллисекунд) қийматини беринг. Шакл асосига оқ ранг беринг.



2. Таймернинг ходисаси кодини очиш учун сичқонча ёрдамида уни икки марта босинг. Бу код хар 200 миллисекунд оралиғида бошқарилади. BFLY1. BMP ва BFLY2. BMP график файлларнинг, LoadPicture функциясида ўзгартиришлар киритиб, ходисага қуйидаги кодни киритинг.

```
Static toggle As Boolean
If toggle = True Then
Set Image1.Picture=LoadPicture("c:\vb5\bfly1.bmp")
toggle=False
Else
Set Image1. Picture=LoadPicture ("c:\vb5\bfly2.bmp")
toggle=True
End If
```

3. Дастурни ишга туширинг ва капалакнинг қанот қоқишини томоша қилинг!

Сиз иловонинг иш вақтида таймерининг Interval хусусиятини ўзгартиришингиз мумкин. Мазкур ҳолда сиз қалқонни тезроқ ёки секинроқ учириш опциясини ишга туширингиз мумкин.

Бу ерда келтирилган график файллар йўлларини уларнинг системангиздаги ҳақиқий йўллари билан алмаштиришни унутманг. Агар сиз бу график файлларни топа олмасангиз, Visual Basicнинг ўрнатиш (Setup) дастурини ишга туширинг ва иловалар (samples) дастурини ўрнатинг.

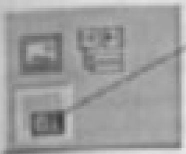
OLE бошқариш элементи

Шу пайтгача сиз Visual Basic асбоблар стандарт панелининг Data (маълумотлар) ва Ole (Ole — Object Linking and Embedding — боғлаш ва объектларни жорий қилиш)дан ташқари барча бошқариш элементлари билан танишиб чиқдингиз.

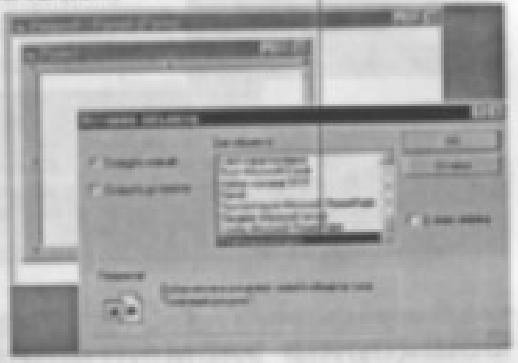
Data бошқариш элементи 5- бобда қараб чиқилади. Visual Basicнинг энг ажойиб бошқариш элементларидан бири OLE дир. У бошқа дастурларда яратилган ҳужжатлар учун контейнер вазифасини бажаради. Фараз қилинг, сизда диаграммалар яратадиган дастур бор, сиз бу дастурда яратилган диаграммани Visual Basic да акс эттиришингиз зарур. Бу масалани Ole бошқариш элементи ёрдамида бажаришингиз мумкин. Шу билан бирга шу диаграммага Ole ёрдамида ўзгартириш киритиш ҳам мумкин.

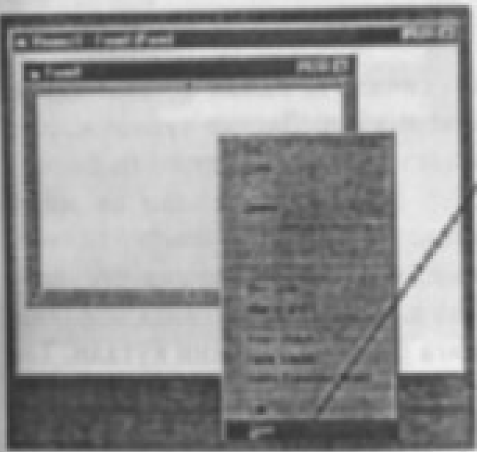
Ole бошқариш элементи ишлаши учун сиз дастур яратиш давомида фойдаланувчи компьютерига ўрнатилган дастурлардан фойдаланишингиз зарур. Шунингдек, бу дастурлар тўла бир-бирига мос келиши керак. Кўпчилик иловалар Ole билан нотўғри ишлайди.

Мисол



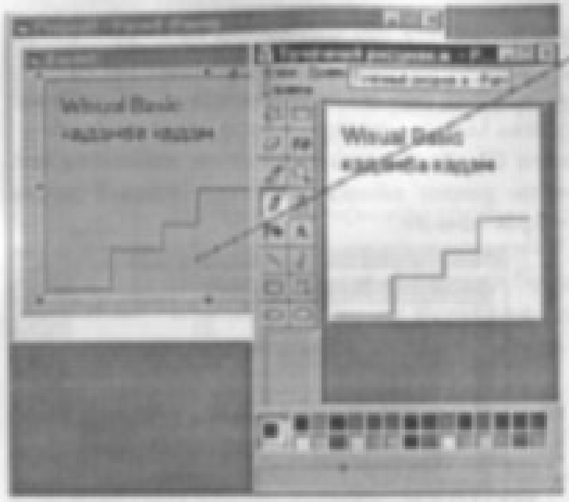
1. Янги Standard Exe дастурини яратинг ва шаклнинг NegotiateMenus хусусиятини False қийматига ўрнатинг. Шаклга Ole бошқариш элементини жойлаштиринг. Очилган диалог ойнасида Bitmap Image (Растрли тасвир)ни танланг.





2. Ole нинг тасвир қилишини ўзгариш учун шаклнинг бўш жойига босинг. Кейин бошқариш элементини ўнг бошқариш билан қайта чақириб чиққан менюда Open (Очиш) сатрини танланг.

Ole бошқариш элементи билан ишлашда аслида нозик ва ҳал қилиниши қийин бўлган муаммолар юзага келиши мумкин. Агар сиз Ole бошқариш элементини Word ёки Excel каби катта дастурлар билан ишлатмоқчи бўлсангиз, дастурингиз ишини секинлашиши мумкин. Ole элементидан жуда зарур ҳолатлардагина фойдаланиш мақсадга мувофиқ.



3. Растрли тасвир муҳаррир режимида очилади. Расм асоси ва Windows Paint график муҳарририни ёпиб, уни Олдо пойдо бўлишини кузатинг. Дастур иши давомида фойдаланувчи расмни таҳрир қилиш учун бошқариш элементини неча марта босиши мумкин.

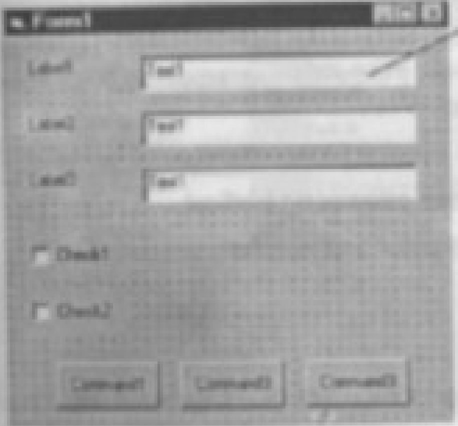
Бошқариш элементларини улаш тартиби

Шуни тан олиш керакки, сичқонча билан ишлаш ҳамма фойдаланувчиларга ҳам ёқавермайди. Дастур тузиш жараёнида, имконият бор жойда дастурни клавиатурадан бошқариладиган қилиш зарур. Бир йўли — тугмалар ва меню сатрлари учун клавишларнинг эквивалент комбинациясидир. Бошқа йўли шуки, фойдаланувчилар бошқа стандарт дастурларда бўлгани каби Tab клавишини босганда бир бошқариш элементидан бошқасига фокус кўчишини кутади. Tab клавиши ёрдамида фокус кўчиши маълум мантиқий кетма-кетликда бўлиши зарур. Бу масалани ҳал қилиш учун бошқариш элементларига TabIndex хусусияти қийматларини бериш зарур.

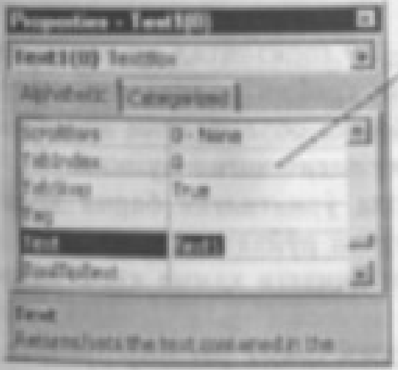
Windows кобиғида ишлайдиган дастурларда "фокус" сўзи актив ойна ёки бошқариш элементини кўрсатиш учун ишлатилади. Мисол учун, агар тугма фокусга эга бўлса, Enter ёки Space клавишларининг босилиши сичқонча тугмасининг босилишига эквивалентдир.

Бошқариш элементларининг фокусни олиш тартибини ўрнатиш жараёни аниқ бўлмаслиги мумкин. Агар сиз бошқариш элементларидан бирортасида TabIndex хусусияти қийматини ўзгартирсангиз, у ҳолда Visual Basic TabIndex хусусиятининг қийматларини бошқа бошқариш элементлари учун ҳам автоматик ўзгартиради. Бунда бошқариш элементларининг фокусни олиш тартиби ўзгармайди, фақат сиз ўзгартирган бошқариш элементидагина фокус тартиби ўзгаради.

Фокусни олиш тартибини қандай белгилаш керак

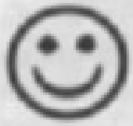


1. Шаклга бир неча бошқариш элементини жойлаштиринг ва қайси элемент биринчи навбатда фокус олиши зарур бўлса, шу элементни белгиланг.



2. Хусусиятлар ойнасида TabIndexга 0 қийматини беринг.

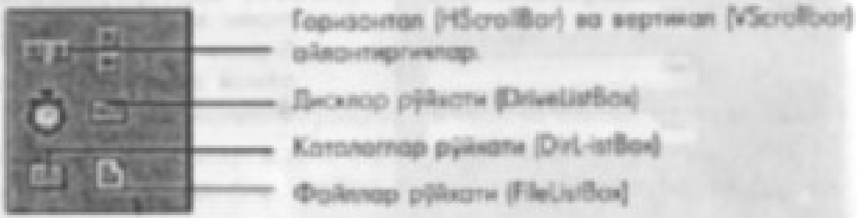
3. TabIndex хусусиятига кетма-кет 1, 2, 3 ва ҳ.к қийматлар бериб, барча бошқариш элементлари учун юқоридаги қадамни қайтаринг



Бошқариш элементларининг ҳаммаси ҳам фокус ола-вермайди. Мисол учун, ёзув элементи фокус олмайди, сиз уни Tab клавиши ёрдамида фаоллаштира олмайсиз. Агар сиз уни фаоллаштиришга ҳаракат қилсангиз, у ҳолда автоматик тарзда тартиб бўйича кейинги бошқариш элементига ўтиб кетасиз.

Бошқа стандарт бошқариш элементлари

Visual Basicнинг стандарт бошқариш панелида бошқа яна бир нечта объектлар бор. Бу объектлар унчалик фойдали эмаслиги сабабли улар ҳақида шу бобнинг охирида энди маълумот берамиз.



Бошқариш элементлари ҳақида батафсилроқ маълумот

HScrollBar ва VScrollBar — бу горизонтал ва вертикал айлантиргичлардир. Сиз уларни график индикатор, юритгич (регулятор) созлагич ёки бошқа объектларни айлантиришни қўлда бошқариш учун ишлатишингиз мумкин.

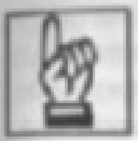
DriveListBox — тизимингизда ўрнатилган барча дискларни кўрсатадиган тушиб қолувчи рўйхат

DirListBox — жорий каталог ичидаги ҳамма қисм каталоглар рўйхати.

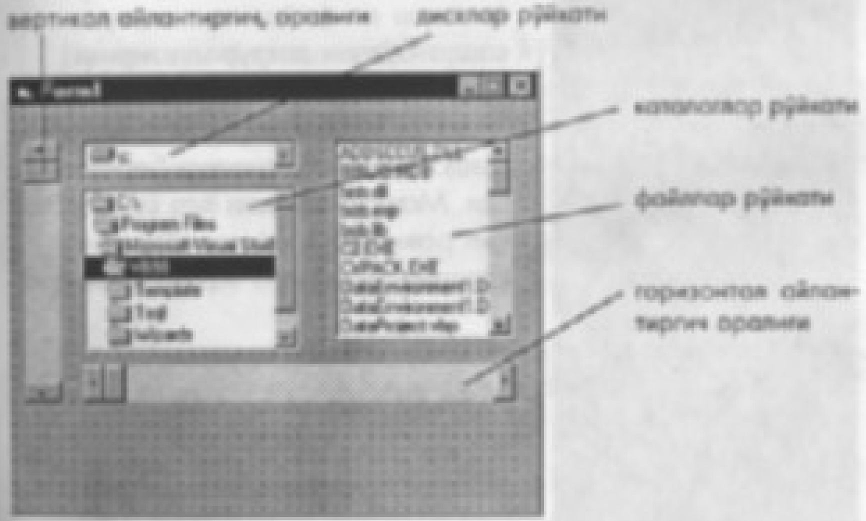
FileListBox — жорий каталогдаги ҳамма файллар рўйхати.

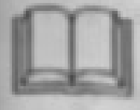
Айлантиргич оралиқлари — бошқариш элементлари кам ишлатилади, чунки кўпчилик объектлар ўз ўрнатилган айлантиргич оралиқларига эга бўлади.

FileListBox ва DirListBox бошқариш элементлари ҳам кам ишлатилади, чунки умумий диалог буларнинг ҳаммасини яхшироқ бажаради.



Объектга айлантиргичларни жойлаштиришдан олдин, объектда ўзининг айлантириш оралиқлари ўрнатилмаганлигига ишонч ҳосил қилинг.





Баъзи объектлар ўрнатилиши бўйича оралиқлари билан пайдо бўлади. Ўрнатилган айлантириш оралиқлари билан ишлаш алоҳида айлантириш оралиқлари билан ишлашга қараганда анча осон.

УЧИНЧИ БОБ

Basic асослари

Visual Basic ишнинг асосий қисмини визуал (кўринувчи) воситалар ёрдамида бажариш имконини берса-да, кичик визуал воситаларни бошқариш кодини ёзишдан воз кечишга имкон бермайди. Бу код ўрганиш ва фойдаланиш учун содда бўлган дастурлаштириш тили Basicда ёзилади.
1- ва 2- бобларда Basic тилидаги коддан кичик парчалар берилган эди. Мазкур бобдан эса сиз мустақил равишда дастурни ёзишни ўрганиш имконига эга бўласиз.

Бошидан бошлаймиз

Дастур (программа) — компьютер бажариши зарур бўлган кетма-кетликда ёзилган йўриқлар кетма-кетлигидир. Дастурлар мураккаб бўлиши мумкин бўлса-да, лекин ҳар бир алоҳида йўриқ соддадир. Компьютер дастурнинг биринчи йўриғидан бошлаб, берилган кетма-кетликда охиргисигача бажаради. Қуйида Visual Basicнинг баъзи муҳим элементлари келтирилган.

Операторлар

Бу маълум бир ишни бажариш йўриғидир. Мисол учун, Beep оператори компьютер овоз карнайида (динамик) овоз сигналини чақиради, FileCopy оператори эса бир диск файлини иккинчисига кўчиради. Операторларни буйруқлар (командалар) деб ҳам аталади.

Функциялар

Бу қийматларни қайтарувчи йўриқлардир. Мисол учун, Now функцияси жорий сана ва вақтни қайтаради.

Ўзгарувчилар

Бу қийматларни сақловчи сўзлардир. Масалан,

myvar = "Visual Basic"

сатри ўзгарувчилар сатрини myvar номи билан сақлайди.

Операциялар (амалар)

Операциялар-бу оддий "+", "−" ва "=" каби стандарт арифметик амаллардир. Оддий кўпайтириш амалини ҳарф билан чалкаштирмаслик учун "*" юлдузча белгисидан фойдаланилади. Бўлиш операцияси эса "/" белги, чизиқ билан ифодаланади.

Объектлар

Объектлар — кўринадиган, мисол учун шакл ёки тугма ёки Windows алмашиш чўнтаги (буфер) га ўхшаш кўринмайдиган бўлиши мумкин.

Хусусиятлар

Хусусиятлар — бу объектнинг тавсифларидир.

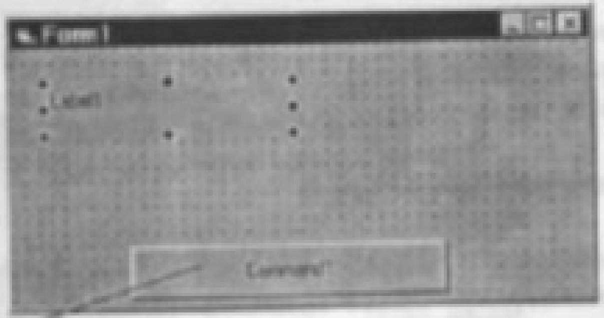
Усуллар

Объект бажариши мумкин бўлган харакатлар усуллар дейилади. Мисол учун, шакллар Hide усулига эга бўлиб, у шаклни кўринмас қилади.

Дастур тузишни энг яхшиси амалиётда ўрганиш лозим. Visual Basic — энг хавфсиз дастур тузиш муҳитидир, сиз бу муҳитда ҳар қандай дастур тузишингиз ва ўз хатоликларингизни амалда тузатишингиз мумкин.

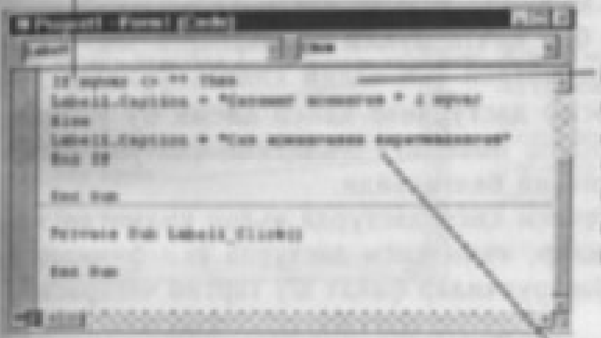
Дастур намунаси

Оддий дастурдан намуна келтирамиз ва унинг таркибий қисмларини муфассал кўриб чиқамиз.



1. Шаклга ёзув ва тугма жойлаштиринг. Кейин Click (босиш) ходисаси учун код муҳарририни очиш мақсадида тугмани икки марта босинг.

Visual Basic коднинг биринчи ва охирги сатрлари бу қисқа тартибнинг бошланиши ва тугаш қолатларини кўрсатади. Sub сўзи асосий сўз бўлиб, Subroutine (қисм дастур) сўзининг қисқартирилган шаклидир. Private (ёпиқ) асосий сўзи мазкур тартиб қанда фақат у ўзи тегишли бўлган шакл томонидангина чақирилиши мумкинлигини англатади.



Мазкур шарт бўлмаган сатр Visual Basicга матн сатрини саклаш учун тувар номли ўзгарувчидан фойдаланмоқчилигингизни хабар қилади.

Бу сатр ўнгдан чапга қараб бажарилади. Олдинига Visual Basic InputBox функциясини чақиради ва у фойдаланувчидан бир сатрни ёзишни сўрайди. Кейин эса натижавий сатр тувар ўзгарувчисида сақланади.

Dim асосий сўзи "Эълон қилиш" маъносини англатади. Visual Basicда илғовчи малакали дастурчилар барча ўзгарувчиларни Dim, Public (Очиқ) ёки Private (ёпиқ) каби асосий сўзлар билан белгилайдилар. Ўзгарувчилар номини ёзишда орфографик хатоликка йўл қўймаслик учун Tools (Сервис) менюсидан Options (Параметрлар) сатрини ва Editor (Муҳаррир) қисмида Require Variable Declaration (Ўзгарувчиларнинг тўлиқ номи) сатри байроқчасини белгиланг. Сиз бу байроқчани ёқиш билан ҳам хатоликларни камайтирасиз, ҳам дастур ишини тезлаштирган бўласиз.

Ниҳоят, Visual Basic тувар сатрини бўш сатр билан солиштиради. Агар киритилган сатр бўш бўлмаса, унинг қиймати Caption хусусиятига тақдим этилади, акс ҳолда эса "Сиз исмингизни киритмадингиз" хабари пайдо бўлади.

Ўзгарувчилар ва кўриниш соҳаси

Олдинги бўлимдаги мисолда MyVar ўзгарувчиси белгилар сатрини сақлашда фойдаланилган эди. MyVarнинг ўзгарувчи дейилишига сабаб шуки, унинг мазмуни дастур иши чоғида ўзгариши мумкин.

Ўзгарувчилар бир нечта муҳим тавсифларга эга. Улардан бири кўриниш соҳаси ёки оддий қилиб кўриниш ҳам дейилади. Бу тавсиф дастурнинг қайси қисми бу ўзгарувчига мурожаат қилиш имконига эгалигини ёки уни ўзгартириши мумкинлигини белгилайди.

MyVar ўзгарувчиси қисм дастурда эълон қилинган эди. Бундай ўзгарувчилар, яъни қисм дастурда ёки функцияда эълон қилинган ўзгарувчилар фақат шу тартиб чегарасидагина кўринади.

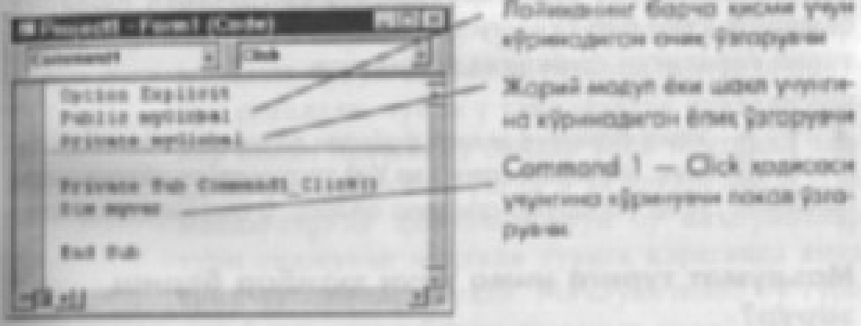
Сизнинг ўзгарувчиларингиз қандай кўриниш соҳасига эга бўлиши зарур? Иложи борича тор киритиш соҳасига эга бўлиши лозим. Кўп миқдорда очиқ ёки глобал ўзгарувчиларнинг бўлиши қулай туюлади, лекин бу ҳолда дастурда хатолик пайдо бўлиш эҳтимоллиги ортади.

Ўзгарувчиларни шаклнинг ёки код модулининг Declarations (Эълонлар) бўлимида ҳам кўрсатиш мумкин. Агар сиз уларни Dim ва Private (Ёпиқ) асосий сўзлари билан белгиласангиз, у ҳолда шу шакл ёки код модулининг исталган тартиби бу ўзгарувчиларга мурожаат қила олади.

Ниҳоят, ўзгарувчилар Declarations (Эълонлар) бўлимида Public (Очиқ) асосий сўзи билан ҳам эълон қилиниши мумкин. Бу ўзгарувчилар лойиҳангизнинг исталган қисмидан чақирилиши мумкин ва уларни глобал ўзгарувчилар деб аталади.

Агар сиз Require Variable (Ўзгарувчиларнинг аниқ тавсифи) байроқчаси белгисини ўрнатмаган бўлсангиз, у ҳолда ўзгарувчига унинг кўриниш соҳаси ташқарисида мурожаат қилганингизда Visual Basic хатолик ҳақида хабар қилмайди. Бунинг ўрнига у янги бўш ўзгарувчини

яратади, бу эса кўпгина муаммолар туғдириши мумкин, шунинг учун Require Variable Declaration (Ўзгарувчиларнинг аниқ тавсифи) байроқчаси доим ўрнатилган бўлишини кузатиб боринг.



Маълумотлар турлари билан танишамиз

Ўзгарувчиларнинг кўриниш соҳаси билан бир қаторда яна муҳим тавсифларидан бири унинг тури ҳисобланади. У ўзгарувчи қайси турдаги ахборотни (маълумотларни) сақлай олишини белгилайди.

Мисол учун, сатр ўзгарувчиси сатрга ёзилган белгиларни сақлайди. (қўштирноқ ичига ёзилган ҳамма нарса, масалан, "123" ҳам сатр ҳисобланади) Бу ўзгарувчилар номлар ва адреслар учун қулай, лекин математик ҳисоблашларда улардан фойдаланилмайди. Агар сонлар устида амаллар бажариш ва уларни сақлаш зарур бўлса, бутун кўрсаткичли (Integer) ўзгарувчилардан фойдаланиш зарур.

Агар маълумот тури аҳамиятга эга бўлмаса, у ҳолда variant деб аталувчи умумий вазифали ўзгарувчидан фойдаланишингиз мумкин. variant туридаги ўзгарувчиларда исталган турдаги маълумот сақланади. Visual Basic да ишлашни осонлаштириш учун барча ўзгарувчилар келишувга биноан variant турига эга. Бошқа турдаги ўзгарувчиларни кўрсатиш учун As асосий сўзидан фойдаланиш зарур. Мисол учун:

Dim MyVar As String

(Сатр ўзгарувчисини эълон қилади).

Масалан, агар сатрни сонга кўпайтиришни бу ишни қилиб бўлмайдиган ишларни ҳатто variant ўзгарувчилари ҳам бажаришга имкон бермайди. Сиз бу ишга уриниб кўрсангиз, Visual Basic хатолик ҳақида "Type mismatch"(Турлар номутаносиблиги) деган ахборот беради. Бундай ишларни бажариш учун бир турдаги маълумотни иккинчисига ўзгартириб берадиган функциялар мавжуд.

Visual Basicнинг мана бу фойдали вазифасига эътибор беринг: Str, (сонни сатрга ва Val) сатрни сонга айлантиради. Агар Val сатрда сон топа олмаса, 0 сонини қайтаради.

Маълумот турига нима учун эътибор бериш зарур?

Агар сиз дастур тузишни variant туридаги ўзгарувчилардан фойдаланиб бошласангиз ҳам, Visual Basic ни ўзлаштириш жараёнида маълумотларнинг аниқ турларини қўллашишга ҳаракат қилишингиз керак. Бу сизга хатолик қилмасликка ёрдам беради ва компьютер ишининг самарадорлигини оширади. Мисол учун, компьютер аниқ ўзгарувчи турини олдиндан билса, уни сақлаш учун хотирадан зарур миқдорда жойни ажратади. Демак, компьютер ўз хотирасидан унумлироқ фойдалана олади.

Маълумотлар турларини ўрганамиз

Visual Basic variant туридан бошқа яна ўн битта маълумот турига эга.

String — 2 миллиардгача белгилар бўлган белгилар сатри. Шу билан бирга сиз ўрнатилган узунликдаги сатрларни ҳам белгилай оласиз.

Integer — 32768 дан 32768 гача бўлган бутун сонлар.

Long — 2147483648 дан 2147483648 гача бўлган бутун сонлар.

Single — силжувчи нуқтага эга бўлган, тахминан 3.4 E38 гача бўлган мусбат ёки манфий сонларни сақ-

ловч...
ража...
Double силж...
маълу...
мусба...
Currency маълу...
масал...
тун де...
лар би...
да 15 ...
маблағ...
тури с...
аниқ ...
пул би...
Date санал...
(Janua...
31, 99...
лайди...
Boolean энг со...
кин бў...
Byte маълу...
мусбат...
Object маълу...
ларин...
лари...

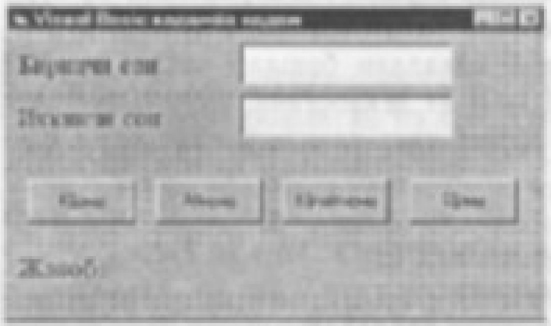
Юқоридагилар...
мот тури мавжуд...
маълумот тури ва...
бу бир ўзгарувчи...
лар гуруҳидир. М...
ҳокама қилинади...

Агар си...
бу ёрдо...
палоти т...
иккилик...
нинг д...

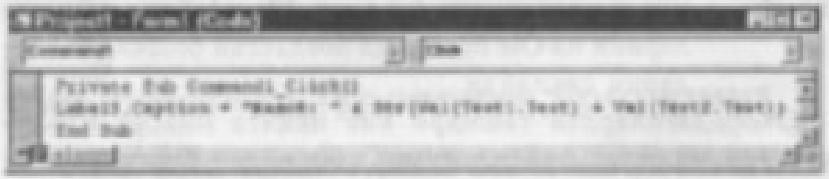
Юқорида санаб ўтилган маълумотлар турларини билгандан кейин, табиийки, бундай савол туғилади, қандай ҳолларда қайси турдаги маълумотдан фойдаланиш зарур. Компьютер нуқтаи назаридан қарайдиган бўлсак, энг қисқаларидан фойдаланган маъқул.

Дастур — калькулятор

Навбатдаги мисол дастур — калькулятордир. Дастурнинг мазмуни шундан иборатки, сиз иккита сон киритсангиз тегишли тугмани босганингизда дастур сонларни мос равишда қўшади, айиради, бўлади ёки кўпайтиради.



1. Янги дастурни тоза шакддан бошланг. Расмда кўрсатилган тартибда унга ёзув, матн ойнаси ва тугмаларни жойлаштиринг



2. Click (Босиш) ходисаси кодини очиш учун "Қўшиш" тугмасини икки марта босинг. Қуйидаги код сатрини ёзинг (бир сатрга):
Label 3. Caption = "Жавоб:"+
Str (Val (Text1. Text)+Val (Text2. Text))
3. Бошқа учта тугма учун ҳам иккинчи белги "+"ни қуйидаги белгилар билан алмаштириб, шунга ўхшаш операторларни ёзинг.
"Айириш" тугмаси учун -
"Бўлиш" тугмаси учун /
"Кўпайтириш" тугмаси учун *
4. Дастурни ишга туширинг ва калькуляторни иш вақтида текшириб кўринг.

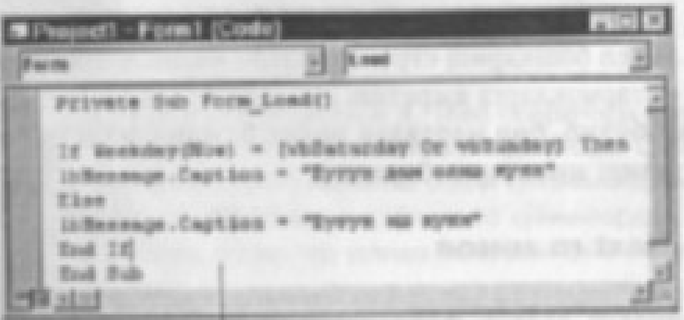
"+" ва "*" белгилари ҳисоблашларда қўлланилади ва операциялар деб аталади. Бу мисоллар келтирилган тўртта операция энг умумий ҳисобланади. Баъзи ҳолларда матннинг иккита сатрини бирлаштириш учун + операцияси ўрнига & операциясидан фойдаланилади.

IF... Then... Else

If оператори дастурга интеллект элементини киритиш имконини беради. Бу оператор сиз берган шартли ифоданинг қийматига боғлиқ ҳолда у ёки бу блокни танлаб, коднинг икки блокидан бирини бажаради.

Уни ишда текширамиз

1. Шаклга ёзув жойлаштиринг ва уни lblMessage деб номланг.



2. Load (Юклаш) ходисаси учун код муҳарририни очиш мақсадида шаклни икки марта босинг. Бу код дастур ишга тушиш чоғида бажарилади.

Код муҳарририни зарур жойда очиш учун сиз ёзувни эмас, балки шаклни икки марта босишингиз зарур.

3. Куйидаги кодни ёзинг ва дастурни ишга туширинг:

```
If Weekday (Now)=(vbSaturday Or vbSunday) Then
lblMessage. Caption = "Бугун дам олиш куни"
Else
lblMessage. Caption = "Бугун иш куни"
End If
```

Юкорида келтирилган код Visual Basic нинг баъзи функцияларини ишлатади. Now функцияси жорий сана ва вактни кайтаради. WeekDay (хафта куни) функцияси уларни хафта кунини ифодаловчи сонга айлантиради. Visual Basicнинг vbSaturday каби константалари хафтанинг хар бир кунига мос келувчи сонларни эсда саклаб колиш зарурлигидан сизни озод килади.

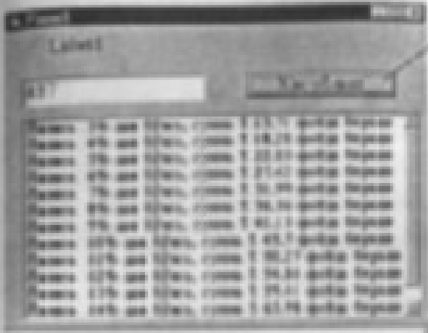
For.... Next

If... Then бошкариш структуралари кодни ташкилий кисмларга — тармокларга ажратиш имконини беради. For...Next — бу цикл бўлиб, бир мартадан ортик бажариладиган код блокини тузиш имконини беради.

For... Next га мисол

Куйидаги дастур фойдаланувчи томонидан киритилган пул маблаги учун йиллик даромадни хисоблайди.

For ... Next цикллари ўзгарувчи-хисоблагичга эга бўлиб, унда бажарилган цикллар сони сакланади. Бу ерда келтирилган мисолда у iCounter деб номланади. Бу ўзгарувчига цикл ичида хеч кандай киймат берманг, акс холда у ишламайди.



1. Шаклга ёзув, матн ойнаси ва рўйхат ойнасини расмда кўрсатилгандек килиб киритинг

2. Тугмани икки марта босинг ва Click (Босиш) ходисаси учун куйидаги кодни ёзинг. Дастурни ишга туширинг, матн ойнасига сон ёзинг ва "Хисоблаш" тугмасини босинг

```
Dim curAmount As Currency
Dim iCoutvar As Integer
List 1. Clear
curAmount = Val (Text1. Text)
If curAmount <> 0 Then
For iCoutvar= 3 To 15
List 1. AddItem ("да" + Str (iCoutvar)
+ " % йиллик фоиз $ даромад келтиради" -
+ Str /curAmount * iCoutvar/ 100)
Next iCoutvar
Else
MsgBox ("Сон нокиний эмас")
End If
```

Бу код For...Next циклини If...Then операторлари ичига кандай килиб кўйиш мумкинлигини кўрсатади. Операторларни ичига кўйиб ишлатиш баъзи кулайликлар беради ни билан, сиз жуда хам кўп ичига кўйишлардан фойдалансангиз, бу дастур устида ишлашни кийинлаштириши мумкин.

Do ... Loop

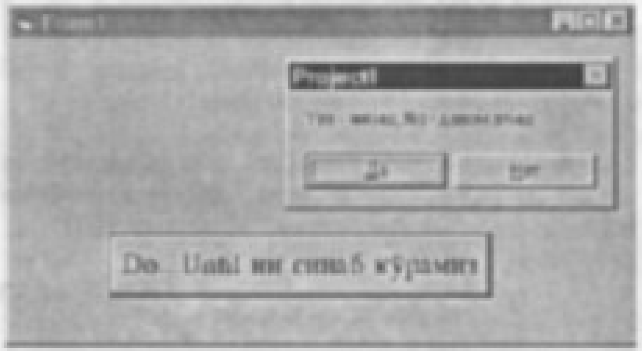
Хамма цикллар хам For....Next дан фойдаланавермайди. Бошка яна иккита имконият мавжуд: Do While...Loop ва Do Until... Loop. Do While... Loop циклни берилган шартли ифода False кийматга эга бўлганда, Do Until...Loop эса берилган шартли ифода True кийматга эга бўлганда бажаради.

Мисол:

1. Шаклга тугмани жойлаштиринг.
2. Тугмани икки марта босинг ва Click (Босиш) ходисаси учун қуйидаги кодни киритинг:

```
Dim answer As Integer
Do Until answer = vbYes
answer=MsgBox("Yes — Чиқиш, No — давом эттириш", vbYesNo)
Loop
```

3. *Дастурни ишга туширинг ва тугмани босинг. Диалог панели пайдо бўлади.* Агар сиз Yes тугмасини боссангиз, диалог йўқолади ва цикл тугайди, чунки answer = vbYes шарти бажарилади. Агар сиз No тугмасини боссангиз, диалог йўқолади ва дарҳол яна пайдо бўлади.



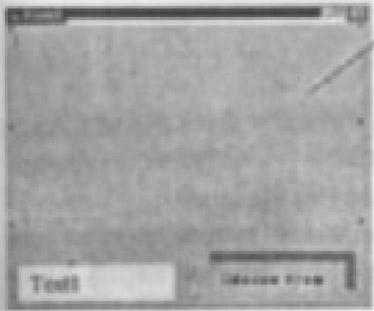
Шу билан бирга сиз шартли ифодани Do... Loop Until кўринишида цикл охирига қўйишингиз мумкин. Бу ҳолда цикл камида бир марта бажарилади.

Do циклининг битта тури бўлса-ю, нега унинг икки тури керак? деган савол туғилиши табиий албатта. Ўзингизга маълумки, While шартини, унинг олдига Not калитсўзини қўйиб Until шартига алмаштириш мумкин. Бунинг асосий сабаби шуки, коднинг тушунарли бўлишига риоя қилиш зарур. Сиз ёки сиздан кейин дастурдан фойдаланувчи киши код ишини яхшиламоқчи ёки хатоликларини бартараф қилмоқчи бўлса, коднинг тушунарли тузилганлиги унинг қандай ишлашини тезроқ англаб олишга имкон беради.

Case ... Else

If...Then оператори кодда тармоқлар хосил қилиш учун қулай, лекин жуда кўп сондаги мумкин бўлган тармоқлар яратиш керак бўлганда, у ноқулай бўлиши мумкин. Бундай холларда Visual Basic Select Case ... End Select тузилмасига эга бўлиб, бу назорат ўзгарувчисининг қийматига боғлиқ равишда турлича кодни бажаришга имкон беради.

Select Case тузилмасига мисол



1. Шаклга расмда кўрсатилган тарзда нишонга ёзув, матн ойнаси ва тугмани жойлаштиринг.
2. Тугмани икки марта босиб, код муҳарририни очинг ва қуйидаги кодни ёзинг:

```
Dim sColour As String
sColour = UCase (Trim(Text1.Text))
Select Case sColour
Case Is ="Қизил"
Label1. BackColor=vbRed
Case Is ="Кўк"
Label1.BackColor = vbBlue
' ва ҳоказо ... Агар зарур бўлса,
' бошқа рангларни қўшинг.
Case Else
Label1.BackColor = vbWhite
Label1.Caption = "Номаълум ранг"
End Select
```

3. Дастурни ишга туширинг ва матн ойнасига ранг номини ёзиб, "Ойнани бўяш" тугмасини босинг.

Бу мисолда апостроф ['] белгисидан бошланувчи сатр борлигига эътибор беринг. Бундай сатр шарҳ сатри ҳисобланади ва дастурни бажариш вақтида ҳеч қандай иш бажармайди. Шарҳнинг асосий вазифаси кодни ўқишда дастурни янада тушунарлироқ қилишдир.

UCase ва Trim функциялари фойдаланувчи бош ва кичик ҳарфлар тартибини аралаштириб юборганда (UCase барча ҳарфларни бош ҳарфга айлантиради) ва киритилган сўздан олдин ва кейин оралиқ қолдирганда фойдалидир (Trim функцияси бу оралиқларни олиб ташлайди.)

Vizual Basicда ранг танлашни кулайроқ усули мавжуд. Бунинг учун ShowColor (рангларни кўрсатиш) усулида умумий диалог бошқариш элементидан фойдаланиш керак.

With ... End With

With...End With блоки бошқа турдаги блоклардан фарқ қилади. Бу блокнинг ягона вазифаси — коднинг ёзилишини ва ундан фойдаланишни соддалаштиришдир.

Vizual Basicнинг истаган объектига қўлланиладиган операторлар кетма-кетлигининг ёзилиш усулини қараб чиқамиз. Ушбу

```
object. thisproperty ="Something"
object. thatproperty ="Something else"
```

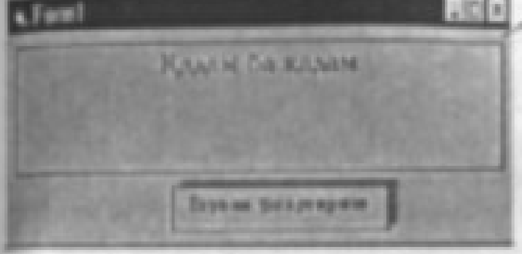
коднинг ўрнига қуйидагини ёзишингиз мумкин:

```
With object
        .thisproperty = "Something"
        .thatproperty = "Something else"
End With
```

With блокига, масалан, Go To операторидан фойдаланиб, кирманг ва ундан чиқманг. Акс ҳолда дастур хатоликка йўл қўйилганлиги ҳақида хабар беради.

Код ёзиш жараёнида бундай қисқартириш номи узун объектлар билан ишлашда қулайлик яратади. Бундай ҳолатлар OLE механизми орқали бошқа дастурларни бошқарувчи дастурларда кўпроқ учрайди.

With...End With операторига мисол



1. Ёзув ва тугмани шаклга жойлаштиринг.

2. Тугмани икки марта босинг ва Click (Босиш) ҳодисасида қуйидаги кодни ёзинг:

```
With Label1
 .Caption = "Қадам ба қадам"
 .Alignment = VbCenter
 .BackColor = VbYellow
 .Font = "Arial"
 .Font Size = 24
 .Font Bold = True
End With
```

With блокларидан устма-уст фойдаланишингиз мумкин. Бундай тузилма ишлай олиши учун ҳар бир With операторига жавоб берувчи End With қисмига эга бўлиши керак.

Массивлар

Бу пайтга келиб сиз дастурнинг бажарилиш жараёнида қийматларни сақлашга имкон берувчи ўзгарувчилардан фойдаланишни ўрганиб олишингиз керак эди.

Амалий иш жараёнида кўпинча бир жинсли ўзгарувчилар билан ишлашга тўғри келади. Мисол учун, сиз ойлик савдо ҳажмларини билган ҳолда йиллик савдо ҳажмини ҳисоблашингиз зарур бўлсин JanSales, FebSales ва ҳ.к. деб номлаб, ўн иккита ўзгарувчи билан ишлаш ўрнига ўн иккита элементдан иборат массив билан ишлаш анча қулай бўлар эди. У ҳолда сиз ҳар ойлик савдо ҳажмига қуйидаги тарзда мурожаат қилишингиз мумкин эди: MonthlySales (0), MonthlySales (1) ва ҳ.к. Мисол келтирамиз.

```
Dim MonthlySales (12) As Currency
Dim TotalSales As Currency
Dim iCountVar As Integer
MonthlySales (0) = 5214. 45
MonthlySales (1) = 4576. 76
MonthlySales (2) = 7142.32
```

ва ҳ.к.

Массив эълон қилишингиз билан массивда йўқ элементга мурожаат қилинмаслигига эътибор беринг. Агар сиз юқорида келтирилган кодда MonthlySales (20) массив элементига йўлланма берсангиз, Visual Basic хатолик ҳақида хабар беради: "Subscript out of range" ("Индекс берилган оралиқдан чиқди").

Массивларнинг афзаллик томони шундаки, уларни цикл ичида қайта ишлаш мумкин. Масалан, қуйида келтирилган код (тартиб) йиллик савдо ҳажмини ҳисоблайди:

```
Dim TotalSales As Currency
Dim iCountVar As Integer
For iCountVar = 0 To 11
Total Sales = TotalSales + MonthlySales (iCountVar)
Next
Label 1. Caption = "Йиллик савдо ҳажми: "+"
                    Str (TotalSales)
```

Кўп ўлчамли массивлар

Массивлар ўлчами бирдан катта бўлиши мумкин. Мисол учун сиз иккита савдо дўкони маълумотларини битта массивда сақлашингиз мумкин:

```
Dim MonthlySales (12,2) as Currehcy
MonthlySales [0,0] = 5621.34' биринчи дўкон
MonthlySales [0,1] = 5214.45' иккинчи дўкон
```

Basic да массивлар элементлари учун ўрнатилган индекс 0 дан бошланади, яъни 12 элементдан иборат массивнинг охирги элементи 11 индексга эга бўлади. Код ёзиш жараёнида сизга индекси 1 дан бошланувчи массив зарур бўлса, қуйидаги сатрни ёзинг:

```
Dim MyArr (1 to 12)
```

Шу билан бирга сиз ўрнатилган катталикларни ўзгартириш учун Option Base операторидан ҳам фойдаланишингиз мумкин

Фойдаланувчи аниқлайдиган маълумот турлари

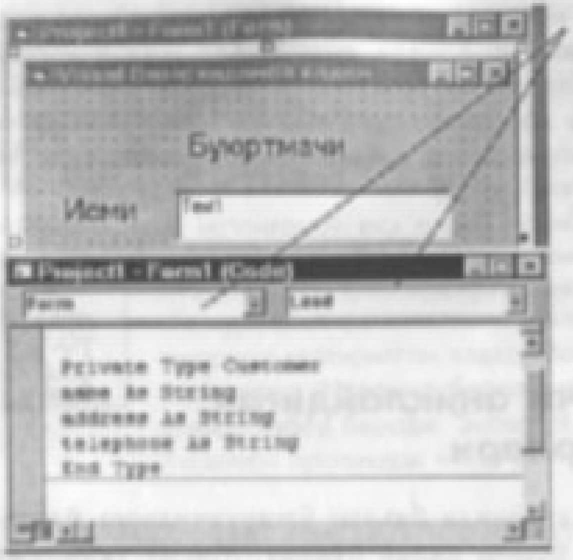
Дастур ёзиш жараёнида баъзан ўзгарувчиларга биттадан ортиқ қиймат бериш қулай бўлади. Мисол учун, сиз яратган дастур маълум маҳсулот буюртмачилари билан ишлайдиган бўлса, ҳар бир буюртмачининг номи, манзили ва телефони каби маълумотларни сақлаш лозим бўлиши мумкин. Бу муаммони ечишнинг бир йўли уч хил ўзгарувчи эълон қилиш, масалан, CustName, CustAddress ва CustTel.

Лекин энг қулай усул Customer ўзгарувчисини алоҳида соҳаларга ажратиш: Customer.Name, Customer.Address ва Customer.Telephone. Visual Basic да бу ишни фойдаланувчи томонидан аниқланадиган маълумот турини кўрсатиш билан ҳал қилинади.

Type билан мисол

Қуйидаги дастур шаклга буюртмачи ҳақидаги ахборотни чиқаради. Дастурда Customer туридаги ўзгарувчидан фойдаланилади. Уни аниқлашни келтирамиз.

```
Private Type Customer
name As String
address As String
Telephone As String
End Type
```



1. Муҳаррир кодини очиш учун шаклни икки марта босинг. Код муҳарриринг юқорисидаги томонидаги рўйхатдан General (Умумий) сатрини, юқори-ўнг томондаги рўйхатдан Declarations (Эълонлар) сатрини танланг. Кейин юқорида келтирилган Customer туридаги таърифни киритинг.

Customer тури мазкур шакл модулида Private (ёпиқ) асосий сўзи билан эълон қилинган. Бунинг маъноси шуки, Customer фақат шу шаклга боғланган код модуллари билан ишлайди. Агар сиз Customer ни бир нечта шаклларда ишлатмоқчи бўлсангиз, янги Basic — модул яратинг ва Private сўзи ўрнига Public (Очиқ) асосий сўзини қўллаб, бу турни аниқланг.

Энди сиз Customer туридаги ўзгарувчини эълон қилишингиз ва ундан ўз кодингизда фойдаланишингиз мумкин. Масалан:

```
Dim Custvar As Customer
CustVar. Name = "Ne'matjon"
CustVar.Address = "Mustaqillik -5"
CustVar. Telephone = "371 139 16 26"
```

Қисм (ёрдамчи) дастурлар тузиш

Шаклнинг Click (Босиш) ходисасини очиш учун тугмани икки марта босганингизда, ходиса коди қуйидагича бошланишини ва тугалланишини кўрасиз:

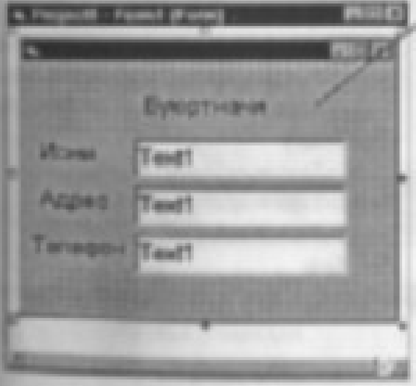
```
Private SuB ... End SuB
```


Sub — Subroutine (қисм дастур) сўзининг қисқартирилган шакли бўлиб, тугмани ҳар гал босганда доимий бажариладиган жараённи англатади. Шу билан бирга сиз, бирор ходисалар билан бевосита боғланмаган шахсий қисм дастурларни тузишингиз мумкин. Бу қисм дастурларни коднинг бошқа қисмларидан зарур бўлганда чақириб олиш мумкин.

Агар сиз дастур ёзиш жараёнида битта кодни дастурнинг бир нечта қисмларида такрор ёзадиган бўлсангиз, қисм дастур тузиш имкониятини ўйлаб кўринг. Бу сизнинг кодингизни кейинги хизмат учун янада самарали ва янада содда қилади.

Ёрдамчи дастур намунаси

Қуйида келтирилган мисол олдинги бўлимда кўрсатилган мисолнинг давоми. Буюртмачилар ҳақидаги ахборот учта матн ойнасида чиқади. Иш жараёнида кўп учрайдиган ҳаракат — учала матн ойнасидан ахборотни ўчириш зарурияти юзага келиши мумкин. Ҳар гал лозим бўлганда битта коднинг бир нечта сатрини ёзиш ўрнига бу ишни бажарадиган қисм дастурни бир марта ёзишингиз мумкин.



1. Олдинги бўлимдаги шаклни очинг, унга расмда кўрсатилгандек матн ойналари ва ёзувлар жойлаштиринг.
2. Код муҳарририни очинг ва Declarations (Эълон) бўлимини яна бир марта топинг. Customer нинг давомидан қуйидаги кодни ёзинг:

```
SuB ClearForm ()
Text 1. Text = " "
Text 2. Text = " "
Text 3. Text = " "
End SuB
```

3. Шаклнинг Load (Юклаш) ходисаси учун қуйидаги сатрни ёзинг:

```
Call ClearForm
```

Дастурни ишга туширсангиз, матн ойналари тозаланади.

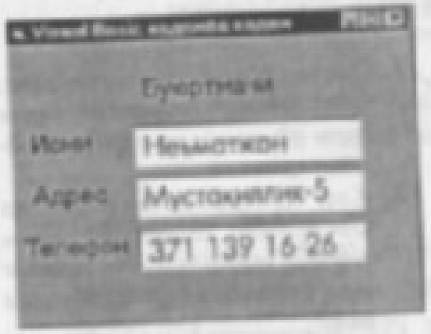
Қисм дастурни ишга тушириш учун Call сўзидан фойдаланиш шарт эмас. Лекин бу сўздан фойдаланиш фойдадан холи эмас, чунки у сизга коднинг қандай ишлашини эслатиб туради.

Қисм дастурлар параметрлари

Ёрдамчи дастур ҳар гал ишга туширилганда сиз унга параметрлар деб аталувчи ахборот беришингиз мумкин. Бу келтирилган маълумотга аниқлик киритиш ва олдинги бўлимларда келтирилган дастурлар ишлаб чиқишни давом эттириш учун буюртмачи ҳақида ахборотни кўрсатувчи ёрдамчи дастурни кўриб чиқамиз.

Параметрли қисм дастур

Қуйида келтирилган қисм дастурни PrivateSub (Ёпиқ) деб эълон қилишингиз зарур, чунки унга ишлаётган Customer тури шаҳс модулида ёпиқ турдир. Акс ҳолда Visual Basic хатолик ҳақида хабар беради.



1. Мати муҳарририни очинг ва олдинги ClearForm қисм дастурни топинг. Қисм дастурнинг давомидан (End Sub дан кейин) қуйида келтирилган кодни қўшиб қўйинг: Private SuB DisplayCust (cust — par As Customer)
Text1. Text = custpar. name
Text2. Text = custpar. address
Text3. Text = custpar. Telephone
End Sub

2. Load (Юклаш) ходисаси учун кодни қуйида келтирилганидек ўзгартиринг ва дастурни ишга туширинг:
Dim custvar As Customer
Call ClearForm
Custvar. name = "Неъматжон"
Cust var. address = " Мустақиллик — 5"
Cust var. Telephone = " 371 139 16 26"
Call DisplayCust (custvar)

Параметрдан фойдаланишнинг бошқа бир намунасини кейинги бўлимдан кўринг.

Функцияларни яратиш

Функциялар қисм дастурларга жуда ўхшайди. Лекин уларнинг муҳим фарқи ҳам мавжуд: функция қийматни қайтаради. Демак, функция ҳам ўзгарувчилар каби маълум турга (типга) эга. Функция тури — функция томонидан қайтарилувчи қиймат туридир.

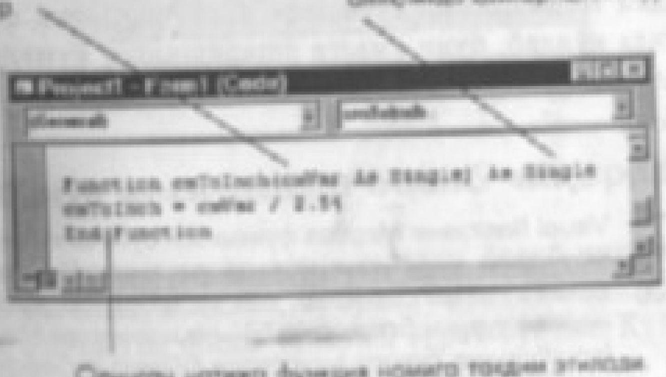
Қисм дастурлар ва функциялар ёзиш билан сиз Visual Basic имкониятларини кенгайтирасиз. Айни шу сифати Visual Basicни кучли қуролга айлантиради.

Дюймга айлантириш: функцияга мисол

Бу мисол сантиметрларни дюймларга айлантирувчи дастурдир. Унинг асосий қисми фойдаланувчи томонидан аниқланган ва қуйидагича эълон қилинган функциядир:

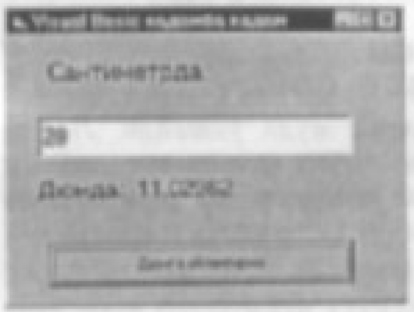
Бу сантиметрларда ифодаланган узунлик қийматини сақловчи параметр.

Бу қисм функциянинг силжувчи нуқтали сонни бир хона аниқликда қайтаришини кўрсатади.



Олинган натижа функция номига тақдим этилади. Функцияни чақирган код шу қийматни олади.

Сиз бирорта фойдали функцияни, мисол учун cmToInchни яратсангиз, уни код модулида сақлашингиз ва зарур бўлганда кейинги лойиҳаларда фойдаланишингиз мумкин.



Бу расмда cmToInch функциясининг натижаси кўрсатилган. Бу дастурни яратиш учун тугмани Click (Босиш) ходисасига матн ойнасидаги дюймга айлантириш қийматини параметр сифатида тақдим этинг. Матнни сонга айлантириш учун Val (функциясидан, тескари жараён учун эса Str() функциясидан фойдаланинг.

MsgBox ва InputBoxдан фойдаланиш

MsgBox ва InputBox функциялари ўхшаш функциялар бўлиб, улар фойдаланувчига ахборот бериш ва фойдаланувчи бериши мумкин бўлган жавобларни олишда ишлатилади.

MsgBox оператор сифатида ёки функция сифатида ишлатилиши мумкин. Энг оддий шаклда у маълумот панелини чиқаради, бу панел фойдаланувчи томонидан OK тугмачаси босилганда ёпилади.

```
MsgBox ("Операция тўла бажарилди")
```

Баъзан фойдаланувчидан кўпроқ маълумот олиш талаб қилинади. Масалан, кўпгина иловалар босмага беришдан олдидан тасдиқлашни сўрашади. Сиз MsgBoxни функция сифатида қўллаб, буни амалга оширишингиз мумкин.

```
If MsgBox ("OK- босилган", "vbOKCancel")= vbOK Then
'Босиш
End If
```

Visual Basicнинг MsgBox функцияси бўйича тўлиқ маълумот олиш учун маълумотлар системасига мурожаат қилинг, у ерда сарлавҳа, сон ва хабарлар панели тугмалари турини бошқариш бўйича,шунингдек MsgBox ва InputBoxнинг бошқа жиҳатлари бўйича ахборот келтирилган.

InputBox дан фойдаланиш

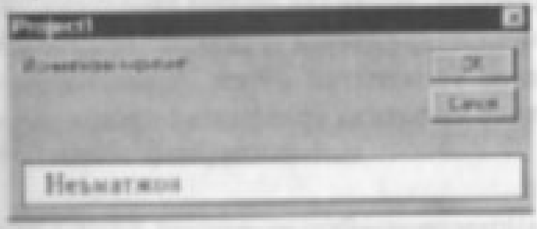
Баъзи ҳолларда тугмани оддий босилганига нисбатан кўпроқ ахборот олиш зарурати юзага келади, бундай ҳолларда InputBox функциясидан фойдаланиш мумкин. Мисол учун, фойдаланувчи томонидан киритилган номи бўйича буюртмачини топиш зарур бўлсин. Қуйидаги код шу масалани ечишга ёрдам беради

```
Dim sName As String
sName= InputBox ("Исмни киритинг")
MsgBox = ("Сиз танлаган" + sName)
```

Энди сиз InputBox га қиймат (кўрсаткич, ном) беришингиз мумкин. Аввалги мисолдаги InputBox кодини қуйидаги код билан алмаштиринг:

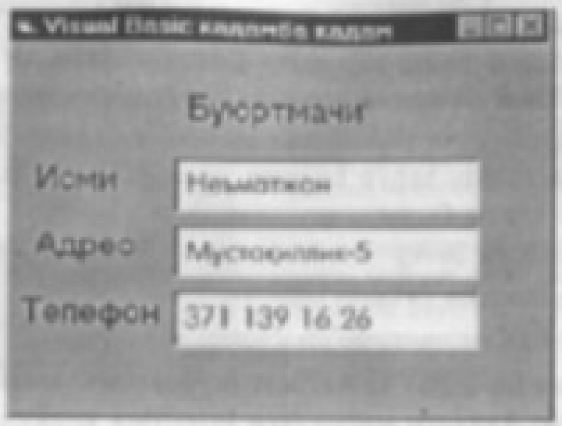
```
sName= InputBox (Prompt:= "Исмни киритинг",
Default: = "Неъматжон"
```

Юқоридаги код ишга туширилганда диалог ойнасида "Неъматжон" ёзуви чиқади. Агар фойдаланувчи OK тугмасини босса, функция "Неъматжонни" қайтаради, агар "Cancel" ("Бекор қилиш") тугмаси босилса, функция бўш сатрни қайтаради.



PrintForm воситасида босиб чиқариш

Visual Basicда содда хужжатларни босиб чиқариш мураккаб иш эмас, масалан, буюртмачи ҳақида шаклга киритилган ахборотни босиб чиқариш керак бўлсин. Қуйида бу масалани ечишнинг икки усули кўрсатилган.

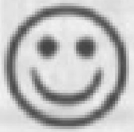


Расмда шу бобда яратилган буюртмачи ҳақида ахборот бўлган шакл келтирилган. Расмга эътибор берсангиз, унга "Босиш" тугмаси қўшилган. Бу ахборотни чоп қилишнинг иккита усули мавжуд. Биринчи ва энг қулай усул PrintForm усулидир. Бу усулдан фойдаланиш учун "босиш" тугмасини Click ҳодисасига қуйидаги кодни ёзинг:

```
Me.PrintForm
```

Сиз PrintForm усулидан фойдалансангиз, Visual Basic шакл тасвирини принтерга узатади. Агар системада бир нечта принтер ўрнатилган бўлса, Visual Basic Windowsга асосий принтер сифатида ўрнатилган принтерда чоп қилади.

Шакллар ҳам Print усулига эга. Бу усул PrintForm усулидан тубдан фарқ қилади. Print усули принтерда эмас, балки шаклда чоп қилади.



Шаклнинг график элементлари AutoRedraw хусусияти True ҳолатига ўтказилгандагина босиб чиқарилади.

PrintForm усули бир нечта муаммога эга:

- Шаклнинг AutoRedraw хусусияти True ҳолатига ўтказилгандагина унинг график элементлари босиб чиқарилади.
- Босиш сифати экраннинг ажратиш қобилиятига мос бўлиб, юқори сифатли принтерларда тасвир кўринарли нуқтавий структурага бўлади.
- Шакллар одатда экранга чиқариш учун мўлжаллангани сабабли дизайни бўйича қоғозга чиқариладиган талабларга жавоб бермайди.

Бу муаммоларни ҳал қилиш учун PrintForm усули ўрнига Printer объектини ишлатиш зарур. Босишнинг бу тури кейинги бўлимда муҳокама қилинади.

Printer объекти ёрдамида босиб чиқариш

Printer объекти Visual Basic да ўрнатилган объект бўлиб, код ёзиш жараёнида унга мурожаат қилиш ва махсус ҳаракатлар қилмасдан туриб (системага ўрнатилган принтер бўлса) чоп қилиш мумкин.

Printer объектидан фойдаланиш

Қуйида буюртмачи ҳақида шаклга илгари чиқарилган ахборотни чоп қилувчи код намунаси келтирилган:

```
Printer. Print "Буюртмачи" ҳақида ахборот"
Printer. Print/ бўш сатр қолдирамиз
Printer. Print "Исм:" + Text1. Text
Printer. Print "Манзил"; + Text 2. Text
Printer. Print "Телефон: + "Text 3.Text
Printer. EndDoc
```

Дастур EndDocга етиб бормагунча, Printer.Print операторларидан бирортаси ҳам маълумотларни чоп этмайди. Шундан кейингина барча маълумотлар принтерга узатилади. Шу вақтга қадар босишни бекор қилмоқчи бўлсангиз, KillDoc усулидан фойдаланишингиз мумкин.

Муаммолар ва ечимлар

Юқорида келтирилган усулда маълумотни чоп қилиш ўзига хос муаммоларни келтириб чиқаради, босиш сифати унчалик кўримли бўлмайди. Шрифт тарихасида ўрнатилган шрифт қабул қилинади. Агар сиз принтер бети (принтер қоғози ўлчамлари) га сиғмайдиган маълумот узатсангиз, бу маълумот узилган ҳолда босиб чиқарилади. Бир сўз билан айтганда, қоғозга чиқариладиган маълумотни сифатли қилиб чоп қилиш учун кўп меҳнат талаб қилинади.

Printer объектида Font (Шрифт) хусусиятини ўзгартириб, принтер шрифтини ўрнатишингиз мумкин, масалан:

```
Printer. Font = "Arial"
Printer. FontSize = 12
```

Агар сиз фойдаланувчи системасида қандай шрифтлар ўрнатилганлигини билмасангиз, Arial, Courier ва Times New Roman шрифтларидан фойдаланинг. Бу шрифтлар Windows дастурларида ўрнатилган бўлади.

Сиз Visual Basic да ўрнатилган шрифтларни Font (Шрифт) хусусиятидан, қоғоз ўлчамларини принтернинг Height (Баландлик) ва Width (Кенглик) хусусиятларидан фойдаланиб аниқлашингиз мумкин. График маълумотни график усуллардан фойдаланиб чоп этишингиз мумкин. Нашр қобиғини эса Current X ва CurrentY хусусиятлари ёрдамида белгилашингиз мумкин. Принтер объектидан фойдаланишда сифатли чоп қилиш учун ёзиладиган сатр узунлиги, уни ажратиш ва бошқа сатрларга ўтказиш каби масалаларни код ичида ёзиш зарур. Умуман, Visual Basic да дастур ёзишни ўзлаштиргунча, принтер объектида фақат содда ҳужжатларни чоп қилган маъқул.

ТЎРТИНЧИ БОБ

Visual Basic нинг асбоблар воситаси

Visual Basic нинг асбоблар воситалари ва усталари (мастерлари) жуда кенг имкониятларга эга, лекин дастур билан ишлашни бошлаганда уларни фарқлаш жуда ҳам қийин. Мазкур бобда меню муҳаррири, созлагич ва йўлланма ойнаси каби муҳим воситалар муҳокама қилинади. Бу ерда сиз яна Visual Basic муҳитини созлаш бўйича ахборот оласиз.

Меню яратиш

Windows қобиғида ишловчи дастурларнинг кўпчилигида фойдаланувчи дастурни тўла бошқариш учун менюдан фойдаланади. Visual Basic лойиҳага меню чизгичини қўшиш имконини берувчи оддий кўриниш (визуал) воситаларга эга.

1. Янги лойиҳа яратинг, менюнинг Tools (Сервис) қисмидан Menu Editor (Меню муҳаррири) сатрини танланг.

2. Caption (Сарлавҳа) ойнасида меню чизгичи биринчи сатрининг номини ёзинг. Кейин Name (Ном) ойнасида менюнинг ўзининг номини ёзинг. (Номни шундай танлангки, у қайси меню кўзда тутилаётганлигини аниқ эслатиб турсин). Менюни рўйхатга қўшиш учун Next тугмасини босинг. Менюда "Файл", "Тузатиш", "Ёрдам" бўлимларини ташкил қилиш учун юқоридаги иш тартибини уч марта бажаринг.

Менюнинг барча бўлимлари ўз номига эга бўлиши зарур. Акс ҳолда Visual Basic менюдан фойдаланиш имконини бермайди.

"&" белгиси меню сарлавҳасидаги бирор ҳарф олдига қўйилса, менюда "тезкор" клавиш ҳосил қилади. Менюни чақиришда бу ҳарф остига чизилган бўлади.

3. Кейин менюнинг "Тузатиш" бўлимини танланг ва Insert (Киритиш) тугмасини босинг. Менюга яна бир бўш сатр қўшилади. Бўш сатрни танланг ва унга "Очиш" сарлавҳасини киритинг. Ўнг томонга йўналтирилган стрелкали тугмани босинг. "Очиш" менюси олдидан бўш соҳа очилади. Меню муҳарририни ёпиш учун OK тугмасини босинг.

Юқорига ва пастга йўналтирилган стрелкали тугмалар ёрдамида меню рўйхатидаги сатрлар тартибини ўзгартириш мумкин.

4. Дастурни ишга туширишингизда, шаклнинг юқори қисмида меню чизгичи пайдо бўлишига эътибор беринг. Менюнинг "Очиш" сатри "Файл" бўлимининг ичига жойлаштирилган, чунки меню муҳарририда "Очиш"дан олдин бўш жой қолдирилган эди.

Менюни созлаш

Visual Basic да менюни созлашнинг бир нечта усули мавжуд. Бу ерда менюни созлаш меню муҳаррири опциялари ёрдамида амалга оширилган.

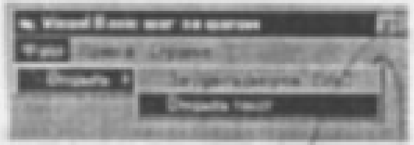
"&" белгиси ёрдамида яратилган "Тезкор" тугма ва меню муҳарририда белгиланган клавишлар бирикмаси ўртасида фарқ бор. "Тезкор" тугмага нисбатан клавишлар бирикмаси тезроқ ишлайди, чунки у менюга кирмасдан, менюнинг мазкур сатрига мос буйруқни дарҳол бажаради. "Тезкор" клавиш бевосита менюга киргандагина ишлайди. Менюга кириш учун уни Alt клавиши ёрдамида ишчи ҳолатга келтирилади (активлаштирилади).

Менюнинг Enabled (Ишончли) ва Visible (Кўринувчи) хусусиятлари опциялари меню бўлими холатларини мос равишда белгилаш учун ишлатилади. Агар Enabled байроқчаси ўчирилган бўлса, мос равишда менюнинг шу бўлими фойдаланувчи томонидан сичқонча орқали ошкор қилинган ҳаракатларга жавоб бермайди. Visible байроқча хусусияти ўчирилса, бу меню умуман кўринмас бўлиб қолади.

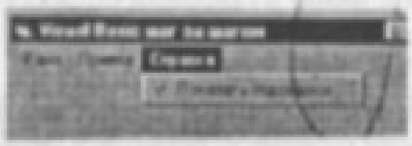
1. Менюнинг жорий ҳолатига мос буйруқни бажариш учун клавишлар бирикмасини танланг.

2. Менюнинг белгилаш имконияти бўлган қисмини яратиш учун Checked (Белгиланувчи) бўлимини танланг.

3. Менюнинг очилувчи қўшимча қисмлари яратиш учун менюнинг барча ("Файл", "Таҳрир", "Ёрдам") бўлимларида бўш оралиқлар ташланг.



Менюнинг қўшимча қисми клавишларнинг Ctrl+O бирикмасига эквивалентдир.



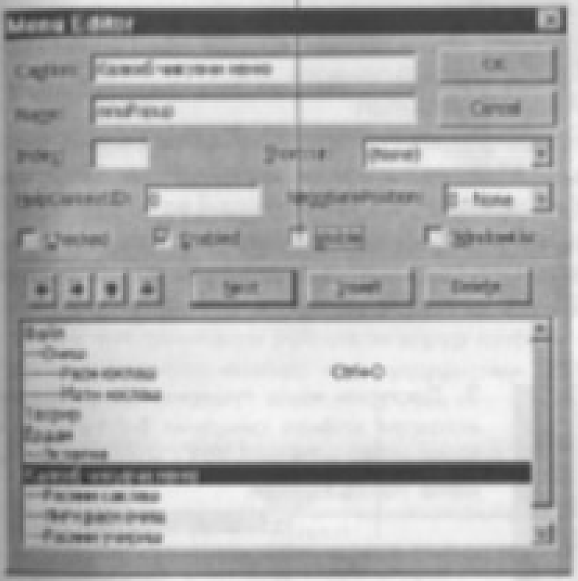
Менюнинг белгиланган бўлимига мисол.

Қалқиб чиқувчи меню яратиш

Windows қобиғида ишловчи дастурларнинг барчасида бирор объектни сичқонча ўнг тугмасини босганда қалқиб чиқувчи менюлардан фойдаланилади. Одатда бу менюнинг

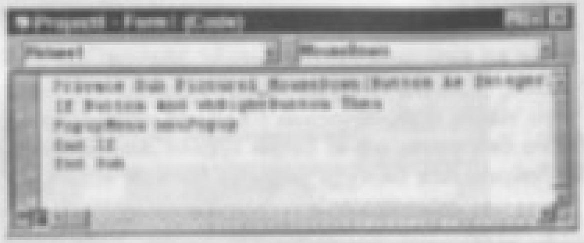
мазмуни танланган объектга боғлиқ равишда ўзгаради. Қалқиб чиқувчи менюни ҳосил қилиш тартибини қараб чиқамиз

1. Меню муҳарририни очинг ва янги меню ҳосил қилинг. Мазкур мисолда бу меню mnuPopup номи, "Қалқиб чиқувчи меню" сарлавҳаси билан берилган. Бу меню учун Visible байроқчасини белгиламай қўйинг. Кейин бўш соҳаси билан бирга бир нечта меню сатри қўшинг. Менюнинг бу сатрлари учун Visible байроқчаси белгиси ўрнатилган бўлиши лозим.



Сичқонча ўнг тугмасининг босилишини қайд этиш учун Click (чертиш) ҳодисасини қўллай олмайсиз. Чунки Click ҳодисаси сичқонча тугмаси босилганда сичқончанинг қайси тугмаси босилганлиги ҳақида ахборот бера олмайди. Click ҳодисаси ўрнига сичқончанинг қайси тугмаси босилганлигини фарқловчи MouseDown ҳодисасидан фойдаланиш мақсадга мувофиқ.

3. Шаклга график ойна жойлаштиринг. График ойна учун сичқончани икки марта босиб код муҳарририни очинг. Юқори ўнг томондаги рўйхатдан MouseDown (сичқонча тугмаси босилган) ҳодисасини танланг. Бу MouseDown тартибини яратади. Расмда кўрсатилгандек коднинг учта сатрини ёзинг.



Мазкур коднинг асосий сатри бир кўринишда қолати тушлиши мумкин. If операторидаги
If button And vbRightButton
шартли ифода разрядли таққослаш жараёнидан фойдаланади. (And операцияси — мантиқий ва операнднинг ҳар бир разряди учун қўлланилади) ва vbRightButton константаси билан берилган Button параметрининг маълум разряди 1 га ўрнатилгандагина шартли ифоданинг қиймати Truega тенг бўлади.

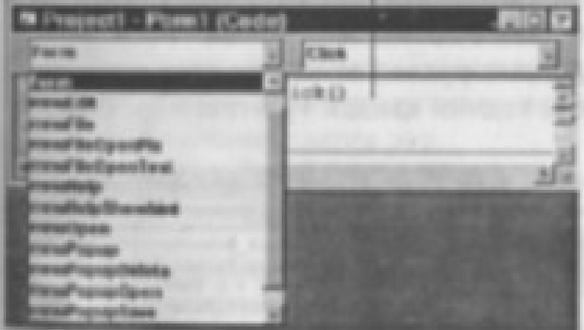
3. Дастурни ишга туширинг ва расмнинг исталган жойида сичқонча ўнг тугмасини босинг, жавоб тариқасида қалқиб чиқувчи меню пайдо бўлади.

Менюни ишга тушириш

Сиз яратган менюларнинг кўриниши чиройли бўлгани билан, улар ҳозирча ҳеч қандай иш бажармайди. Бу бўлимда менюнинг танланган вазифасига (сатрига) мос равишда иш бажарувчи код ёзамиз.

Меню ҳойкасини тугатгач, меню муҳарририни ёпинг. Сиз қосил қилган меню шаклда кўринади. Менюнинг зарур қисмини меню ишчи ҳолатида танлангандек қилиб танланг. Click (Босиш) ҳодисаси учун код муҳаррири ойнаси очилади.

Агар сиз кўринмас меню, масалан, қалқиб чиқувчи меню яратган бўлсангиз, код муҳарририни юқоридаги тартибда оча олмайсиз. Бунинг ўрнига код муҳарририни очиш учун шаклни ихтиёрий жойига икки марта босинг ва юқори чап томондаги рўйхатдан зарур сатрини танланг. Visual Basic жавоб ҳаракати тариқасида код муҳарририни Click (Босиш) ҳодисаси учун очади.



Visual Basicнинг исталган объекти учун код муҳарририни очишда ёки янги тартиб ёки функцияларни яратишда юқорида келтирилган усулни қўллаш мумкин. Баъзи ҳолатларда бу усул объектни икки марта босишга нисбатан қулайроқдир.

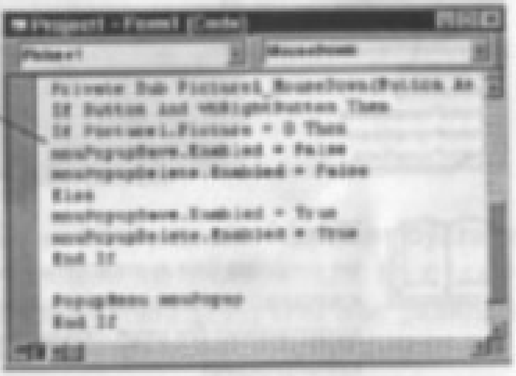
Дастурни бажариш вақтида менюни ўзгартириш

Windows қобиғида ишловчи энг яхши дастурлар дастурни жорий ҳолатида аҳамиятга эга бўлмаган, яъни жорий ҳолатга алоқадор бўлмаган меню бўлимларини яшириш ёки уларни умуман кўринмайдиган қилиб қўйиш имконига эга. Масалан, дастурда (Word, Excel) биронта ҳам ҳужжат очилмаган бўлса, унинг Save (Сақлаш) бўлими ишламайдиган ҳолатда бўлиши зарур. Чунки биронта ҳам ҳужжат ёки лойиҳа очилмаган шароитда уни сақлашнинг мазмуни йўқ. Бундай ҳолатларда менюнинг зарур бўлмаган қисмларини яшириш ҳолатга ўтказиш ёки уларни ишламайдиган қилиб қўйиш мумкин.

Қуйида келтириладиган мисол "Қалқиб чиқувчи меню яратиш" бўлимида келтирилган мисол асосида бажарилади. Сиз расмга ўнг босиш қилганингизда менюнинг "Расмни сақлаш" (Save), "Расмни ўчириш" (Delete) бўлимлари пайдо бўлади. Агар жорий ҳолатда график ойнага расм юкланмаган бўлса, у ҳолда менюнинг юқорида келтирилган бўлимлари ишламайдиган ҳолатга ўтказилган бўлиши зарур.

Бу ишнинг сирини менюнинг Enabled хусусиятини True ва False ҳолатларини ўзгартирадиган код ёзиш ташкил этади. Қуйидаги менюнинг "Расмни сақлаш" ва "Расмни ўчириш" сатрларига киришга рухсат этиш ёки тақиқлаш учун MouseDown ҳодисаси кодини қандай тузатиш кўрсатилган.

График ойнага нихта босиб, код муҳарририни очинг ва MouseDown ҳолати учун қуйидаги кодни ёзинг.



Мазкур код фақат меню номлари код ичида ёзилган номлар билан тўғри келгандагина ишлайди. Меню сатрларининг номини меню муҳаррири ёрдамида ўзгартиришингиз мумкин.

Дастурни ишга туширгач ва ўнг тугмани босгач, расм юкланмаган бўлса, менюнинг "Расмни сақлаш" ва "Расмни ўчириш" бўлимлари ишламайдиган ҳолатда бўлади.

Шу билан бирга менюнинг Visible (Кўриниш) хусусиятини False ҳолатига ўтказиб, уни кўринмайдиган қилиб қўйишингиз мумкин.

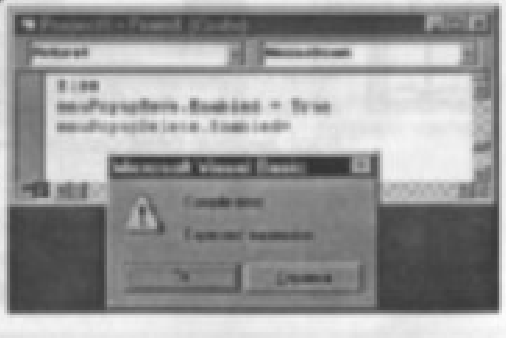
Бошқа фойдали хусусиятлардан бири Checked (Белгиланадиган) хусусиятидир. Бу хусусиятни False ёки True ҳолатига ўтказиб, меню ёнида пайдо бўлувчи белгини бошқариш имкониятига эга бўласиз. Фойдаланувчига бу белги мазкур бўлим ишчи ҳолатда ёки ишламайдиган ҳолатдалигини кўриш имкониятини беради, шу сабабли мос диалог панелини очмаса ҳам бўлади.

Код муҳаррири

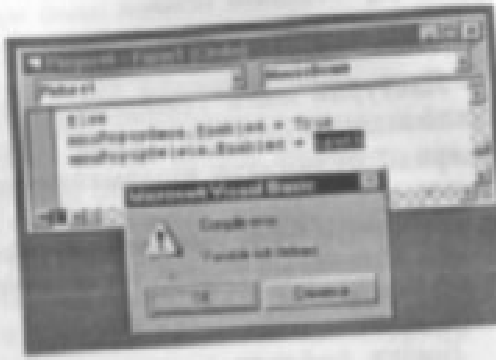
Visual Basic билан ишлаш жараёнида энг кўп вақт талаб қилинадиган ишлардан бири бу код муҳарририда код ёзиш ва кодни таҳрир қилишдир. Бу муҳаррир кўриниши жиҳатидан содда бўлса ҳам ишингизни осонлаштирувчи жуда кўп махсус имкониятларга эга.

Синтаксисни текшириш

Дастур ёзиш жараёнида код муҳарририда код сатрини ёзиб, Enter тугмасини босишингиз билан Visual Basic кодининг синтактик хатоликларини текширади ва аниқланган хатоликни ажратиб кўрсатади. Масалан, "=" дан кейин ўзгарувчи ёки ифоданинг йўқлиги ва ҳ.к.

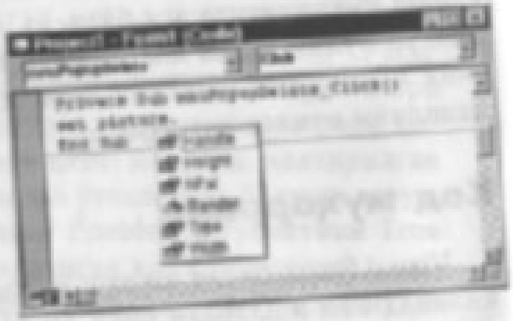


Дастурни бажаришдан олдин компиляция қилиш



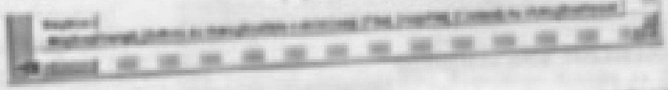
Дастурни ишга туширишда менюнинг Run (ишга тушириш) бўлимидан Start with full compile (тўла компиляция қилиб бошлаш) сатрини танласангиз, Visual Basic автоматик тарзда кодни компиляция қилишга имкон бермайдиган орфографик хатоларни текширади.

Тезкор маълумот ва компонентлар рўйхати



Муҳаррир иш саҳифасида нуқта ёзсангиз, у мавжуд услуб ва хусусиятлар рўйхатини кўрсатади. Рўйхатда зарур компонент танлангач, унинг номини автоматик киритиш ва рўйхатни ёпиш учун Tab ёки (Space — пробел) клавишини босинг.

Тезкор маълумот ойнаси ҳам автоматик тарзда функция хусусиятларини акс эттиради.



Агар юқорида келтирилган бу имкониятларнинг ҳаммаси сизга зерикарли туюлса, сиз истаган вақтда уларни менюнинг Tools (Сервис) бўлими Options (Параметрлар) диалог ойнасидан ўчиришингиз мумкин.

Сизга тезкор маълумот зарур бўлса, керакли сўзни танлаган ҳолда сичқонча ўнг тугмасини босинг ва қалқувчи менюдан Quick Info (тезкор маълумотнома) сатрини танланг.

Созлагич ҳақида

Одатда кўпчилик дастурлар биринчи марта ишга туширилганда кутилган натижани бермаслиги мумкин. Компьютер дастурларининг тўғри ишлаши дастурга киритилган белгиларнинг тўғрилигига боғлиқ: минглаб сатрларга ёзилган кодлар орасига битта нотўғри белги қўшилиб қолган бўлса, дастур тўхтайди. Бошқа холатлар дастурнинг секин ишлаши, дастур кўп вақт тўғри ишлаб туриб, кутилмаганда хато натижалар кўрсатиши ва ҳ.к. бўлиши мумкин.

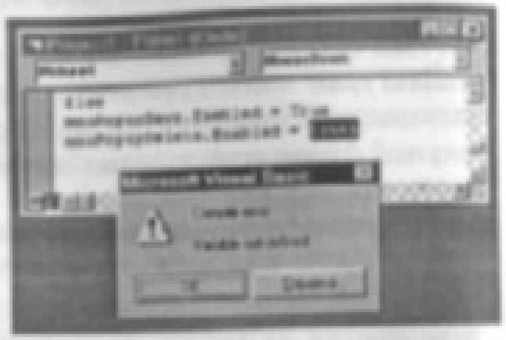
Дастурнинг оддий иш жараёнида унинг коди кўринмайди. "Созлагич" эса дастур бажарилиши жараёнида кодни сатрма-сатр кузатиш имконини беради. Созлагич ёрдамида зарур холатларда ўзгарувчиларнинг жорий қийматларини текшириб кўриш учун дастурнинг бажарилишини тўхтатиш хам мумкин.

Агар дастурингиз чексиз цикл ҳолатига тушиб қолса, нима қилиш керак? Бундай ҳолатда Ctrl+Break(Pause) клавишлар бирикмасини босинг: дастур ишини тўхтатади ва созлагични очади.

Созлаш асослари

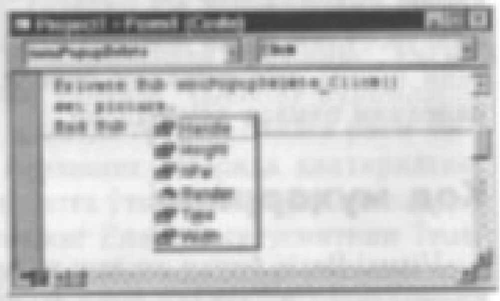
1. Дастурнинг сатрма-сатр ишлашини кузатиш учун менюнинг Debug (Созлагич) бўлимидан Step Into (кириш билан қадам) сатрини танланг.

Дастурни бажаришдан олдин компиляция қилиш



Дастурни ишга туширишда менюнинг Run (ишга тушириш) бўлимидан Start with full compile (тўла компиляция-лашдан бошлаш) сатрини танласангиз, Visual Basic автоматик тарзда кодни компиляция қилишга имкон бермайдиган орфографик хатоликларни текширади.

Тезкор маълумот ва компонентлар рўйхати



Муҳаррир иш соҳасида нукта ёзсангиз, у мавжуд услуб ва хусусиятлар рўйхатини кўрсатади. Рўйхатда зарур компонент танлангач, уни номини автоматик киритиш ва рўйхатни ёпиш учун Tab ёки (Space — пробел) клавишини босинг.

Тезкор маълумот ойнаси ҳам автоматик тарзда функция хусусиятларини акс эттиради.

Агар юқорида келтирилган бу имкониятларнинг ҳаммаси сизга зерикарли туюлса, сиз исталган вақтда уларни менюнинг Tools (Сервис) бўлими Options (Параметрлар) диалог ойнасидан ўчиришингиз мумкин.

Сизга тезкор маълумот зарур бўлса, керакли сўзни танлаган ҳолда сичқонча ўнг тугмасини босинг ва қалқувчи менюдан Quick Info (тезкор маълумотнома) сатрини танланг.

Созлагич ҳақида

Одатда кўпчилик дастурлар биринчи марта ишга туширилганда кутилган натижани бермаслиги мумкин. Компьютер дастурларининг тўғри ишлаши дастурга киритилган белгиларнинг тўғрилигига боғлиқ: минглаб сатрларга ёзилган кодлар орасига битта нотўғри белги қўшилиб қолган бўлса, дастур тўхтайди. Бошқа ҳолатлар дастурнинг секин ишлаши, дастур кўп вақт тўғри ишлаб туриб, кутилмаганда хато натижалар кўрсатиши ва ҳ.к. бўлиши мумкин.

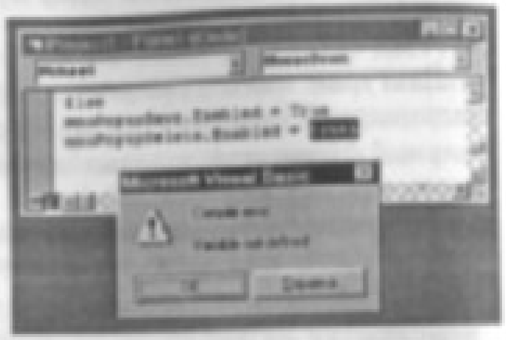
Дастурнинг оддий иш жараёнида унинг коди кўринмайди. "Созлагич" эса дастур бажарилиши жараёнида кодни сатрма-сатр кузатиш имконини беради. Созлагич ёрдамида зарур ҳолатларда ўзгарувчиларнинг жорий қийматларини текшириб кўриш учун дастурнинг бажарилишини тўхтатиш ҳам мумкин.

Агар дастурингиз чексиз цикл ҳолатига тушиб қолса, нима қилиш керак? Бундай ҳолатда Ctrl+Break(Pause) клавишлар бирикмасини босинг: дастур ишини тўхтатади ва созлагични очади.

Созлаш асослари

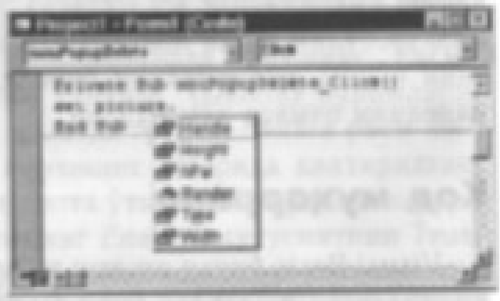
1. Дастурнинг сатрма-сатр ишлашини кузатиш учун менюнинг Debug (Созлагич) бўлимидан Step Into (кириш билан қадам) сатрини танланг.

Дастурни бажаришдан олдин компиляция қилиш



Дастурни ишга туширишда менюнинг Run (Ишга тушириш) бўлимидан Start with full compile (тўла компиляциялашдан бошлаш) сатрини танласангиз, Visual Basic автоматик тарзда кодни компиляция қилишга имкон бермайдиган орфографик хатоликларни текширади.

Тезкор маълумот ва компонентлар рўйхати



Муҳаррир иш соҳасида нуқта ёзсангиз, у мажмуа, услуб ва хусусиятлар рўйхатини кўрсатади. Рўйхатда зарур компонент танлангач, унинг номини автоматик киритиш ва рўйхатни ёпиш учун Tab ёки (Space — пробел) клавишани босинг.

Тезкор маълумот ойнаси ҳам автоматик тарзда функция хусусиятларини акс эттиради.

Агар юқорида келтирилган бу имкониятларнинг ҳаммаси сизга зерикарли туюлса, сиз исталган вақтда уларни менюнинг Tools (Сервис) бўлими Options (Параметрлар) диалог ойнасидан ўчиришингиз мумкин.

Сизга тезкор маълумот зарур бўлса, керакли сўзни танлаган ҳолда сичқончанинг ўнг тугмасини босинг ва қалқувчи менюдан Quick Info (тезкор маълумотнома) сатрини танланг.

Созлагич ҳақида

Одатда кўпчилик дастурлар биринчи марта ишга туширилганда кутилган натижани бермаслиги мумкин. Компьютер дастурларининг тўғри ишлаши дастурга киритилган белгиларнинг тўғрилигига боғлиқ: минглаб сатрларга ёзилган кодлар орасига битта нотўғри белги қўшилиб қолган бўлса, дастур тўхтайди. Бошқа ҳолатлар дастурнинг секин ишлаши, дастур кўп вақт тўғри ишлаб туриб, кутилмаганда хато натижалар кўрсатиши ва ҳ.к. бўлиши мумкин.

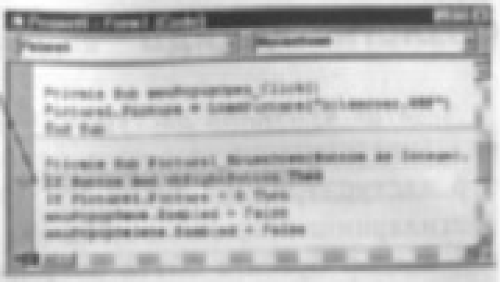
Дастурнинг одций иш жараёнида унинг коди кўринмайди. "Созлагич" эса дастур бажарилиши жараёнида кодни сатрма-сатр кузатиш имконини беради. Созлагич ёрдамида зарур ҳолатларда ўзгарувчиларнинг жорий қийматларини текшириб кўриш учун дастурнинг бажарилишини тўхтатиш ҳам мумкин.

Агар дастурингиз чексиз цикл ҳолатига тушиб қолса, нима қилиш керак? Бундай ҳолатда Ctrl+Break(Pause) клавишлар бирикмасини босинг: дастур ишини тўхтатади ва созлагични очади.

Созлаш асослари

1. Дастурнинг сатрма-сатр ишлашини кузатиш учун менюнинг Debug (Созлагич) бўлимидан Step Info (қадам билан қадам) сатрини танланг.

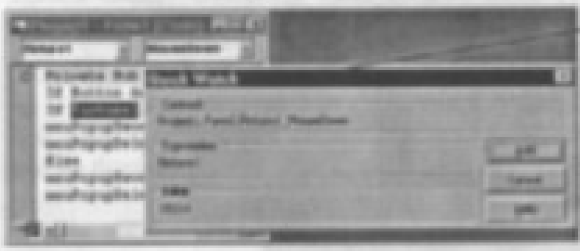
2. Visual Basic жорий қаторда бажарилаётган код сатрига борилди билан шу сатр тўғрисида актив сатрни кўрсатиб турувчи пайдо бўлади. Код сатр-ма-сатр бажарилиши учун F8 клавишадан, созлагич ишини тўхтатиш учун эса F5 клавишадан фойдаланинг.



Созлаш ҳақида батафсилроқ маълумот

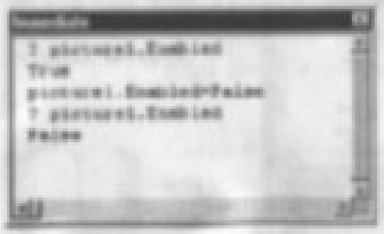
Созлагич ишчи ҳолатда турганида, созланаётган дастур ҳақида бир нечта усулда ахборот олиш мумкин.

Ўзгарувчининг жорий қийматини кўриш учун унинг устида сичқонча курсорини бир оз тўхтатинг.



Сўз ёки ифодани белгиланг ва Shift+F9 клавишлар бирикмасини босиб, Quick Watch (Тез кўриб чиқиш) ойнасини очинг.

Immediate (Созлаш) ойнаси алоҳида буйруқларни бажариш ҳамда дастурнинг бажарилиш вақтида ўзгарувчилар қийматини текшириш ва ўзгартириш имкониятини беради. Агар созлаш ойнаси кўринмаса, уни View менюсидан танлашингиз мумкин.



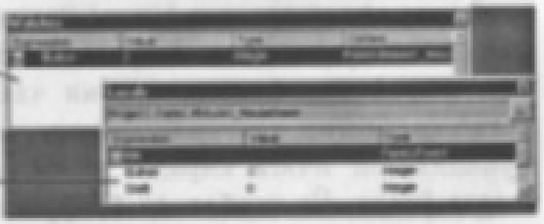
Ўзгарувчига тақдим этилган қийматни кўриш учун ўзгарувчидан олдин "?" белгисини ёзинг. Ўзгарувчига янги қиймат бериш учун эса "=" белгисидан фойдаланинг.

Locals ойнасида (Локал ўзгарувчилар ойнасида) жорий жараён ёки функция учун барча локал ўзгарувчилар ва уларнинг қийматлари рўйхати келтирилган. Wathces ойнаси (Назорат қиймати ойнаси) дастурни созлаш жараёнида қийматларини кузатиш зарур бўлган ўзгарувчиларни кузатиш имконини беради. Бу ойнага танланган ўзгарувчилар уларнинг кўрсаткич қийматлари билан бирга чиқарилади ва дастурнинг бажарилиш жараёнида улар автоматик тарзда янгиланади. Юқорида кўрсатилган ойналарни менюнинг View (Кўриниш) бўлимидан топиш мумкин.

Watches ойнасига ўзгарувчини жойлаштириш учун уни код муҳарриридан танланг ва Watches ойнасига тортиб ўтказинг. Бу ишни Watches ойнасига ўнг босишни бажариб ва қалқиб чиқувчи менюдан Add Watch сатрини танлаб ҳам амалга оширишингиз мумкин.

Watches ойнаси (Локал қиймат ойнаси)

Locals ойнаси (Маҳаллий ўзгарувчилар ойнаси)



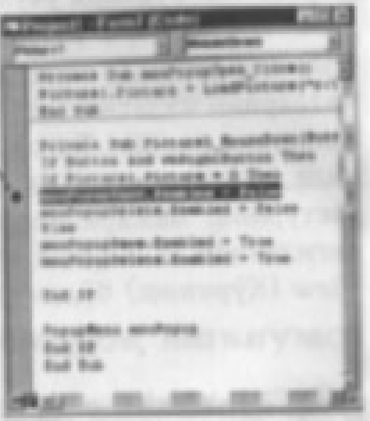
Узиш нуқталаридан фойдаланиш

Агар яратган дастурингизнинг ҳажми катта бўлса ва унинг асосий қисми хатосиз ишласа, уни созлагич ёрдамида сатрма-сатр бажариш жуда ҳам кўп вақтни олади.

Visual Basic узиш нуқталарини қўйиш имконини беради. Узиш нуқталари қўйилган сатргача дастур нормал режимда ишлайди. Бу вақтда дастур ишини тўхтатади ва созлаш режимига киради

Узиш нуқтасини ўрнатиш

Узиш нуқтасини ўрнатиш учун код муҳарририни очинг. Сичқонча ёрдамида чап томони босинг. Унда узиш нуқтасининг қўйилганлигини кўрсатувчи нуқта пайдо бўлади. Дастур ишга туширилгач, узиш нуқтаси қўйилган сатрга тўхтамасдан ишлайди. Узиш нуқтасини олиб ташлаш учун нуқтага яна бир марта босинг.



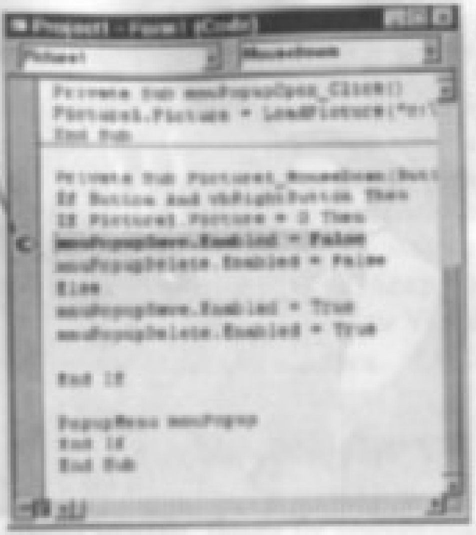
Узиш нуқтасини ўрнатиш ёки олиб ташлаш учун F9 калитидан фойдаланишингиз мумкин.

Step Over буйруғи (Айланиб ўтиш қадами)

Фараз қилайлик, сиз Visual Basic да ўзингизнинг функциянгизни яратдингиз. Агар сиз кодни сатрма-сатр бажариш жараёнида функциянгизни чақирувчи кодга рўпара келиб қолсангиз, дастур бу функциянинг бажариш учун шу функциянинг ичига киради, уни бажаради, кейин бошқа сатрга ўтади. Ўз навбатида бу функция катта код асосига эга бўлиши мумкин. Бундай ҳолларда кодни текширишни айланиб ўтиш мақсадида Step Over (Айланиб ўтиш) буйруғидан фойдаланиш зарур. Бу буйруқни қўллаш билан бир қаторда Shift + F8 бирикмасидан ҳам фойдаланиш мумкин. Бу ҳолда Visual Basic функция кодини уни сатр режимида ўтмасдан бажаради.

Step Out буйруғи (Чиқиш қадами) буйруғи Step Over билан ўхшашдир. Step Out коднинг қолган қисмини оддий, сатрма-сатр бажариш режимида амалга оширади. Бу буйруқни ишга тушириш учун Shift+Ctrl+F8 бирикмасидан ёки менюнинг Debug (созлаш) бўлимидан фойдаланинг.

Кейинги ҳолатда коднинг мазкур сатри функция кодини чақиради, уни бажармасдан ўтиб кетиш учун F8 тугмасидан, функция созлагичига кирмасдан, кодни бажариш учун Shift+F8 тугмасидан фойдаланинг.



Хатоликларни кузатиш

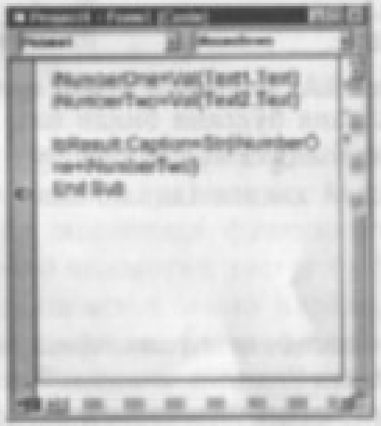
Баъзи дастурлар яхши компиляция бўлгани билан баъзи бир жараёнларни бажаришда хатоликка йўл қўяди. Бундай ҳолатларда Visual Basic хатолик ҳақида (диалог панелида) ахборот беради.

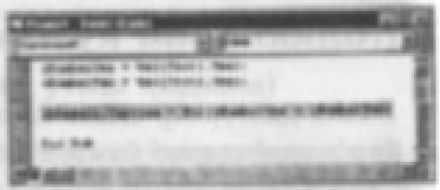
Visual Basic да яратилган дастур, дастурдан ташқари муҳитда ишлаётган ҳолатларда хатолик ҳақида хабар берса, бу муҳитда одатда дастур ўз ишини тўхтатиб қўяди. 7-бобда қараб чиқилган хатоликларни қайта ишловчи коддан фойдаланиб бундай ҳолатларнинг олдини олиш мумкин.

106-бетдаги расмда кўрсатилган ҳолатда дастур хатолик ҳақида ахборот берган. Мазкур ҳолатда муаммони ечиш учун расм йўлини тўғри кўрсатиш зарур.

Дастур ишини тезда тўхтатиш учун End (Тугаллаш) тугмасини босинг.

Хатолик юзага келган код сатрини очиш учун Debug (Созлаш) тугмасини босинг. Агар хато манбаи ойнаи код кўрсатиб турган сатрда бўлса, хатоликни тузатишингиз ва хатолик тузатилгач, дастурни яна ишга тушириш учун F5 тугмасидан фойдаланишингиз мумкин.





Хатолик ҳақида тўла маълумот олиш учун Help (Маълумот) тугмасидан фойдаланинг.

ActiveX бошқариш элементлари

Visual Basic нинг энг яхши имкониятларидан бири бу дастур яратиш жараёнида ActiveX элементларидан фойдаланиш мумкинлигидир. Visual Basic нинг барча версиялари ActiveX элементлари билан бирга етказиб берилади, қолган элементларни бошқа манбалардан ҳам олиб ишлатиш мумкин.

ActiveX бошқариш элементлари нима?

Агар сиз Visual Basic нинг стандарт объектлари ҳисобланувчи тугма ва ойна элементлари билан ишлаш тажрибасига эга бўлсангиз, ўз навбатида ActiveX элементлари билан ҳам ишлайоласиз. Улар асбоблар панелида жойлашган ва сиз бу элементларни шаклга жойлаштиришингиз, бошқаришингиз, хусусиятларини ўрнатишингиз ва услубларини чақиришингиз мумкин. Бу бошқариш элементлари ҳам Visual Basic нинг бошқа элементлари каби хусусиятларга, услубларга, баъзилари эса жуда ҳам кенг имкониятларга эга.

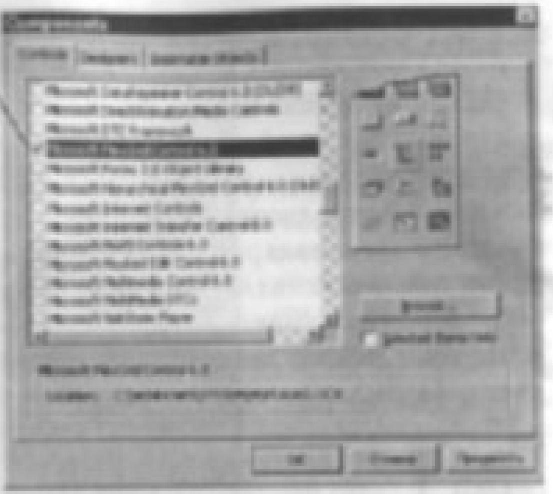
Иш жараёнида сиз системангизда Visual Basicда ишлата олмайдиган ActiveX бошқариш элементлари мавжудлигини кўришингиз мумкин. Лекин бу элементлардан фойдаланиш учун сизда рухсатнома (лицензия) бўлиши зарур. Кўпинча бу элементлар имкониятини реклама қилиш учун имкониятлари чекланган ҳолда ўрнатилади. Баъзилари эса Visual Basic билан ишлашга мўлжалланмаган бўлиши мумкин. Windows ни нотўғри ўрнатилиши ҳам баъзи элементларнинг ишламаслигига олиб келиши мумкин.

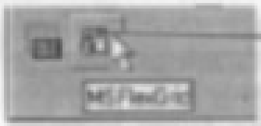
Компонентлар ойнаси

ActiveX бошқариш элементи билан ишлаш учун дастлаб уни Vizual Basic нинг асбоблар панелига ўрнатиш зарур. Бу ишни амалга ошириш учун қуйидаги ишларни бажаринг.

1. Асбоблар панелида ўнг босишни бажаринг ва калқиб чиқувчи менюдан Components (Компонентлар) бўлимини танланг.

2. Controls (бошқариш элементлари) бўлимида системангизда ўрнатилган барча ActiveX бошқариш элементларининг рўйхати акс эттирилган. Намуна сифатида Microsoft FlexGrid Control ни танланг ва OK ни босинг.





3. Асбоблар панелининг пастки қисмида ActiveX янги бошқариш элемент пайдо бўлади. Энди бу элемент Visual Basic нинг бошқа стандарт элементлари каби ишлайверади ва зарур ҳолатларда ундан фойдаланиш учун асбоблар панелига мурожаат қилинг.

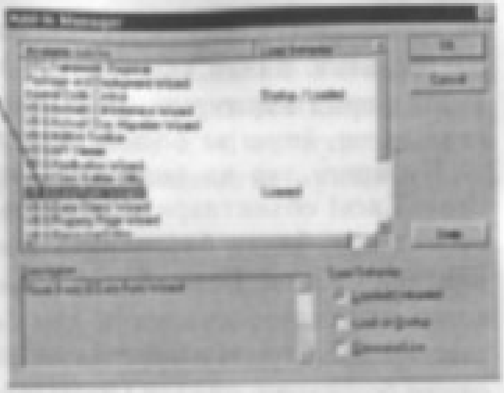
Кенгайтмалардан фойдаланиш

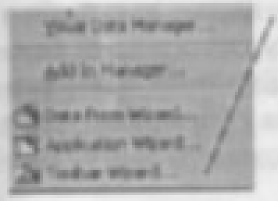
Visual Basic нинг имкониятларини кенгайтирувчи воситалардан яна бири кенгайтмалар (add-ins)дан фойдаланишдир. ActiveX элементларидан фарқли ўлароқ, кенгайтмалар дастурни бажариш чоғида қўлланилмайди. Улар Visual Basic нинг лойиҳа жараёнидаги имкониятларини кенгайтириш ва мавжудларидан унумлироқ фойдаланиш учун қўлланилади.

Кенгайтмаларни бошқариш ойнаси (Add-In Manager)

Ишлаш имкони бўлган кенгайтмалар рўйхати Visual Basic нинг системангизда ўрнатилган версиясига боғлиқ. Шу билан бирга кенгайтмаларни алоҳида маҳсулот сифатида ҳам сотиб олиш мумкин.

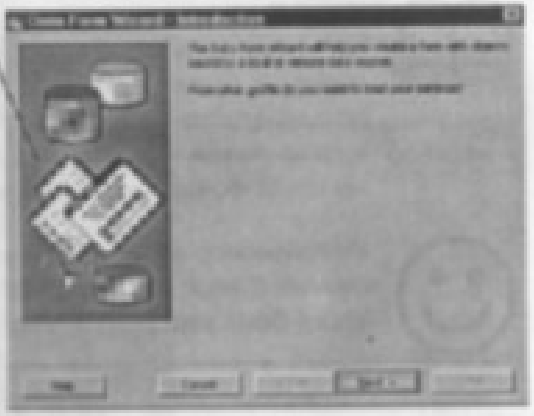
1. Add-in менюсидан Add-in Manager сатрини танланг. Ишлаш имконияти бўлган кенгайтмалар рўйхати очилади. Ўрнатиш зарур бўлганларини белгиланг.





2. OK тугмасини босингач, Add-in менюсининг пастки қисмида янги сатрлар сифатида танланган кенгайтмалар пайдо бўлади.

3. Кенгайтмадан фойдаланиш учун уни Add-in менюсидан танланг. Баъзи кенгайтмалар Visual Basic га янги панель ёки менюнинг бошқа сатрларини қўшиши, баъзилари типик вазифаларни автоматлаштириб, уста (Мастер) сифатида ишлаши мумкин.



Мурожаатларни бошқариш ойнаси

Visual Basic да яратилган баъзи бир мураккаб дастурлар бошқа дастур объектларидан ёки Visual Basicнинг ўзининг объектидек фойдаланади. Бундай дастурларга Excel жадвалига автоматик қиймат берувчи дастур мисол бўла

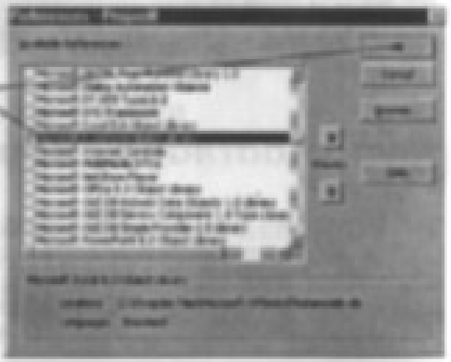
олади. Бу дастур ёрдамида Excel жадвали ячейкаларига ахборот киритиш билан чегараланиб қолмасдан, балки ячейкаларни форматлаштириш, ҳисоблашларни бажариш, иш варағини очиш, ёпиш ва босиш ишларини бажариш мумкин.

Бу ишни тез ва сифатли амалга ошириш учун Visual Basic Excel объектларининг тавсифини уларнинг барча хусусиятлари билан биргаликда ёзиб олиши зарур. Бу, шунингдек, Visual Basic Excel объектларини бошқарадиган кодни текшириш имконини ҳам беради. Объектларининг бу тавсифлари "турлар кутубхонаси" дейилади ва мурожаатларни бошқариш ойнаси (Reference Manager) орқали бошқарилади. Агар сиз Visual Basicда унинг бошқа дастурлари учун дастур тузаётган бўлсангиз, яна мурожаатларни бошқариш ойнасига мурожаат этасиз. Сиз Excelдаги каби ўзининг шахсий "турлар кутубхонасига" эга бўлган дастурлар тузишингиз ва бу кутубхоналарни мурожаатларни бошқариш ойнасига юклашингиз мумкин.

Visual Basic да дастурлашни энди ўргонаётган бўлсангиз, мурожаатларни бошқариш ойнасидан фойдаланиш мақсадга мувофиқ эмас. Бу бўлимда келтирилган ахборотдан эраттан дастурингизни Word, Excel каби дастурлар билан бирга қўшиб ишлатиш зарурати юзага келганда фойдаланишингиз мумкин.

Кенгайтмалар ва ActiveX элементлари учун бўлгани каби, мумкин бўлган мурожаатлар рўйхати системангиздаги Visual Basic версиясига боғлиқ.

Мурожаатларни бошқариш ойнаси "турлар кутубхоналарига" мурожаатни ўрнатиш ёки ўчириш учун ишлатилади. Керакли кутубхонани улаш учун белгини унинг номи ёнига ўрнатинг ва ОК ни босинг

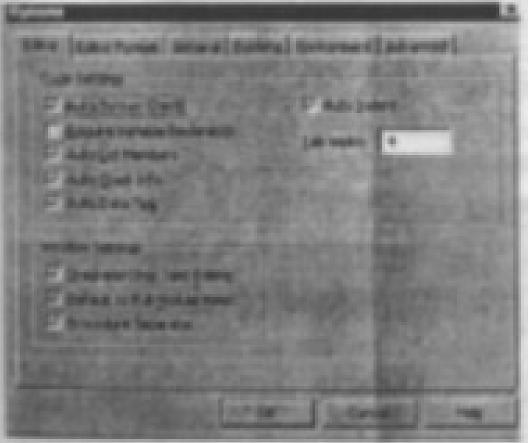


Сиз мурожаатни киритганингизда ёки ўчирганингизда бирор фарқни сезмайсиз. Мурожаатлар Сизнинг Visual Basic кодингизда нимани қилиш мумкин, нимани эса қилиш мумкинмаслигини билдиради. Масалан, Exel га мурожаатни киритган заҳотингиз ушбу сатрни ёзишингиз мумкин.

```
Dim xl As Excel.Application
```

Visual Basic параметрларини созлаш

Options (параметрлар) диалоги Tools (Сервис) менюсида жойлашган бўлиб, Visual Basicни сизга қулай қилишга имкон беради. Масалан: Editor (Муҳаррир) иловаси код муҳаррирининг қўшимча имкониятларини, масалан, Auto Syntax Ghek (Синтаксисни текшириш кабиларни бошқаради.

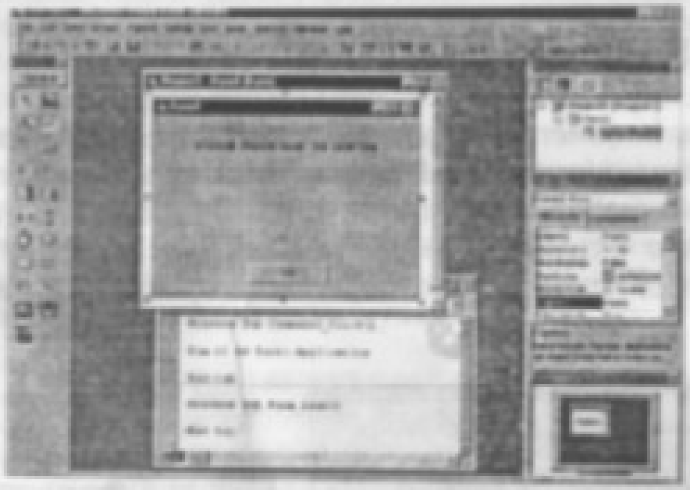


Visual Basicда ишлашингиз қулай бўладиган қилиб уни созлаш учун Options (Параметрлар) диалогидан фойдаланинг.

Visual Basicда ўрнатилган параметрлар кўпчилик холларда яроқлидир. Агар сиз бирор нарсани ўзгартирмоқчи бўлсангиз, бир галнинг ўзида кўп сонли параметрларни ўзгартирманг, шунда параметрларнинг нагори ўрнатилган қийматларига қайтиш осон бўлади.

SDI ва MDI режимлари

Advanced (қўшимчалар) иловаси Visual Basic муҳити учун энг муҳим параметрлардан бири — SDI Development Environment (бир хужжатли ишлаб чиқиш муҳити)ни ўз ичига олади. Бу параметр ўчирилган ва Visual Basic MDI режимда (кўп хужжатли) режимда ишлайди. Бу режимда ойналар мослашганда мумкин, максималлаштирилган ойна эса Visual Basic бош ойнасининг фақат ўрта қисмини тўлдиради.



SDI-режимда барча ойналар Windows ни столи бўйлаб эркин кўчиб юради. Visual Basicнинг эски версиялари фақат SDI-режимда ишлайди.

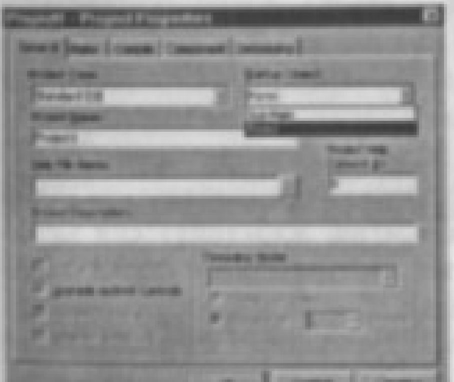
Фойдали параметрлардан яна бири Save Changes (Ўзгаришларни сақлаш)дир. У менюнинг Environment (Муҳит) бўлимида жойлашган. Бу кўрсаткични ишлатиб қўйсангиз, лойиҳага киритилган ҳар қандай ўзгаришларнинг йўқиб кетиши эҳтимоли камаяди. (Электр таъминоти узилиб қолса ва ҳ.к.)

Лойиҳа хусусиятларини ўрнатиш

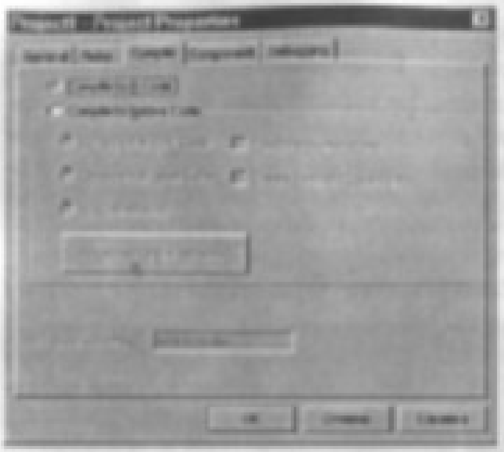
Tools>Options (Сервис>Параметрлар — кўрсаткичлар) диалог ойнаси Visual Basic ни исталган лойиҳа учун ростлаш имконини беради. Бошқа муҳим параметрларни лойиҳанинг фақат юкланган ҳолатида Project Properties (Лойиҳа хусусиятлари) диалог ойнасидан топиш мумкин. Бу диалог ойнасини менюнинг Project бўлимидан ишга тушириш ёки лойиҳалар ойнасининг номига ўнг босиш билан ишга тушириш мумкин.

Лойиҳа хусусияти

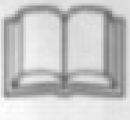
Project Properties (Лойиҳа хусусиятлари) ActiveX компонентларини яратишда ишлатилади. Шунга қарамасдан, уларнинг баъзилари умумийлик хусусиятига эга бўлиб, бу диалогнинг бу ерда сиз билишингиз керак бўлган иккита параметрини кўриб чиқамиз.



Агар сиз юкланган лойиҳа бир нечта шаклларга эга бўлса, у ҳолда Startup Object (Бошланғич объект) рўйхатидан дастур юкланаётган пайтда пайдо бўладиган шаклни танлашингиз мумкин. Бу рўйхатдаги Sub Main Visual Basic код сатри модулидаги Main (Асосий) номли жараённи қидиришни англатади ва уни биринчи навбатда бажара бошлайди, яъни сиз зарур шаклларни Main жараёни орқали акс эттиришингиз ёки шакллардан умуман фойдаланмайдиган дастур яратишингиз мумкин.



Агар сизга Visual Basic да мустақил (алоҳида) ишловчи дастур яратиш зарур бўлса, сиз уни p-код(pseudo-code стандарт чиқувчи) гоҳо ёки процессорнинг кўрсатмаларига (native code — фақат Visual Basicнинг Professional ва Enterprise версияларидагина ишлайди) компиляция қилишингиз мумкин. p-коднинг модуллари ихчамроқ, лекин секин ишлайди. Native code эса хотира жиҳатидан каттароқ, лекин тезроқ ишлайди. Бу шаклларга ва ActiveX элементларига тааллуқли эмас.



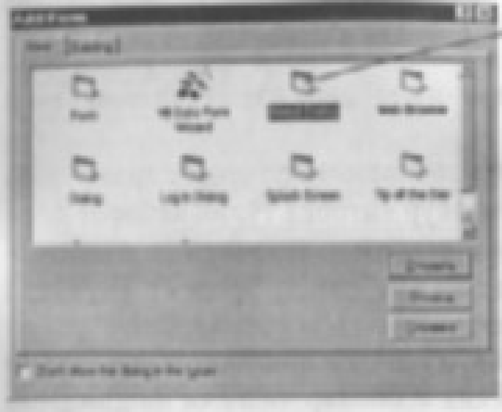
Мазкур бобнинг охиррогида Visual Basicда мустақил ишлайдиган дастурлар яратиш ҳақида батафсилроқ маълумотлар берилган.

Шакллар андозалари (шаблонлари)

Visual Basic нинг кўпгина лойиҳалари бир нечта шакллардан иборат. Одатда дастур ишга тушиши жараёнида пайдо бўладиган экран ёрлиғи (заставкаси) дастур ҳақида ахборот берувчи About (Дастур ҳақида) панели, фойдаланувчи томонидан ростланадиган параметрлар панели ва бошқа қўшимча диалог ойналаридан иборат бўлади.

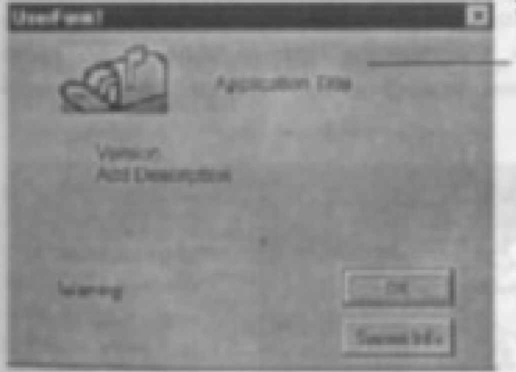
Юқорида санаб ўтилган шаклларни Visual Basic автоматик тарзда ҳосил қилиши мумкин. Улар стандарт шакллар ҳисобланади, лекин лойиҳа яратиш жараёнида сиз тайёр андазалардан фойдалансангиз, дастурлашнинг биринчи босқичи бажарилган ҳисобланади. Дастурлашга бу тарзда ёндошиш иккита афзалликка эга: биринчидан, андаза вақтингизни тежаш имконини берса, иккинчидан, яратган дастурингиз сифатлироқ ва стандарт кўринишда чиқади.

Шакл андозаларидан қандай фойдаланилади?

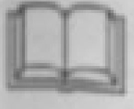


1. Менюнинг Project бўлимидан Add Form (Шаклни киритиш) сатрини танланг. Шакл андозалари рўйхати пайдо бўлади. Лойиҳага стандарт шаклни жойлаштириш учун рўйхатдан биринчи элемент Form (Шакл) ни танланг. Қолган ҳолларда андозани танланг. (Масалан) About (Дастур ҳақида) диалоги андозаси танлаб олинган.

Add Form (Шаклни киритиш) диалогидаги шакллар рўйхати ҳам системангизга ўрнатилганVisual Basic версиясига боғлиқ.



2. Шаклнинг андозаси пайдо бўлгач, зарур объектни танлайсиз ва таҳрир қиласиз. Шаклнинг экранда кўриниши учун эса шаклнинг Show (Кўрсатиш) усулидан фойдаланинг. Масалан, frmAbout шаклини қуйидагича чақиринг:
frmAbout. Show
Ёки дастурни давом эттириш учун фойдаланувчи шаклни ёпиши керак бўлса, frmAbout. Show vbmodal дан фойдаланинг.



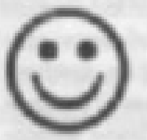
Агар андозани танлаш рўйхати очилмаса, Tools> Options>Environment (Сервис>"Муҳит">"Параметрлар" диалог ойнасида Form Templates (Шакл андозалари) параметр белгиланмаган бўлиши мумкин.

Лойиҳа андазалари

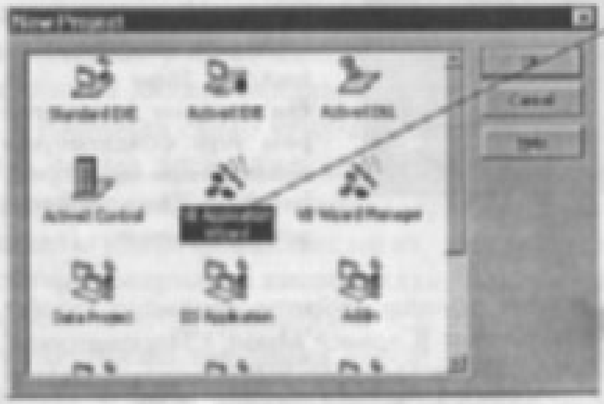
Лойиҳа андазаларининг шакл андазаларидан фарқи шуки, лойиҳа андазалари шакл андазасига ўхшаб шакл ҳосил қилмайди, балки шакл учун турли стандарт параметрларни ўрнатади.

ActiveX бошқариш элементларининг турли параметрларини ўрнатиш тўғрисида мазкур китобда тўлиқ ахборот бера олмаймиз. Лекин шаклнинг асосий ва About диалог ойнасини яратувчи дастур устаси (Application Wizard) ўқувчида маълум қизиқиш уйғотади.

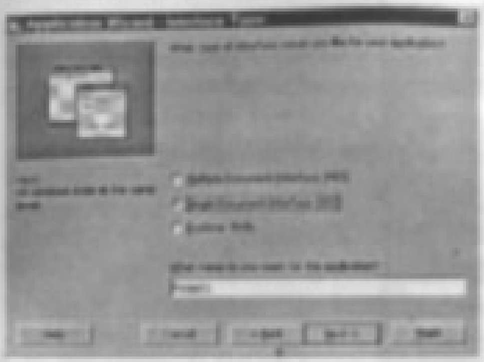
Бу уста сиз учун унча кўп иш бажармайди, асосий ишни ўзингиз бажарасиз. Агар сиз асбоблар чизғичи ишончларидан биттасини ёки меню сатрларидан бирортасини боссангиз "...code goes here" (код шу ерга ёзилади) қабилида ахборот оласиз. Шунинг учун Visual Basic билан ишни бошлашда энг осон лойиҳа андазаси сифатида Standard EXE (стандарт EXE-файл) дан фойдаланинг.



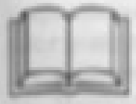
Лойиҳа андазасидан умуман фойдаланмаслигингиз мумкин, чунки лойиҳанинг барча параметрларини Visual Basic нинг бошқа диалог ойналарида танлаб олиш мумкин.



1. Дастур устасини (Файл) ишга тушириш учун менюнинг File бўлимидан New Project (янги лойиҳа) сатрини танланг. Кейин VBApplication Wizard (VB дастурлари устаси) мулоқот ойнасини танланг.



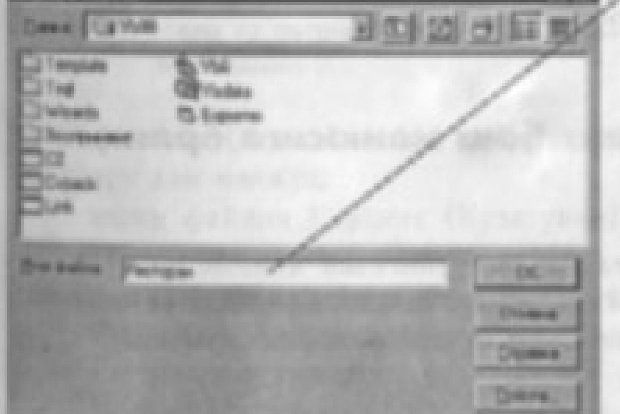
2. Лойиҳа учун зарур параметрларни ўрната бориб, кейинги диалог ойнасига ўтиш учун Next (Кейинги) тугмасини босинг. Охирги қадамда Visual Basic дастур яратиши учун Finish (Тугаллаш) тугмасини босинг.



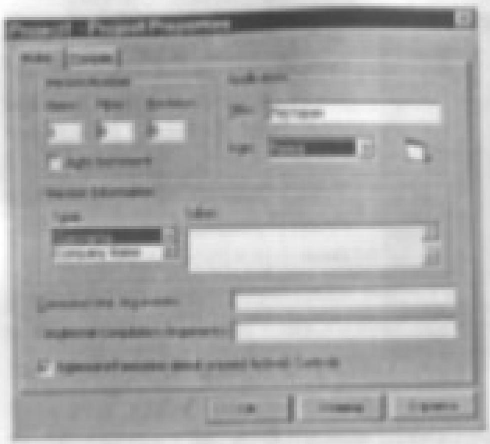
Агар диалог ойналарида пайдо бўлган параметрлар қийматлари мазмунини билмасангиз, уларни ўрнатилган ҳолатда қолдирган маъқул.

Мустақил ишловчи (автоном) дастур яратиш

Visual Basic да дастур яратиш жараёнида бошқа фойдаланувчилар учун ҳам ишлайдиган дастур ярата бошлайсиз. Бундай ҳолатларда Setup Wizard устасидан фойдаланиб, ўрнатиладиган дастур мажмуасини яратиш зарур, у 7-бобда тасвирланган.



1. Мустақил ишлайдиган файл яратиш учун лойиҳани юкланг ва менюнинг File (Файл) бўлимида Make.EXE сатрини танланг. Бажариладиган файл номини ёзинг ва зарур каталогга жойлаштиринг.



2. Лойиҳа хусусиятларини ўзгартириш учун менюнинг Make Project (Лойиҳа яратиш) қисмидан Project Properties (Лойиҳа хоссалари) ни танланг ва Options (Параметрлар) тугмасини босинг. Бу ерга муаллифлик ҳуқуқлари, дастур версияси ва бошқа маълумотларни киритиш мумкин. Compile (Компиляция) сатрида сиз p-код компиляция усулини ёки бевосита процессор йўриқларини танлашингиз мумкин.

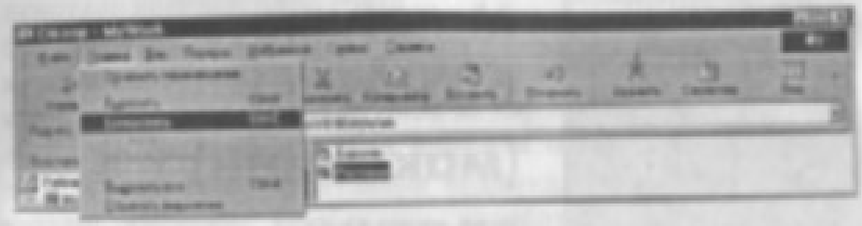
3. Барча ростлаш ишлари оҳирига етказилгач, Make EXE диалог ойнасига қайтинг ва OK тугмасини босиб, файл яратинг. Энди Visual Basic дан чиқиш ва яратилган файлни Windowsнинг бошқа ҳар қандай дастури каби ишга туширишингиз мумкин.

Агар дастурингизда шакллардан бирортаси учун махсус нишонлардан фойдалансангиз, Project Properties (Лойиҳа хоссалари) диалог ойнасида Icon кўрсаткичини зарур шакл учун ўрнатиб қўйинг. Бу нишон Windows Explorer (Кузатувчи)да дастур нишони сифатида кўринади. Visual Basic билан бирга нишонлар тўплами ҳам ўрнатилади, бу нишонлардан дастурингизда фойдаланишингиз мумкин.

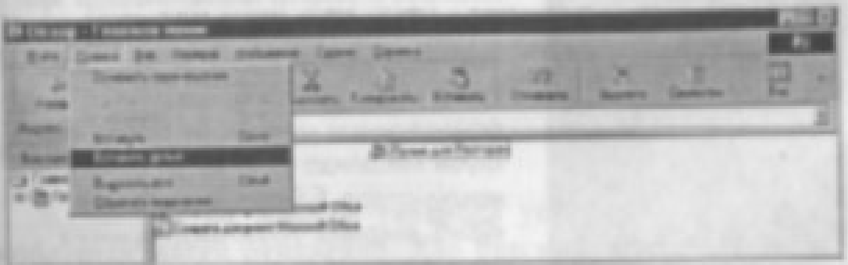
Windows нинг бош менюсига ёрлиқ қўшиш

Яратган мустақил дастурингизни компьютерда тез топиш учун, уни Windowsнинг асосий менюсига (Startmenu) ни қўшиб қўйинг. Буни бундай бажариш мумкин.

1. Windows Explorer (Кузатувчи) ни ишга туширинг. Яратган дастурингизни сақлаган каталогга киринг ва Explorer менюсининг Edit (Тузатиш) бўлимидан Copy (Нусха кўчириш) сатрини танлаб, уни кўчиринг.



2. Windows нинг масалалар панелида (Taskbar) Start менюсини танланг, сичқонча ёрдамида ўнг босиш қилинг ва менюдан Explorer сатрини танланг. Explorer (Кузатувчи) ойнасида Start Menu (Асосий ойна) ойнаси очилади, файл ёрлиқ жойлаштирилиши зарур бўлган жойда Paste Shortcut (Ёрлиқ қўйиш) сатрини танланг.



Setup Wizard устаси ёрдамида ўрнатиладиган дастур мажмуини ҳосил қилишда, дастурингиз ёрлиғини Start menu га автоматик тарзда жойлашадиган қилиб яратиш мумкин.

Start menu (Асосий меню) га ёрлиқ қўшишнинг бошқа усуллари ҳам мавжуд:

- ишчи файлни Explorer (Кузатувчи) ёрдамида Пуск (Start) менюсига ўтказиш;
- масалалар панелига ўнг босиш қилиб, менюдан Properties (Хусусиятлар) сатрини танланг. Start Menu Programms (менюни созлаш) ни танланг ва Add (қўшиш) тугмасини босинг.

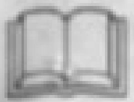
БЕШИНЧИ
БОБ

Маълумотлар базасини (маконини) яратиш

Visual Basic маълумотларни бошқариш учун энг қулай восита ҳисобланади, чунки у сиз учун энг оғир ишларни бажарувчи дастур элементларига эга. Visual Basic томонидан сақланган маълумотлардан Microsoft Access ва Word каби дастурлар ҳам фойдаланиши мумкин. Бу бобда спорт клуби мисолида унча катта бўлмаган маълумотлар базасини яратишни кўриб чиқамиз. Маълумотларингиз асосида ҳисобот тайёрловчи Crystal Reports дан қай тарзда фойдаланишни ҳам билиб оласиз.

Маълумотлар базасига кириш

Маълумотлар базаси — маълум турдаги ахборот ёки маълумотлар йиғиндисидир. Матн процессори ҳужжатидаги рўйхат ёки жадвал ҳам маълумот базасига мисол бўла олади. Лекин маълум миқдордаги ҳажмга етгач, бундай рўйхатларни бошқариш қийинлашади. Visual Basic да ҳар қандай ҳажмдаги маълумотлар базаларини қайта ишловчи оддий воситалар мавжуд. Visual Basic воситалари ёрдамида маълумотни қайта ишлаш, қидириш, ўзгартириш, ҳисобот тайёрлаш, уни чоп қилиш каби ишларни бажаришингиз мумкин. Маълумотлар базаси билан ишлайдиган дастурчилар маълум атамаларни яратишган. Улардан баъзилари билан танишайлик.



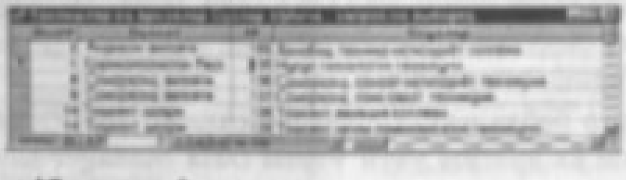
Visual Basicда маълумотлар базаси билан ишлашни ўрганишнинг энг яхши йўли — маълумотлар базасини яратишдир. Шунинг учун мазкур бобда унча катта бўлмаган маълумотлар базасини яратасиз.

Жадвал

Жадвал — бу маълум тартибда устун ва майдонларга жойлаштирилган рўйхатдир. Ҳар бир майдон одатда белгиланган ўлчамга эга.

Маълумотлар базасини яратишда Windows қобиғида ишлайдиган исталган дастурда яратилган базадан фойдаланиш мумкин.

Microsoft Access томонидан яратилган жадвал.



Ёзувлар (Сатрлар)

Ёзув — бу рўйхатнинг бир сатридир. Баъзан жадваллар тўрва рўйхат кўринишида чиқарилмай, балки сатрларга ажратилган ҳолда ташкил қилинади.

Маълумотлар базаси

Маълумотлар базаси маълум кетма-кетликда жойлаштирилган жадваллар мажмуасидир. Масалан, сизда мижозлар учун бир жадвал, уларнинг буюртмалари учун эса иккинчи жадвал бўлиши мумкин. Баъзи маълумотлар базалари битта жадвалдан иборат бўлади.

Тузилмалаштирилган сўровлар тили (SQL)

Visual Basic да маълумотлар базаси билан ишлаш жараёнида доимо SQLга мурожаатга дуч келасиз. SQL — маълумотларни янгилаш ва сўровлар яратиш учун қўлланиладиган махсус тилдир. Бошланғич босқичда SQL ни ўрганиш унча ҳам зарур эмас, лекин ўқувчи SQL нинг фақат Visual Basic да эмас, балки умуман дастурлашда маълумотлар базаси билан ишлашда кенг қўлланилаётганлигини билиб қўйгани фойдадан холи эмас.

Visual Data Manager дан қандай фойдаланилади

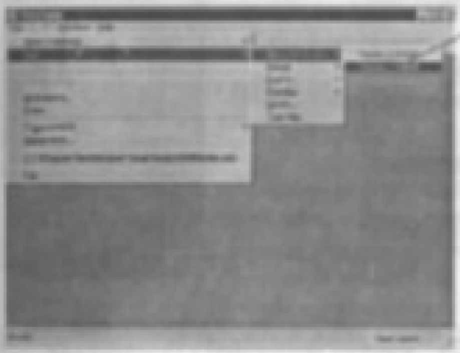
Visual Data Manager (Маълумотлар визуал диспетчери) — бу Visual Basic кенгайтмасидир. Уни сиз Add-Ins (Кенгайтмалар) менюсидан топасиз. Бу сервис дастури Visual Basic да ёзилган бўлиб, маълумотлар базасини яратиш ва уни бошқариш учун қўлланилади. Сиз бу элементни Microsoft Accessнин соддалаштирилган версияси сифатида қабул қилишингиз мумкин.

Мазкур бобда маълумотларни сақлайдиган файлни қайта ишлаш ва яратиш учун Visual Data Manager дан фойдаланишни билиб оласиз. Visual Data Manager ёрдамида спорт клуби маълумотлар базасини яратасиз. Уни манзиллар китоби, кутубхона маълумотномаси ёки исталган бошқа маълумотлар тури учун тез ўзгартириш мумкин.

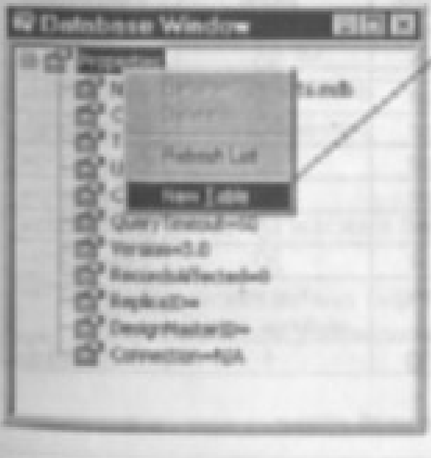
Visual Data Manager Visual Basic нинг Professional ёки Enterprise версиялари билангина қўйилади. Агар сизда Standard версияси ўрнатилган бўлса, сиз маълумотлар базасини ярата олмайсиз, лекин мавжуд базалар билан ишлаш имкониятига эга бўласиз. Маълумотлар базасини яратишнинг энг қулай йўли Microsoft Accessдан фойдаланишдир.

Агар сизда ишлаш учун маълумотлар базаси бўлса, мазкур бобнинг "Маълумотлар билан ишлаш учун шакл яратиш" бўлимига ўтинг.

1. Windows Кузатувчи (Explorer)идан фойдаланиб маълумотлар базаси учун C:\MYDATA номли (бу мисолда шундай номланди) янги каталог яратинг.



2. Менюнинг Add-Ins (Кенгайтмалар) қисмидан Visual Data Manager ни, файл менюсидан New>Microsoft Access>Version 7.0 MDB (Янги>Microsoft Access>Version 7.0) сатрини танланг. File Selection диалог ойнаси пайдо бўлади. Мазкур диалог ойнасидан C:\MYDATA каталогини топинг ва маълумотлар базаси файлига SPORTS.MDB деб ном беринг.



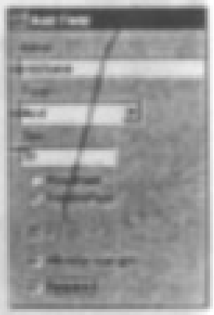
3. Маълумотлар базасининг (Database Window) ойнасида Properties (Хусусиятлар) сатрига ўнг босиш қилсангиз, қалқиб чиқувчи меню пайдо бўлади. Бу ерда New Table (Янги жадвал) сатрини танланг. Маълумотлар базасини яратишни давом эттириш учун кейинги бўлимга ўтинг.

Жадвални ишлаб чиқиш

Жадвалнинг тузилишини аниқлаб олиш вақти келди. Маълумотлар базасини яратишда бу қадам масъулиятли ва муҳимдир, чунки жадвалнинг ташкилий тизими унда сақланадиган маълумот турини белгилайди.

1. New Table [Янги жадвал] сатрини танлаганингизда Table Structure [Жадвал таркиби] диалог ойнаси пайдо бўлади. Table Name [Жадвал номи] ойнасида Members [Аъзолар] ёзувини ёзинг.

2. Add Field [Майдон қўшиш] тугмасини босинг. Name [Ном] ойнасида LastName [Фамилия] ни теринг. Type [Тур] рўйхатидан Text [Матн] сатрини танланг. Size [Ўлчам] ойнасида 30 ни киритинг. Required байроқчаси [Бўш бўлмаган мажбурий қиймат]ни белгиланг.



3. OK тугмасини боссангиз, янги бўш майдон жадвалга қўшилади. Лекин Add Field диалог ойнаси очиқ қолади, зарур бўлса, яна қўшимча майдон қўшиш мумкин. Бу ойна — Close [Ёпиш] тугмасини босгунча актив ҳолатда бўлади. Қуйида сиз киритишингиз керак бўлган майдонларнинг тўлиқ рўйхати келтирилган.

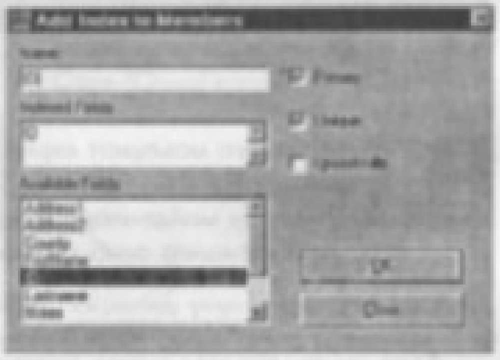
Name	Type	Size	Required
Last Name(Фамилия)	Text	30	Ёқилган
FirstName(Исм)	Text	30	Ёқилмаган
Address 1 [Манзил1]	Text	50	Ёқилмаган
Adress 2 [Манзил2]	Text	50	Ёқилмаган
Town(Шаҳар)	Text	50	Ёқилмаган
Country(Мамлакат)	Text	50	Ёқилмаган
Postcode(Почта индекси)	Text	15	Ёқилмаган
Telephone(Телефон)	Text	30	Ёқилмаган
Notes(Эслатма)	Memo	Кириш мумкин эмас	Ёқилмаган
ID (Идентификатор)	Long	4	Ёқилган *

*AutoIncrField байроқчасини ҳам белгилаб қўйинг.

Агар системангизда Microsoft Access ўрнатилган бўлса, сиз уни Visual Data Manager ўрнига жадвал яратиш учун қўллашингиз мумкин. Microsoft Accessнинг Visual Data Manager га нисбатан қулайлиги шундан иборатки, у кириллица шрифтини қўллайди.

ID майдони учун AutoIncrField байроқчасини белгилашни унутманг.

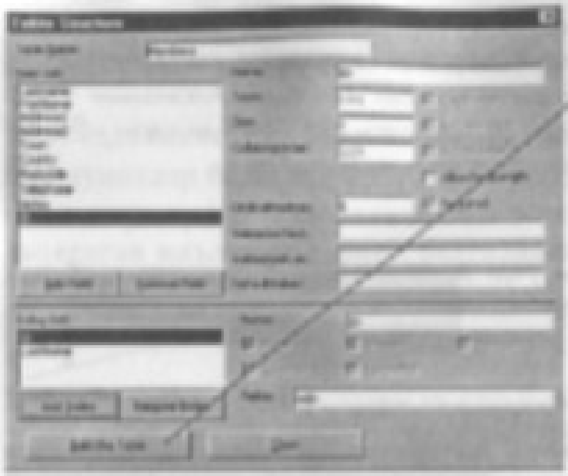
4. Янги жадвалнинг индексларини қўшиш учун Add Index [Индекс қўшиш] тугмасини босинг. Индекслар маълумотларни саралаш ва қидиришни осонлаштиради. Сиз қуйидаги икки индексни қўшишингиз зарур.



Name (Исм)	Indexed Fields (Индексланган майдонлар)	Primary (Бирламчи)	Unique (Ўзига хос)
ID	ID	Ёқилган	Ёқилган
LastName (Фамилия)	LastName (Фамилия)	Ўчирилган	Ўчирилган

Ниҳоят, Close тугмасини босиб Add Index диалогини ёпинг

Бу диалогда яна бир параметр IgnoreNulls бўлиб, уни ўрнатилган (Ёқилмаган) ҳолатида қолдиринг.



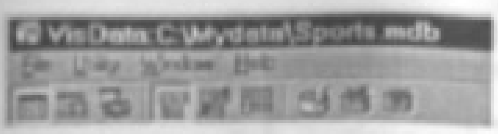
5. Бу расмда Table Structure (Жадвал тузилиши) ёрдамида олинган диалог ойнасининг натижасини кўрасиз. Энди Build the Table (Жадвални тузиш) тугмасини босиб, берилган хусусиятлар асосида жадвал тузинг.

Майдонга маълумот киритилганидан кейин майдон тугмасини (Remove Field тугмасини) босмаслигингиз лозим. Чунки бу майдонлардаги ҳамма маълумотларнинг ҳам ўчиб кетишига олиб келади.

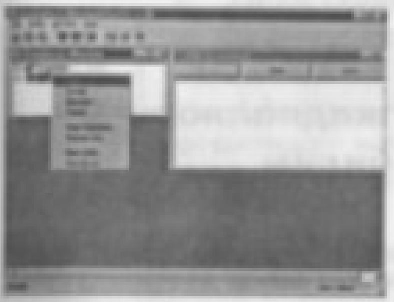
Агар жадвалга маълумот киритиш жараёнида хатоликка йўл қўйилган бўлса, хатоликка йўл қўйилган майдонни ажратиб, унинг баъзи хусусиятларига ўзгартириш киритиш мумкин. Масалан, Required (Бўш бўлмаган мажбурий қиймат). Баъзи хусусиятлар, масалан, Size (Ўлчам) хусусияти эса соҳани ўчириб, янгисини қўйиш йўли билан ўзгартирилади.

Маълумотларни киритиш

Ҳозирча сиз яратган маълумотлар базаси бўш. Мазкур бўлимда Visual Data Manager ёрдамида базага маълумот киритиш ва уни ўзгартиришни ўрганамиз.



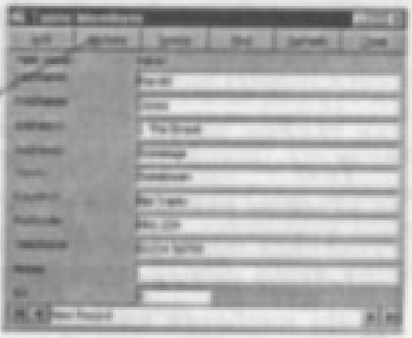
1. Visual Data Manager ни ишга туширинг. Асбоблар чизиғида Table Type Recordset (Жадвал кўринишидаги ёзув) ва Use Data Control on New Form (Янги шаклда Data бошқарув элементидан фойдаланиш) нишонлари босилган ҳолатдалигини текшириб кўринг.



2. Менюнинг File>Open (Файл>Очиш) қисмидан фойдаланиб, SPORTS.MDB маълумотлар базасини очинг. Members сатрини ўнг босиш қилиб Open (Очиш) сатрини танланг.

Жадвалга ўзгартириш киритиш учун унинг номини сичқонча ёрдамида икки босиш қилиб очиш мумкин.

3. Жадвал бўш бўлгани учун шаклнинг пастки қисмида New Record (Янги ёзув) ахбороти пайдо бўлиб у жорий ёзув янги эканлиги ҳақида хабар беради. Мисол тариқасида шаклга маълумотлар киритинг. Эътибор берган бўлсангиз, ID тартиб рақамини киритишга зарурат йўқ. Update (Янгилаш) тугмасини босинг. Буйруқни тасдиқлашни сўровчи шакл пайдо бўлади. Янги ёзувни сақлаб қолиш учун Yes (Ха) тугмасини босинг.



4. Visual Basic ёзувнинг тартиб рақамини автоматик белгилайди. Жадвалга маълумот киритилгач, шаклнинг пастки қисмида 1/1 ёзуви пайдо бўлади.

Сиз бошқа ёзувлар ҳам қўшишингиз мумкин. Бунинг учун Add (Қўшиш) тугмасини, зарур маълумот киритилгач эса Update тугмасини босишингиз зарур.

Маълумотлар базаси жадваллари ҳақида батафсилроқ билим

Visual Data Manager билан амалий ишлаш жараёнида маълумотлар базаси жадвалларининг баъзи асосий хусусиятлари ҳақида билиб олдингиз.

Майдон тури (Field Type)

Visual Basic ўзгарувчилари каби маълумотлар базаси жадвали майдони ҳам маълум турга эга. Жадвал майдонлари турини сақланиши зарур бўлган маълумотлар турига асосан танлаш зарур. Энг муҳим турлар қуйидагилардир:

Text — белгилар сатри, у, масалан, исмлар, манзиллар ва телефон рақамлари учун; Сиз сатрни 255 белгигача тенг узунликда беришингиз мумкин.

Integer ва *Long* — бутун сонлар учун. Integer тури Long турига нисбатан қисқароқ ва 32767 гача бўлган сонларни сақлаши мумкин. *Long* турдаги майдонлар — қўшимча AutoIncrement (Қийматни автоматик ошириш) хусусиятига ҳам эга. Агар бу байроқ ёқилган бўлса, жадвалга ёзиладиган ҳар бир ёзувга навбатдаги сатр рақами берилади.

Memo — узунлиги 65535 белгигача бўлган сатрлар учун. Бу тур учун сатр узунлигининг маълум бир қийматини кирита олмайсиз.

Boolean — True ва False қийматлари учун.

Single ва *Double* — силжувчи нуқтали сонлар учун.

Double тури — Single турига нисбатан каттароқ сонларни ва каттароқ аниқликда сақлаш мумкин.

Currency — пул қийматлари учун.

Date / Time — сана ва вақт қийматлари учун.

Бу ерда келтирилган маълумотлар турлари Microsoft Access ва Visual Basic MDB форматидаги жадвалларда қўлланилади.
Visual Basic да яна бошқа турдаги, масалан, dBase форматидаги маълумотлар жадвалларидан ҳам фойдаланишингиз мумкин.

Майдонларнинг бошқа хусусиятлари

Икки майдон, яъни, ID ва Last Name учун сиз Required байроқчасини белгиладингиз. Бунинг мазмуни шундан иборатки, жадвалдаги ёзувлар шу майдонларда, албатта бирор қийматга эга бўлиши зарур.

Бирламчи калит (Primary Key)

ID майдони Primary (Бирламчи) ва Unique (Ноёб) кўрсаткичлари билан индексланган. Шу билан бирга унга AutoIncrement (Қийматни автоматик орттириш) хусусияти ҳам берилган. Бу комбинациянинг асосий мазмуни шундан иборатки, жадвалдаги ҳар бир ёзув ID кўрсаткичига қараб сараланади. Бу жуда муҳим имконият бўлиб, яхши ишлаб чиқилган маълумотлар базаларига хосдир.

Primary Key (Бирламчи калит) — бу ўзига хос атама бўлиб, маълумотлар базаси билан ишлашда жадвалга ёзиладиган ҳар бир индексни белгилаш учун ноёб белги сифатида қўлланилади.

Data бошқариш элементи билан танишамиз

Жадвалдаги маълумотларни шаклга солиш Visual Basic да Data Control (маълумотлар — бошқариш элементи) объекти ёрдамида амалга оширилади. Data бошқариш элементи исталган жадвал ёки сўров асосида ташкил қилиниши мумкин.

Сиз ўзингиз учун зарур шаклни яратаётган вақтингизда Data бошқариш элементи хусусиятларини ўрнатасиз, шу билан сиз шакл мурожаат қиладиган жадвал, жадвалдан олинадиган маълумот турларини белгилайсиз.

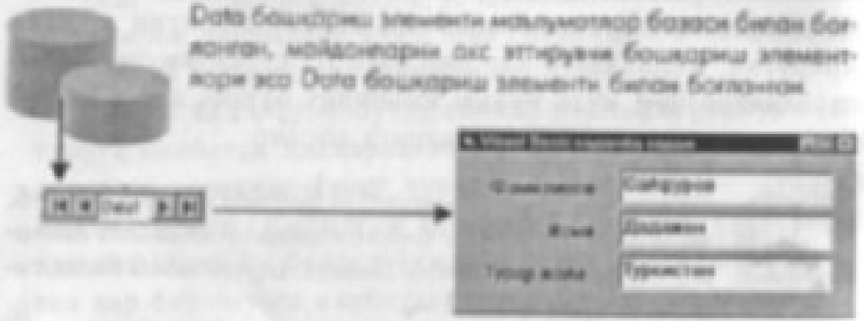
Кейинги босқичда сиз шаклга бошқа бошқариш элементларини, масалан, матн ойналарини ёки рўйхат ойналарини жойлаштирасиз. Матн ойнаси Data Source (Маълумот манбаи) хусусиятига эга бўлиб, сиз уни Data Control объекти билан боғланадиган қилиб белгилашингиз мумкин.

Матн ойнасининг — DataField (маълумот майдони) хусусияти қайси майдонни акс эттириш зарурлигини белгилайди. Юқоридаги қадамлар бажарилгач, шакл ишга туширилади, шаклда сўралган маълумот пайдо бўлади.

Data бошқариш элементи жадвалда олдинга ёки орқага юришни бажарадиган боғланган тугмаларга эга. Уларни боғланган дейилишига сабаб, тугмани босилганда, мос равишда жадвалда бир сатр олдинга ёки орқага силжиш бўлади. Боғланган элементларда кўрсатилган майдонлар бундай кўчишда автоматик янгиланади.

Кейинги бўлимда Data бошқариш элементининг қандай ишлаши кўрсатилган.

Data бошқариш элементи маълумотлар базаси билан боғланган, майдонларни акс эттирувчи бошқариш элементлари эса Data бошқариш элементи билан боғланган.

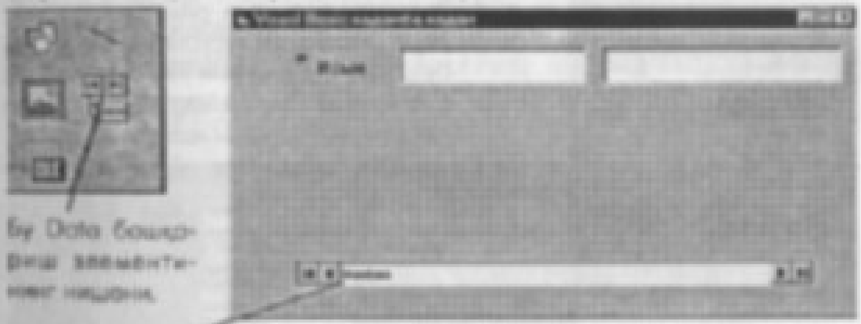


Сўров нима?

"Сўров" (query) ҳам ўзига хос атама бўлиб, маълум турдаги жамланган маълумотлар наборини акс эттиради ва у маълумот базасига қилинган мурожаат натижасидир. Масалан, маълумотлар базасидаги "Smith" фамилиялари бор бўлган барча ёзувлар қизиқтириши мумкин бўлиб, уни кўриб чиқиш зарурати туғилди. Бундай ҳолатларда Data элементи жадвални эмас, балки сўров шартларини акс эттирадиган қилиб ростланади. Сўров шаклини бир вақтнинг ўзида маълумотларни бир нечта жадвалдан оладиган қилиб ҳам ростлаш мумкин, демак, сўровлар етарлича мослашувчан элементлардир.

Маълумотлар билан ишлаш учун шакл яратиш

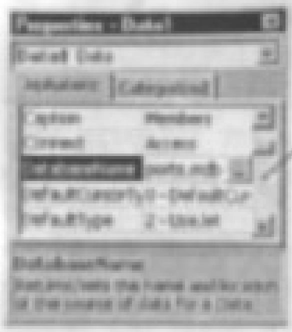
Қуйидаги мисолда олдинги бўлимларда яратилган спорт клуби маълумотлар базасидан фойдаланамиз.



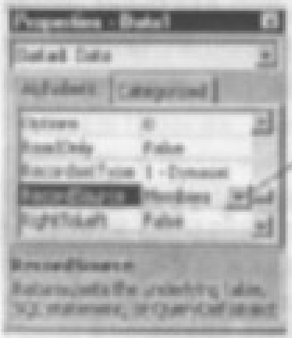
Бу Data бошқариш элементининг нишони.

1. Янги шакл ҳосил қилинг, унга Data бошқариш элементи, иккита матн ойнаси ва ёзув жойлаштиринг. Матн элементи сарлавҳасига "Исм", Data бошқариш элементи сарлавҳасига эса "Members" деб ном беринг.

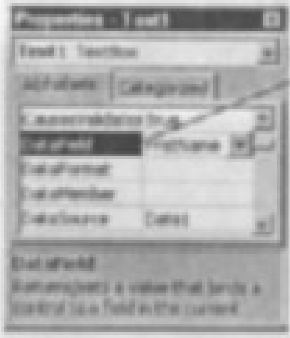
Маълумотлар базаси устаси (Data Form Wizard) Visual Basic да автоматик тарзда шакл ҳосил қилувчи элементдир. Бундай восита мавжудлигига қарамасдан, сўров шаклини яратишни, у қандай ишлашини билиб олиш учун бошидан бошлаб ўрганган маъқул.



2. Data бошқариш элементини танланг ва унинг хусусиятларини кўриб чиқинг. DatabaseName (маълумотлар базаси номи) сатридаги уч нуқтали тугмани босинг ва уни Sports маълумотлар базаси билан боғлаш учун SPORTS.MDB файлини танланг.



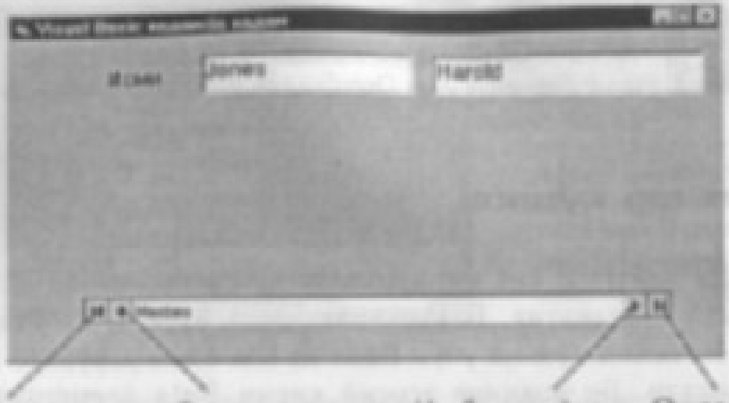
3. Data бошқариш элементининг RecordSource (Ёзув манбаи) хусусияти ёнида очилувчи рўйхат стрелкаси бор, уни босинг ва ундан Membersни танланг.



4. Энди ҳар бир матн ойнасини танлаб, унинг DataSource хусусиятини Data1 га, DataField хусусиятини FirstName га ва LastNameни DataFieldга мос равишда ўзгартиринг. Шаклни ишга туширсангиз, шу майдонлар матн ойнасида пайдо бўлади.

Маълумотлар шаклини такомиллаштириш

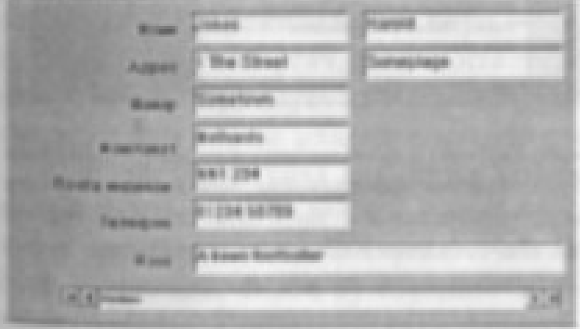
Сиз яратган маълумотлар шакли баъзи имкониятларга эга. Members жадвали бўйлаб олдинга ва орқага юрсангиз, матн ойналарида мос равишда спорт клуби аъзосининг исми кўринади.



Майдонлар қўшиш

Шаклга янги майдонлар қўшиш учун уларга мос ёзув ва матн ойналарини жойлаштиринг. Матн ойналарининг DataSource (Маълумотлар майдони) хусусиятини Data элементи билан, FieldName (Майдон номи) хусусиятини эса керакли майдон билан боғланг.

ID майдонини шаклда кўрсатиш шарт эмас, чунки бу майдон одатда фойдаланувчига керак эмас.



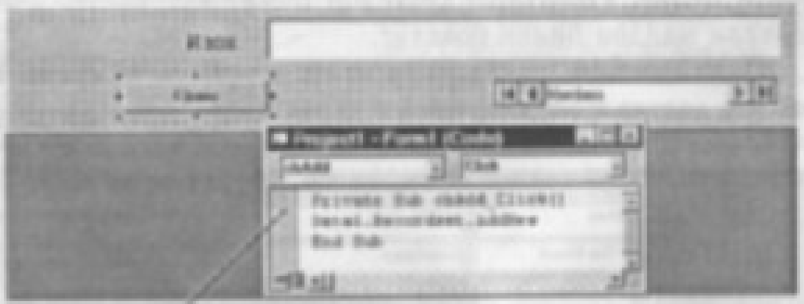
Қўшимча боғланган матн ойналарини қўшиб, шаклга Members жадвалининг барча майдонларини чиқариш мумкин. Notes (Изоҳ) майдонининг MultiLine хусусияти True ҳолатга, ScrollBars хусусияти Vertical (Вертикал) ҳолатга ўтказилса, ойнада кўп сатрли матнларни кўриш имконияти вужудга келади.

Вақтни тежаш учун шаклда бир вақтнинг ўзида бир нечта матн ойнасини танлашингиз ва бир мартада уларнинг барчаси учун DataSource хусусиятларини ўрнатишингиз мумкин.

Ёзувлар қўшиш

Шу пайтгача сиз маълумотлар шаклини коддан фойдаланмай яратдингиз. Баъзи ҳолатларда ёзувларни ўчириш ёки янгисини қўшиш учун Basic да код ёзиш зарурати юзага келади. Бу коднинг асосий қисми Data элементининг Recordset (Ёзувлар тўплами) хусусиятидан фойдаланади. Бу хусусият жорий ёзувлар тўпламини акс эттирувчи объект ҳисобланади. Қурилаётган мисолимизда эса у бутун Members жадвалидир, лекин баъзи ҳолатларда бу объект жадвал эмас, балки сўров натижаси ҳам бўлиши мумкин.

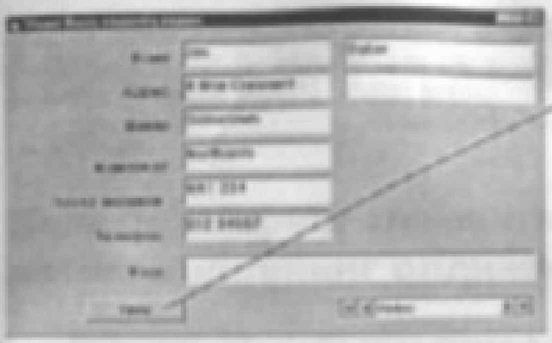
Ёзув қандай қўшилади?



1. Шаклнинг пастки қисмига бир нечта тугма жойлаштириш учун жой қўшишингиз керак. Шаклга битта тугмани жойлаштиринг. Тугмани босинг, Унга cbAdd, Caption (Сарлавҳа) хусусиятига эса "Янги ёзув" деб ном беринг. Кейин эса тугманинг Click ҳодисаси учун ушбу кодни ёзинг:

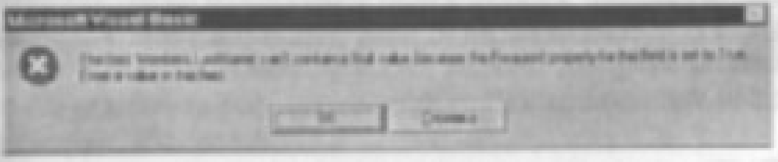
Data1.Recordset.AddNew

Агар шаклга тугмалар жойлаштиришни хохламасангиз, унга меню жойлаштиришингиз мумкин.



2. Шаклни ишга туширгач, янги тоза ёзув ҳосил қилиш учун тугмасини босинг. Унга маълумот киритинг. Кейинги ёзувга ўтиш тугмасини босишингиз билан Data бошқариш элементи маълумотни сақлайди.

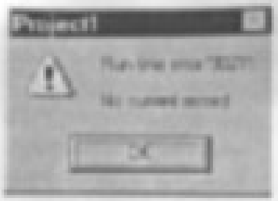
LastName майдонига маълумот киритмасдан, "Янги ёзув" қўшмоқчи бўлсангиз, Visual Basic хатолик ҳақида хабар беради. Бунинг асосий сабаби, LastName майдони учун жадвални яратишда Required байроқчаси ёқилган эди.



Ёзувни ўчириш

Ёзувни ўчириш услуби янги ёзувни қўшиш услубига ўхшаб кетади, лекин бу ерда айрим муаммоларни ҳисобга олиш керак. Муаммолардан бири — тасдиқлашдир. Сиз ёзувни ўчириб юборгач, уни қайта тиклай олмайсиз. Шунинг учун фойдаланувчи "Ўчириш" буйруғини бергач, уни тасдиқлашни сўраш мақсадга мувофиқдир.

Бошқа муаммо унча аҳамиятга молик бўлмаса ҳам фойдаланувчи учун ноқулайлик туғдириши мумкин. Recordset хусусиятининг Delete (Ўчириш) тугмаси босилгач, жорий ёзув ўчирилади, бироқ бу ёзувнинг бўш майдонини акс эттириб туради. Ўчирилгач, ёзув олдинга ёки орқага бошқа ёзувга ўтилгандагина акс этишдан тўхтайди. Демак, бу муаммони ҳал этиш учун ўчириш амали бажарилгач, ўчирилган ёзувдан кейинги ёзувни чақирувчи кодни ёзиш зарур.



Ёзувни ўчирганингиздан кейин ҳам шакл хато тўғрисида уни акс эттириб турса, Visual Basic "No current record" (Жорий ёзув йўқ) кабилида ахборот беради.

Ёзувни ўчириш коди

Қуйида "Ўчириш" тугмасининг Click (Босиш) ходисаси учун код келтирилган:

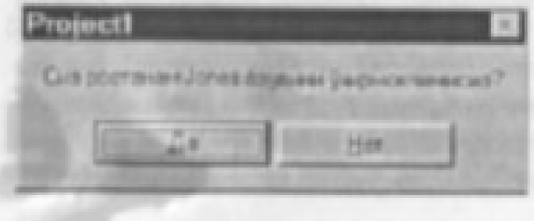
```
With Data1.Recordset
If MsgBox ("Ҳақиқатда ўчирайми"+ –
    .Fields ("LastName")+ "?", vbYesNo)=vbYes Then
    .Delete' Ёзувни ўчириш
    .MoveNext ' Кейинги ёзувга ўтказиш
    If.EOF Then.MoveLast 'Охирги ёзувга ўтиш
End If
End With
```

With операторига эътибор қилинг, у бу блокка кирувчи барча хусусиятлар ва услублар Data1.Recordset объектига алоқадорлигини англатади.

Майдон қийматини қайтариш

Олдинги кодда LastName майдонининг қиймати ёзувни ўчиришни тасдиқловчи диалогга ёзилган эди. Фойдаланилган ифодага эътибор беринг:

```
Data1.Recordset.Fields("LastName")
```



"Ўчириш" буйруғини тасдиқловчи диалог ойнаси.

Нима учун бу код хатоликларни юзага келтиришини мумкинлиги ҳақида кейинги бўлимда батафсил маълумот берилган.

Data бошқариш элементи хатоликларининг олдини олиш

Data бошқариш элементи ёрдамида шакл қуриш тез бажарилгани билан бу элементдан фойдаланишда жуда нозик хатоликларни кузатишингизга тўғри келади. Хатоликлар юзага келишининг асосий сабаби баъзи бир харакатларнинг чекланганлигидир. Масалан, жадвалда бўш сатр мавжуд бўлса, олдинга ёки орқага юришлар хатоликни юзага келтиради. Агар охирги сатрга борилса ва олдинга юришга харакат қилинса ҳам дастур хатолик ҳақида хабар беради. Одатда, бундай муаммолар юзага келмаслиги учун код ёзиш жараёнида баъзи-бир шартларнинг бажарилишини текшириш зарур. Масалан:

```
If Data1.Recordset.EOF Then
Data1.Recordset.MoveLast
End If
```

Агар сиз охирги ёзув чегарасидан ўтиб кетган бўлсангиз, у ҳолда шу майдонда жорий ёзув мавжуд бўлмайди ва EOF (End Of File — Файл охири) хусусияти True ҳолатига ўтказилади. Бу хусусият биринчи ёзув ва BOF (Beginning Of File — Файлнинг бошланиши)учун ҳам ўринлидир.

Ёзувлар жамланмаси мавжуд бўлмаса, EOF хусусияти ҳам, BOF хусусияти ҳам True ҳолатида бўлади, шунинг учун қуйидаги шартнинг бажарилишини текшириб кўришингиз мумкин.

```
If not (Data1.RecordSet.EOF and_
        Data1.RecordSet.BOF) Then
    Data1.RecordSet.MoveFirst
End If
```

"Жорий ёзув йўқ" шарти ҳақида батафсилроқ

Visual Basicнинг маълумотлар базаси билан ишловчи дастурлари жорий ёзувнинг йўқлиги туфайли юзага келадиган хатоликларга жуда ҳам сезгирдир. Бундай хатолик юзага келишига асосий сабаб қуйидагичадир: EOF ва BOF хусусиятларини True ҳолатига ўтказилган, жорий ёзув Delete ёрдамида ўчирилган ёки AddNew усулби қўлланилган, лекин маълумотлар базаси ёзувларини бошқа ёзувга кўчириш билан янгиланмаган. Охирги муаммо энг мураккаб. Қуйида бу муаммони ҳал қилишнинг учта услуби келтирилган:

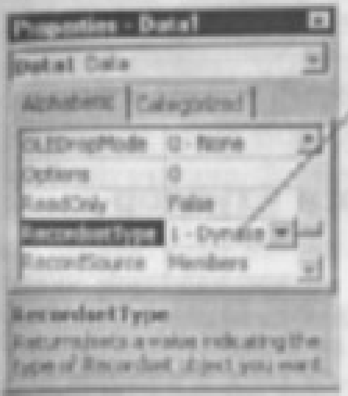
1. Сиз хатоликларни юзага келтириши мумкин бўлган ҳаракатларни такиқлаб қўйишингиз мумкин. Масалан, "Янги ёзув" тугмасининг босиб, AddNew усулини чақиришда "Ўчириш" тугмасини Enabled хусусиятини False ҳолатига ўтказиб қўйишингиз мумкин.
2. Бу ҳолат юзага келмаслиги учун AddNew усулидан кейин дарҳол Update (Янгилаш) усулини қўллаш зарур. Бунда LastName майдони учун вақтинчалик қиймат, масалан, NewName (Янги ном) номи билан бериш зарур. Акс ҳолда Update буйруғи хатолик билан бажарилади.
3. Сиз хатоликларни қайта ишловчи код ёзишингиз мумкин. 7-бобга қаранг, у ерда хатоликларга ишлов бериш услуби келтирилган.

Маълумотлар базаси билан ишлашда базага бир вақтда бир нечта фойдаланувчи мурожаат қилса, ёки сиз маълумотларни тармоқдан (сеть) олган вақтингизда юзага келиши мумкин бўлган хатоликлар ҳақида ахборотни Visual Basic нинг маълумотлар тизимидан олишингиз мумкин.

Ёзувларни берилган тартибда кўриб чиқиш

Спорт клуби учун яратган шаклингиз маълумотларни жадвалга қай тарзда ёзилган бўлса, шаклга шу тарзда чиқаради. Бу эса унчалик ҳам қулай эмас. Клуб аъзоларининг

алифбо тартибидаги рўйхатини олишни истасангиз, буни қуйидаги икки усулда бажариш мумкин.



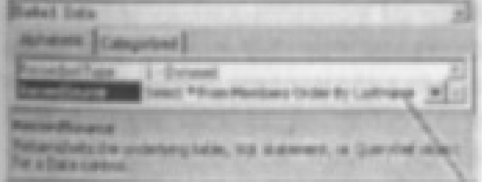
1. Data бошқариш элементини танланг, унинг RecordsetType (Ёзув тури) хусусияти Table (Жадвал) қолатидалигини текшириб кўринг.
2. Шакл учун код муҳарририни очинг. Юқори ўнг томондаги рўйхатдан Activate (Активлаштириш) ҳодисасини танланг. Кодга қуйидаги сатрни ёзинг:

Data1.Recordset.Index="LastName"

Сиз Index хусусиятини фақат Table (жадвалли) ёзувлар туридаги териш учунгина беришингиз мумкин.

Энг яхши услуб

Қуйидаги услуб мураккаброқ бўлгани билан энг яхши ва қулай усул ҳисобланади.



1. Юқорида 2-қадамда киритилган кодни ўчиринг. Data элементини танланг ва унинг RecordsetType (ёзувлар тури) хусусиятини Dynaset турига ўтказинг. Кейинги босқичда RecordSource (Ёзув манбаи) хусусиятини очинг ва қуйидаги сатрни ёзинг:

Select * from Members Order By LastName

Юқорида ёзилган код аслида SQL тилида ёзилган содда сўровдир. У Members жадвали барча ёзувларини LastName майдони қиймати бўйича тартибда сўрайди. Бу услуб жуда

ҳам мослашувчан бўлиб, у LastName соҳаси индексланмаган ҳолатда ҳам ишлайди.

Сиз индексланмаган майдонларни тартибга солиш учун Order By (Бўйича тартиблаш) операторидан фойдаланишингиз мумкинлигига қарамай, мазкур ҳолатда бундай саралаш унуми жуда паст бўлади.

Table ва Dynaset туридаги ёзувлар тўплами

Олдинги бўлимдаги мисолда Dynaset ёзувлар тўпламидан фойдаланган эдик. Visual Basic да Recordsetнинг уч турдаги объекти мавжуд. Recordset — бу Data бошқариш элементи томонидан таклиф қилинадиган ёзувлар тўпламидир.

Table тури

Table (жадвалга оид) туридаги ёзувлар тўплами — маълумотлар базасини ифодаловчи жадвалнинг ўзидир. Жадвал кўринишидаги ёзувлар тўпламининг қулайлиги шундан иборатки, ёзувлар тўплами осон шакллантирилади ва улар устида бажариладиган баъзи бир операциялар жуда тез бажарилади. Шунингдек, жадвал кўринишидаги ёзувларга Index хусусиятини ҳам тақдим этиш мумкин. RecordsSet объектининг бошқа турлари билан эса бу ишни қилиш мумкин эмас. Шу билан бирга Seek (Қидириш) усулини ҳам қўллаш мумкин, бу усул керакли ёзувни излашнинг энг тезкор усулидир.

Юқоридаги қулайликларига қарамасдан, жадвалларнинг ўзига хос ноқулайликлари ҳам бор. Агар сиз жуда катта ҳажмдаги маълумотлар базаси билан ишлайдиган бўлсангиз, базани бошқариш жуда қийинлашиб кетади. Бу фойдаланувчиларга ноқулайлик туғдириши билан бир қаторда иш унумининг пасайишига ҳам олиб келади.

Агар сиз Visual Basic да етарлича мураккаб маълумотлар базаси билан ишлайдиган дастур яратаётган бўлсангиз, ёзувлар тўпламларини Data бошқариш элементисиз яратиш мумкинлигини ҳам ҳисобга олинг.

Dynaset тури

Dynaset (Dynamic set — динамик тўплам) — бу энг фойдали умумий вазифали ёзувлар тўплами. Бу турдаги ёзувлар тўплами SQL тилида тузилган сўровдан фойдаланишда маълумотлар манбаи сифатида ишлатилади. SQL ривожланган сўров тили бўлганлиги сабабли динамик турлар мослашувчандир. Бу ёзувлар тўпламининг динамик дейилишига сабаб шуки, маълумотларга ўзгартириш киритилиши билан улар автоматик тарзда янгиланади. Тизимлар билан ишлашда эса бу турдаги ёзувлар тўпламини оддий усулларини қўллаган ҳолда янгилаш мумкин.

Юқорида саналиб ўтилган ёзувлар тўпламларининг барчаси учун Data бошқариш элементини қўллашингиз мумкин.

Snapshot тури

Snapshot (Статик) турдаги ёзувлар тўплами ҳам SQL сўровлар ёрдамида ҳосил қилинади. Бироқ DynaSet туридан фарқли ўлароқ, бу ёзувлар тўпламидаги маълумотлар тўплам очилган заҳоти ўрнатилади. Сиз бошқа фойдаланувчилар киритган ҳеч бир ўзгаришларни кўрмайсиз, статик тўпламнинг маълумотларини янгилай олмайсиз ва янги ёзувлар қўша олмайсиз. Статик тўпламлар ёзувлар тўпламини, унча катта бўлмаган жадвалларни "Фақат ўқиш учун" режимида ишлатишда қулайдир, бошқа ҳолатларда эса Dynaset дан фойдаланган маъқул.

Ёзувларни қидириш

Маълумотлар базасидаги ахборот маълум ҳажмга етгач, ёзувларни тезкор қидириш эҳтиёжи энг долзарб вазифага айланади. Қуйида Sports маълумотлар базасида ёзувни қидиришнинг содда усули келтирилган.

Мазкур услуб Table туридаги ёзувлар тўпламлари билан ишламайди.

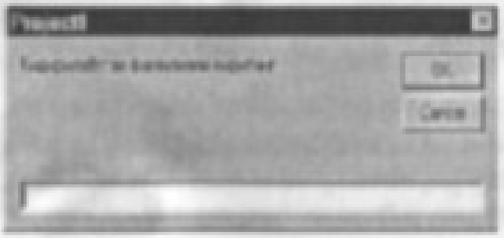
1. Шаклга қидириш тугмасини қўшинг. Унга cbSearch деб ном беринг. Caption хусусиятига "Қидириш" матнини ёзинг. Кейин Click (Босиш) ходисаси учун код муҳарририни тугмани икки марта босиб очинг.

2. Қуйидаги кодни киритинг:

```
Private Sub cbSearch_Click()
Dim searchvar As String
Dim sBookMark As String
searchvar = InputBox ("Қидириладиган фамилияни ёзинг")
searchvar = Trim$(searchvar) ' ортиқча бўш белгиларни ўчириш
    If searchvar <>"" Then ' хеч нарса киритилмаган бўлса, бекор қилиш
    With Data1.Recordset
    sBookmark = .Bookmark
    .FindFirst "LastName LIKE ' "+searchvar + "* "
        If .NoMatch Then ' ёзув топилмади
        MsgBox "Талабга жавоб берадиган ёзув йўқ"
        .Bookmark = sBookmark
        End If
    End With
    End If
End Sub
```

Ёзувлар тўпламида жорий ҳолатни белгилаш ва қидиришни тугаллагандан сўнг бу ҳолатга қайтиш учун фойдаланилган BookMark услубининг қўлланилишига эътибор беринг.

FindFirst нинг қай тарзда ишлаши ҳақида қуйида тўлиқ маълумот келтирилган.



Дастурни ишга туширинг ва "Қидириш" тугмасини босинг. Фамилияни киритиш учун сўровни кўрасиз. Қидириладиган фамилияни ёзинг ва OK тугмасини босинг. Агар сиз қидирган ёзув мавжуд бўлса у шаклга чиқади, акс ҳолда шаклдаги ёзув ўзгармайди.

Visual Basic да қидириш операцияси ҳарфлар регистрига боғлиқ эмас.

Visual Basic да ёзувлар тўпламини шакллантириш учун кўпинча Dynaset туридан фойдаланилади. Бу тур қўлланилганда қидиришнинг осон йўли FindFirst (Биринчи киришни топиш) ҳисобланади. FindFirst усулининг параметрлари бир қарашда ғалати туюлиши мумкин. FindFirst типик параметри қуйидагичадир:

```
FindFirst "LastName" = 'Расулов' "
```

Эътибор қилган бўлсангиз, қидириладиган фамилия битталик тирноқ ичига олинган. Фойдаланувчига қидириш одатда дастур ишлаб турган вақтда зарур бўлади. Демак, ўзгарувчига фойдаланувчи томонидан киритилган қийматни сақлаб қолиш зарур, чунки бу қийматлар FindFirst иши учун зарурдир. Бунинг сири бундай:

```
FindFirst "LastName = ' " + sSearchvar + " ' "
```

Қўштирноқ ичига битталик қавсларни қуйидагича киритинг:

Агар сиз SQL тили билан таниш бўлсангиз, FindFirst параметри SQL- сўровдаги Where операторига тўғри келади.

Агар Table туридаги ёзувлар тўпламидаги ёзувларни қидириш зарур бўлса, у ҳолда Seek услубидан фойдаланинг.

LIKE нинг қўлланилиши

FindFirst усулининг энг кучли томонларидан бири "=" нинг ўрнига LIKE операторининг ишлатиш мумкинлигидир. Бунинг асосий мазмуни шуки, қидириш жараёнида ажратувчи белгилардан фойдаланиш мумкин. Мисол учун фойдаланувчи "Jon" дан бошланадиган фамилияларни қидирмоқчи бўлса, ёзув охирига "*" белгини ёзиш зарур. Бу ишни амалга ошириш учун қуйидаги қадамни бажаринг:

```
FindFirst "LastName LIKE ' " + sSearchvar + " * ' "
```

AND нинг қўлланилиши

FindFirst услуби — сўровнома ташкил қилишнинг кучли усулидир. Сиз AND, OR, AND NOT операторларидан фойдаланиб мураккаб шартли сўровлар хосил қилишингиз мумкин. Кодга аниқлик киритиш учун сўровнинг хар бир шартини алохида қавс ичига ёзиш мақсадга мувофиқ.

```
FindFirst "[LastName = ' Артур ' ] AND [FirstName = ' Жахонгир']"
```

FindNext, FindPrevious, FindLast

Бу услублар ўз номларига мувофиқ равишда ишлайди.

FindNext (Навбатдагини топиш) жорий ёзувдан кейинги киришни излайди. FindPrevions (Олдингини топиш) жорий ёзувдан олдинги киришни қидиради. FindLast ёзувлар тўпламининг охиридан бошига томон йўналишда сўнгги киришни қидиради.

Барча Find... услубларининг ихтиёрий биттасидан фойдаланилгандан кейин NoMatch қиймати ёрдамида, қидириш ишлари муваффақиятли якунланганлигини текшириб кўришингиз мумкин.

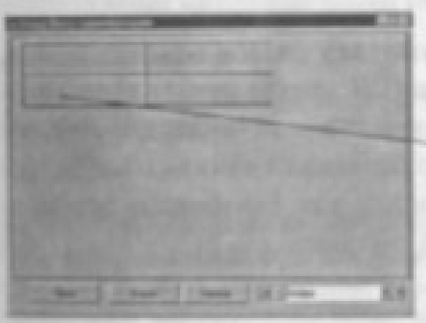
Ёзувларни тўрда кўриб чиқиш

Ёзувларни кўриб чиқишнинг яна бир оммавий усули бу тўрдир. Тўрнинг афзаллиги шундан иборатки, сиз бир вақтнинг ўзида бир нечта ёзувларни кўришингиз мумкин. Visual Basic да боғланган тўр (Microsoft Data Bound Grid Control) — бошқариш элементи мавжуд, бўлиб, у ёрдамида маълумотларни кўриш учун тўрни осон яратиш мумкин.

Маълумотлар тўрда қай тарзда акс эттирилади?

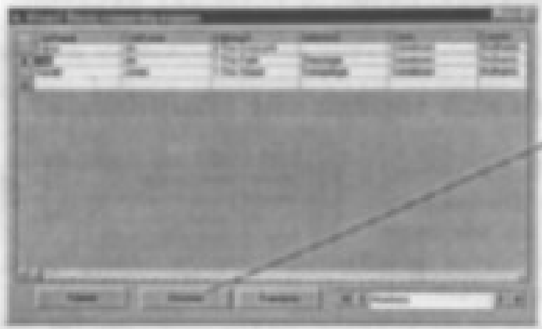
1. Асбоблар панелида ўнг босиш кейинг, менюнинг Components (Компонентлар) сатрини танланг ва Microsoft Data Bound Grid Control сатри белгиланганлигини текшириб кўринг.

Тўр асосида маълумотлар базаси хосил қилиш Visual Basic да маълумотлар базаси билан ишлайдиган дастурни яратишда энг тезкор усул хисобланади.



2. Расмда кўрсатилган шакл олдинги бобда келтирилган мисоллардан олинган, фақат мазкур холатда ёзув ва матн ойналари олиб ташланган. Бундай қилишдан олдин, сиз яратилган шаклни сақлаб қолишга қарор қилишингиз хам мумкин. Бунинг учун лойиха ойнасида шаклни ўнг босиш келинг ва менюни Save As сатрини танланг ва мавжуд лойихани бошқа ном, масалан, GridForm.frm номи билан сақланг. Шундан сўнг ёзувларни ва матн ойналарини ўчиринг ва шаклга DBGrid бошқариш элементини жойлаштиринг. Унинг DataSource хусусиятини Data Control да, AllowAddNew (Янги ёзувлар қўшишга рухсат) хусусиятини эса True холатига ўтказинг.

Сиз боғланган тўр элементини бошқа боғланган бошқариш элементлари билан бирга ишлатишингиз мумкин. Масалан, шаклга Notes (Эслатма) соҳаси билан боғлаб матн ойнасини жойлаштиринг. Тўрнинг исталган сатридаги ёзувлардан бирортасини танласангиз, Notes майдони автоматик тарзда янгиланади.



3. Дастурни ишга туширинг. Тўрдаги барча ёзувлар ойнаштирилмаган тўрда акс эттирилади. Тўрга янги сатр қўшиш учун энг пастдаги сатрга боринг. "Қўшиш" ва "Ўчириш" тугмалари ҳам ўз номи билан ишлайди.

Ёзувни ахборот алмашиш буферига (чўнтагига) кўчириш

Агар Сизда адресли маълумотлар базаси мавжуд бўлса, уни Windows нинг нусха олиш (Clipboard) воситалари ёрдамида алмашиш буферига кўчириш қулай бўларди. Бундай ҳолатда исталган маълумотлар базасидан кўчирма олиб, уни Office дастурларидан бирортасида хат ёки бошқа хужжатларга қўйиш имкониятига эга бўласиз.

Мазкур бўлимда келтирилган мисол сизнинг шаклингизда тўр мавжуд бўлган ёки оддий ёзувни кўриш ойнаси мавжуд бўлган ҳолатларда ҳам ишлаш имконига эга.

Clipboard объекти

Visual Basic кўринмайдиган Clipboard объектига эга. Бу объект ёрдамида нусха олинган маълумотни ўқиш, зарур ҳолатларда унга ўзгартириш киритиш имконияти мавжуд.

Қуйида келтирилган мисол ёрдамида сиз маълумот алмашиш буферини (Clipboard) тозаловчи Clear ҳамда унга янги сатр қўшувчи SetText усулларидан фойдаланишни билиб оласиз.

Шаклга "Буфер" тугмасини қўшиш

1. Маълумотлар шаклига cbClip номи билан тугма қўшинг ва уни Caption хусусиятига "Буфер" деб ном беринг.

2. Тугманинг Click ҳодисаси код муҳарририни очиш учун икки марта босинг ва қуйидаги кодни ёзинг:

```
Private Sub cbClip_Click()
Dim cr As String
Dim sAdress As String
cr = Chr(13) & Chr (10) ' янги сатр белгисини қосил қилади
    With Data1.Recordset
    sAddress = .Fields (FirstName") "+"
    sAddress = sAddress + . Fields ("LastName") + cr
    sAddress = sAddress + . Fields ("Address1") + cr
    ' зарур ҳолатларда бошқа янги майдонларни қўшиш
    End With
Clipboard.Clear
Clipboard.SetText (sAddress)
MsgBox "Нусха олинди"
End Sub
```

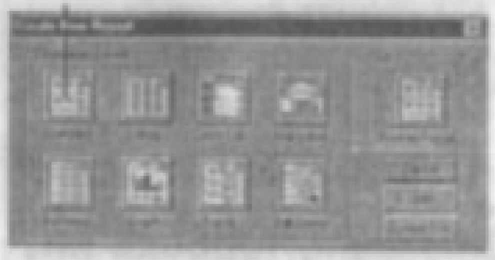
Энг яхши натижага эришиш учун бу кодни навбатдаги майдон бўш ёки бўшмаслиги шартини текшириш билан тўлдириш мумкин. Бу ишни sAddress сатрига соҳа қўшгунча амалга ошириш зарур. Бундай текшириш хужжатга адресни киритишда бўш сатрлар пайдо бўлишининг олдини олади.

Crystal Reports билан танишамиз

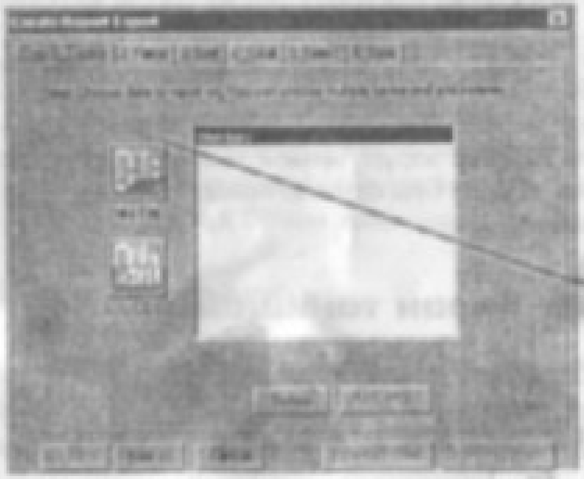
Маълумотлар базалари лойиҳаларининг энг муҳим хусусиятларидан бири ҳисобот тайёрлаш қобилиятидир. Масалан, спорт клуби учун клубнинг барча аъзолари рўйхати-

ни чоп этиш зарурати юзага келиши мумкин. Visual Basic да ҳисобларни чоп этишга чиқариш учун Crystal Reports воситаси мавжуд. Бу восита ёрдамида ҳар хил мураккаблик даражасидаги ҳисоботларни чоп этишга чиқариш мумкин. Қуйида ҳисоботни яратиш кетма-кетлиги баён этилади.

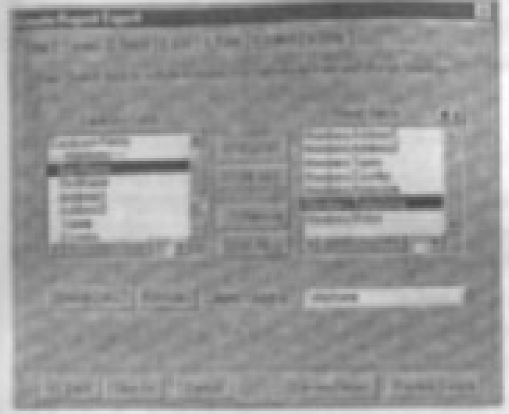
1. Менюнинг Add-Ins (Кенгайтмалар) сатридан Report Designer (Ҳисоботлар яратувчи) ни танлаб, Crystal Reportsни ишга туширинг. Crystal Reports ойнаси пайдо бўлиб, Crystal Reports ишга тушгач, унинг File(Файл) менюсидан New (янги) сатрини, танланг ва Standard (Стандарт) тугмасини босинг.



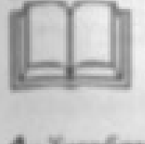
Add-Ins менюсида Report Designer сатри мавжуд бўлмаса, Crystal Reports тўғри ўрнатилмаган. Уни ўрнатиш учун Visual Basic нинг Setup дастурини ишга туширинг ва Crystal Reportsни ўрнатинг.



2. Ҳисобот тайёрловчи дастур (Create Report Expert) ойнаси пайдо бўлиши билан Data File (Маълумотлар файли) тугмасини босинг ва дискдан маълумотлар базаси файли SPORTS.MDB ни танланг.

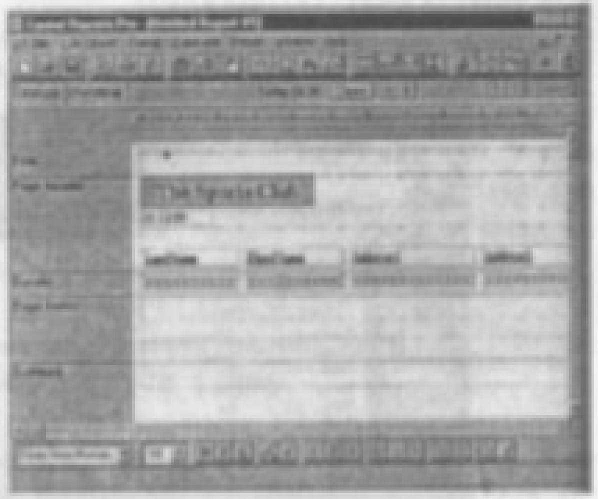


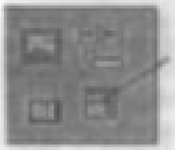
3. Members жадвалининг барча майдонларини ҳисоботга қўшиш учун Fields (Майдонлар) тугмасини танланг ва All (ҳаммаси) тугмасини босинг. Зарур бўлмаган майдонларни, масалан, ID майдонини ва бошқа, Notes майдонини ўчиринг. Клуб аъзоларини LastName (Фамилия) бўйича саралаш учун Sort тугмасини босинг, кейин ойнадаги Add тугмасини босинг. Style тугмасини босиб, ҳисобот номини киритинг: "The Sports Club"



Crystal Reports да ҳисобот қай тарзда тайёрланиши ҳақида тўлиқ маълумотни Crystal Reports маълумотлар тизимидан олиш мумкин.

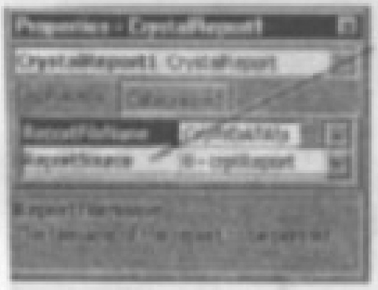
4. Ҳисоботнинг қоғозга қандай ҳолатда чиқишини кўриш учун Preview Report (Олдиндан кўриш) тугмасини босинг. Бу ерда ҳисоботга ҳеч қандай ўзгартириш киритманг, балки File>Save As (Файл>Сақлаш) менюсига кириб, уни C:\MYDATA каталогида SPORTS.RPT номи билан сақланг. Энди ҳисоботлар яратувчисини ёпинг.





Бу асбоблар панелидаги Crystal Reports бошқариш элементининг нишончаси.

5. Sports маълумотлар шаклини очиш учун Visual Basic га қайтинг. Асбоблар панелига ўнг босиш қилинг ва шаклнинг Components (Компонентлар) қисмида Crystal Reports Control 4.6 ишлаб туриганлигини текшириб кўринг.

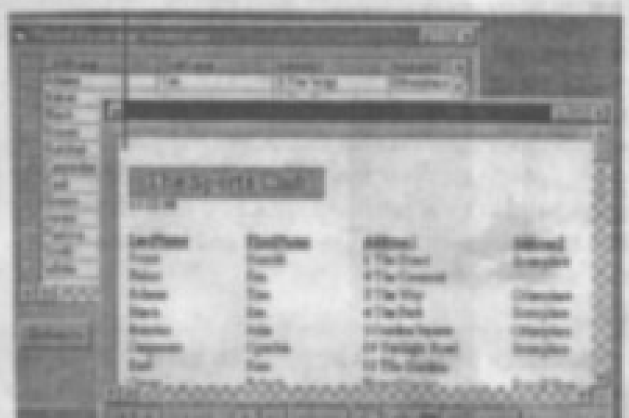


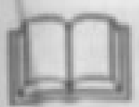
6. Шаклга Crystal Reports бошқариш элементини жойлаштиринг. Дастур бажарилиши жараёнида бу элемент кўринмас бўлгани учун уни шаклнинг истаган жойига жойлаштириш мумкин. Унинг ReportFileName хусусиятини ўзингиз яратган SPORTS.RPT ҳисоботига ўрнатиб кўйинг.

7. Энди шаклга "Ҳисобот" (Отчёт) тугмасини жойлаштиришингиз мумкин. Бу тугманинг Click (Босиш) ходисаси учун қуйидаги кодни ёзинг:

```
CrystalReport1.PrintReport
```

Дастурни ишга туширганингиздан кейин "Ҳисобот" тугмасини боссангиз, сиз яратган ҳисобот ўз номи билан ўз ҳисобот ойнасида очилади.





Дастур иши жараёнида ҳисобот ойнаси навиргa чиқариш, масштабни ўзгартириш ва саҳифалар бўйича кўчиш тугмаларига эга бўлади.

Маълумотлар базаси билан ишлашдаги кейинги қадам

Visual Basic — маълумотлар базалари билан ишлашда автоном дастурга, масалан, Microsoft Access га нисбатан бир оз мураккаброқ муҳит, лекин бир қатор афзалликларга эга. Хусусан Visual Basic маълумотлар базаси билан ишловчи дастурлар яратишда кучли восита ҳисобланади. Яна бир афзаллиги шундан иборатки, сиз бошқа фойдаланувчиларга тугалланган дастурни тарқатмоқчи бўлсангиз, Visual Basic да бу ишни амалга ошириш қулайдир. Агар сиздан маълумотлар базасининг хавфсизлигини таъминлаш талаб қилинса, Visual Basic да фойдаланувчи ҳаракатларини чегараловчи дастурлар ҳам яратиш мумкин.

Visual Basic ни Access билан бирга ишлатиш

Visual Basic ва Microsoft Access — жуда ҳам яхши ҳамкорлардир. Сиз дастур яратиш жараёнида маълумотлар базасини яратиш, маълумотларни текшириш қоидаларини киритиш ва интерактив таҳрир қилишда Accessдан, тугалланган натижавий дастур яратиш учун эса Visual Basic дан фойдаланишингиз мумкин. Шу билан бирга Visual Basic да ёзилган дастур кодини Access га тақдим этиш мумкин.

Бир нечта ўзаро боғланган (реляцион) маълумотлар базаларига ўтиш

Мураккаб маълумот базалари билан ишлашда бир нечта ўзаро боғланган жадваллардан фойдаланишга тўғри келади. Масалан, компания буюртмачилар, буюртмалар ва

буюмлар учун алоҳида жадвалдан фойдаланиш мумкин. Бундай реляцион маълумотлар базаларидан фойдаланиш фойдаланувчига кўпроқ маълумот билан ишлаш имкониятини бергани билан бундай базаларни бошқариш битта жадвалли базаларга қараганда жуда ҳам мураккаб. Лекин қулайлик томонларидан бири шундаки, бундай базалар жуда ҳам мослашувчандир.

Мижоз — сервер технологиясига ўтиш

Visual Basic ODBC номи билан аталувчи технологияни қўллайди. Visual Basic нинг бу имконияти Accessнинг MDB маълумотлар базалари, SQL Server, Oracle ва DB2 маълумотлар базалари серверлари билан ишлаш имконини беради. Мижоз-сервер технологияси маълумотлар базаларини дастурлашнинг энг мураккаб усулларидан бири ҳисобланади. Лекин тармоқда 10 тадан ортиқ фойдаланувчи ишласа, бундай ҳолатларда иш самарадорлигини ошириш учун мижоз — сервер технологиясидан фойдаланиш зарур.

Visual Basic ёрдамида содда маълумотлар базаларидан тортиб, то бир нечта фойдаланувчи билан ишлаш имконияти бўлган тизимлар — мураккаб мижоз-сервер яратиш мумкин.

ODBC — Open Database Connectivity — Очиқ маълумотлар базасидир.
Маълум бир маълумотлар базасига уланиш учун сизга шу форматдаги ODBC-драйвери (йўналтирувчи файл) зарур. Баъзи бир ODBC драйверлари Visual Basic нинг Enterprise версияси билан бирга етказиб берилади.

ОЛТИНЧИ БОБ

Office да Visual Basic

Visual Basic нинг энг қизиқарли имкониятларидан бири шундаки, у бир оз ўзгартирилган ҳолатда Office дастурлари билан бирга қўйилади.
Office дастурларидан фойдаланувчилар бу бирликнинг афзалликларидан фойдаланиши мумкин, мазкур бобда Office дастурларида Visual Basic дан фойдаланиш, макрос ёзиш, шакл ва диалог ойналарини ҳосил қилишни кўриб чиқамиз.
Шу билан бирга Visual Basic ёрдамида Office дастурларини бошқариш, Word ва Excel дастурларидан шахсий лойиҳаларингизда фойдаланишни билиб оласиз.

Ёрдамчи (илова) дастурлар учун Visual Basic нақли — версияси

VBA — Visual Basic for Application — бу ном Microsoft компанияси томонидан Office дастурларининг дастурлаш тилига берилган номдир. VBA, шунингдек, бошқа ишлаб чиқарувчилар учун ҳам ярокли бўлиб, улар бу тилдан ўз дастурларида фойдаланишлари мумкин. VBA ва Visual Basic нинг автоном ишловчи версиялари орасидаги фарқни билиш учун икки ташкил этувчини фарқлаш керак:

1. Visual Basic тилининг негизи.
2. Объектлар, яъни тугма, рўйхатлар ойнаси ва ActiveX бошқарув элементлари.

VBA да ҳам, Visual Basic нинг автоном ишловчи версиясида ҳам тил негизи (асоси) ўзгармайди. Шу билан бир вақтда бу тил — объектлар тўплами бўлиб, уларнинг таркиби ўзгариб туради. VBA билан ишлаш имкони бўлган дастурда ўзига хос дастур объектлари бўлади ва улардан сиз ўз кодингизда фойдаланишингиз мумкин.

Мисол учун, Wordда Document (Ҳужжат) ва Paragraph (Абзац) объектлари бор. Сиз бу объектлар билан ишлашни ўрганиб олган бўлсангиз, VBA коди ёрдамида бу объектлар билан ишлаш имконига эга бўласиз.

Шуниси эътиборга лойиқки, системангизда Office дастури ўрнатилган бўлса, Office дастурларининг исталган элементи учун VBA кодини ёзиш мумкин. Масалан, Excelда ёзилган VBA кодини Word учун ёки Visual Basic автоном дастурида ёзилган коддан Office дастурининг исталган элементида фойдаланиш мумкин.

Бошқа фарқлар

VBA да Visual Basicнинг автоном версиясидагига ўхшаш матн муҳаррири ва созлагич (отладчик) бор, лекин бу элементлар ўз имкониятлари бўйича бир нечта чеклашларга эга. VBA да кодни автоном EXE-файлгача компиляция қилиш

имконияти йўқ, бундан ташқари, VBA процессорнинг (native code) бевосита йўриқномасигача қадар компиляция қила олмайди.

Шу билан бирга VBA да Data бошқариш элементи ҳам йўқ. Лекин маълумот базалари орқали "соф" VBA коди билан ишлаш имконингиз бор.

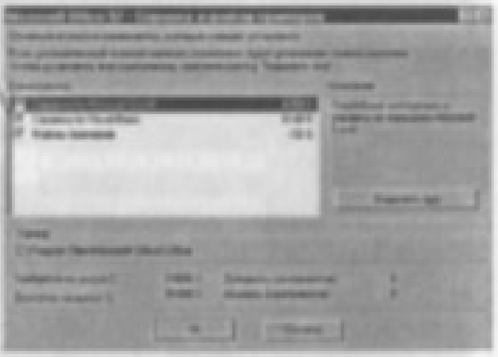
VBA нинг ноёб имконияти ҳам бор: дастур тузишни макрослар ёзишдан бошлашингиз мумкин, яъни Office нинг ўзи сиз учун код ёзади.

VBA Excel, Word, Access ва PowerPoint дастурларининг бир қисми ҳисобланади. Outlook дастури эса Visual Basic нинг бошқа, имконияти камроқ VBScript версиясидан фойдаланади.

VBA ни ўрнатиш

VBA версияси Office дастурларини ўрнатишганда автоматик тарзда ўрнатилади. Лекин VBAнинг маълумотномаси ўрнатилмайди. VBA да дастур тузиш жараёнида маълумотнома зарур бўлиши мумкин. Шу билан бирга VBA билан ишлашда ахборотлар базаси билан ишлаш имконини берувчи объектларни ўрнатиш зарурияти ҳам юзага келиши мумкин.

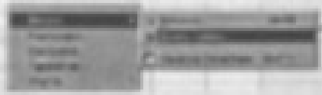
VBA компонентларини ўрнатиш



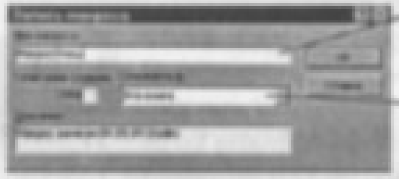
VBA маълумотномаси файлларини ўрнатиш учун Officeни ўрнатиш дастурини ишга туширинг. "Добавить [Удалить компоненты]" (Add/ Remove components) буйруғини танланг. Word, Excel ва PowerPoint учун "Справка и файлы примеров" (Help files and samples) сатрини белгиланг. Access учун "Справка по программированию" (Programming Help) сатрини белгилаш лозим. Ҳар бир дастур ўз объектлари учун маълумотнома ўрнатади, шу билан бирга бу маълумотнома файллари турличадир.

Visual Basic ва VBA да эквивалент тугмалар бир-бирига тўғри келади. Масалан, дастурни ишга тушириш учун F5 тугмасини босинг.

Excel да макрос ёзиш

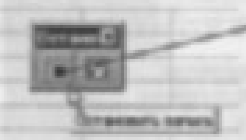


1. Excel ни ишга туширинг ва янги иш китоби очилганлигини текширинг. Сўнгра менюдан Сервис>Макрос>Начать запись [Tools>Macro>Record New Macro] ни танланг.

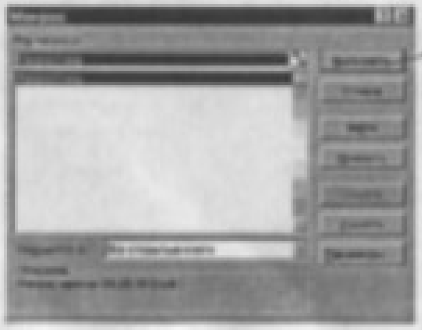


2. Пайдо бўлган диалог ойнасида макрос номини ёзинг, масалан, "Очиқ макрос". "Сохранить в" (Store Macro in) рўйхатида "Эта книга" ("This Workbook") сатри танланганлигини текшириб кўринг. OK ни босинг.

VBA да дастурлашни бошлашнинг энг қулай йўли макрос ёзишдир, чунки макрос сизнинг ўрнингизга Visual Basic кодини ўзи ёзади.



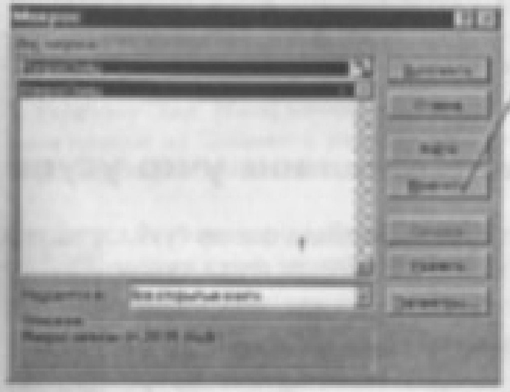
3. Макрорекордер ойнаси пайдо бўлганлигига эътибор қилинг. Бу ойна актив ҳолатда бўлса, сизнинг барча ҳаракатингиз ёзилади. Энди менюдан Excel Faile>Open [Файл>открыть]ни танланг ва камдикингиз бўйича истаган ишчи китобни очинг. Сиз танлаган китоб очилгандан "Остановить запись" (Stop Recording) тугмасини босинг.



4. Сиз макрос ишини осонлик билан текшириб кўришингиз мумкин. Очган ишчи китобингизни ёпинг. Сервис > Макрос > Макросы [Tools > Macro > Macros] менюсига киринг "Макрос открыть" номини танланг ва "Выполнить" (Run) тугмасини босинг. Ишчи китоб яна бошқатдан очилади.

Макрос ёзиш жараёнида макроснинг қаерда сақланаётганлигига эътибор беринг. Макрос ишга тушиши учун макрос ёзилган китоб ишга туширилган бўлиши зарур. Макросни истаган вақтда ишга тушириш учун эса уни "Личная книга макросов" [Personal Macro Work book] — Макрослар шахсий китобида сақлаш лозим.

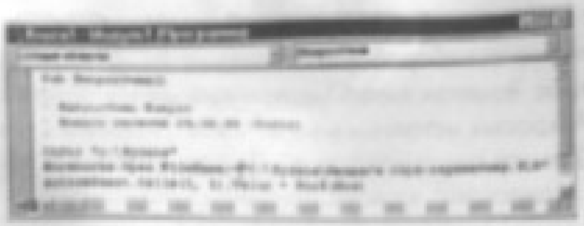
Макроснинг тузилмаси



1. Ўзингиз яратган макросни кўриш учун, макрос ёзилган китобнинг очиқлигини текширинг. Сўнгра макрослар рўйхатини очиш учун Сервис>Макрос> Макросы [Tools> Macro>Macros]ни танланг. Макрос танланг ва уни ўзгартириш учун "Изменить" [Edit] тугмасини босинг.

2. Visual Basic энди очиқ қолда, унинг ойнасида эса сизнинг "Макрос очиқ" номли макросингиз турибди.
У иккита сатри кодда эга:
ChDir "C:/mydata"
сатри сизнинг ишчи китобингиз жойлашган каталогни жорий каталог деб эълон қилади.
WorkBooks. Open FileName:= "C:\mydata\myfile.xls" сатри эса файлни очади. WorkBooks — бу Excel объектларининг коллекцияси бўлиб, Excelда очилган барча китобларни кўрсатади. Open — WorkBookнинг услуби бўлиб, у Excelга кўрсатилган ишчи китобни очишни буюради. FileName — бу номланган аргумент бўлиб, унга":=" операцияси ёрдамида бериладиган қиймат, файлнинг тўлиқ номи ва унинг жойлашган жойи кўрсатилиши мумкин.

Excel объектлари номлари ҳақида тўлиқ маълумот олиш учун Excelда Visual Basic маълумотномасидан фойдаланинг.



3. Сиз макрос кодини ўзгартиришингиз ёки янги йўриқномалар ёзиб, макросни кенгайтиришингиз мумкин.
Масалан, макросга қуйидаги кодни ёзсангиз, Excel биринчи уяга бугунги санани ёзади:
Active Sheet. Cells (1,1). Value = Str$ (Now)
Бошқа ишчи китобни очиш учун кодни осонликча ўзгартиришингиз мумкин. Кодни бу тарзда, яъни қўлда таҳрир қилиш имкони билан ёзиш катта имкониятларга эга.

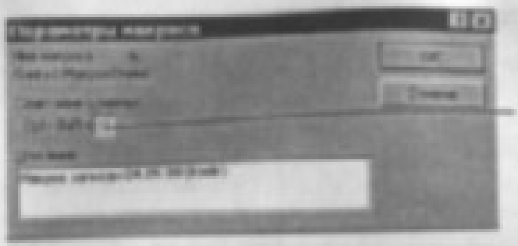
Макросни ишга туширишнинг учта усули

Макрос яратилгандан сўнг табиийки, савол туғилади, уни қай тарзда ишга туширилади? Макросни ишга туширишнинг бир нечта усули бор. Олдинги бўлимда макросни менюдан Сервис> Макрос> Макросы (Tools>Macro>Macros) сатрини танлаб ишга туширишни кўрдингиз. Қуйида макросни ишга туширишнинг яна учта усулини келтирамиз.

Бу бўлимда асосан Excel ҳақида гапирилса ҳам Officeнинг бошқа дастурларида ҳам шу бўлимдаги маслаҳатлардан фойдаланиш мумкин.

1. Макросни клавишлар бирикмаси ёрдамида ишга тушириш

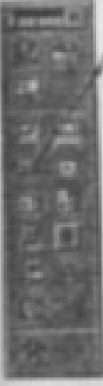
Макрослар рўйхатини очинг ва "Параметры" (Options) тугмасини босинг. "Сочетание клавиш" (Short Key) ойнасига исталган белги, масалан "O"ни киритинг. OK тугмасини босинг ва макрослар рўйхатини ёпинг. Энди Ctrl+Shift+O клавишлар бирикмаси бу макросни ишга туширади.



2. Макросни ишчи варақдан ишга тушириш

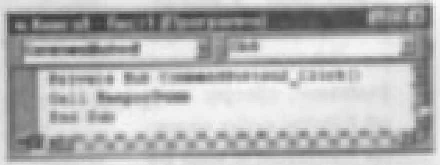
Макросни ишга тушириш учун тугмани бевосита ишчи варағига жойлаштириш мумкин.

1. Excelнинг "Вид" (View) менюсидан "Панели инструментов" (Toolbars) сатрини танланг ва "Элементы управления" (Бошқариш элементлари) (Control Toolbars) сатри белгиланганлигига ишонч ҳосил қилинг.



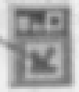
2. Ишчи варақда тугмани жойлаштиринг ва тугма чегараларини кенгайтиринг.

3. Сичқонча ўнг тугмасини босинг ва қалқиб чиқадиган менюдан "Исходный текст" (View code) сатрини танлаб, тугма Click ҳодисаси учун кодни очинг.



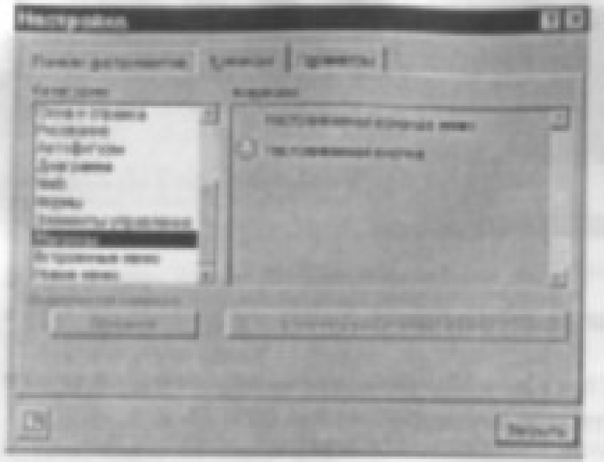
Қуйидаги сатрни ёзинг:
(Call Makros Открыть, ёки сизнинг макросингиз номи).
Макрос очиш (ёки макросингиз номи) код муҳарририни ёпинг, энди сиз жойлаштирган тугма макросни ишга туширади.

Excelда дастурлаш ҳолатидан чиқиш учун унинг ишчи варағида мана шу тугмани босинг.



3. Макросни менюдан ишга тушириш

Сиз ўзингиз яратган макрос учун менюда қуйидагича алоҳида сатр яратишингиз мумкин:

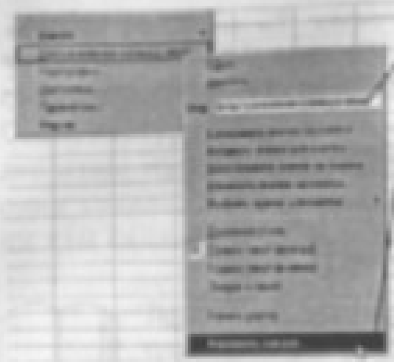


1. Excel нинг "Сервис" (Tools) менюсидан "Настройка" (Customize) сатрини танланг. Сўнгра "Команды" (Command) ни белгиланг. Чап томондаги ойлантирилувчи рўйхатдан "Макросы" (Macros) ни танланг.

2. Энди ойнанинг ўнг томонида "Настраиваемая команда меню" (Custom Menu Item)ни танланг ва сичқонча чап тугмасини босилган ҳолда ушлаб туринг.

3. Менюнинг "Сервис" (Tools) сатрини Макрос менюсига тортиб ўтказинг, сўнгра пайдо бўлган қора чизиқни кузатган ҳолда бу сатрни пастга туширинг, сичқонча тугмасини қўйиб юборсангиз, менюнинг янги сатри қора чизиқ турган майдонга жойлаштирилади.

Агар сиз ўзингиз яратган бир нечта макрос билан ишласангиз, у ҳолда юқори даражадаги янги асбоблар чизиғи ва янги меню ҳосил қилишингиз мумкин. Бу ишларни амалга ошириш учун Excel маълумотномасидан фойдаланинг.

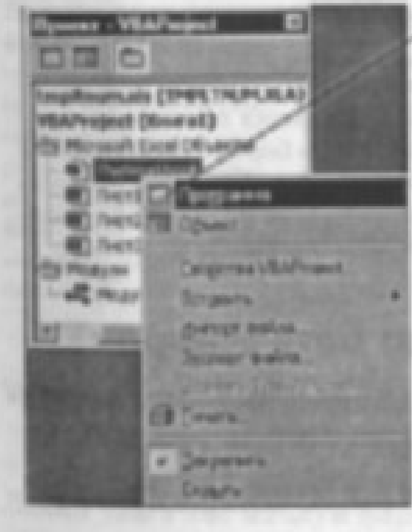


3. Менюнинг янги сатрига ўнг босинг. "Имя" ("Name") ойнасида меню сатри номини ёзинг. Кейин "Назначить макрос" (Assign Macro) сатрини босинг ва ўз макросингизни танланг. Юқоридаги қадамларни амалга оширгач, "Настройка" (Customize) диалог ойнасини ёпинг.

Автоматик тарзда бажарилувчи макрослар

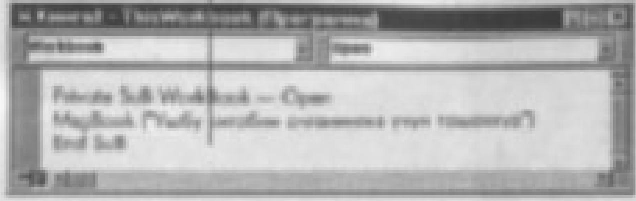
Бирорта ҳужжат очилганда ёки маълум бир ҳаракатга жавоб тариқасида автоматик бажарилиши мумкин бўлган макрос яратишингиз мумкин. Қуйида ана шундай макрос яратишга мисол келтирилган.

Excel ишчи китобини ишга туширишга жавоб тариқасида бажарилувчи макрос.



1. Excel ишчи китобларидан бирортасини очинг ёки янгисини яратинг. Visual Basic муҳарририни очинг. Агар лойиҳа ойнаси очилмаган бўлса, "Вид" (View) менюсидан "Окно проекта" (Project Explorer) сатрини танланг. Лойиҳа ойнасида This Workbook сатрини топинг ва ўнг босинг. Қалқиб чиқувчи менюдан "Программа" (View Code) сатрини танланг.

2. Чап рўйхатдан Workbook сатрини танланг. Ўнг томондаги рўйхатда Open (Открыть) ходисасон кўрсатилганлигини текшириб кўринг, кейин эса истаган кодни, масалан учун, ташаккур эълон қилувчи панель кодини ёзишингиз мумкин.



Word хужжатлари хам Open ходисасига эга.

Рўйхатда кўрсатилган бошка кўпгина ходисалар учун хам код ёзишингиз мумкин.

3. Visual Basic мухарририни ёпинг ва ишчи китобни сакланг. Сўнгра уни очинг, китоб очилиши билан код ишга тушади ва ташаккур эълон қилинган панель пайдо бўлади. Бундан ташқари, вактни, санани кўрсатувчи ёки глобал ўзгарувчиларнинг бошланғич қийматларини кўрсатувчи код ёзиш мумкин.

Сиз макросга эга хужжатни очаётганингизда огохлантирувчи панель пайдо бўлади. Бу диалог ойнасини макровируслардан химояланишга имкон беради. Агар сиз ишчи китобни ва макросни ўзингиз яратган бўлсангиз ёки улар олинган манбага ишончингиз комил бўлса, бу огохлантирувчи диалог ойнасини бекор қилишингиз мумкин.

Лойиха ойнаси хақида батафсилроқ

Visual Basic For Application нинг лойихалар ойнаси (Project Explorer) сиз яратган шакл ва кодлар қаерда сакланаётганлигини кўрсатади хамда уларни тахрир қилиш учун

очиш имконини беради. Office матнидаги Visual Basic коди ва шакллари хужжатларда ёки андазаларда сакланади. Яъни Office дастурларида яратилган хужжатларни VBA маъсули деб аташ мумкин. Сиз Office хужжатларини ёки андазаларини саклаш жараёнида уларда Visual Basic элементларини хам саклайсиз.

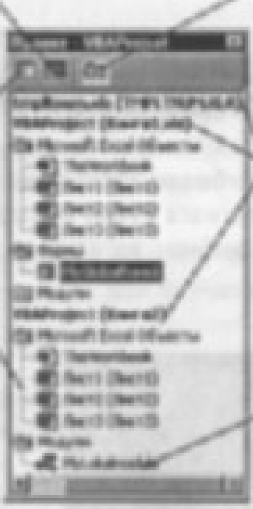
Лойихалар ойнасининг қай тарзда ишлашини билиб олиш учун уни амалда синаб кўрган маъқул.

Excel да лойиха ойнаси

"Объект" (Object) тугмасини танланган объектни кўриш учун босинг.

Танланган объект кодини очиш учун "Программа" (View Code) тугмасини босинг.

Танланган ишчи варок ёки китобни икки марта босинг ёки Open (очиш) ёки Change (Алмаштириш) ходисаларига жавоб тариқасида бажарилувчи код ёзиш учун "Программа" (View Code) тугмасини босинг.



Папкалар шаклида ташкил қилинган лойихаларни кўриш учун "Папка" (Folder) тугмасини босинг.

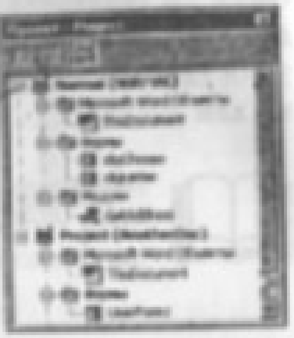
Excelда очиладиган хар бир хужжат (документ) лойиха хисобланади. Макрослар шахсий китоби "Personal Macro Workbook" хам шу ерда кўрсатилган.

Модуллар функциялар, кетма-кетликлар ёки глобал ўзгарувчилар қийматлари кодига эга бўлиши мумкин.

Word да лойиха ойнаси

Word лойиха ойнаси Excel лойиха ойнасига ўхшашдир. Лекин ўзига хос маълум фарклари хам бор. Масалан, Word

лойихасида Normal андазаси бор. Ундан ташкари, бу ерда мурожаат хисоби бошка усулда ташкил этилган.

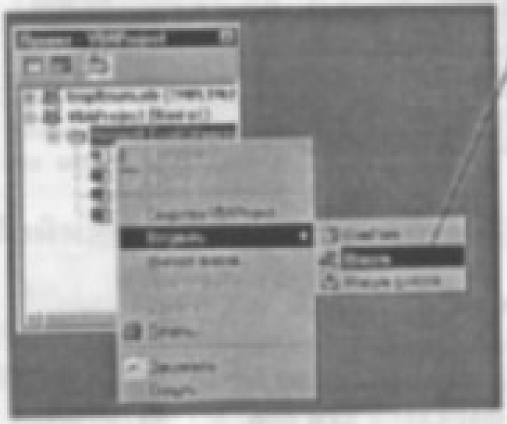


Normal андазаси коди ва шаклини хар қандай ҳужжатга кўчириш мумкин.

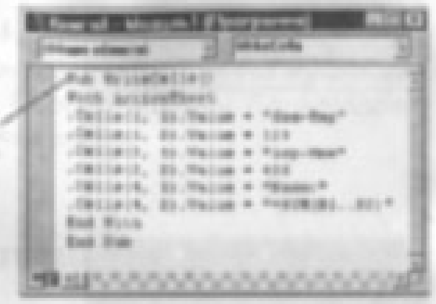
VBA га мурожаатлар Visual Basic автоном версиясидаги каби ишлайди.

VBA да ячейкалар қийматларини ёзиш

Excel да Visual Basic билан ишлаш жараёнида кўпинча ячейкаларга қиймат беришингизга тўғри келади. Қуйида келтирилган мисолда ишчи китобга қиймат ва формула ёзиш тартиби келтирилган.



1. Янги китоб яратинг ва Visual Basic муҳарририни очинг. "Вид" (View) менюсидан "Окно проекта" (Project Explorer) сатрини танланг. Лойиҳани сичқонча ўнг тугмаси ёрдамида босинг ва қалқиб чиқувчи менюдан Вставить» Модуль (Insert» Module) сатрини танланг. Бу модуль лойиҳа ойнасида Модуль 1 (Module 1) номи билан чиқади.



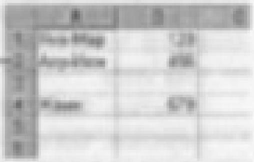
2. Код муҳаррири автоматик тарзда очилади. Сиз янги тартиб (процедура) яратишингиз зарур. Бунинг учун сиз Insert менюсидан Procedure (Процедура) сатрини танлашингиз мумкин эди, лекин энг қулай йўли "Sub WriteCells" ни муҳаррирга ёзишдир. Enter тугмасини боссангиз, VBA End Sub операторини автоматик тарзда ёзиб қўяди. Хозирлиқча сиз WriteCells номи билан бўш макрос яратдингиз. Унинг кодини ёзинг:

```
With Active Sheet
.Cells (1,1) . Value = "Яна-Мар"
.Cells (1,2) . Value = 123
.Cells (2,1) . Value = "Апр-Июн"
.Cells (2,2) . Value = 456
.Cells (4,1) . Value = "Жами"
.Cells (4,2) . Value = " = Sum (B1..B2)
End With
```

Номланган диапазон — VBA кодида ячейкаларни белгилашнинг қулай усулидир. Мисол учун, агар сизни "TotalVal" диапазон қизиқтирса, кодда бу ораликка Range ("TotalVal") номли мурожаатларини бериш мумкин.

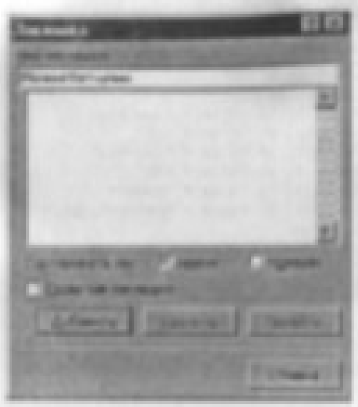
Excel да формула ёзишда, формулани оддий матндан фарқлаш учун "=" дан фойдаланиш зарур.

3. Макрос ишга туширилгач, автоматик тарзда Excel да код ичига ёзилган қийматлар пайдо бўлади.



Word макросидан матн киритиш

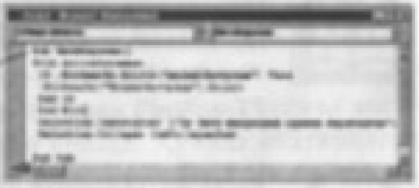
Word макросидан матн киритиш — бу, Word билан ишлаш жараёнида ҳужжатга макрос ёрдамида матн қўйишдир. Бу бўлимда Word макроси ёрдамида матн киритишга оид содда мисол келтирилган.



1. Word нинг янги хужжатини яратинг, Вставить (Insert) менюсидан (Хатчўп) (Bookmark) сатрини танланг. Закладкага ном беринг, масалан, "Менинг хатчўпим".

Word да хатчўплардан фойдаланиш Excel да номланган диапазонлардан фойдаланишга ўхшашдир. Visual Basic билан ишлашда иккаласи хам фойдалидир.

2. Visual Basic муҳарририни очиб, лойиҳа ойнаси очилган-очилмаганлигини яна бир марта текшириб кўринг. Сиз уни "Вид" менюсидан танлашингиз мумкин. Сичқонча ўнг тугмасини хужжат лойиҳаси устида босинг ва менюдан Вставить> Модуль (Insert> Module) сатрини танланг. Код муҳарририни очиб, Sub 'матн киритинг.

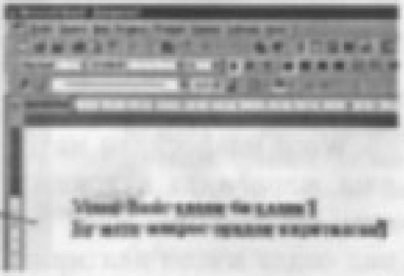


End Sub ни муҳаррирнинг ўзи ёзади.

Энди қуйидаги кодни ёзинг:

```
With ActiveDocument
If.Bookmarks.Exists ("Менинхатчўпим") Then,
Bookmarks ("Менинг хатчўпим"). Select
End If
End With
Selection. Insert After ("Бу матн макросдан қўйилган")
Selection. Collapse (wdCollapseEnd)
```

3. Макросни ишга туширинг ва Word хужжатини текшириб кўринг. Матн автоматик киритилган эди.



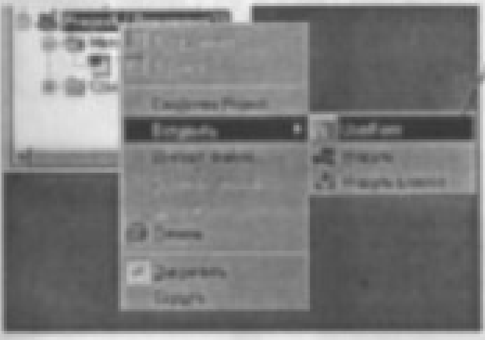
Сиз макросни ёки Visual Basic муҳарриридан, ёки бевосита Word хужжатидан ишга туширишингиз мумкин, бунинг учун Word нинг Сервис>Макросы (Tools>Macros) менюсидан фойдаланиш зарур.

Word да диалог яратиш

Visual Basic "Визуал" деган номни оқлайди, чунки сиз шакллардан ва диалог ойналаридан фойдаланувчи дастур яратишингиз мумкин. Мазкур бўлим мисоли содда бўлишига қарамасдан, VBA имкониятларидан фойдаланиш бу мисолларда чекловларсиз намойиш қилинган. Сиз шаклда ActiveX элементларидан фойдаланишингиз мумкин, булар хужжатга диаграммалар, мультимедиа ва анимацияларни киритиш имконини беради.

Диалог ойнасини яратиш ва ундан фойдаланиш

Мазкур лойиҳадан мақсад, макросни тузиш бўлиб, у иккита андазадан бири асосида янги хужжат яратади.



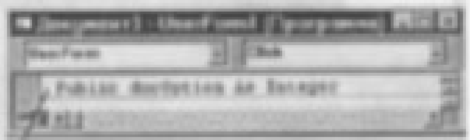
1. Word да янги хужжат яратинг ва Visual Basic муҳарририни очинг. Янги хужжат билан боғлиқ лойиҳа ойнасига ўнг босинг ва менюдан Insert> UserForm ни танланг.
Жавоб тариқасида янги шакл ойнасида ва асбоблар панели пайдо бўлади.

Агар сиз макросдан мунтазам фойдалансангиз, уни хужжатда эмас, балки Normal.dot андазасида сақлаган маъқул. Эксперимент тариқасида яратилган макросларни эса хужжат (документ)нинг ўзида сақлаш мақсадга мувофиқ.

2. Шаклни Visual Basic нинг олтинчи версиясида ишлайдиган қилиб яратинг. Расмда кўрсатилган тарзда шаклга қайта улаш тугма ва иккита сарлавҳали тугма жойлаштиринг. Юқоридаги қайта улаш тугманинг Value хусусиятини True ҳолатига ўтказинг. "OK" тугмачаси учун Default, "Cancel" тугмаси учун эса юқоридаги кўрсаткични True ҳолатига ўтказинг.

Enter ёки Escape клавишлари босилганда диалогнинг қандай ишлашини ҳодисани Default ва Cancel тугмалари белгилайди.

3. Шакл код муҳарририни очиш учун икки марта босинг ва чап томондаги рўйхатдан "Умумий соҳа" (General) сатрини танланг. Ўнг томондаги рўйхатдан "Тавсиф" (Declarations) сатрини танланг. Код муҳарририни очиб қуйидаги кодни ёзинг:



Public DocOption As Integer

Бу ўзгарувчи, фойдаланувчи томонидан диалогда танлаш натижаси қийматини сақлайди.

4. "OK" тугмасининг Click (Босиш) ҳодисаси учун код муҳарририни очинг, қуйидаги кодни ёзинг:

```
If Option Button 1. Value = True Then
User Form 1. DocOption = 1
ElseIf OptionButton 2. Value = True Then
UserForm 1. DocOption = 2
Else
UserForm 1. DocOption = 0
End If
Me. Hide
```

"Отмена" тугмачасини икки марта босинг ва ягона сатрни киритинг:

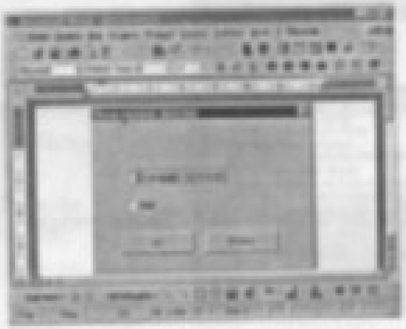
```
Me. Hide.
```

5. Сўнгра қолипнинг модуль қўйинг ва агар бор бўлса, уни очинг. Код муҳарририни очинг ва ёзинг:

```
SuB DocStart ()
User Form 1. Doc Option=0
UserFopm 1. Show
Select Case UserForm 1. Doc Option
Case1 'стандарт документ
Documents Add Normal. DOT фойдаланади
Case 2 ' Документ письма
Documents. Add Template:= "C:\Program Files\Microsoft Office\Шаблоны
\ письма и факсы\ Стандартное письмо. dot"
End Select
End SuB
```

Системангизда Word нинг ўрнатилган жойига қараб, андоза йўлини ва номини тўғри кўрсатишингиз керак.

6. Visual Basic ни ёпинг ва Wordга қайтинг. Кейин DocStart макросини ишга туширинг, у янги танланган андоза асосида ҳужжат яратади.



Office-Ёрдамчини дастурлаш

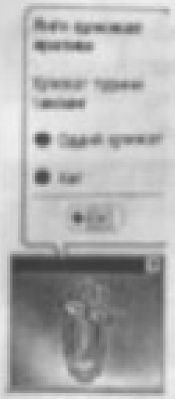
Скрепка (Қистиргич) ва Барбос номли Office ёрдамчилар баъзи фойдаланувчиларга ёқмаслиги мумкин. Агар сизга Office-Ёрдамчи (Office Assistant) ёкса, уни Visual Basic кодида бошқаришингиз мумкин. Бу бўлимда келтирилган мисол махсус диалог ойнаси ўрнига Office ёрдамчисидан фойдаланишни ўргатади.

Ёрдамчи — бу диалог ойналари хосил килишнинг осон ва тезкор усулидир, лекин у махсус шакл каби универсал мослашувчан эмас.

1. DocStart макроси ёзилган хужжатга қайтинг ва Visual Basic ни ишга туширинг. DocStart ни сакловчи модулни икки марта босинг ва кодга қуйидаги ўзгартиришларни киритинг:

```
Sub DocStart ()
Dim OptionChosen As Integer
With Assistant
. Visible = True
. Animation =
msoAnimationGetAttentionMinor
    With.NewBalloon
    .Heading= " Янги хужжат яратиш"
    .Text = "Хужжат турини танланг"
    .Labels(1). Text = "Ушумий хужжат"
    .Labels(2). Text = "Хат"
    . Mode =msoModeModal
    Option Chosen=. Show
    End With
End With
'... Doc Start нинг эски давоми.
```

Office-Ёрдамчини унинг "Параметры" (Options) диалог ойнасидан танланади.



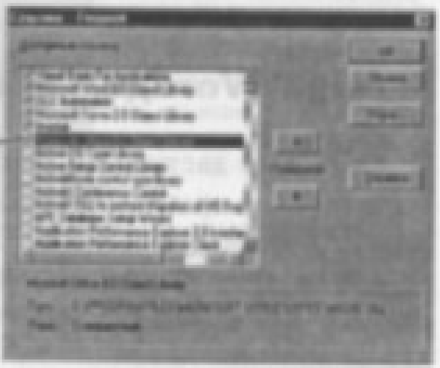
2. Хужжатга қайтинг ва DocStartни бажаринг. Бу гал Ёрдамчи пайдо бўлади. Макрос эса олдингидек ишлайверади.

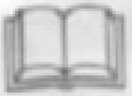
Ёрдамчи ҳақида батафсилроқ маълумот

Office-Ёрдамчини дастурлаш Visual Basicнинг Office билан ишлашига яхши мисол бўлади. Ўзингизнинг Ёрдамчингизни яратиш учун жуда кўп вакт ва катта тажриба керак бўлади. Лекин Officeга киритилган Ёрдамчини асос қилиб олиб унинг барча қулайликларидан фойдаланишингиз мумкин.

Office-Ёрдамчи ичкаридан ActiveX ёрдамида бошқарилади. Visual Basic нинг у билан ишлашига имконият яратиш учун "Ссылки" (References) диалогида Microsoft Office 8.0 Objest Library сатрини белгилаш зарур. Агар сиз Visual Basicни Word дан ишлатсангиз, бу сатр белгиланган бўлади.

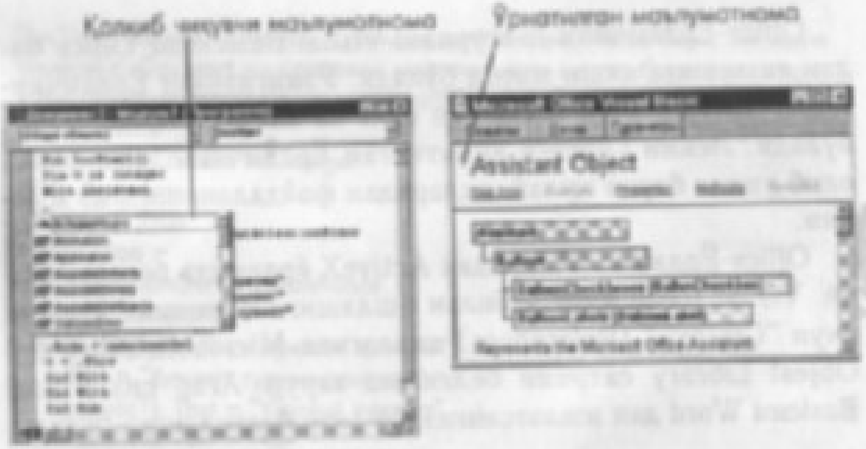
Бу сатрни белгилаб қўйинг, у Visual Basic учун Office-Ёрдамчини ишлатиш имконини беради.





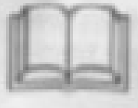
Office-Ёрдамчини дастурлашни ўрганиш осон ишдир, уни ўргангач, сиз Office ни дастурлашни янада чуқурроқ ўрганишингиз мумкин.

Мурожаат ўрнатилиши билан Office-Ёрдамчи Assistant нинг хусусиятлари ва услубларини Visual Basic нинг ташкил этувчиси каби ишлатиш мумкин. Асосий қийинчилик қил этувчиси каби ишлатиш мумкин. Асосий қийинчилик ёрдамчи хусусиятлар ва услублар қандай ишни бажаришини билиб олишдир. Бунинг учун код муҳаррири ичида нуқта ёзишингиз билан мавжуд ҳаракатлар рўйхати пайдо бўлади. Бошқа ахборот манбаи ўрнатилган маълумотномадир. Ўз кодингизда Assistant сўзини танланг ва маълумотноманинг керакли бўлимини очиш учун F1 клавишини босинг.

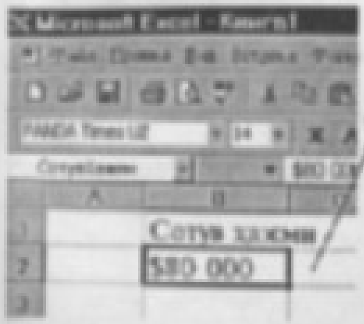


Excelни Wordдан бошқариш

Сиз Office дастуридан бажариладиган макросни муваффақият билан яратганингиздан сўнг, энди, навбатдаги қадамингиз бир дастурни бошқа дастурдан бошқаришни ўрганиш бўлади. Бунга мисол қилиб Word хужжатига Excel жадвалидан ахборотни киритиш талаб қилинадиган вазиятни қараш мумкин.



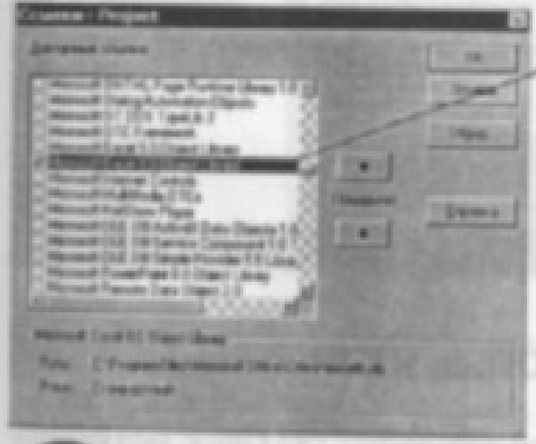
Excel жадвалларига Word хужжатларидан мурожаат қилишнинг бошқа усуллари ҳам мавжуд, лекин бу услуб энг универсалдир, чунки бу ерда сиз Excelнинг барча имкониятларини назорат қила оласиз.



1. Бу мисолни бажариш учун Excelда янги китоб яратинг, уни Sales (савдо) деб ном беринг, унда "Сотув ҳажми" номи билан ячейка тонланг (Total Sales). Бу ячейкага катта миқдор (пул бирлиги) маблағ жойлаштиринг, китобни сақланг ва Excel ни ёпинг.

2. Янги Word хужжатини яратинг ва "Сотув ҳажми" ячейкасига мурожаат қилувчи закладка (катчўп) матнини ёзинг. Бу ерда ячейка номини айнан Excelдагидек ёзиш шарт эмас, бу иш фақат қулайлик учун қилинган.

3. GetSalesTotal(Савдо ҳажмини олиш) макросини яратиш учун Сервис>Макрос>Макросы> Создать (Tools>Macro>Macros>Create) — Сервис>Макрос> Макрослар> Яратиш)ни танланг.



4. Макрос кодини ёзишдан олдин Visual Basic Object Library кутубхонаси Microsoft Excel 8.0 га мурожаат қилаётганлигини текшириб кўринг. Visual Basic менюсидан Сервис>Ссылки (Tools>References) ни танланг.

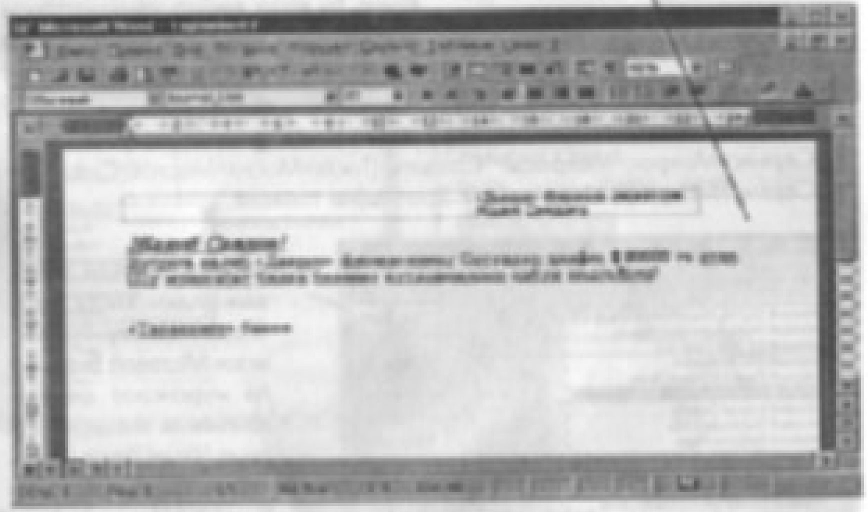
Код ёзиш жараёнида Sales. XLS файлининг жойини тўғри кўрсатинг.

5. Энди сиз Excel ни бошқарувчи кодни ёзишингиз мумкин. Қуйида келтирилаётган — Create Object коди Excelни фон режимида ишга туширади, ўз навбатида сиз Excelда "Иш китобини очиш" йўриқномасини бажариш имконига эга бўласиз.

```
Sub GetSales Total ()
Dim xl As Excel. Application
Dim sAmount As String

Set xL= CreateObject ("Excel. Application")
XL.Workbooks. Open ("C:\MYDATA\Sales.XLS")
sAmount= Str (xl. ActiveSheet.Range ("Сотув ҳажми"). Value Active
Dokument. Bookmarks ("Сотув ҳажми). Select
Selection.InsertAfter ("$" + Trim$ (sAmount))
Set xl=Nothing
End Sub
```

6. Wordга қайтинг ва макросни ишга туширинг. Агар код ёзишда хатоликка йўл қўйилмаган бўлса, ҳужжатингизда сотув ҳажми пайдо бўлади.

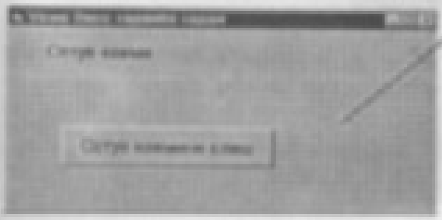


Visual Basicда Office ни ишлатиш

Visual Basic автоном версиясида сиз Office дастурларининг биридан бошқасини (Wordдан Excel ни) бошқарибгина қолмасдан, балки исталган Office элементини бошқариш имконига эгасиз. Қуйида келтирилган мисолда Word VBA

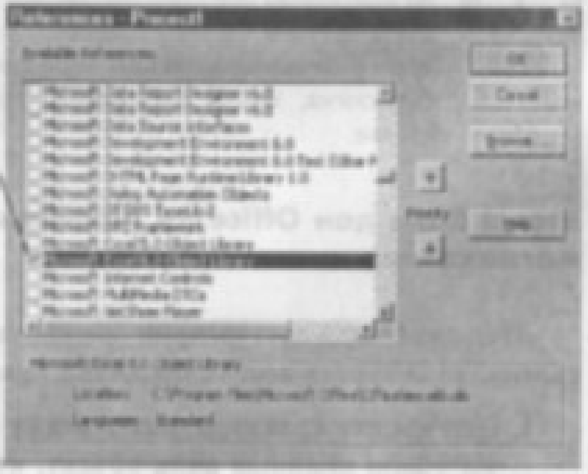
учун ёзилган кодни Visual Basic нинг автоном версияси учун қўллаймиз.

Ҳозирги кунда фақат Office дастурларигина эмас, балки бу системани қўллаб қувватлайдиган бошқа дастурларни ҳам автоматлаштириш мумкин.



1. Visual Basic тўлиқ версиясини ишга туширинг ва янги лойиҳа тузинг. Шаклга расмда кўрсатилгандек, буйруқ тугмаси ва ёзув жойлаштиринг.

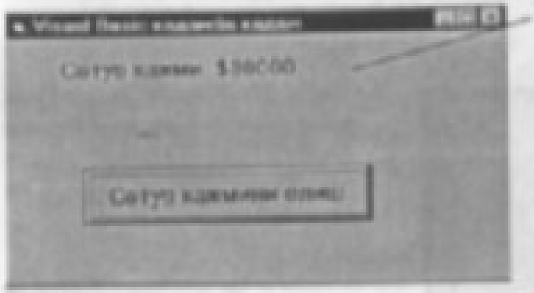
2. Project (лойиҳа) менюсининг Reference (Йўлланма) сатрида Microsoft Excel 8.0 Object Library га йўлланма кўрсатилганлигини текшириб кўринг. Кейин диалогни ёпинг.



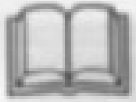
3. Тугмачанинг Click (босиш) ҳодисаси учун қуйидаги кодни ёзинг:

```
Dim xl As. Excel. Application
Dim sAmount As String
Set xl-CreateObject ("Excel. Application")
xl. Workbooks. Open ("C:\MYDATA\Sales.XLS")
sAmount= +Str (xl.ActiveSheet. Range ("Сотув ҳажми").Value)
Label1. Caption= "Сотув ҳажми: "&" "$" Trim $ (sAmount)
Set xl = Nothing
```

Бундай кодни ишлатишдан олдин, системангизни имкониятларини текшириб кўришингиз зарур. Чунки Excelни фон режимида ишга тушириш системанинг иш унумини пасайтиради. Бошланғич даражадаги системаларда эса умуман ишламаслиги ҳам мумкин.



4. Дастурни ишга туширинг ва тугмани босинг. Visual Basic Sales.XLS китобидаги қийматни ёзув-бошқарув элементига чиқаради.



Дастур бир нечта объектларга мурожаат қилган ҳолатда ишлаганда аввалига адашишингиз мумкин. Лекин тажриба ортиши билан бу муаммо йўқолади. Агар шундай тушунмовчиликлар юзага келишининг олдини олмоқчи бўлсангиз, Visible хусусиятини True ҳолатига ўтказиб қўйинг.

Visual Basic дан Office ни ишлатиш ҳақида маслаҳатлар

Агар сиз бошқа дастурларни бошқарувчи кодга назар солсангиз, унда тўртта асосий босқични кузатишингиз мумкин:

1. Dim калити ёрдамида ўзгарувчини эълон қилиш. Доимо эълон қилишда аниқ объект, яъни умумий объект "As Object" ўрнига айнан "As Excel.Application"ни қўлланг, бу ўз навбатида дастур иш унумини оширади.

2. Объект ўзгарувчисини тўғри ўрнатинг, яъни унинг нусхасига ҳам мурожаат қилинг. Буни амалга ошириш учун Visual Basic иккита мукобил функциядан фойдаланади. Create Object функцияси ишлаб турган объектга мурожаат қилади. Get Object функцияси файл номини кўрсаткич сифатида ҳам олиш имкониятига эга, бу эса ўз навбатида вақтни тежаш имконини беради.

3. Объектнинг услуби ва хусусиятларига мурожаат. Бу ерда ишнинг асосий қисми бажарилади.

4. Объектга Nothing ўзгарувчисини бериш, бу ўзгарувчидан фойдаланилса, объект ишлатилмаган тақдирда уни хотирадан ўчириш имкони яратилади. Баъзи ҳолларда бу қадамни бажаришга зарурият бўлмаслиги мумкин, лекин ундан фойдаланиш дастурлашда яхши услуб ҳисобланади.

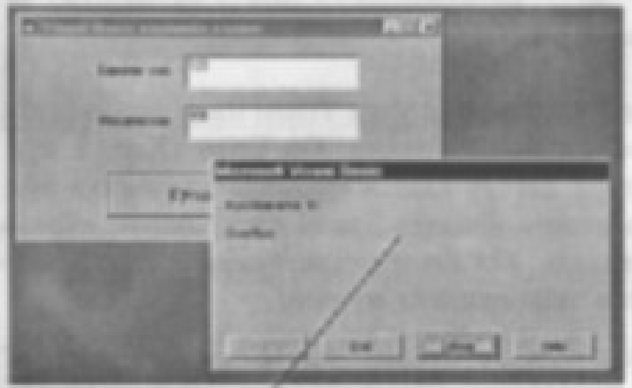
ЕТТИНЧИ БОБ

Малакали дастурчиларга маслаҳатлар

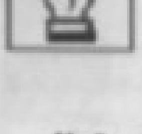
Visual Basic да дастур яратиш бир иш, яхши дастур яратиш эса бошқа иш. Бу бобда тез ва қулай ишловчи дастур яратиш асослари, хатоликларни қамраб олиш усуллари ва шу каби бошқа маълумотлар келтирилган.

Хатоликларни қамраб олишга кириш

Бошқа муқобил дастурлаш тизимларига нисбатан Visual Basic муҳити фойдаланувчи учун хавфсизроқ муҳит ҳисобланади. Visual Basic системани камдан кам ишдан чиқариши мумкин. Лекин бу ерда ҳам хатоликлар рўй бериши учун етарлича сабаблар мавжуд. Булар код ёзиш жараёнида йўл қўйилган ички хатоликлар бўлиши мумкин.



Visual Basic ни ишга туширишда юзага келган хатолик.



Агар код ёзишда хатоликка йўл қўйилган бўлса, дастур ишга туширилиши жараёнида Visual Basic ўз ишини тўхтатади, дастурнинг бу тарзда ишлаши фойдаланувчилар учун, айниқса. Улар ўз иш натижаларини хотирада сақланмаган ҳолларда мақбул эмас.

Қуйидаги кодни кўриб чиқиб, йўл қўйилган хатоликнинг манбаини кўришингиз мумкин:

```
Dim myint As Integer
myint= Val(Text1.Text)* Val (Text2. Text)
Label1.Caption= "Жавоб:"+Trim(Str$(myint))
```

Бу мисолда Myint тури Integer туридаги ўзгарувчи каби эълон қилинган. Лекин Myint унча катта бўлмаган сонлар, яъни 32767 гача бўлган сонлар билан ишлайди, агар кўпайтириш натижаси 32767 дан ортиб кетса, Visual Basic 6 рақамли (Error 6) хатоликни кўрсатади. Бу сон хоналарининг ошиб кетганлигини кўрсатувчи хатолик ҳисобланади. Код ёзишда Myint ни Long кўринишда ёзсангиз ҳам, сон етарлича катта бўлса дастур хатолик ҳақида хабар бериши мумкин.

On Error оператори

Юқоридаги муаммони ечиш учун Visual Basic да On Error операторидан фойдаланиш мумкин. On Error (Хатога йўл қўйилса) оператори дастур иши жараёнида хатоликка йўл қўйилса, дастур ишини қаердан давом эттиришини кўрсатиш учун ишлатилади. On Error операторининг иккита вариантидан бирини танлашингиз мумкин.

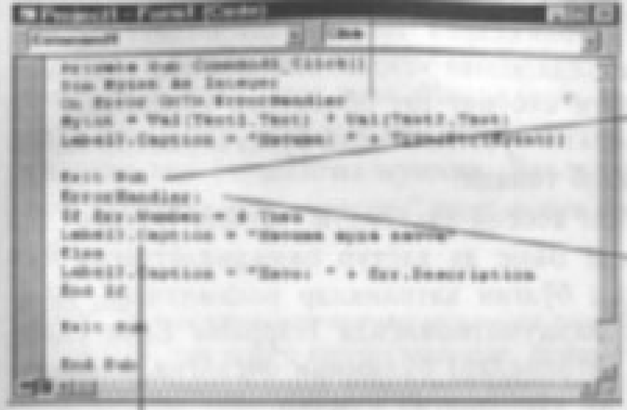
On Error Resume Next оператори хатоликни ўтказиб юборади ва дастурнинг хатоликдан кейинги сатри бажарилаверади.

On Error Goto Label оператори label белгисига ўтади ва ишни шу ердан бошлайди. Юқорида келтирилган дастур ишини яхшилаш учун кодга хатоликларни тузаткич элементни қўшиб қўйиш зарур.

"Хатоликларни тузатгич" (error-handler) — код бажарилиши жараёнида хатоликка йўл қўйилган тақдирдагина ишга тушувчи код. Унинг вазифаси хатоликни юзага келтирилган вазиятни ҳал қилишдир.

Кейинги бўлимда Error объекти ҳақида тўлиқ маълумот келтирилган.

Бу сатр хатоликларни тузатгичга мурожаат қилади



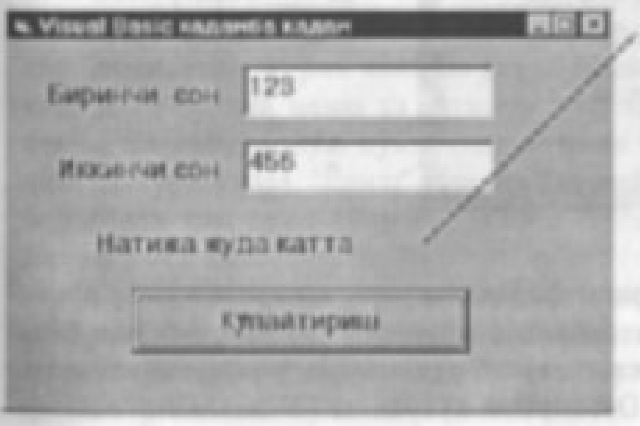
Агар хатоликлар бўлмаса, бу сатр жараён охирига етганини билдиради ва хатоликларни тузатгич кодини ишга туширмайди.

Бу сатр белги [метка] вазифасини ўтайди [сатр охиридаги икки нуқтага эътибор беринг], у хатоликни тузатгичнинг бошланишини белгилайди.

Number хусусияти текширилган, мос равишда, яъни Error объектининг хатолик характерига қараб маълумот олинади.

"Хатоликни тузатгич" ни синаб ишга тушириш

Коддаги хатолик бартараф қилингач, дастурни ишга туширинг:

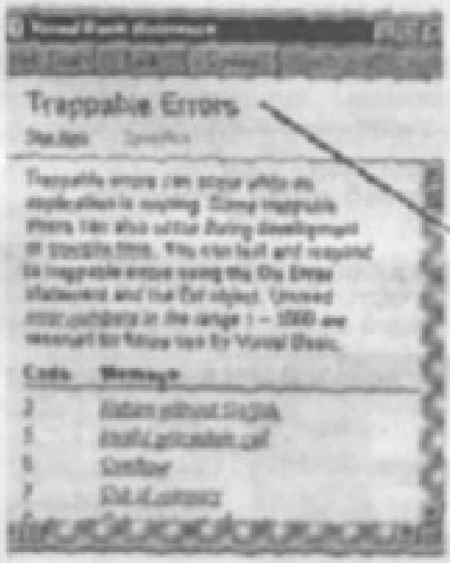


Хатоликка дуч келгач, дастур ўз ишини тўхтатмасдан, хатолик характери ҳақида ахборот беради ва ишини давом эттиради.

Err объекти ҳақида муфассал ахборот

Visual Basic хатоликларни топувчи Err объектига эга. Дастур иш жараёнида юзага келган хатоликлар, уларнинг характери ҳақидаги ахборот Err объектида тўпланади. Ўз навбатида "Хатоликни тузатгич" коди бу маълумотни қайта ишлаб, тўғри жавоб топади.

Err объектининг асосий хусусияти Number (Номер) хусусиятидир. Visual Basic да дастур бажарилаётганда йўл қўйилиши мумкин бўлган хатоликлар рақамланган. Агар сиз Visual Basic маълумотномасида Trappable Error (Қамраб олинадиган хатоликлар) бўлимини очсангиз, хатоликлар бош рўйхатини кўришингиз мумкин.



Visual Basic маълумотномасида хатоликлар рўйхати

Err объектининг фойдали хусусиятларидан яна бири — Description (Хатоликнинг тавсифи) ҳақида ахборот беришидир. Хатоликлар тартиб рақами сизга зарур маълумотни бера олмайди. Description хусусияти — хатонинг тавсифи ҳақидаги қисқача маълумотдир. Баъзи ҳолларда бу маълумот ҳам камлик қилиши мумкин.

Хатоликларни текширувчи код ёзганда ҳақиқатга яқин хатоликларни, яъни амалиётда кўп учрайдиган хатоликларни текширадиган код ёзиш зарур.

Баъзи ҳолларда хатоликларни тузатгич хатолик ҳақида ахборот бериши шарт эмас. Мисол учун, хатолик унча аҳамиятга эга бўлмаслиги ёки уни фойдаланувчи ёрдамисиз тузатадиган код ёзишингиз мумкин. Баъзан дастурнинг ишлаши "хатоликларни тузатгич" коди ишига боғлиқ бўлиши мумкин.

Хатоликлар ҳақида маълумот очиқ бўлиши зарур. Хатолик юзага келган тақдирда, фойдаланувчи нима қилиши зарурлигини кўрсатинг, имконият бўлса, техник ёрдам телефон рақамларини ҳам бериб қўйинг.

Фойдаланувчига ёрдам

Хатоликка йўл қўймасликнинг энг яхши йўли — бу фойдаланувчига хатоликка йўл қўйиш имконини бермасликдир.

1. Enabled ва Visible хусусиятидан фойдаланинг

Дастурларнинг кўпчилигида шундай тугмалар ва менюлар борки, улар жорий ҳолатда керак бўлмаслиги ҳам мумкин. Агар бундай меню ва тугмани ноқулай вазиятда танланса, хатоликка йўл қўйилиши мумкин. Айтайлик, сизнинг дастурингизда "Ёпиш" тугмаси бор, агар ҳужжатлардан бирортаси ҳам очиқ бўлмаса, "Ёпиш" тугмаси ҳеч қандай маънога эга эмас.

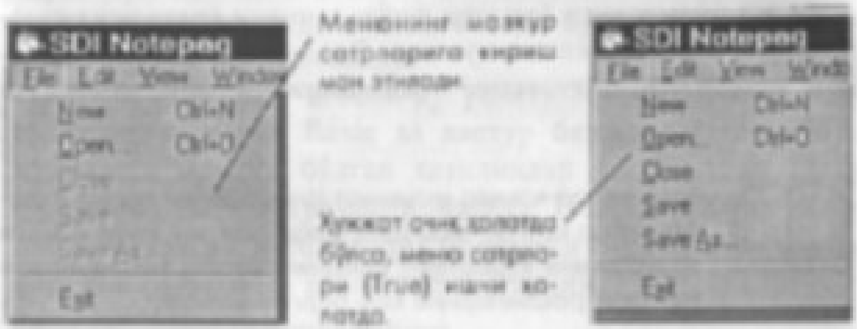
- Бу масалани ечиш учун, потенциал хавфли меню ва тугмаларни яшириш ёки уларни ишламайдиган қилиш зарур. Айтайлик, сизнинг кодингизда, "Очиш" тугмаси учун қуйидаги сатр ёзилган бўлсин:

```
mnuFileClose.Enabled = True
```

Агар охирги хужжат ёпилган бўлса,

```
mnuFileClose.Enabled = False
```

коди бажарилади. Агар сиз меню сатрини йўқотмоқчи бўлсангиз, Visible (Кўринувчи) хусусиятидан фойдаланинг.



Visible хусусиятини дастурнинг жорий ҳолатига таъсир қилмайдиган меню сатрлари ва тугмалар учун қўлланг. Enabled хусусиятининг ҳолати ўзгартирилса (бирорта ноқулай ҳолат юз берса), фойдаланувчи ёпиб қўйилган меню сатрларига мурожаат қилиш имкониятлари ҳақида ахборот топиши мумкин. Масалан, Edit>Cut (Тузатиш, кесиб ташлаш) Visible хоссасини эмас, балки Enabled хоссасини қўллаш учун яхши номзоддир.

2. Маълумотнома, жараён йўлкаси ва ҳолат сатридан фойдаланинг

Сичқонча тугма ёки менюга олиб борилганда пайдо бўладиган қисқача маълумот фойдаланувчига жуда қулай. Элементларнинг деярли барчаси ToolTipText (Асбоб номини кўрсатувчи матн) га эга.

Дастур иш жараёни давомида бир жараён анчагина вақт ичида бажарилса, шу жараён бажарилиши давомида сичқонча кўрсаткичини қумсоатга ёки бошқа шу каби элементга алмаштириб қўйинг. Фойдаланувчи бирорта жараён бажарилиши вақтида маълум бир харакатни кўриши зарур, акс ҳолда дастур ишдан чиққан деб ўйлаши мумкин.

Шакл, бошқариш элементи ёки Screen объекти учун сичқонча курсори шаклини ўзгартириш Mouse Pointer хусусияти ёрдамида амалга оширилади.

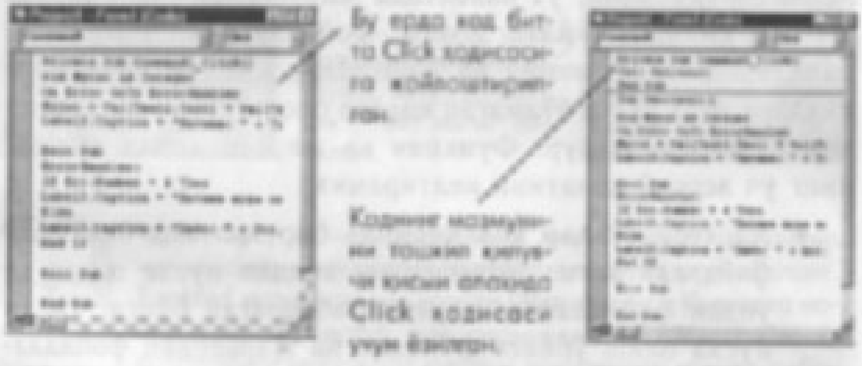
Код қандай ташкил қилинади

Дастурингиз маълум даражада мураккаб дастурга айланиши билан, кодни қандай ташкил қилишни пухта ўйлаш лозим. Функцияни, ўз навбатида жараёнларни қай тарзда ташкил қилишни ўйлаш лозим. Фойдаланувчи томонидан аниқланадиган функция ва жараёнлар маълум кетма-кетликдаги унча катта бўлмаган кодлар блоки тариқасида ташкил қилиниши зарур. Функция ва жараён хосил қилишнинг уч асосий ҳолатини келтирамиз:

- Дастурнинг ҳар хил қисмида бир хил код блокидан фойдалансангиз. Агар сизни коддан нусха олиш ва ундан фойдаланиш қизиқтириб қолган бўлса, ундан нусха олиш ўрнига функция ва жараёндан фойдаланинг. Коднинг биринчи нусхасида хатоликка йўл қўйилган бўлса, ундан фойдаланадиган барча кодларни тузатиш лозим. Бир коддан кўп фойдаланиш, кодни биринчи ёзишда йўл қўйилган хатоликни қайтариш демакдир.
- Дастур ёзишда катта код блоклари хосил қилмаслик зарур, чунки катта блокларни кўриб чиқиш қийин, ўз навбатида хатоликка йўл қўйиш эҳтимоллиги ҳам ортади.
- Кодга янги ўзгартириш киритиш ва соддалаштириш учун, айтайлик, сизнинг дастурингиз диаграмма чизадиган бўлимга эга бўлса, бу бўлимни ишга тушириш учун менюдан ёки тугмадан фойдаланиш лозим. Бу вазифани бажарадиган кодни Click (Босиш) ҳодисаси учун ёзиш мумкин, лекин бу фойдали эмас. Қулайлик учун битта DrawChart (Диаграмма чизиш) жараёни-

ни ёзиш мақсадга мувофиқдир. Бу жараённи шаклнинг умумий қисмида ёки алоҳида модулда жойлаштириш мумкин. Алоҳида модулда жойлаштирилган кодни ўзгартириш осон.

Кодни алоҳида модулда сақлашнинг қулайлиги шундаки, сиз функциялар кутубхонасини ташкил қилишингиз ва зарур ҳолатда бу функцияларга мурожаат қилишингиз мумкин.



Кодни қандай қилиб тушунарлироқ қилиш мумкин

Лойиҳа яратиш жараёнида сиз кўплаб кодлар ёзишингиз мумкин, лекин уни тузатиш, йўл қўйилган хатоликларни бартараф қилиш вақти келганда бирмунча қийинчиликларга дуч келишингиз мумкин. Бу қийинчиликларнинг олдини олиш учун баъзи маслаҳатларни келтирамиз.

1. Кўпроқ бўш жой қолдиринг

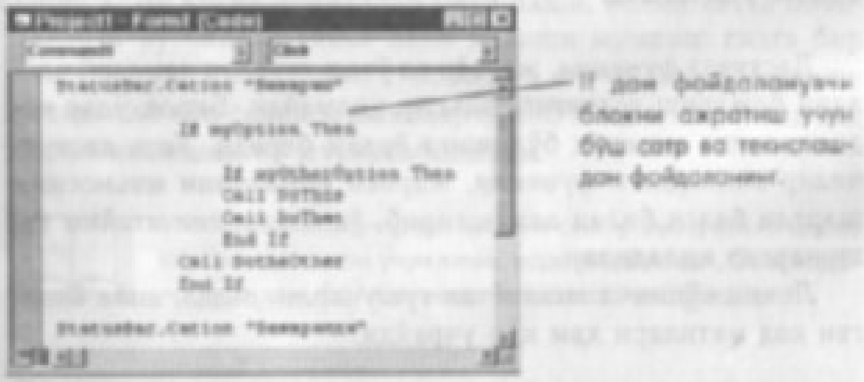
Код ёзиш жараёнида маълум бир жараёнларга мўлжалланган мантиқий блоклар учун сатрлар ёки бўш жойлар қолдириш мумкин. Коднинг ичидаги бўш жойларни Visual

Basic ўтказиб юборади, улар деярли жой ҳам эгалламайди. Ёзишнинг бу усули коднинг мантиқий блоклари билан алоҳида ишлаш имконини беради.

Функция, жараён ва ўзгарувчилар учун танланадиган ном ҳам аниқ равшан бўлиши лозим. Баъзи маслаҳатларни кейинги бўлимдан қаранг.

2. Блокларни диққат билан ростланг

Кодни яна ҳам тушунарлироқ қилиш, кўп даражали If ёки Select Case блокларида чалғиб кетмаслик учун ҳар бир янги даража блокини текисланг ва у бутун код бўйича зидиятсиз бўлсин.



3. Шарҳларни кўпроқ ёзинг

Код ёзишда шарҳлар ҳам бўш жой каби дастур ишига халақит бермайди, бинобарин дастур ишини секинлатмайди.

Аслида функция ёки жараён ёзишда уларнинг ҳар бири қандай иш бажариши шарҳлаб берилади. Агар гуруҳ томонидан ёзиладиган дастур бўлса, ҳар бир функция нима иш бажариши, тузатиш вақти, ким томонидан тузатиш кири-

тилган ва шу каби бошка зарур маълумотларни изоҳ сифатида ёзиб қўйган маққул.

Дастур яратиш жараёнида маълум бир муаммони ҳал қилиш учун код ёзсангиз, яъни мавжуд кодларга яна қўшимча код киритсангиз, у ҳолда албатта бу коднинг вазифаси ва уни нима учун киритилганлиги ҳақидаги маълумотларни ёзиб қўйиш зарур. Акс ҳолда гуруҳдаги бошка дастурчилар ўзи билмаган ҳолда номаълум кодни ўчириб юбориши мумкин.

Бир элементни қўшиш ёрдамида ёзилган ҳар қандай маълумот шарҳдир. Сиз шарҳни, шунингдек, сатр охирига яна ҳам қўшишингиз мумкин.

Ном танлаш

Дастурда функция, жараён ва ўзгарувчилар номлари иловани бажариш тезлигига таъсир қилмайди. Бироқ улар кодингизнинг аниқроқ бўлишига ёрдам беради. Уста дастурчилар кодлардаги функция, жараён номларини маъносини шартли белги билан алмаштириб, ўз кодларини атайин тушунарсиз қиладилар.

Лекин кўпинча амалиётда тушунарли, содда, аниқ ёзилган код матнлари ҳам кўп учрайди.

1. Маъноли ном танланг

Қисқа номлардан кўра маъноли номлар маъқул.

Масалан (номерли Дўкондан Даромад олиш) функциясини мана бу қисқа вариант

GetIncDosLong()

дан кўра бундай номлаш яхшироқ:

Get Income From Shoped odong()

2. Ном келишувлардан фойдаланинг

Ном танлаш схемалари ном келишувлар деб аталади. Масалан, объект ёки ўзгарувчи ички зиддиятсиз префиксда фойдаланиб идентификациялаш фойдалидир.

str Something	String тури учун
iSomething	Integer тури учун
sgl Something	Single тури учун
txt Something	матнли блоклар учун

Бу ёндашувнинг афзаллиги шундаки, Сиз, масалан, бутун турдаги ўзгарувчига силжувчи нуқтали сонларни ўзлаштириш каби хатоликлар камроқ ўйлайсиз. Бу китобдаги мисолларда бошқариш элементлари учун Visual Basicда ўрнатилган номлардан фойдаланилади, масалан, Command 1 ёки Lobel 1. Бу қисқа лойиҳалар учун яхши, лекин катта лойиҳаларда бу чалкашликка олиб келиши мумкин: сизга бир хил Command 1 ва Lobel 1 номларга эга бўлган бошқариш элементли кўп сондаги шакллар бўлиб қолди, натижада код билан ишлашингиз мураккаблашади.

Visual Basic ҳарфлар регистрига сезгир эмас, лекин шунга қарамасдан, ном учун кичик ҳарфлардан ҳам, бош ҳарфлардан ҳам фойдаланиш мақсадга мувофиқдир.

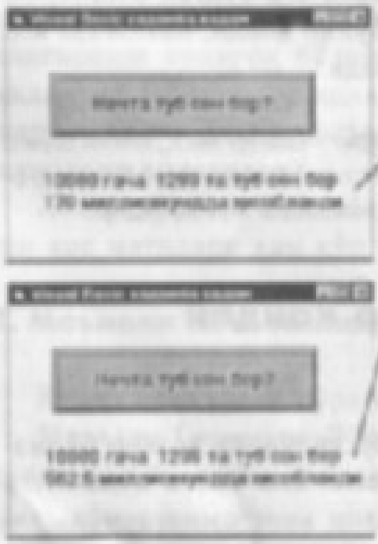
Тез ишловчи дастурлар қандай яратилади?

Сиз фойдаланаётган компьютер қанчалик тезкор бўлмасин, яратилган дастур унумдорлиги ҳам катта аҳамиятга эга. Агар яратадиган дастурингизни кенг оммага мўлжалласангиз, дастур тезлиги ҳақида кўпроқ ўйлашга тўғри келади, чунки фойдаланувчиларнинг барчаси ҳам тезкор системага эга эмас. Қуйида шу борада бир нечта маслаҳатлар келтирамиз.

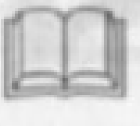
1. Барча ўзгарувчиларни эълон қилинг

Дастур тузиш жараёнида ўзгарувчиларни Dim калити ёрдамида эълон қилмасангиз, Visual Basic эълон қилинмаган ўзгарувчи тарзида Variant турини тақдим этади. Variant ўзгарувчиси бошқа ўзгарувчиларга қараганда секин қайта ишланади. Демак, лойиҳа ишини тезлаштиришнинг энг оддий йўли барча ўзгарувчиларни, ҳар бир ҳолда ярокли турдан фойдаланиб, ошкор эълон қилишдир.

Visual Basic компилятори барча ўзгарувчиларни эълон қилишни талаб қилиши учун Option Explicit операторини мавжуд кодларга киритиш зарур. Менюнинг Tools>Options [Сервис> Параметры] қисмида Require Variable Declaration байроқчасини белгилаб қўйинг ва шунда Option Explicit оператори эълонлар бўлимига барча янгидан яратиладиган модуллар, шакллар ва класслар учун автоматик тарзда қўйилади.



Бу иловада дастур (илова) туб сонларни топади. Бу мисолларнинг коди бир хил бўлишига қарамасдан, юқоридаги мисол пастдагига нисбатан уч марта тез бажарилади. Сабаби иккала мисолдаги кодлар бир хил бўлса-да, секинроқ ишлайдиган версияда ўзгарувчилар ошкор эълон этилмаган эди.



Компьютер бутун сонларга тезроқ ишлов бергани учун имкони борича бутун сонлардан фойдаланинг.

2. ActiveX бошқариш элементларидан кам фойдаланинг

ActiveX бошқариш элементлари тез ишлаши мумкин, лекин улар секин юкланади. Яна бир муаммо шуки, ActiveX нинг баъзи бошқариш элементлари Сизга керагидан анча кенгроқ имкониятларга эга бўлиши мумкин. Шу сабабли дастур ҳажмини ихчамроқ ва унинг ишини тезлатиш учун ActiveX бошқариш элементлари бу аслида ҳам зарур бўлган ҳоллардагина фойдаланинг.

3. График ойналардан иложи борича фойдаланинг

График ойна бошқариш элементи расм бошқариш элементига ўхшаш бўлганлиги билан унинг ички тузилиши анча мураккабдир. Расм бошқариш элементи бир қатор масалаларни ечишда кўпроқ самаралидир.

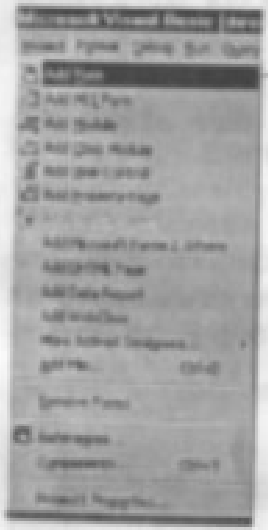
4. Бошқариш элементи ўрнига имкони борича коддан фойдаланинг

Шаклда чизиқ ёки расм чизишнинг икки усули мавжуд. Бир усул — бу тегишли бошқариш элементидан (чизиқ ёки расм) фойдаланиш. Иккинчи усул эса график усуллардан, масалан, Line ва Cercle дан фойдаланиш.

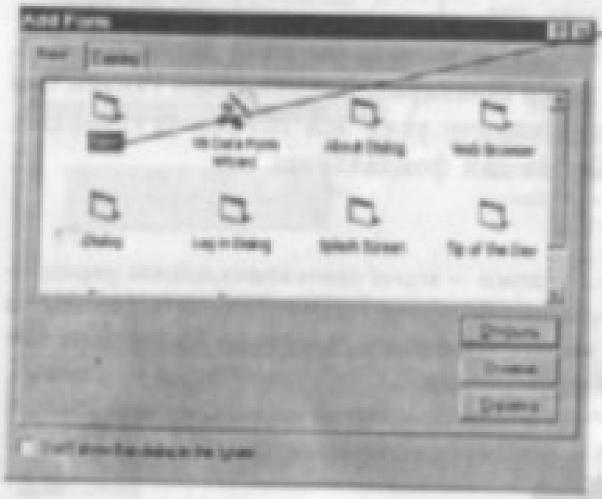
Бажариш тезлиги — ягона омил эмас: меҳнат унумдорлигини ҳам ҳисобга олиш лозим. Кўпинча қисқа вақтда ёзилган ишончлироқ кодни ёзиш унинг бажарилиш тезлигидан муҳимроқдир.

Бир нечта шакллар ишини ташкил қилиш

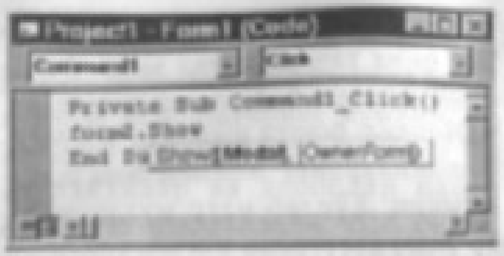
Мазкур китобда келтирилган мисолларнинг кўпчилигида фақат бир шаклдан фойдаланилади. Тўлиқ комплекс масалаларни еча оладиган дастур яратиш учун бир нечта шакллар билан ишлашга тўғри келади.



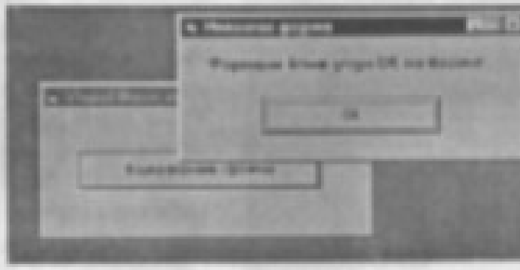
1. Project (Лойиҳа) менюсидан Add Form (Шакл қўшиш) сатрини танланг.



2. Керакли турдаги шаклни танланг. Бу ойнадаги кўплаб уста ва сҳоналардан ўзингиз учун зарурини танлаб олишингиз мумкин. Мазкур расмда стандарт шакл танланган.



3. Янги шаклни кўриш учун Show шаблонинг усулидан фойдаланинг. Бу усул иккита параметр олади: улардан биринчиси Visual Basic: vbModal ёки vbModeless — бу параметр дастурингиз кейинги ҳаракатини бошқариш учун шаклнинг ёпилишини кутишини билдиради.



4. Дастур иши жараёнида сиз шаклни бир нечта усул билан ёпишингиз мумкин: ҳар юқори томондаги Close тугмасини босиш; кодда Unload (юксизлантириш) буйруғини бериш; Hide (Яшириш) усулидан фойдаланиш.

Шаклни қандай ёпган маъқул — Hide ми ёки Unload дан фойдаланишми?

Иккала ҳолатда ҳам шакл ёпилади, лекин бу усуллар ҳар хил ишлайди. Hide шаклни хотирада қолдириб яширади, бу эса сизга зарур ҳолатда унга мурожаат қилиш ёки қисқа вақт ўтгандан сўнг зарур бўлганда шаклни экранга чиқариш имконини беради. Unload шаклни хотирадан чиқариб ташлайди, бу усул система имкониятларини тежашда қўл келади.

MDI га кириш

Бир нечта шакллардан фойдаланадиган дастурлар, айникса, номодал режимда тартибсиз кўринади. Шакл бошка шаклга ўтилганда экранда колади. Агар системада бир вактнинг ўзида бир нечта дастурлар очилган бўлса, кайси шакл кандай дастурга тегишлилигини ажратиш қийин бўлиб колади. Бундай холатларда MDI (Multiple Document Interface —

кўп ҳужжатли интерфейс) бошқариш усулидан фойдаланган маъқул. MDI — бу бош ойна ташқарисида эмас, балки ичида қамраб олинадиган ойналарни бошқариш усулидир. Visual Basicда бош ойна MDI-шакл, бошқа шакллар эса MDI-мижозлар деб аталади, бу шаклларнинг жойини бош шаклнинг ичида ўзгартириш мумкин. Кўплаб матн муҳаррирлари ва электрон жадваллар MDI-дастурлар ҳисобланади.

MDI-дастур битта ёки бир нечта шакллардан иборат бўлиши мумкин. Расмда намуна сифатида Visual Basic пакети таркибига шу соллардан бири сифатида кирадиган MDI-дастур келтирилган.

MDI-шакллар билан ишлаш

Мазкур китобда MDI ни тўлиқ тавсифлашга жой камлик қилади, лекин имконият даражасида тавсифлашга ҳаракат қиламиз. Янги дастур яратинг. Кейин менюнинг Project (Лойиҳа) қисмидан Add MDI Form (MDI-шаклни қўшинг) сатрини танланг. Кейин эса биринчи шаклнинг MDI хусусиятини MDIChild (ички шакл)га ўзгартиринг. Дастурни ишга туширсангиз, MDI-шакл, унинг ичида эса унга боғланган ички шакл пайдо бўлади.

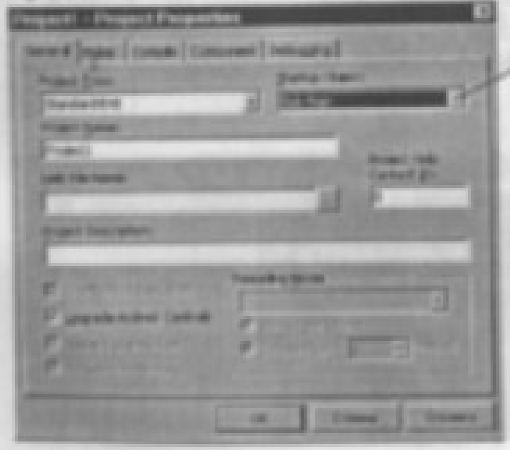
MDI-шакллар баъзи махсус имкониятларга эга. MDI-шаклга бевосита график ойна ва махсус бошқариш элементларини жойлаштириш мумкин. Бу элементлар Align (Текислаш) хусусиятига эга. Улар ўзларини бир томондан текислайдилар ва асбоблар панели ёки ҳолат сатрлари сифатида ишлатилади. Шу билан бирга сиз MDI-шаклга меню пунктларини ҳам жойлаштиришингиз мумкин.

MDI-дастурларнинг асосий мураккаблиги шундан иборатки, ишга туширилган код "тўғри" ойна билан, яъни кодга бевосита боғлиқ ички ойна билан ишлаши зарур. MDI-шакллар ActiveForm хусусиятига эга.

Visual Basic да дастурлар устасининг бир бўлими MDI-дастурлар (Application Wizard).

Sub Main ни қўллаш

Visual Basic да барча лойиҳалар дастур ишга тушиши билан бажариладиган бошланғич объектга эга бўлади. Бошланғич объект Project Properties (Лойиҳа хусусиятлари) диалог ойнасида ўрнатилади ва у лойиҳанинг исталган шакли ёки махсус Sub Main номига эга бўлган жараён бўлиши мумкин.



Бошланғич объект сифатида биринчи шаклни эмас, балки Sub Main ни ўрнатиш учун расмда кўрсатилган тугмани босинг.

Sub Main ни мустақил равишда яратишингиз зарур. Олдинига қуйидаги код модулини жойлаштиринг. Кейин модулга Sub Main номи билан жараён қўшинг. Одатда у шаклни чиқариш учун буйруқни ҳам ўз ичига олади.

Sub Main нима учун керак?

Мураккаб дастурларда Sub Main нинг қўлланилиши бир қанча қулайликларга эга. Одатда Sub Main қуйидаги ишларни бажаради:

- ўзгарувчиларга ўрнатилиши бўйича қиймат берилади;
- маълумотлар базаси очилади;
- дастур файлларининг тўлиқлигини текшириб чиқади ва дастурнинг бошланғич элементи шаклини ишга туширади.

Экран-ёрлиқни ишга тушириш

Sub Main жараёни экран- ёрлиқни (Splash Screen) ишга туширишни осонлаштиради. Экран- ёрлиқ — бу дастур ишга тушиш чоғида акс этадиган шаклдир. Агар дастурнинг ишга тушиши маълум вақт талаб қилса, экран-ёрлиқ фойдаланувчига жараён тўғри кечаётганлиги ҳақида ахборот беради. Экран — ёрлиқни ишга тушириш учун Sub Main бошланғич қисмига қуйидаги кодни қўшиб қўйинг:

```
frmSplash.show
DoEvents' шакл ясашга ёрдам беради.
```

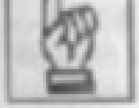
Дастур ишга тушгандан кейин экран-ёрлиқни хотирадан чиқаринг:

```
unload frm Splash
form1.show' дастурнинг биринчи шаклини экранга чақиради.
```

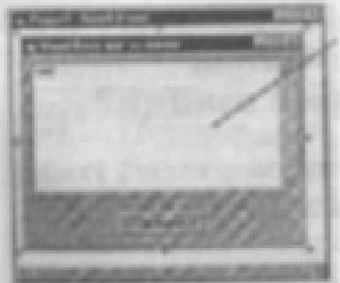
Sub Main дан экран-ёрлиқни ишга тушириш учун фойдаланиш шарт эмас. Агар сиз яратган дастур шаклларга эга бўлмаган тақдирдагина Sub Mainсиз ишлаши мумкин эмас. Лекин яратган дастурингиз маълум тартибда, унинг бўлишини хохласангиз, бошланғич кодни Sub Mainга жойлаштиринг. Load ходисасидан эса бевосита шаклни юклаш учун фойдаланинг.

Дискдаги файл қандай ўқилади?

Яратиладиган компьютер дастурларининг кўпчилигини дискдан ўқиш ёки дискка ёзиш зарур. Кўп ҳолларда бу жараён фойдаланувчига кўринмайди, яъни сиз Visual Basic дан фойдаланаётган бўлсангиз, Visual Basic дискдаги маълумотлар базаси билан ишлашда маълумотлар базасини бошқарувчи ядродан фойдаланади. Бу ҳолда сиз файлларни очиш ёки ёпиш ҳақида ташвишланмасангиз ҳам бўлади. Бироқ ахборотни ўқиш ва дискка ёзишни билиб қўйиш керак бўлиб қолиши мумкин. Масалан, иш жараёнида дастур ташқилий файлларидан бирортасини келгусида ахборотни чиқариш учун дискка кўчириб олиш зарурияти юзага келиши мумкин. Қуйидаги мисол матн файлини ойнага қандай юклашни кўрсатади.



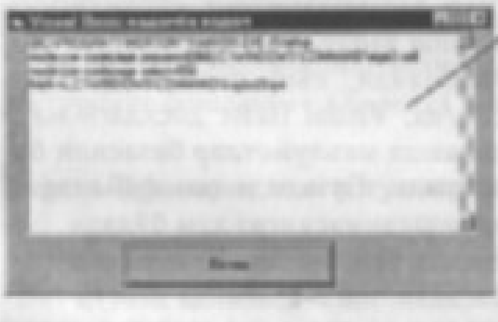
Бу бўлимдаги маълумот сизга оғирлик бўлмаслиги мумкин, лекин бу ерда код ёрдамида файлларни ўқиш имконияти жуда осонлиги кўрсатилган.



1. Янги дастур яратинг, унга матн ойнаси ва буйруқ тугмасини жойлаштиринг. Матн ойнасининг Scrollbars хусусиятини Vertical қийматига, Multiline хусусиятини эса True қийматига ўтказинг.

2. Тугманинг Click (Босиш) ходисаси учун қуйидаги кодни ёзинг:

```
On Error GoTo ErrorHandler
Dim FileNum As Integer
FileNum = FreeFile
Open "C:\AUTOEXEC.BAT"For Input As #FileNum
Text1.Text = Input (LOF(FileNum), # FileNum)
Close # FileNum
Exit Sub
Error Handler:
MsgBox"Хатолик:"&Error Description
```



3. Дастурни ишга туширинг ва тугмани босинг. Матн ойнасида AUTOEXEC.BAT файлининг мазмуни чиқади.

Бу мисолдаги коднинг ишлаши ҳақида бир нечта фикрлар. Visual Basic очиқ файлларни номери бўйича идентификациялайди. Free File функцияси бўш номерни қайтаради, Open ... For Input оператори файлни очади, кейин эса Input функцияси файлни ўқийди ва сатрни қайтаради. LOF (Length of File — файл узунлиги) функцияси Input функцияси ўқиши учун файл узунлигини аниқлайди. Ўқиш операцияси сўнгида Close # функцияси файлни ёпади.

Файлга маълумот қандай ёзилади?

Файлга маълумот ёзиш коди файлни ўқиш кодига ўхшаш. Аслида, файл ўқиш\ёзиш режимида очилганда сиз файлни ўқишингиз ва, зарур бўлса, унга маълумот ёзишингиз мумкин.

Қайд қилувчи файлни яратиш

Бу бўлимда келтирилган мисолда дастурни ишга туширилган ҳамда ёпилган санаси ва вақтини ўзига оладиган қайд файли яратиш намунаси келтирилган.

Энг муҳими файлни тўғри режимда очиш зарур. Агар сиз файлдан иккилик маълумот ўқимоқчи бўлсангиз, файлни (иккилик маълумотларини) ўқиш (Open for Binary) режимида очинг. Шу билан бирга сиз файлни қайси режимда очилишидан қатъи назар маълумотлар базасига мурожаат қилиш режимини ҳам белгилашингиз мумкин.

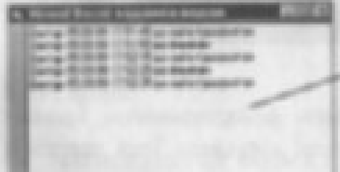
1. Янги шакл яратинг ва Load (Юклаш) ходисаси учун қуйидаги кодни ёзинг:

```
On Error GoTo ErrorHandler
Dim FileNum As Integer
FileNum = FreeFile
Open "C:\EASYSTEP.LOG"For Append As #FileNum
Print # FileNum, "Дастур ишга тушди:"&Str$(Now)
Close # FileNum
Exit Sub
ErrorHandler;
MsgBox" Хатолик: " & Err.Description
```

2. Unload (Хотирадан чиқариш) ходисаси учун Print оператори билан бошланувчи сатрни қуйидаги сатр билан алмаштиринг:

```
Print # FileNum, "Дастур ёпилди:"&Str$(Now)
```

Бу мисол дастур биринчи марта ишга туширилганда қайд қилувчи файл ҳосил қилади. Кейин ишга тушганда эса ахборотни мавжуд файлга ёзиб кетади. Агар сиз файлни фақат ёзиш режимида очсангиз (Open ...for Output), шу ном билан очилган ҳар қандай файл ўчирилади.



Бу ойнада юқорида келтирилган код асосида ишлайди.

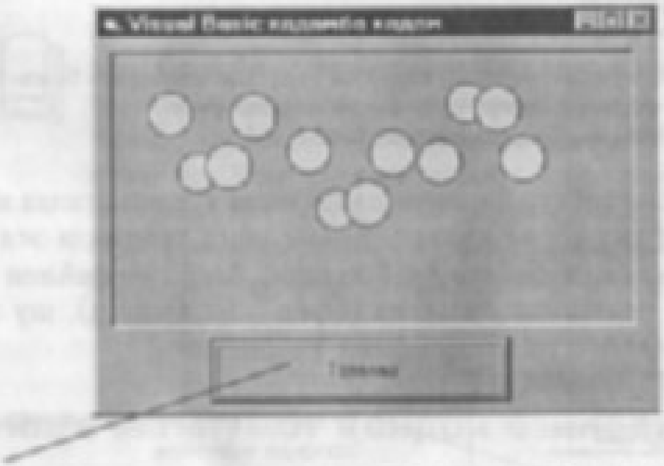
Бу код асосида WriteLog жараёни учун код ёзинг.

График чизиш

Фойдаланувчилар Windows қобиғида ишловчи дастурларда ахборот ва маълумотномаларни график режимда кўришга харакат қиладилар. Диаграмма ёки график қандай чизилади?

График чизиш учун ActiveX бошқариш элементларидан фойдаланиш ёки OLE бошқариш элементига Excel диаграммасини жорий қилиш мумкин. Бу етарлича яхши натижалар бериши мумкин бўлса-да, лекин бошқача усул хам мавжуд. Visual Basic да унинг график усулларидан фойдаланиб шаклларда ва график ойналарга тезда чиройли графиклар ясашингиз мумкин.

График ойнада расм қандай чизилади?



1. Бу мисолда шакл, график ойна ва тугмадан фойдаланилган. График ойнанинг AutoRedraw (Автоматик янгилаш) хусусияти True қийматига ўзгартирилган бўлиши зарур.
   Тугманинг Click ходисаси учун битта сатр ёзилган
   Picture1.cls

2. Mouse Down (Сичқонча тугмаси босилганда) график ойна ходисасига қуйидаги кодни ёзинг:

```
Private Sub
Picture1_MouseDown(Button As Integer,
Shift As Integer, X As Single, Y As-Single)
Picture1.DrawWidth=2
Picture1.CurrentX=X
Picture1.CurrentY=Y
Picture1.FillColor = vbYellow
Picture1.FillStyle= vbSolid
Picture1.Circle Step(0,0),200, vbRed
End Sub
```

3. Лойихани ишга туширинг. График ойнанинг исталган қисмида сичқонча тугмасини боссангиз, график ойнада бўялган доиралар чизилади.

Бу мисолда Click ходисаси ўрнига MouseDown ходисаси ишлатилди. Чунки MouseDown ходисаси объектга нисбатан сичқонча кўрсатган жорий координаталарни қайтаради.

Visual Basic график усуллари (охирги мисолда уларнинг озгина қисмидан фойдаланилди) амалда график ахборотни шаклга ёки график ойнага чиқаришни тўлиқ бошқаришга имкон беради. Қуйида асосий график хоссалар ва усуллардан баъзилари келтирилган.

DrawWidth, DrawStyle

DrawWidth — объектни чизадиган қаламнинг қалинлиги(эни)ни белгилайди. Сон қиймат қанча катта бўлса, чизиқ шунча қалин (энли) бўлади.

DrawStyle — чизиқ турини белгилайди, яъни узлуксиз, штрихли, нуқтали ва х.к.

FillColor

Чизиладиган объект бўяладиган рангни белгилайди. Бунда фақат FillStyle хусусиятига шаффоф "ранг" ўрнатилмаган бўлиши зарур.

> Visual Basic маълумотномасидан RGB функция хақида маълумот олинг. Бу функция рангни тўлиқроқ бошқариш имконини беради.

FillStyle

Бу хусусият ёрдамида шаффоф ёки ношаффоф ранг билан бўяшни шунингдек, бошқа олтита бўяш усулларидан бирортасини танлаш мумкин, булар орасида вертикал ёки диагональ чизиқлар хам бор.

CurrentX, CurrentY

Объект координаталарини белгилайди.

Circle, Line

Бу усуллар берилган турдаги объект (айлана, чизиқ) чизади. Шарт бўлмаган Step параметри эса бошқа CurrentX, CurrentY хоссалари билан бериладиган нисбий координаталар фойдаланилиши лозимлигини кўрсатади.

DrawMode

Бу услуб ёрдамида мавжуд объектларни (ёки тусни) чизилаётган объект билан аралаштириш мумкин. Масалан, бир объект бошқа объектни қоплайди.

AutoRedraw

AutoRedraw хусусияти (автоматик янгиланиш) ёрдамида график объектнинг янгиланиш жараёнини автоматик тарзда бошқариш мумкин. AvtoRedraw False холатига ўтказилган бўлса, расм вақтинчалик характерга эга бўлиб, расмнинг навбатдаги янгилашида ўчирилади. Мисол учун, AutoRedraw хусусияти True холатига ўтказилган бўлса,

Visual Basic ўзгаришни автоматик бажариш масъулиятини ўзига олади. Агар AvtoRedraw False холатига ўтказилган бўлса, у холда тасвирни зарур бўлганда янгилаш учун Point ходисасидан фойдаланиш мумкин.

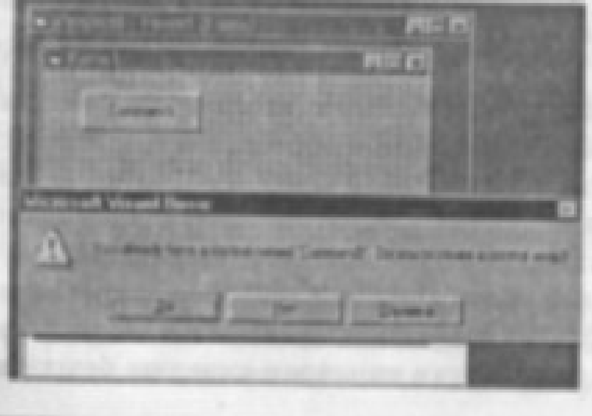
> График усуллар билан иш бошлашда AutoRedraw хусусиятини True холатига ўтказиб қўйинг.

Бошқариш элементлари массивлари

Visual Basic билан ишлаш давомида бошқариш элементини кўчириб кўрган бўлсангиз, шаклга ўрнатаётганингизда Visual Basic қуйидаги саволни беради: "Do you want to create a control array?" (Бошқариш элементи массивини хосил қилмоқчимисиз?) Yes ни танласангиз, шаклда иккинчи бошқариш элементи пайдо бўлади. Иккала бошқариш элементи бир хил номга эга бўлади, лекин уларнинг индекслари хар хил бўлади. Буларни қандай ишлатишни кўрайлик-чи.

Бошқариш элементлари массивининг қўлланиши

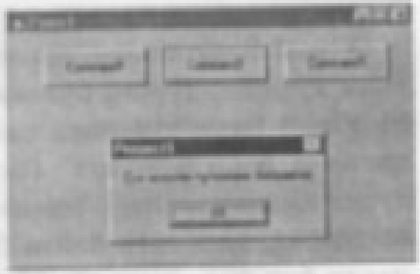
1. Шаклга буйруқ тугмасини жойлаштиринг, уни танланг ва менюдан Edit»Copy (Тахрир»Кўчириш)ни танланг. Кейин Edit»Paste (Тахрир»Ўрнига қўйиш)ни танланг. "Бошқариш элементи массивини хосил қилмоқчимисиз?" саволи пайдо бўлади. Yes тугмасини босинг.



2. Edit>Paste ни яна бир марта танланг, шундан яна бир буйруқ тугмаси пайдо бўлади. Ушбу тугманинг номи бир хил бўлгани билан, массивлар ойнасидан Index хусусиятнинг қийматларини қарасангиз, уларнинг индекслари бошқа-бошқалар.

Ctrl+C — кўчириш учун;
Ctrl+V — бошқариш элементи қўйиш учун; бу иккала комбинациядан ҳам кўчириш учун фойдаланиш мумкин.

3. Тугмачалардан бирини икки марта босиб, Click ходисаси кодини очинг. Тугмачалар бир хил номга эга бўлгани билан уларнинг Index кўрсаткичи ҳар хилдир. Фойдаланувчи қайси тугмани босганлигини аниқлаш учун Index параметридан фойдаланинг. Масалан:



```
Select Case Index
Case 0
MsgBox "Биринчи тугмани босдингиз"
Case 1
MsgBox "Иккинчи тугмани босдингиз"
```
ва ҳ.к.

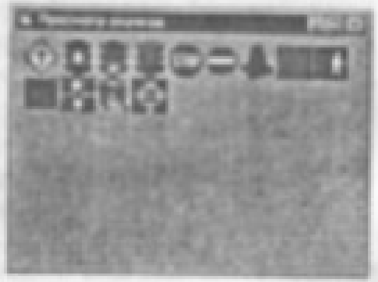
Бошқариш элементлари массиви ҳақида батафсилроқ маълумот

Бошқариш элементлари массивидан фойдаланишнинг икки сабаби бор, улардан бири равшан, иккинчиси эса унчалик эмас; равшан сабаб шундан иборатки — массив элементлари кодининг алгоритми самаралидир. Мисол учун: манзилгоҳлар китобини яратиш дастурида маълум бир ҳарф билан бошланувчи номларни мос равишда тугмачаларга бириктириб қўйиш мумкин. Тугмача босилганда ўрнатилган ҳарфдан бошланувчи номларни қидириш бошланади.

Бошқариш элементлари массивида тугмалар бир хил номга эга бўлгани билан, уларнинг сарлавҳаси ҳар хил бўлиши мумкин.

Менюлар ҳам массивлар кўринишида ташкил қилиниши мумкин. Меню муҳаррирининг Index майдонидан фойдаланиб, менюлар массивини яратинг.

Илова (дастур) бажарилиши жараёнида бошқариш элементларини яратиш



Бошқариш элементлари массивининг қулайликларидан бири, дастур ишлаётган вақтда ҳам бошқариш элементини яратиш мумкинлигидир. Чунки лойиҳалаш даврида дастур учун зарур бўладиган бошқариш элементларининг сонини билмаслигингиз мумкин. Қуйида келтирилган мисолда расм, тугма ва умумий диалог каби бошқариш элементларидан фойдаланилган. Дастурнинг бажарилиш давомида фойдаланувчи бу дастур кодини ёзиши учун жой етмайди, шунинг учун энг муҳим ҳолатларни кўрсатамиз. Зарур бўлган белгиларни тугмача ёрдамида шаклга жойлаштириши мумкин.

1. Олдинги яратилган бошқариш элементи индексини сақлаш учун статик (Static) ўзгарувчидан фойдаланинг.
2. Янги расм — бошқариш элементини яратиш учун NewIdx ўзгарувчисининг қийматини биттага оширинг ва қуйидаги код сатрини ёзинг:

   Load Image1(NewIdx)

3. Янги бошқариш элементини Top ва Left хусусиятларидан фойдаланиб шаклга жойлаштиринг. Янги элементни олдинги элементга ёнма-ён жойлаштириш учун

Image1 (NewIdx) бошқариш элементининг Top ва Left хусусиятлари қийматларидан фойдаланинг. Жорий бошқариш элементи шакл қобиғидан ташқарига чиққан-чиқмаганлигини Scale Width хусусияти қиймати ёрдамида текшириб кўринг. Агар чиқмаган бўлса, расмларнинг янги сатрини бошланг. Агар жорий бошқариш элементи шаклнинг чегарасидан чиқиб кетгудек бўлса, у ҳолда ишни бекор қилинг ва фойдаланувчига бу ҳақда ахборот берувчи сатр кодини ёзинг.

4. Охирида белгини умумий мулоқот ойнаси ёрдамида юкланг ва унинг Visible (кўринадиган) хусусиятини True ҳолатига ўтказиб қўйинг.

Бу масалани ечишнинг бошқа йўли шаклга кўринмайдиган бошқариш элементларини жойлаштиришдан иборат. У ҳолда керакли бошқариш элементларини зарур бўлган ҳолатда кўринадиган (Visible) қилиш керак. Бироқ бошқариш элементлари массивидан фойдаланиш бошқариш элементларини сиз учун зарур миқдорда яратиш имконини берувчи қулай ечим ҳисобланади.

DoEvents оператори ёрдамида тўхтатиш

Дастурлардан фойдаланиш жараёнида беҳосдан бирорта (нотўғри ҳаракат натижасида) узоқ вақт бажарилувчи ва бекор қилиб бўлмайдиган жараённи ишга тушириб юборишдан ёмони бўлмаса керак. Бундай ҳолатларда компьютерни қайтадан юклаш мумкин, лекин сифатли дастурлар бу усулдан фойдаланмаслиги зарур.

Тасаввур қилинг, катта ҳажмдаги ҳисоблашни бажарадиган цикл коди ишга туширилган бўлса, ишга туширилган код цикли тугамагунча, Visual Basicнинг бошқа ёрдамчи дастурлари ҳеч қандай ҳаракатга жавоб бермайди. Cancel тугмачани боссангиз ҳам цикл тугагунча дастур жавоб ҳаракатини бажармайди. Энг ёмони, дастурнинг хатоси чексиз циклга айланиб кетиши ва фойдаланувчи дастурни тузатиш имконияти бўлмаслигидир.

Умуман олганда, Windows қобиғида ишловчи дастурларда хатолик юз берса, Ctrl+Alt+Del тугмачаларини босиб, дастур ишини тўхтатиш ва уни Windows нинг масалалар диспетчерида [Task Manger] тугаллаш мумкин. Лекин яхши дастурларда бундай бўлмаслиги лозим.

DoEvents операторидан фойдаланиш

Юқорида келтирилган муаммони ҳал қилишнинг бир нечта усули бор. Булардан энг соддаси DoEvents дан фойдаланишдир. Бу буйруқ оператори Windows бошқарувини қайтариш учун қўлланилади. Демак, Click ҳодисаси ёки бошқа ҳодиса цикл ичида ишлаб чиқилиши мумкин. Қуйида бу оператордан фойдаланиш намунаси келтирилган.

1. Шаклнинг Declorations (Эълон) бўлимида ўзгарувчи эълон қилинг:

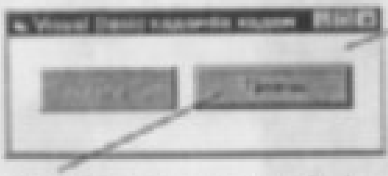
```
Dim CancelFlag as Boolean
```

2. Узун жараённи бошлаб берувчи тугманинг Click ҳодисасига қуйидаги кодни ёзинг:

```
CommandButton1. Enabled = False
CancelFlag = False
```

3. Циклнинг ўрта қисмига қуйидаги сатрларни ёзинг:

```
DoEvents
If CancelFlag then
Exit Sub
CommandButton1 Enabled = True
End If
```



Бу мисол учун цикл жараёнининг бажарилиши давомида "Отмена" (Тўхтатиш) тугмачаси DoEvents ёрдамида ишга тушириш учун қўлланилади.

4. "Отмена" тугмасининг Click ҳодисаси коди қуйидагичадир:

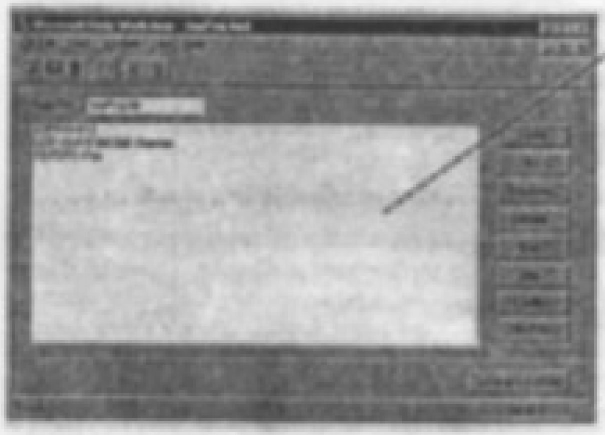
```
CancelFlag = True
```

Агар сиз DoEvents буйруғидан фойдалансангиз, фойдаланувчи тугмани босиб, циклни бошқатдан ишга тушириб юбориш эҳтимоли бор. Бунинг олдини олиш учун тугмани ишга туширишни тақиқлаб қўйишингиз (бу бўлимдаги мисолда шундай қилинган) ёки байроқчани фақат бир марта циклни ишга туширадиган қилиб қўйишингиз зарур.

Маълумотнома файллари қандай пайдо бўлади?

Кўпчилик профессионал дастурлар маълумотнома файллари билан бирга яратилади. Диалог ойнаси панелида F1 тугмасини боссангиз, бу панель учун тегишли маълумот пайдо бўлади.

Мазкур бўлимда маълумотнома файлларини яратиш тўғрисида тўла маълумот берилмаган, балки фақат бу файлларни яратишнинг умумий услуби тасвирланган холос. HTML форматидаги маълумотноманинг янги версияларини Visual Basic нинг кейинги нашрларида топишингиз мумкин.



Microsoft Help Workshop — бу маълумотнома файлларини бошқаришга ёрдам берувчи дастурдир.

Сиз яратган дастур учун маълумотнома файлларини тайёрлаш бир нечта босқични ўз ичига олади. Маълумотнома файлларини Word ёки шу каби бошқа дастурлар ёрдамида ёзиб олишингиз мумкин. Кейинги босқичда эса ҳар бир элемент учун ёзилган маълумотномага ўзига хос кодланган ном бериш зарур. Ундан кейин эса маълумот ҳужжатларини .HLP файлига компиляция қилинади.

Охирги, учинчи босқичда эса маълумотнома файлини дастурга боғлаш ишлари амалга оширилади. Бу ишлар Project Properties (Лойиҳа хусусияти) диалог ойнаси орқали HelpContextId хусусиятидан фойдаланиб амалга оширилади.

Учинчи ишлаб чиқарувчилар томонидан фойдаланишга чиқарилган автоматлаштирилган маълумотнома файлларини яратиш дастурларидан фойдаланиш қулай.

Project Properties диалог ойнасида маълумотнома файли номини кўрсатиш.

API ҳақида

Visual Basic лойиҳалари билан ишлашда баъзи муаммоларни ҳал қилиш учун API (Application Programming Interface — амалий дастурлар интерфейси) — Windowsнинг умумий кутубхона кодидан фойдаланишга тўғри келади. Бу бўлимда API ҳақида маълумот келтирилган.

Windows API си нима?

API (Application Programming Interface — амалий дастурлар интерфейси) — бу Windowsнинг умумий кутубхона

коди бўлиб, ойналар ҳосил қилиш, матн ва график элементларини экранда чизиш, сичқонча ҳаракатларига нисбатан жавоб ҳаракатларини амалга ошириш ва шу каби бошқа ҳаракатларни (визуал) кузатилувчи элементларга айлантириш учун ишлатилади.

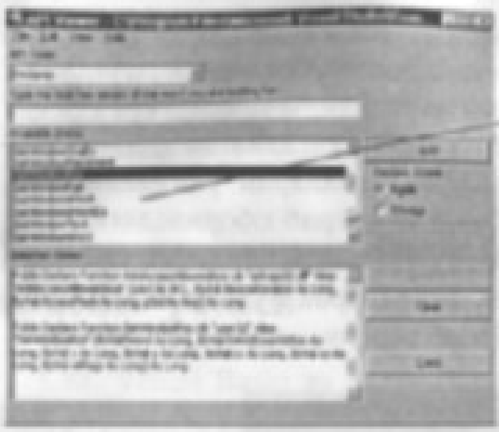
Windowsнинг асосий қисми Visual Basic дан мураккаброқ бўлган C ёки C ++ дастурларида ёзилган. Visual Basic да API дан фойдаланиш учун Basicда код ёзиш керак холос, кодни API элементига айлантириш ишларини Visual Basic нинг ўзи бажаради.

Лекин баъзи холатларда API функциясидан фойдаланмасликнинг имкони бўлмайди. Мисол учун, актив ойна бошқа лойиҳа ойналари томонидан тўсиб қўйилмаслиги учун унинг хусусиятини олдинги соҳага ўтказиб қўйиш зарур. Visual Basicнинг ўзида бундай имконият бўлмаганлиги учун API дан фойдаланишга тўғри келади.

API кучли восита ҳисобланади, лекин API дан фойдаланишда жуда эҳтиёт бўлиш зарур, эҳтиётсизлик туфайли системани оғир ҳолатига келтириб қўйиш мумкин.

Visual Basic да API ни ишга тушириш

Visual Basic да API функциясини ишга тушириш учун Visual Basic таркибига кирувчи API Text Viewer (API матнларини кўриш воситаси) дан фойдаланиш мумкин. Visual Basic да Declare оператори ёрдамида функцияларни эълон қилиш ва ташқи манбалардан олинадиган маълумотларни ҳам Visual Basicнинг функцияси сифатида ишлатиш мумкин.



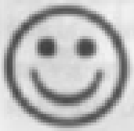
Internet билан ишлаш

Замонавий дастурларнинг деярли кўпчилиги зарур холатларда дастур учун техник ёрдам олиш, ёрдам файллари ҳамда дастурнинг янги версияларини олиш учун Internet га уланиш (Web-серверлар) имкониятига эга.

Visual Basic да ҳам бундай имоният мавжуд. Тўлиқроқ маълумот олиш учун Help менюсининг Microsoft on the Web сатридан фойдаланиб, Internet га уланишингиз мумкин.

Visual Basicнинг Internet билан ишлаш имкониятлари

1. Visual Basicнинг бошқариш элементлари таркибига Internet билан ишловчи бошқариш элементлари WebBrowser (Web — браузер, яъни Web ресурсларини кўриб чиқувчи элемент), протоколларни FTP шаклда узатувчи элементлар киради.



HTML (Hyper Text Markup Language) — гиперматнларни жойлаштириш тили, бу тилдан (Web-саҳифалар) график ва актив йўлланмаларга, шакллаган матнларга эга Web-файллар яратиш учун фойдаланилади.

2. Сиз Visual Basic учун HTML-саҳифаларда сценарий (Script) ёзишингиз мумкин. Бу сценарий Web-файл ишга туширилганда бажарилади. Web-файл орқали кириб келган фойдаланувчи ўзи англамаган ҳолатда ёки тушуниб туриб, ёпилган файлларни ўчирмаслиги ва қаттиқ дискдаги ахборотни ўчириб юбормаслиги учун, Web-сценарий ёзишда баъзи функционал имкониятлари камайтирилган, хавфсизроқ бўлган Visual Basic нинг Visual Basic Script версиясидан фойдаланади.

Visual Basic Web-браузери.

Лойиҳага шакл қўшаётганингизда Browser Form ни танласангиз, Visual Basic сизнинг дастурингиз учун мини-браузер яратади.

3. Microsoft нинг Visual InterDev воситасидан фойдаланиб, Web-серверда бажариладиган Visual Basic сценарийсини ёзишингиз мумкин. Бу восита ёрдамида яратилган сценарий ҳар қандай браузер билан ишлай олади ва зарур ҳолатда маълумотлар базасига мурожаат қилиб, мурожаат натижаларини HTML шаклда акс эттиради.

Visual Basic сценарийлари фақат Microsoft браузерлари томонидангина қўллаб-қувватланади. World-Wide-Web учун ишлатиладиган сценарийларни ёзишда юқоридаги маълумотни инобатга олинг.

Ўрнатиладиган дастур мажмуасини яратиш устаси

Сиз, Visual Basic да дастур яратиб бўлганингиздан кейин, табиийки, уни бошқа фойдаланувчиларга ҳам тарқатасиз. Лекин Windows муҳитида ишлайдиган барча дастурлар ўзига хос (Dinamyc Link Libraries — ёки DLL) файлларидан фойдаланадилар, бу файллар ҳар бир мақсадли система учун ўзига хос бўлади. Бошқа системаларда бу файллар бўлмаслиги мумкин. Лекин кутубхона (DLL) файллари системада бўлмаса, уларни кўчириш катта хатолик ҳисобланади.

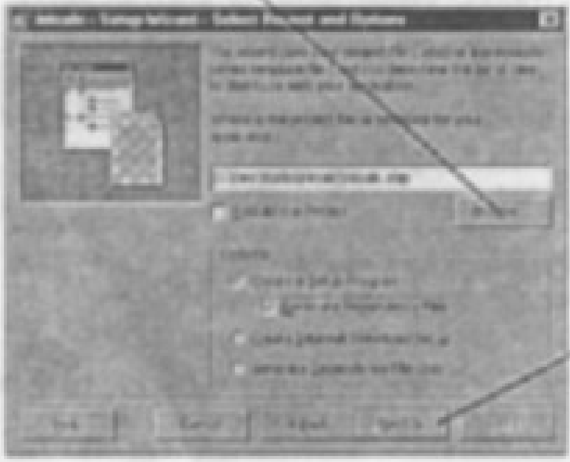
ActiveX элементлари DLL файлларининг бир тури ҳисобланади. ActiveX элементлари ишлаши учун уларни рўйхатдан ўтказиш (регистрация қилиш) зарур. Windows да (Registry) маълумотлар базаси системага ўрнатиладиган дастурларнинг хусусиятлари ва усулларини рўйхатга олади.

Бир сўз билан айтганда, Visual Basic да яратган дастурингиз бошқа системаларда ҳам ишлаши учун дастурингиз ўрнатиладиган мажмуани яратишингиз лозим. Бу ишни амалга ошириш учун Visual Basic да Application Setup Wizard воситасидан фойдаланиш мумкин.

Setup Wizard ни ишга туширишдан олдин жорий лойиҳани ёпиш лозим. Акс ҳолда хатолик тўғрисидаги хабарни кўрасиз.

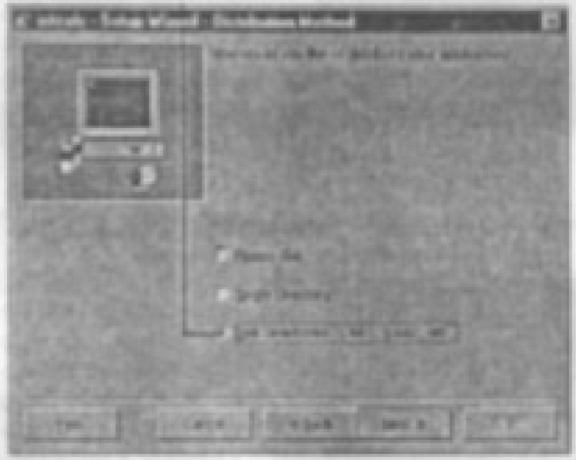
1. Яратган дастурингизда катапик ўрнатишга тўлиқ ишонч хосил қилгач, Visual Basic ни ёпинг ва Windows даги Setup Wizard Application ни Start Menu нинг Visual Basic 5,0 бўлимидан ишга туширинг.

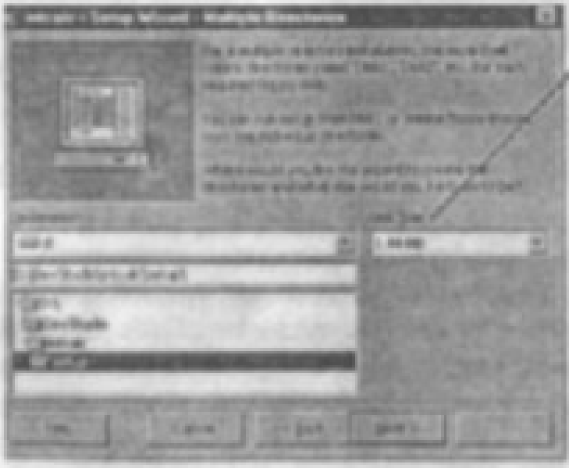
2. Уста ишга тушгач, Browse тугмасини босинг ва лойиҳангизнинг .VBP-файлини топинг. Бу лойиҳангиз EXE-дастур эмас, балки уста кўриши керак бўлган файли.



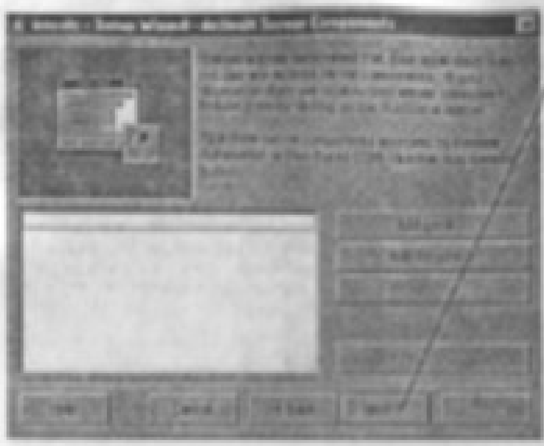
3. Next тугмасини босинг.

4. Кейинги босқичларда эса дастурни тарқатиш усулини танлашингиз лозим. Эгилувчан дискларда ўрнатиш пакетини яратиш учун Floppy Disk ёки тармоқдан ўрнатиладиган пакет яратмоқчи бўлсангиз, Single Directory режимини танланг.
Охирги усул — бу ўзингизни системангизда текширишни ишларини амалга оширишнинг энг қулай усулидир, яъни ўрнатиш пакетининг ишини ўзингизнинг системангизда яратасиз ва ўрнатиб кўрасиз.

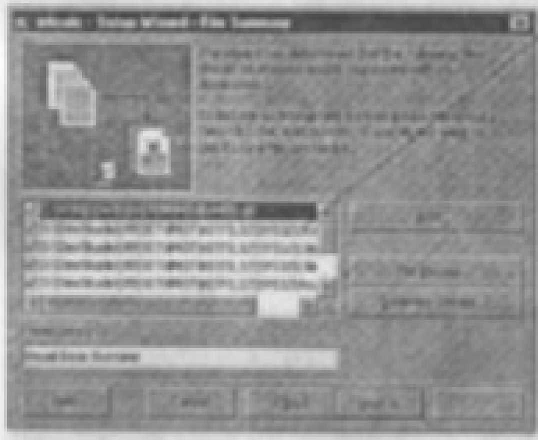




5. Ўрнатиш пакети учун диск ҳажмини танланг (Disc Size). Одатда 1,44 Mb ли ҳажм танланади



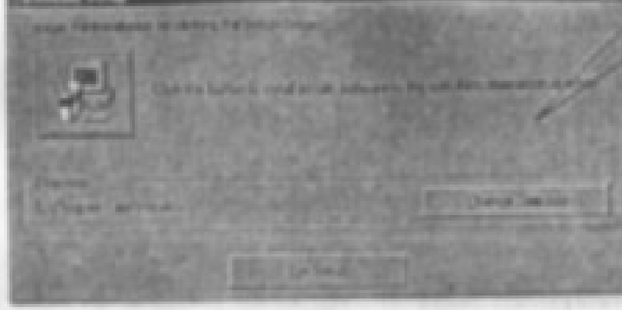
6. ActiveX Server Components мулоқот ойнаси Visual Basic да қўлланилган ActiveX элементларини аниқлаш имконини беради. Мазкур диалог ойнаси ишини кўриш, тавсифлаш учун жой камлиги муносабати билан, тушунтиришларини кейинги нашрга қолдирдик. Бу диалог ойнасидан Next тугмасини босиб ўтиб кетинг.

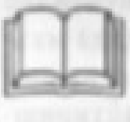


7. Бу диалог ойнаси дастурингиз иши учун зарур бўлган файллар рўйхатини акс эттиради. Application Setup Wizard зарур файлларнинг барчасини танлаб ола олмайди, шунинг учун бу рўйхатни диққат билан кўздан кечиринг ва етишмаётган файлларни ўзингиз белгиланг. Мазкур ҳолда бунга зарурат йўқ.

Бу диалог ойнасида уста томонидан танланган файлларни ўчириб қўйишдан эҳтиёт бўлинг. Файллар орасидаги боғлиқликни билиш учун Windows ни чуқурроқ ўрганинг.

8. Яратган дастурингизнинг Setup.EXE файлини ишга тушириб кўринг. Шу билан бирга ўрнатиш пакетини Visual Basic ўрнатилмаган бошқа системада ҳам текшириб кўринг.





Ўрнатиш устоси Uninstall дастурини ҳам яратади. Бу дастурни Windowsнинг бошқариш панелидаги (Control Panel>Add/Remove Programms) "Дастурларни ўрнатиш/Ўрнатилганларини бекор қилиш" бўлимидан ишга туширишингиз мумкин.

Бундан кейин нима қилиш керак?

Мазкур китобнинг охирига етдингиз; лекин Visual Basic нинг бу китоб саҳифаларига туширилмаган имкониятлари кўп.

1. Лойиҳани бошланг

Агар сизда аниқ мақсад бўлмаса, дастур тузишдан кийин иш йўқ. Уйда, ишда ёки исталган ҳаётий муаммони Visual Basic ёрдамида дастурлаб ечим топишга ҳаракат қилинг. Биринчи уринишларда ҳеч қандай натижа чиқмаса ҳам сиз Visual Basicда маълум муаммо учун аниқ ечим топишга ўрганасиз.

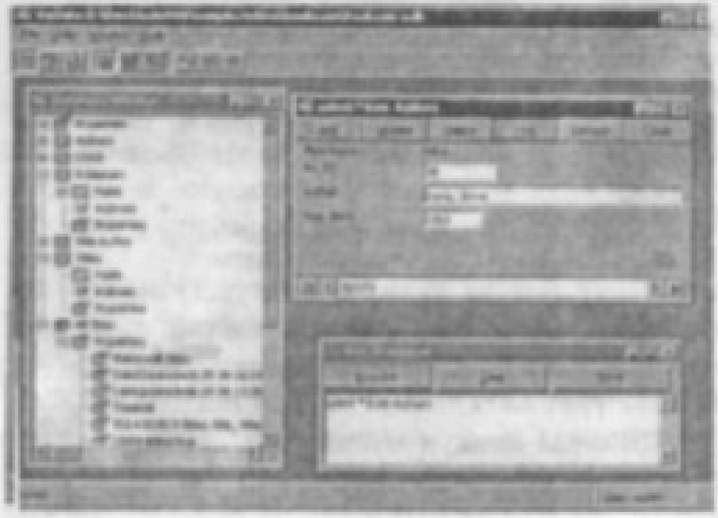
2. Ёрдам олиш учун мурожаат қилинг

Visual Basic да қандай масалани ҳал қилишингиздан қатъи назар, сизда шу масалани аввал кимдир ҳал қилгани ҳақида маълумот олиш имконияти бор. Чунки Visual Basic жуда ҳам оммавий дастурлаш муҳити ҳисобланади. Баъзи масалалар юзасидан китобдан ва Internet дан ҳам маълумот топиш мумкин.

3. Текширинг ва яхшиланг

Бизнесда харидорнинг хоҳиши ҳар доим ҳақлидир. Дастурлар бозорида эса фойдаланувчининг хоҳиши катта рол ўйнайди. Сиз яратган дастурдан бошқа фойдаланувчилар ҳам фойдаланса, улардан дастур камчиликлари ҳақида маълумот беришларини талаб қилинг, яхшиси уларнинг ишини кузатинг ва баъзи бир ортиқча ҳаракатларни камайтиришга, визуал элементлардан кўпроқ фойдаланишга ҳаракат қилинг.

Visual Data Manager дастури Visual Basic да яратилган. Унинг кодини очиб, баъзи бир масалаларни қай тарзда ечилишини кўриб чиқишингиз мумкин.



VISUAL BASIC
ҚАДАМ ВА ҚАДАМ

Маълумотнома нашр

Тим Андерсон

VISUAL BASIC.
ҚАДАМ БА ҚАДАМ

Муҳаррир М. Турсунова
Бадиий муҳаррир *У. Салохов*
Муқова мусаввири *У. Салохов*
Тех. муҳаррир *Т. Харитонова*
Компьютерда тайёрловчи *Е. Гильмутдинова*

Теришга берилди 15.02.2002. Босишга рухсат этилди 27.05.2002.
Бичими 84×108 1/32. "Антиква" гарнитурада офсет босма
усулида босилди. Шартли б.т. 11,76. Нашр т. 8,94.
Нусхаси 5000. Буюртма № 84.

"Ўзбекистон" нашриёти 700129. Тошкент, Навоий кўчаси, 30.
Нашр № 34—2002.

Ўзбекистон Республикаси Давлат матбуот кўмитаси
Тошкент китоб-журнал фабрикасида босилди
700129, Тошкент, Юнусобод даҳаси, Муродов кўчаси, 1 уй.

Анденсон Тим.

А 65 Кадам-ба-кадам.-Т.: Ўзбекистон, 2002.— 222 б.

ISBN 5-640-03134-4

32.973

№ 156-2002

Алишер Навоий номидаги

Ўзбекистон Республикасининг

Давлат кутубхонаси

$А\frac{2404040000-46}{M351(04)\ 2002}2002$

"ЎЗБЕКИСТОН"