

Нұрпейісова Т.Б.



**ТАУ-КЕН МАШИНАЛАРЫН
ПАЙДАЛАНУ КЕЗІНДЕГІ
ЖҰМЫС ОРНЫН
АВТОМАТТАНДЫРУ**



6771.2-00 (07515)

Н86

ЖОК 622.6:004(075.8)

ББК 33.16.Я73

Пікір жазғандар:

Б.Д. Хисаров техн.ғыл.канд., профессор АЭЖБИ.

Г.А. Самизулина техн.ғыл.канд, доцент ЕМКИЖБПИ.

Р.К. Ускенбаева техн.ғыл.докторы, профессор ҚазҰТУ.

Н 86 НҮРПЕЙІСОВА Т.Б.

Тау-кен машиналарын пайдалану кезіндегі жұмыс орнын автоматтандыру. Оқу құралы. – Алматы: Эверо, 2015. -112 б. Сурет – 17. Кесте - 13. Әдебиеттер тізімі - 5 атау.

ISBN 978-601-240-239-1

Оқу құралы 050707, 050712 және 050724 мамандықтарының стандартына қойылған талаптарға, мемлекеттік стандартқа сай құрастырылған.

Оқу құралында *Visual Basic* жүйесінде қолданушы қосымшасын жобалауға қажетті материалдар жинақталған. Бұл материалдар оқу процесін жақсартуға, оқытушыға дәрісті жоғарғы педагогикалық деңгейде оқуға, сонымен қатар студенттердің өзіндік жұмыстарын орындауда тақырып материалдарын терең оқып-үйренулеріне болады.

Оқу құралы 050707 Тау-кен ісі, 050712- Машинажасау және 050724 – Технологиялық машиналар мен жабдықтар мамандықтарының бакалавр студенттеріне арналған.

ЖОК 622.6:004(075.8)

ББК 33.16.Я73

Қазақстан Республикасы Білім және ғылым министрлігінің 2009 жылғы жоспары бойынша басылды.

ISBN 978-601-240-239-1

© Нұрпейісова Т.Б., 2015

© Эверо, 2015



КІРІСПЕ

Қазіргі кезде ақпараттану - ғылым ретінде кеңінен даму үстінде.

Қоғамды ақпараттандыру процесі ауқымды сипаттама алып, іс жүзінде әлемдегі барлық елдерді қамтыды. Сондықтан әр адамға ақпараттанудың фундаментальды негізін, оның мүмкіндіктерін және даму келешегін білу қажеттілігі туды.

«Тау-кен машиналарын пайдалану кезіндегі жұмыс орнын автоматтандыру» пәнінің негізгі мақсаты - студенттердің ақпараттанудағы жаңа технологияның негізгі бағытын оқып-үйрену. Ал соңғы қолданылатын дербес компьютерлерді қолдану барысында техникалық машиналарды пайдаланудың техника-экономикалық талдауын жүргізетін объектілі бағытталған бағдарламалауды іс жүзінде пайдаланып, деректер базасын құра білу.

Жоғарғы деңгейлі мамандарды дайындау талаптарына сай, студенттерге «Тау-кен машиналарын пайдалану кезіндегі АЖО» пәнін оқып-үйрену үшін төменде аталған міндеттер көзделген:

Білім міндеттері:

- объектілі бағытталған бағдарламалаудың негізгі түсініктерін, бағдарламалаудың визуальды орта түсінігін, Visual Basic бағдарламалау тілінің негізгі операторларын білу.

Іскерлік міндеті:

- бағдарламалау тілінің ерекшеліктерін ескере отырып, объектілі бағытталған бағдарламалаудың визуальды ортасын қолданып, бағдарламаларды және жоба бағдарламаларын құра білу;

- визуальды компонент кітапханаларын қолданып, бағдарлама өнімдерін құру.

1. ТАУ-КЕН МАШИНАЛАРЫН ПАЙДАЛАНУ КЕЗІНДЕГІ ИНЖЕНЕР ҚЫЗМЕТКЕРЛЕРІНІҢ ЖҰМЫСЫН АВТОМАТТАНДЫРУ

1.1. Автоматтандырылған жұмыс орнын (АЖО) дайындау

Автоматтандырылған жұмыс орны (АЖО) дегеніміз - белгілі бір пән аймағында дербес кәсіби ЭЕМ-ді пайдаланатын қолданушылардың жұмысын қамтамасыздандыруға арналған әдістемелік, тілдік және бағдарламалық құрылғылар жиыны. АЖО құрамына ЭЕМ-мен пайдаланушының арасындағы диалогті жүргізуге арнайы тіл және қажетті деректер базасын құруды қамтамасыздандыратын тілдер жатады.

Қазіргі деңгейдегі халық шаруашылығындағы тау-кен саласын жылдам дамытудағы негізгі шарттардың бірі – есептеу техникасын кеңінен қолдану барысында ұйымдастырылған басқару жұмыстарын жетілдіру болып табылады. Бұл жетілдірудің техникалық негізін құраушы - ДК болып саналады. Бірақ бұл жинақталған ақпарат көлемін өңдеу мәселелерінің бір бөлігін ғана шешеді. Екінші бөлігі (аса күрделі бөлігі) – мекеменің ұйымдастырушы құрылымы мен тау-кен бөлімше инженерлерінің өз жұмысында есептеу техникасын кеңінен қолдануы.

Қазіргі кезде әртүрлі мамандардың ой және басқару еңбегін интенсификациялау үшін ДЭЕМ базасында жұмыс істейтін автоматтандырылған жұмыс орындары құрылып, олар кеңінен қолданылуда.

1.2. АЖО-ның құрылымы

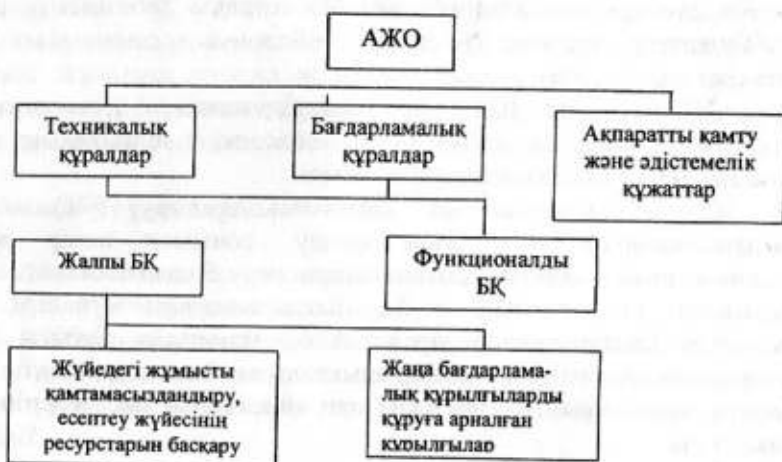
АЖО-ның біріктірілген құрылымды сұлбесі 1.1- суретте келтірілген.

Дербес электронды есептеу машинасы (ДЭЕМ) АЖО-ның техникалық негізін құрайды, оның сұлбесі 1.2 - суретте келтірілген.

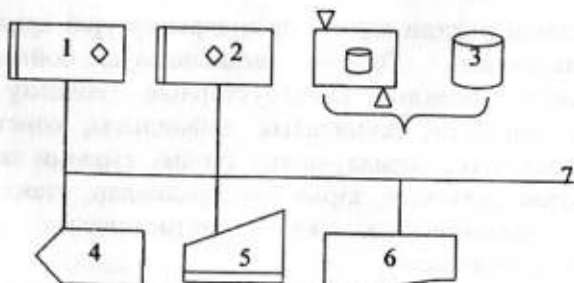
ДЭЕМ-ның негізгі құрылғысы – микропроцессор, ол бағдарлама құрамындағы әртүрлі операциялардың орындалуын қамтамасыз етеді. Микропроцессорлар: разрядтілігі және тактілі жиілігі бойынша ажыратылады. Разрядтілігі және тактілі жиілігі үлкен болса, микропроцессордың өнімділігі арта түседі.

ДЭЕМ-нің жады екі компоненттен (бөліктен) тұрады – тұрақты сақтау құрылғысы (ПЗУ) және оперативті еске сақтау құрылғысы (ОЗУ). ОЗУ-да бағдарламаның динамикалық ақпараты мен өңделетін ақпарат сақталады. ДК өшірілгенде, ондағы ақпарат жойылады. ПЗУ-дағы ақпаратты өзгертуге болмайды, себебі онда жүйені бағдарламалық қамтамасыздандырудың (БҚ) көпшілік бөлігі жазылған.

Сыртқы сақтау құрылғылары ретінде иілгіш және қатты магниттік дискілі жинақтауыштар қолданылады. «Жүйелік диск» типті қатты дискілердің негізгі жетістігіне: оның үлкен көлемділігі және мәліметтерді тез арада алу жылдамдығы жатады.



1.1-сурет. Автоматтандырылған жұмыс орнының құрылымды сұлбесі



1.2-сурет. ДЭЕМ сұлбесі

1-микропроцессор; 2-жады; 3-сыртқы есте сақтау құрылғысы; 4-дисплей; 5-пернелер тақтасы; 6-баспа құрылғысы; 7-жүйелік магистраль

Дисплей - ақпаратты бейнелеудің негізгі құрылғысы болып саналады.

Экрандағы курсорды басқару үшін «мауыс» манипуляторы қолданылады. «Мауыс» - шамды қаламға ұқсас, айырмашылығы оның жылжуы горизонтальды жазықтықта бақыланып, вертикальды экранда – қозғалысқа түрлендіріледі.

Пернелер тактасы, мәліметтер мен бағдарламаларды енгізуге арналған. Қазіргі таңда кеңінен таралған лазерлі баспа құрылғылары жұмыс жылдамдығы мен мәтін баспасының типографиялық сапасын арттырады.

ДЭЕМ-нің барлық құрылғыларының байланысы жүйе магистралі арқылы атқарылады. Магистраль дегеніміз - ДЭЕМ компоненттері арасындағы дыбыс байланысын қамтамасыз ететін электрлі желі торабы.

АЖО базасында ДЭЕМ-ні қолдануды нәтижелі ұйымдастыруда, нақты мамандарды (бухгалтер, басшы, техникалық білім инженерлерін, т.б.) дайындау аса тиімді.

Жалпы бағдарламалық қамтамасыздандыру (БҚ) - жаңа бағдарламаларды қосу және өңдеу, сонымен қатар есептеу техникасының қызметін қамтамасыздандыру болып табылады. Оның құрамына операциялық жүйе, бағдарламалау жүйелері және қызметші бағдарламалар кіреді. АЖО мамандық бағыты БҚ-ың функционалды бөлігімен (ФБҚ) анықталады. Осы жерде нақты маман бағыты қарастырылып, нақтылы пән аймағының мәселелерін шешу көзделген.

1.3. АЖО-ның классификациясы

Автоматтандырылған жұмыс орындары әртүрлі принциптермен классификацияланады. Оларды тағайындау бойынша келесі салаларға бөлуге болады: конструкторлық (қолдану аймағына байланысты), өндірістің техникалық дайындығы, конструкциялық құжаттарды дайындау, бақылау-сынау бөлімі, ғылыми эксперимент, бағдарламаларды дайындау, кірме бағдарламалар, стандартты емес функциялар, экономикалық және статистикалық құжаттарды дайындау, бейнелерді өңдеу.

Функционалды белгісі бойынша: мәтіндік және символды ақпаратты өңдеу, графикалық ақпаратты өңдеу, аналогті және дискретті ақпаратты өңдеу болып бөлінеді.

Қуаттылығы бойынша: кіші, орта және үлкен өнімді болып бөлінеді.

Жинақтау принципі бойынша: автономды, көп пульттілі немесе топтық және терминалды болып бөлінеді.

Қазіргі кезде АЖО құрудың функционалды түрі кең қолдау табуға, енді оның негізін қарастырайық. Таңдалған мамандық бойынша жұмыскерлердің атқарған жұмысы зерттеліп, оның ішінен жиі орындалатын және типтес жұмыстар таңдалып, ол функцияларды орындау жұмысын автоматтандыратын бағдарламалық құралдар құрылады.

ДЭЕМ-ға арналған бағдарламалық құралдар каталогінің құрамында мыңдаған атаулар бар, сондықтан жаңа ФБҚ құрмай-ақ, бұрын құрылған БҚ нақтылы қолдану шарттарына өзгертіп қолдануға болады. Ең алғашқы құрылған бағдарламалық құралдар, техникалық саладағы жұмысшылардың жұмысын автоматтандыруға арналған. Жиі кездесетін функциялардың біріне – мәтіндік құжаттарды дайындау мен математикалық есептеу жұмыстарын орындау жатады.

Бақылау сұрақтары:

1. Автоматтандырылған жұмыс орны дегеніміз не?
2. АЖО негізгі құрамын атаңыз.
3. АЖО-ның классификациялану принциптерін атаңыз.
4. Тау-кен машиналарын пайдалану кезіндегі инженер-механиктердің автоматтандырылған жұмыс орны қандай міндеттерді көздейді?

2. ОБЪЕКТИЛІ БАҒЫТТАЛҒАН БАҒДАРЛАМАЛАУҒА КІРІСПЕ

2.1. Қазіргі қоғамдағы ЭЕМ жаңа маңыздылығы

15 жыл ішінде компьютер экзотикалық есептеу орталығының құрылғысынан, миллиондаған адамдардың күнделікті жұмыс құралына айналды. Ауқымды ақпараттық төңкеріс микропроцессор мен дербес компьютерлерді ойлап табуына, цифрлы байланыс құралдарының пайда болуына байланысты болды. Ол компьютерді қолдану, өңдеу және тасымалдау құрылғыларының жаңартылған түрлері және адам қызметінің әр саласында ақпаратты тасымалдаумен сипатталады. *Ақпараттану* деп аталатын өндіру процесінің өзі, адам қоғамының жаңа эволюциялық даму кезеңінің басы болды.

Қазақстанда ақпараттану процесі білім жүйесінің Мемлекеттік ақпараттану бағдарламасы пайда болғаннан басталды. Оның алдына қойған мақсаттары:

- ақпараттық қоғамда оқытушыларды өмірге бейімдеу;
- оқу, тәрбиелеу және басқару процестеріне ақпараттық және телекоммуникациялық технологиялардың негізінде білім жүйесін *реформалау*;

- Қазақстандық білім жүйесін әлемдік білім кеңістігіне қосу.

Соңғы он жылда әртүрлі ғылыми және әлеуметтік құбылыстарды сипаттау үшін ақпарат пен ақпараттану процесінің түсінігі қолданылады. “Экономикалық ақпарат”, “ақпараттық бизнес”, “ақпараттық құқық”, “ақпаратты өңдеу” түсініктерінің пайда болуы, “ақпарат” түсінігінің қазіргі кездегі ғылыми әлемтанудың негізгі категориясы ғана болып қоймай, сонымен қатар қазіргі кезде ең маңызды сала болып отыр.

Ақпараттық төңкеріс ХХ ғасырдың орта шенінен басталған ақпараттық дағдарыстан туып отыр. Ақпарат адам баласына ақпаратты толық қабылдау мүмкіндігін бермей, көшкін тектес ағыммен ағылды. Бағдарламалау қиындай түсті. 1990 жылдан бастап адам баласының жинақтаған жалпы білім қосындысы әр 50 жыл ішінде, 1970 жылы- әр 5 жыл, 1990 жылдан бастап әрбір жылда екі еселеніп отырды. Соңғы жылдары жүйедегі басылымдағы құжат саны 2000 жылға қарай 850 млн болды, ал 2010 жылға қарай ақпараттық кеңістік 10 млрд құжатқа дейін кеңейеді деп күтілуде. Адамдардың

шектелген қабылдау мүмкіндіктері мен ақпаратты өңдеуде, барлық ағыммен сақталған ақпарат массивтерінің қайшылықтары пайда болды. Ақпараттық төңкеріс адамдар өмірінде экономикаға да әсерін тигізді. Олар білімге негізделген жаңа экономикаға көше бастады, экономика - тауарлардың, ресурстардың және материалдардың қозғалыстары емес, білім мен идея базасының негізінде қарастырылады. Бұл процесс барлық елдерде жүріп жатыр, оны үшінші экономикалық төңкеріс деп атауға болады. Мұндай төңкеріс, XVIII ғасырдың соңында бу машиналарын ойлап табумен байланысты, электрлендіруді құру мен қолдану төңкерісі болды.

Ақпаратты әрекеттердің өсу мәні - жұмыс орындарының құрылымдарын қайта қарастыруға әсерін тигізді: материалдық саладан еңбек ресурстары ақпараттық түрге ауысып, ақпаратты өңдеумен байланысты жаңа мамандықтар пайда болды.

Қазіргі кезде дамыған елдерде өндірістік мамандықтағы жұмысшылар 16%, ал ақпаратты өңдеуші жұмысшылар (яғни, олардың негізгі мамандық қызметі - ақпаратты өңдеу) - шамамен 60% құрайды. Жақын аралықта, негізгі өндірістік функциясы - ақпаратты өңдеу болып табылатын адамдар саны өсе түседі деп күтілуде.

Ақпараттық қоғамның қалыптасуы тек өндіріс саласында өзгерістерді ғана емес, сонымен қатар адамның қалыптасқан өмірін, оның құндылық жүйесін де өзгертті. Өндірісте, тауар алмасуына бағытталған индустриалдық қоғаммен салыстырғанда, ақпараттық қоғамда интеллект (зерде) білімі өңделіп, қолданылады. Бұл ой еңбегінің үлесін арттырады. Адамның шығармашылық қабілетін қажет етіп, оның білімге деген талабын арттырады.

2.2. Бағдарламалаудың даму тарихы

1960 жылдардың орта шенінде Кеңестер Одағы есептеу техникасы саласында әлемдегі ең алдыңғы қатардағы ірі мемлекеттердің бірі болды. Бұл елде ондаған мың жоғарғы деңгейлі бағдарлама құрушылар жұмыс істеді. Олардың еңбегінің нәтижесінде тез арада алғашқы ЭЕМ бірнеше буындары құрылды және ол ЭЕМ батыс нұсқасынан анағұрлым жоғары болды. Ол уақытқа дейін өздерінің бағдарламашылар мектебі құрылды. Оның ерекшелігі-жаңашылдықта, стандартсыз шешімдерді таба білуде және ең бастысы; жұмыстағы командалық рух болды. Біздің мектеп әлемде

жоғары бағаланды, бүгінгі таңда да ол Германия мен АҚШ-та жоғары бағаланады.

КСРО кезінде барлық бағдарлама мемлекеттік тапсырыс бойынша, не мемлекеттік жобалармен жұмыс істеу кезінде құрылды. Бұл кезде жай бағдарламалар мен машиналық код дамыды: Ассемблер, Бейсик, Qbasic, Алгол, Фортран, Кобол, Паскаль, Ассемблер, Си Си++.

Жай бағдарламалау тілдері оператор тізімінен, операторларды қолдану синтаксисінен (ережелерден) және компиляторлардан тұрады. Компилятор дегеніміз не?

Бағдарламалауда әріптерді, символдарды және ағылшын тілінің сөздерін қолдану дәстүрі бар. Бұл орыс немесе қазақ тілінде бағдарлама құруға болмайды деген сөз емес, басқа тіліндегі операторларды ЭЕМ-ға қабылдату қиындатылады. Себебі, компьютер процессоры қазақ, орыс, не ағылшын, басқа да әріптерді түсінбейді. Ол тек сандардың көмегімен жазылған сигналдарды ғана түсінеді. Ол үшін, компилятор деп аталатын арнайы бағдарлама бар, бұл бағдарлама сөздерді машиналық кодтарға түрлендіреді. Компиляторды құру бірнеше жылдарды қажет ететін өте күрделі жұмыс және ол ондаған, жүз миллиондаған доллар тұрады. Сондықтан, бүкіл әлемге танымал, әмбебап тілдерді меңгерген жөн, әрі ағылшын тілін білу қажет. Жай бағдарламалау тілдері бағдарлама ядросын түпнұсқалы және талапты түрде жазуға мүмкіндігін береді, ол шамамен жұмыстың 10% құрайды. Ал шамамен 60% алғашқы деректерді рәсімдеуге және нәтижені алуға кетеді. 1980 жылдан бастап, батыс елдерінде компьютер кеңінен тарала басталды. Бағдарламаға арнайы талаптар қойылды. Олар Windows операциялық жүйелердің мүмкіндіктерін ескере отырып тартымды, мәнерлі, әрі дұрыс қалыпты болуы тиіс. Әлемде қолданушы жобаларды рәсімдеумен байланысты, бағдарламаны құрумен кәсіби айналасушы компаниялар пайда болды. 1990 жылдардың екінші жартысында әдеттегі бағдарламалау тілдеріне: бағдарламалау жүйесі деп аталатын бағдарламалар қосылды.

Бағдарламалау тілінен, бағдарламалау жүйесінің артықшылығы - ол тек компилятордан ғана тұрмай, сонымен қатар бағдарламалауды жеңілдететін қосымша бағдарламамен толықтырылған. Оның көмегімен бағдарламаларды жазбай-ақ, оларды балалар компьютерінің компоненттерінен неше түрлі ойыншықтар жинақтауға болатындай, дайын компоненттерден жинақтауға болады.

Атап айтқанда: Basic - Visual Basic бағдарламалау жүйесіне, Паскаль - Delphi-ге, C++ - Borland C++Builder-ге және MS Visual C-ке айналды. Ең қарапайым бағдарламалау жүйесі - Visual Basic болып табылады. VB ең танымал тілдердің бірі. VB бағдарламалау жүйесі Windows қосымшасының офистік бағдарламаларына қосылған.

Бірінші курстан сіздер “Ақпараттану” пәнінде BASIC тілінің идеологиясымен таныссыздар.

2.3. Объектілі бағытталған бағдарламалау

Объектілі бағытталған бағдарламалау (ОББ) - бұл бағдарламаны құру методологиясы. Ол әрқайсысы анықталған класс объектісі болатын, объект жиыны түріндегі бағдарлама көрсетіліміне негізделген. Бүгінгі таңда ОББ - құрылымды бағдарламалау тәсілдерін ығыстыра отырып, сенімді түрде оның үстемдігі артты.

Алғаш қарағанда, бағдарламалауға ешқандай қатысы жоқ, есептеу техникасын өндіру аумағынан белгілі мысалды алайық. Comrag, HP, Gateway атты компаниялардың тез арада технология саласында әлем жетекшілері мен дербес компьютерлерді негізгі өндеуші болуына не себеп болды? Бұл компаниялар көптеген болмашы функциялардан бас тартып, өз жұмысын негізгі күшті соңғы өнім технологиялық стратегиясын құруға бағыттады. Олар көптеген дайын компоненттерді беделді мамандандырылған өндірушілерден сатып алып, олардың негізінде өз компьютерлерін құрды. Олар қоректендіру көзін, дискжинақтауыштарды, аналық тақшаны, т.б. элементтерді өндіруге және жобалауға көп уақыты мен ақшаларын жұмсамады. Қысқаша айтқанда, дербес компьютерді дайындаушылар “функционалды буманы” сатып алатын. Қоректендіру көзін сатып алу кезінде, олар өздеріне қажетті қасиеттерін (өлшемін, пішімін, т.б.) және кейбір функционалды сипаттамаларын (тұрақты кернеу, қуаттылығы) таңдап алады. Comrag компаниясы, компьютердің барлық бөліктерін жобалаудан, оның көптеген құрылғыларын тәжірибелі өндірушілерден сатып алғанда соңғы нәтижесінің анағұрлым жақсаратынына көз жеткізіп, осы тәсілді негіз ретінде қолданды.

ОББ негізінде де, сол идея жатыр. Қолданбалы бағдарламалар - объектілер орындай алатын, объектінің кейбір қасиеттері мен операцияларының негізінде құрылады. Объектінің ағымдағы күйі, біраз уақыттан кейін өзгереді, бірақ бағдарлама - объектілердің бір-

бірімен ерікті түрде қарым-қатынаста болмауына байланысты болады. Жаңа түпнұсқалы объектіні құруды немесе құрылған объектіні қолдануды бағдарламашы өзі шешеді. Бірақ құрылған объектілер, атап айтқанда не орындай алатын және сыртқы ықпалға қалай әсер ете алатын талаптарды қанағаттандырса, онда бағдарламашы оның функционалды мүмкіндіктері қалай жүзеге асырылғанына көңіл бөлмейді. ОББ тілінде оны объектінің бере алатын мүмкіндіктері ғана қызықтырады.

2.3.1. Объектілер

Visual Basic-те *объект* деп - біртекті технологияны қолдана отырып, онымен ұқсастыруға және басқаруға болатын жүйенің әр компонентін айтамыз. Жобалаушы - жұмыс істейтін барлық объектілердің объект типіне қарамастан, стандарттық қасиеттері, объект тәртібін басқару сипаттамалары, көбінесе, әдістері (объект әрекетінің бағдарламалық сипаттамасы) - қолданушы және стандартты болады. Visual Basic объект қосымшаларын қолданушы, толық “түсіне” алатын бола алады, мысалы диалог терезелері, командалық батырмалар, мәтіндік өрістер, т.б арқылы. Объектілер, өздері қолданушы интерфейсінің бөлігі болмайтын және оларды маус батырмаларымен “қозғауға” болмайтын, кейбір категориялардың көрсетімі бола алады. Оларға мысал ретінде, жұмыс аймақтарын, байланыстарды және жазбалар жиынын (Workspace, Table Def, Connection, Recordset объектілері) жатқызуға болады. Қосымшаларды құру кезінде оны құрушы объект әрекеттерін бағдарламалайды, немесе бұл объектілердің кейбір оқиғаларға, орта күйіне немесе қосымша әрекеттерін бекітеді. Мысалы, объектінің “командалық батырмасының” сипаттамасы болатын стандартты әдіске бағдарламалық кодты қосып, қолданушы ОК батырмасын шерткен кезде, пішін орындайтын әрекетті тапсыруға болады.

Visual Basic-тегі объектінің анықтамасы - “Объект дегеніміз - бірлік ретінде қабылданатын және оны әртүрлі түрде басқаруға болатын бағдарламалық код пен деректер комбинациясы”. Мысалы, объект ретінде мәтіндік өрісті алсақ, деректерді енгізу түстерін өзгертіп рәсімдеу, қаріптерін бекіту, т.б. түрде басқаруға болады. Объектінің модельдік көзқарасында объект деп - есептеу жүйесінің

аппаратты құрылғыларын да атауға болады. Мысалы, принтер, монитор, дискілерді бағдарламалық код арқылы басқаруға болады.

Бір объект ерікті түрде басқа объектінің ішкі деректерін басқармауы тиіс. Объектаралық барлық байланыс тек хабарлар арқылы атқарылады. Кейбір сәйкес хабарлар жиынымен басқаруға болатын объектілерді құру кезінде және объектілердің өздеріне олардың ішкі деректерін басқаруға мүмкіндік берумен, біз объектіні көптеген ретте қолдану мүмкіндігін өсіріп, оны жөндеу уақытын азайтамыз. Visual Basic-те құрылатын объектілер тек қасиеттерін өзгерту мен әдістерін шақыру арқылы басқарылады.

Басқару элементтерінің объектілері - басқару элементтері (Controls) деп аталады. Онымен жұмыс істеу кезінде қолданушы белгілі оқиғаларды орындап, нәтижесінде бағдарламаны басқара алады.

Объектілер кластарға біріктіріледі. Бір класқа қасиеттері, әдістері және оқиғасы бірдей жиыны бар объектілер жатады. Әр объект анықталған параметрлерімен сипатталады. Оларды үш категорияға бөлуге болады:

- оқиғалар;
- әдістер;
- қасиеттер.

2.3.2. Оқиғалар, әдістер және қасиеттер

Visual Basic-те объектілерді екі тәсілмен басқаруға болады:

- 1) объект қасиеттерін өзгерту арқылы;
- 2) сол объектіге байланысты әдісті активтеу арқылы.

Бұл екі тәсіл, кейбір қолданушы немесе жүйелік оқиға орнауымен байланысты болады. Оқиғалар қолданушының анықтаған әрекетімен байланысты, және осы оқиғаны өндейтін бағдарламаны іске қоса алады. Бұл - батырманы шерту немесе пернені басу болуы мүмкін. Оқиғалар, сонымен қатар, бағдарламалық код қосымшасында (пішінді жадыға жүктеу) немесе тікелей жүйелік ортада орындалады. Оқиғаларды өндеу үшін автоматты түрде шақырылатын оқиғаны өндеу процедурасында өзіндік бағдарламалық коды анықталады.

Дәстүрлі немесе “процедуралық” бағдарламалау модельдерінде бағдарламалық кодтың кейбір бөлігін орындау логикасы мен тізбегінде басқару тікелей қосымша арқылы жүргізіледі. Орындалу кодының бірінші қатарынан басталып, қажет кезінде процедураларды шақыра отырып, қосымша арқылы стандартты түрде атқарылады.

Оқиғалық – басқару қосымшасында код алдынала анықталған, қатаң кодталған түрде жүрмейді. Оның орнына, кодтың әртүрлі бөлімдері оқиғаға жауап ретінде орындалады. Бұл оқиғалар тізбегінде, кодтың орындалу тізбегін анықтаумен бірге, код қосымшасының орындалу жолы да әр кезде өзгеше болады.

Әдістер - объектімен байланысты, кейбір өңдеулерді орындайтын бағдарламалық процедуралар. Мысалы, командалық батырманы шерту арқылы пішінді ашу керек болса, командалық батырма арқылы Clickt процедурасының денесіне сәйкес бағдарламалық кодты қосу керек.

Visual Basic-тің стандартты тәсілдері екі категорияға бөлінеді:

1) стандартты оқиғалар әсер ететін процедуралар. Оқиға жиыны әр объект қосымшаның компоненттері үшін автоматты орындалатын Procedure объект кодының терезесіндегі тізімде келтірілген;

2) нақты құрушының бағдарламалық кодында шақырылатын стандартты әдістер. Visual Basic-те стандартты әдістер кеңейтілген диапазонда орналасады.

Сонымен қатар, бағдарлама құрушы объектіге тән және қосымша орындалу кезеңінде кейбір есептерді шығаратын өзінің қолданушы процедураларын құра алады.

Қасиеттер - объектінің көрсетілімін, тәртібін, т.б. түрін анықтайды. Пішіндегі жазбаларды көру көзіне пішін фонының түсі, деректер базасын ашу сеансында қолданылатын байланыс қатары және деректер базасының кестелері - мұның бәрі кез келген объектілердің қасиеттері болып саналады. Қосымшаны құру кезінде өзгертуге болатын қасиеттер тізімі, қасиеттерді басқару элементтерінің терезесінде бейнеленеді. Сонымен қатар, қасиеттер мен әдістер объект интерфейсі деп аталады.

Әдіс немесе қасиет пе? Қойылған бір мақсатқа әртүрлі жолдармен жетуге болады. Мысалы, Mov әдісі объект позициясын өзгертеді және кейбір қасиеттер арқылы да (Top, Left) осы нәтижеге

жетуге болады. Әдістер мен қасиеттердің негізгі айырмашылығы: қасиеттерді жобаны құру кезінде де, қосымшаны орындау кезінде де өзгертуге болады. Ал тәсілдерді тек орындау кезінде өзгертеміз. Кейбір қасиеттерді қосымшаны құру кезінде пайдалануға болмайтынын, ал кейбіреуін жұмыс кезінде тек оқуға болатынын ескерте кету керек.

Объектілі бағытталған бағдарламалауда мына түсініктер негізгі рөл атқарады:

- кластар;
- инкапсуляция;
- туындылар;
- полиморфизм.

2.3.3. Кластар

Класс – негізінен соңғы нәтижесінде объект құрылатын үлгі немесе жоба ретінде сипатталады. Бұл жағдайда әр объект класс түрінде сипатталып, олар элементтер панелінде орналасады.

2.3.4. Инкапсуляция

Инкапсуляция – ОББ-ның барлық версиясында объектімен жұмыс істеудің негізгі концепциясы болып табылады. Формальды түрде *инкапсуляция* дегеніміз - деректер мен объект тәртібінің түрлерін бір пакетке біріктіру, оны жүзеге асырудың анық-қанығын қолданушыдан жасыра білу. Объект деректері - айнымалы-мүше, айнымалы түрі немесе түр өрісі деп те аталады. Класс түрі болып келетін объект айнымалы түрінің кейбір мәндерінен тұруы керек. Бұл мәндер объектінің ағымдағы күйін анықтайды. Сонымен қатар:

Visual Basic класындағы функциялар мен процедуралар объектінің әдістері мен қасиеттеріне сәйкес келеді;

Қолданушы объектінің ағымдағы күйіне (түрі, өрісіне) тек әдістер мен қасиеттер арқылы ғана қол жеткізеді.

Бағдарламалар да бұл деректермен тек объект қасиеттері мен әдістері арқылы қатынаста болуы керек.

Деректерді private-айнымалылар түрінде қолдану - объектілерді көптеген қолданудағы сын шарттары мен қосымшаның сенімділігіне, объект қасиеттері мен тәртібіне “қара жәшіктің” түрін берудің жалғыз тәсілі.

2.3.5. Туындылар

ОББ-да кластарды құру процедурасы бұрынғы кластар базасының негізінде құрылса, онда ол айтарлықтай жеңілдейді. Базалық (аталық) кластың жалпы концепциясының кеңеюі - *туындылау* деп аталады. Туынды қарым-қатынасы объектінің мамандығын бекітеді. Егер А класы В класын кеңейтетін болса, онда ол оның тәсілдерінен туындайды. Бірақ, мүмкіншіліктері көбірек болады, яғни туынды объектілеріне аталық класқа тән әдістерге ешқандай қатысы жоқ жаңа әдістер мен қасиеттер болуы мүмкін. Visual Basic ОББ-ның классикалық түсінігінде ол туындауды қолдамайды. Дегенмен, бұрын құрылған басқару элементтерінен, басқару элементін құру процесі туындыны іске асыруы сияқты болғанымен, бұл екі объект арасында ешқандай аталық-туынды қарым-қатынасы орнамайды. Себебі, ОББ-да туындаудың жай версиясы туындаумен инкапсуляцияның арасындағы сәйкестік болмайды деп ұйғарады. Егер В объектісі А объектісінің барлық қасиеттерінен туындап, А объектісі өзгерсе, онда В объектісінің функционалды мүмкіндіктерін абайсызда бүлдіріп алу қаупі бар (бұл кейде нәзік базалық кластың проблемасы деп аталады).

Visual Basic туындылау кезінде нәзік базалық класс проблемасын айналып өтеді, ол үшін интерфейс концепциясын қолданады. Интерфейстер (әдіс, қасиет және оқиға жиыны) тура туындау механизімін қолданбай-ақ, функционалды мүмкіндіктердің көпшілігінің көрсетілімін сипаттайды. Ол нәзік базалық класс проблемасын қозғамауға мүмкіндік береді. Visual Basic-те интерфейссті бағдарламалау жұмысының көбін орындайтын Active Control Interface Wizard мастери бар.

2.3.6. Полиморфизм

Полиморфизм (грек тілінен “көпжақтылық”) дегеніміз - туындалған объектілердің, туындау тізбегінде орналасқан орнына байланысты, қандай тәсілді қолдану керек екенін анықтайды. Мысалы, мотоцикл мен автомобиль (екеуі де “көлік құрылғылары” абстракты класынан туындалған) “оңға бұрылу” тәсілін атқаруы керек, бұл екі объектінің бұрылу тәсілі әртүрлі. Мәселен, объект алынған хабарға негізделіп, қай тәсілді орындау керектігін таниды. Бұл -

қолданушы мен бағдарлама құрушы хабар берілген объектінің қай класқа жататынын білуі шарт еместігін білдіреді.

2.3.7. Объектаралық қатынас

Жеке объектілердің сипаттамасын өзгертуді толықтыру үшін, қосымшада объектаралық қатынас анықталған болуы тиіс. Мысалы, пішін мен кестені байланыстыру кезінде қосымша объектісімен деректер базасының объектісімен қарым-қатынасы анықталған болуы тиіс. Егер кестеде таңдалған және пішімде көрсетілген деректер қайта реттелсе, онда осы екі объект арасындағы қатынас қайта анықталады. Осы кезде, басылған командалық батырма пішімді ашқанда, байланыс әрекеті басқаша болады; Click() процедурасына командалық батырмамен қосылатын әдістің кодын анықтайды.

2.3.8. Қосудың иерархиялық құрылымы

Объектілі бағдар қосымшасында кейбір объект кластарының құрамында басқа объектілер болуы мүмкін. Мысалы, пішін басқару элементтерінен, статикалық объектілерден, т.б. контейнерлерден (мысалы, кадрлерден) тұруы мүмкін. Құру саймандар құрылғылары, қай объектілердің құрамында басқа объектілер бар екенін анықтап және олардың иерархиялық құрылымын жоғарыдан (қосымшалар) төменге қарай (мысалы, пішін ішіндегі командалық батырма) құрады.

2.3.9. Модульділік

Объект модульділігі - инкапсуляциямен тығыз байланысты. Объектілі-бағытты құру ортасында объектілер құрамында олардың толық анықтамасы болуы керек. Процедуралар мен объект қасиеттерінің барлық анықтамалары тек объект ішінде орналасуы керек. Бұл қасиет - объектіні басқа объектіге көшіру мүмкіндігін: пішіндерді, есеп беруді, қолданушы кластарын, түр көрсеткіштерін жоғалтпай, тәртібі мен негізгі сипаттамасын береді.

327591(4) 17



Бакылау сұрақтары:

1. Объектілі бағытталған бағдарламалау дегеніміз не?
2. Объектіні анықтау параметрлерін атаңыз.
3. Инкапсуляция дегеніміз не?
4. Объектілерді басқару тәсілдерін атаңыз.

3. VISUAL BASIC 6 ЖОБАЛАУ ЖҮЙЕСІНДЕ ЖҰМЫС АТҚАРУ

3.1. Visual Basic 6-ны іске қосу

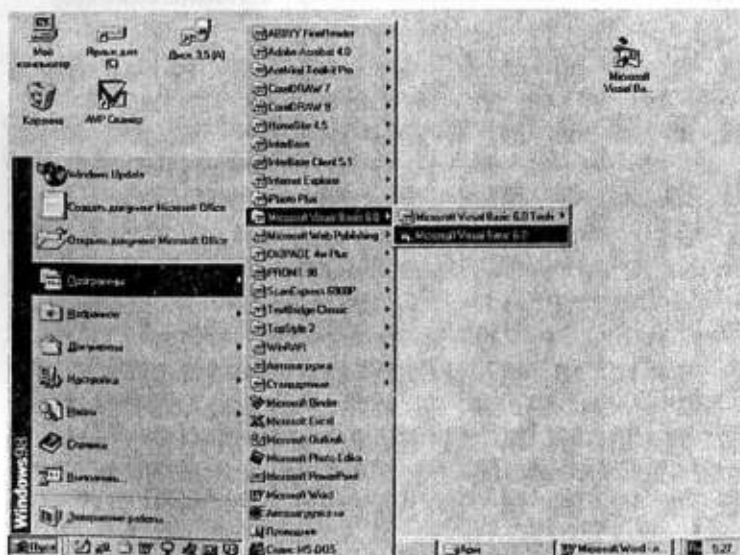
Visual Basic іске төмендегі тәсілдер арқылы қосылады:

- маустың сол жақ батырмасымен Visual Basic 6.0 пиктограммасын (3.1- сурет) екі рет шерту керек;

- Windows-та “Іске қосу” (“Пуск”) белгісін шертіп, шыққан менюден “Программаны” ашып, оның ішіндегі Microsoft Visual Studio бағдарламалар тобынан Microsoft Visual Basic 6.0 пунктін іске қосу керек (3.1-сурет);

- “Сілтеуіш” (“Проводник”) көмегімен Visual Basic 6.0. exe файл орналасқан буманы тауып, оны іске қосу керек.

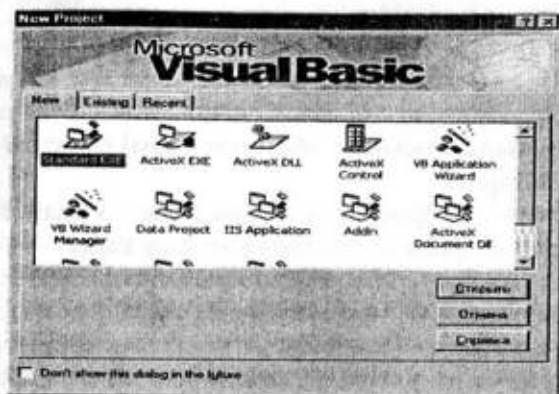
Жоғарыда аталған VB іске қосу тәсілдерінің бірін орындағанда, монитор экранында VB жобалау ортасының негізгі панелі пайда болуы керек. Ол панель үстінде New (жаңа жоба), Existing (жұмысқа қосылған жоба) және Recent (жақында құрылған жоба) терезелері орналасады.



3.1-сурет. Visual Basic-ті іске қосу

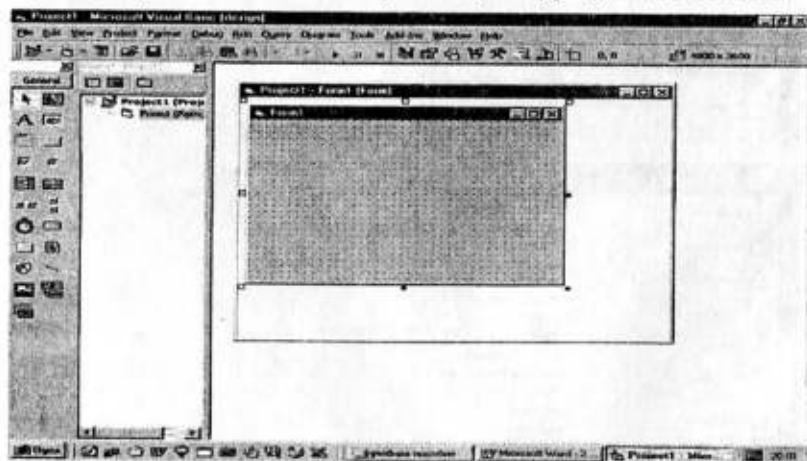
3.1.1. Жаңа жоба құру (New бетбелгісі)

3.2-суреттегі бетбелгіден Standard.exe пиктограммасын таңдап, біз VB жобалау ортасына стандартты тәсілмен кіреміз. Бұл жұмысты бастаудың ең оңай тәсілі, әрі ол жаңадан жұмыс істеп үйренушілерге (студенттерге) арналған.



3.2-сурет. VB жобалау ортасын стандартты тәсілмен іске қосу

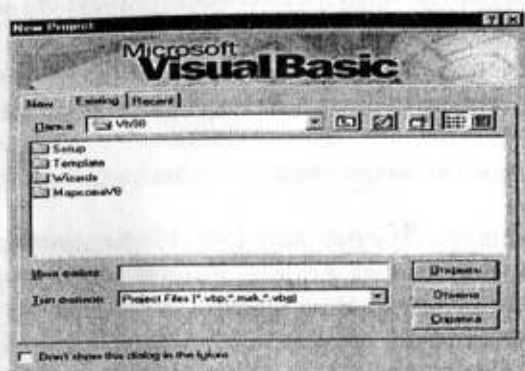
Standard.exe пиктограммасын маус батырмасымен екі рет шертсек, экранда VB негізгі терезесі (3.3-сурет) пайда болады.



3.3-сурет. Стандартты тәсілмен VB6 кіру кезіндегі жобалау ортасының негізгі панелі

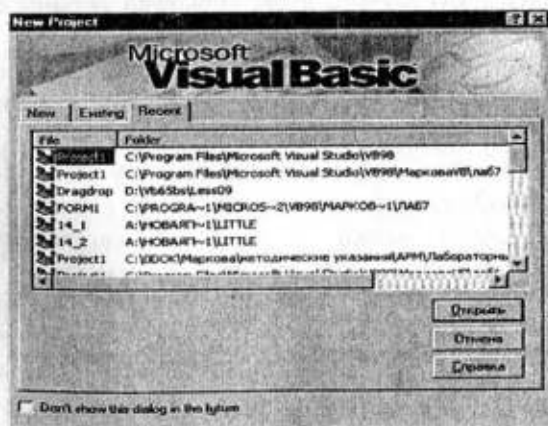
3.1.2. Жұмысқа қосылған (Existing) немесе жақында құрылған (Recent) жобаның терезелерін ашу

Existing бетбелгісін қолдансақ, VB жобасында ағымды бумада орналасқан көптеген файлдарды қарастырып, оның ішінен өзіңізге қажетті файлды (3.4-сурет) таңдап алуға болады.



3.4-сурет. Existing бетбелгісі

Recent бетбелгісін қолдансақ, алдымен жақында жұмысқа қосылған жобалардың файлдарын көруге болады (3.5-сурет). Кезекте бірінші болып, алдыңғы сеанста жұмысқа қосылған жоба тұрады.



3.5-сурет. Recent бетбелгісі

Екінші кезекте, оның алдындағы жоба тұрады, т.с. Жұмыс кезінде бұл кезекті қолдану өте ыңғайлы, себебі бір жобаны құру кейде бірнеше күнге созылады.

3.2. VB жобалау ортасының негізгі панелі

VB жобалау ортасының негізгі панелі (3.6-сурет) келесі панелдер мен терезелерден тұрады:

- *тақырып қатары* – жоба аты, берілген жоба пішінінің аты, жақша ішінде жұмыс істеу режимі: design - жобалау режимі, run – орындау режимі;

- *меню панелі* – меню панелі тақырып қатарынан кейін орналасқан.

Меню панелінде Негізгі менюдің 13 батырмасы (опциясы) орналасқан: **File, Edit, View, Project, Format, Debug, Run, Query, Diagram, Tools, Add-Ins, Window, Help.**

Асты сызылған әріп, маус жұмыс істемей қалғанда, пернелер тақтасынан Alt+ <асты сызылған әріп > пернелерін бірге басу арқылы қажетті опциямен жұмыс істеу мүмкіндігін береді.

Файл (File) менюі - Microsoft фирмасының басқа қосымшасындағыдай стандартты командалардан тұрады, мысалы, *Print* (баспаға алу), *Exit* (жұмысты аяқтау). Бұл меню командаларының арасында басқа қосымшаларда жоқ командалар да пайдаланылады: жаңа жобаны құру, құрылған жобаны *ашу* (жүктеу), ағымдағы жобаны *сақтау*.

Түзету (Edit) менюі - түзетудегі кеңінен танымал командалардан тұрады, мысалы: *киып алу*, *көшіру*, *қою*, *табу*, *орнын ауыстыру*, *жою*, *жылжыту*, *қалпына келтіру*.

Түр (View) менюі - жұмысты жалғастыру үшін *жобаның негізгі панелінде саймандар терезесін* ашу мүмкіндігін береді.

Жоба (Project) менюі - жұмысты жалғастыруға арналған командалардан тұрады. Мысалы, *Add* ағымдағы жобаға жаңа элементті (*пішінді*, *файлды*, *модульді*, *басқару элементін*, т.б.) қосу.

Формат (Format) менюі - экран пішінінде орналасқан объектілерді форматтау мүмкіндігін береді. Мысалы, олардың өлшемін өзгерту, белгіленген объект тобының өлшемдерін реттеу, т.б..

Жөндеу (Debug) менюі - бағдарламаны *жөндеу* мүмкіндігін береді, бағдарламадағы қателерді табуға арналған командалар жиынынан тұрады.

Орындау (Run) менюі – бағдарламаны орындау режимін басқару мүмкіндігін беретін (бағдарламаны орындау, жұмысты доғару, бағдарлама жұмысын аяқтау) командалардан тұрады.

Саймандар (Tools) менюі - жаңа процедуралар мен функцияларды құру, *түзету менюінің терезесін* ашу, жобалау ортасының параметрлерін баптау командаларынан тұрады.

Утилиттерді қосу (Add-Ins) менюі – қосымша мүмкіндіктерін қосу арқылы жобалау ортасының мүмкіндіктерін арттырады.

Терезе (Window) менюі - *негізгі панельдегі* терезенің орналасу ретін (*жазық, тік* немесе *каскадты* түрін) өзгерту мүмкіндігін береді.

Анықтама-көмек (Help) менюі - Visual Basic-тің *анықтама жүйесімен* жұмыс істеу мүмкіндігін береді.

- *Саймандар тақтасы*. Саймандар тақтасы - *негізгі меню* қатарынан кейін орналасқан. Бұл панельдегі саймандардың (батырмалардың) атқаратын қызметі - *менюдің* атқаратын кейбір командаларын орындайды. Батырмалар *негізгі менюдің* жиі қолданылатын командаларын жылдам орындауға арналған. Мұнымен қатар, батырмаларды кейде *пиктограмма* немесе *саймандар* деп те атайды (олардың саймандар панелі – деп аталатыны да сондықтан).

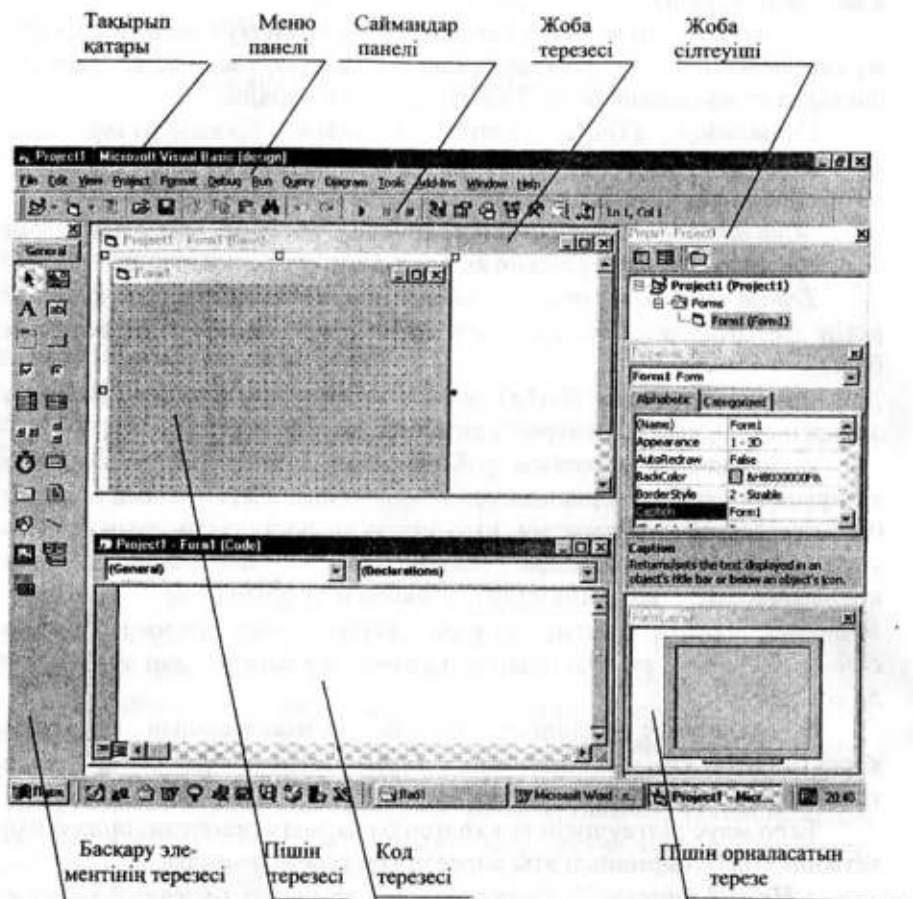
3.1- кестеде берілген панель саймандарының барлығы көрсетілген. Әр батырманың түрі, ағылшынша аты және тағайындалуы қысқаша түрде берілген.

Егер маус сілтеушісін кез келген батырмаға орнатсақ, онда оның астында сол батырманың аты көрсетілген терезе шығады.

- *Негізгі панель*. Жобалау процесінде негізгі панелде 13 терезе ашылады:

Басқару элементтерінің терезесі (Toolbox), пішін терезесі (Object), жоба терезесі (Project Explorer), код терезесі (Code), қасиеттер терезесі (Properties), сипаттамаларды қарастыру терезесі (Object Browser) және пішін орналасатын терезе (Form Layout). Екі редакциялау (түзету) терезесі - (пішінді редакциялау – Form Editor және кодты редакциялау - Edit), бағдарламаны жөндеу терезесі – Debug, қалған үш терезе (шұғыл орындау терезесі –

Immediate, бақылау терезесі – Watch, локалды айнымалылар терезесі – Locals) тек жобаны іске қосу және жөндеу кезінде пайда болады.



3.6-сурет. Visual Basic 6 жобалау ортасының негізгі панелі

Басқару элементінің терезесі (Toolbox)

Toolbox терезесі жобаның негізгі панелінде болуы да, болмауы да мүмкін. Бұл терезені ашу үшін:

- *View/Object* командасының көмегімен саймандар тақтасын шерту керек.

Саймандар тақтасындағы батырмалар

Саймандар тақтасындағы батырмалар	Батырма аты және тағайындалуы
Add Standard EXE Project	Ағымдағы жобаға жаңа жоба қосу
Add Form	Ағымдағы жобаға жаңа пішім қосады
Menu Editor	Редактор (түзету) менюін шақыру
Open project	Жобаны жүктеу
Save Project	Жобаны сақтау
Cut	Редакциялау: буферге қиып алу
Copy	буферге көшіру
Paste	буферден көшіру
Find	Іздеу
Can't Undo	Доғару (орындалған қызметті доғару)
Can't Redo	Қалпына келтіру
Start	Іске қосу
Break	Жұмысты уақытша тоқтату
End	Бағдарлама жұмысын аяқтау
Project Explorer	Жобалау ортасында инструментальды <i>Жоба сілтеуші</i> терезелерін ашу
Properties	Жобалау ортасында инструментальды <i>Объект қасиеттері</i> терезелерін ашу
Form Layout	Жобалау ортасында инструментальды <i>пішімді орналастыру</i> терезелерін ашу
Object Browser	Жобалау ортасында инструментальды <i>объектілерді және олардың сипаттамасын қарау</i> , терезелерін ашу
Toolbox	Жобалау ортасында инструментальды <i>басқару элементтері</i> терезелерін ашу

3.2-кестеде VB стандартты басқару элементтерінің комплексті аттары көрсетілген. Кестеде көрсетілген саймандар тізімі, олардың *Toolbox* терезесінде орналасқан тізімге сәйкес келеді.

Экран пішімінің терезесі - Form 1

Object экран пішімінің терезе құрамы *Form 1* экран пішіні және *Project1* жоба терезелерінен тұрады.

Экран пішімінің терезесін төмендегідей ашуға болады:



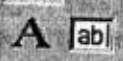







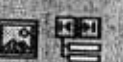

- *View/Object* командасының көмегімен;
- Саймандар панеліндегі батырма көмегімен;

Алғашқы жүктелгенде экран пішіні бос болады. Ал оның қасиеттерінің мәндері *Visual Basic* жүйесінің үнсіз бекітілген

мәндеріне тең болады. Үнсіз жарияланған кезде *Visual Basic* экран пішініне *Form 1* атын, қасиеттер мәні ретінде (*Name*) бекітеді. Ал бүкіл жобаға *Project1* атын меншіктейді.

3.2-кесте

VB басқару элементінің стандартты комплексі

Аталымы	Таңба	Аталымы
Нұсқағыш (Pointer)	 	Графикалық өріс (Picture Box)
Жазба (Label)		Мәтіндік өріс (TextBox)
Жақтау (Frame)		Командалық батырма (CommandButton)
Жалауша (CheckBox)		Ауыстырып – қосқыш батырмасы (Option)
Терезе комбиннациясы (ComboBox)		Терезе тізімі (ListBox)
Жазық айналдыру сызғышы (HscrollBar)		Тік айналдыру сызғышы (VscrollBar)
Таймер (Timer)		Құрылғы тізімі (DriveListBox)
Каталог тізімі (DirectoryList)		Файл тізімі (FileListBox)
Фигура (Shape)		Түзу сызық (Line)
Бейне (Image)		Деректер (Data)
Контейнер OLE (OLEContainer)		

Экран пішінінің өлшемдері төмендегідей орнатылады:

- Экран пішінінің оң немесе төменгі қатарын мауспен “ұстап” алып, оны көз мөлшермен жазық немесе тік бағытта созуға не қысуға болады.

Бұл жұмысты аса бір дәлділікпен де орнатуға болады. Ол үшін қасиеттер терезесінен *Width (Ені)*, *Height (Биіктігі)*, *Left (Сол жақ*

шеті) және Тор (Оң жақ шеті) қасиеттерінің қажетті мәнін таңдау керек.

Саймандардың *негізгі панелінің* оң жақ бұрышындағы саймандар панелінде орналасқан объектінің *орналасу ретінің индикаторы* мен *өлшемінен* экран пішінінің орналасу реті мен өлшемі туралы мәлімет алуға болады.

Пішін торлар желісімен бөлінген – ол пішінде неше түрлі басқару объектілерін орналастыру жұмысын ыңғайлайды. Көрші нүктелер арасы анықталған ара қашықтықта (үнсіз жариялауда 120 твин) бекітілген және пішіндегі объектіні созу немесе орнын ауыстыру қалқыма түрде емес, нүктеден нүктеге дейінгі серпілісте орындалады. Мұнымен пішіндегі объектінің әртүрлі комбинацияларын өте әдемі етіп тұрғызуға болады.

Project Explorer жоба сілтеуішінің терезесі

Жоба сілтеуішінің терезесі - жоба құрамының графикалық кескінінен тұрады. *Жоба түрі* – жаңадан құрылатын (жаңа жоба) не Visual Basic ортасында өзгертуге немесе толықтыруға арналған (құрылған жоба) болады.

Графикалық кескін дегеніміз – жоба құрамының бұтақ тәріздес немесе барлық файлдар тізімімен сипатталуын айтамыз (егер жоба бірнеше экрандық пішіннен тұрса, онда осы сияқты бірнеше файлдардың экрандық пішіні). Әрине, бұтақ тәріздес құрылымда (немесе тізімде) экрандық пішіннен (немесе бірнеше пішіннен) тұратын жобаның өзіндік файлы кіруі керек. Жоба құрамында файлдардың басқа да түрі болуы мүмкін, мысалы бағдарламаның модульдік файлдары.

Жоба сілтеуішінің терезесі - *Project Explorer* жобаның негізгі менюіндегі *View* менюі көмегімен немесе саймандар тақтасынан – *Project Explorer* батырмасын шерту арқылы ашуға болады.

Жоба сілтеуішінің терезесі, азғана батырмасы бар саймандар тақтасынан тұрады:

- *View Code* - *бағдарлама код терезесін* ашу;
- *View Object* - *Экран пішінінің терезесін* ашу;
- *Toggle Folder* – бумаларды ауыстыру, егер батырма басылған болса, онда жоба құрамы бұтақ, егер батырма басылмаса - құрамы тізім түрінде сипатталады.

Code бағдарламалық код терезесі

Жаңа құрылатын жоба үшін бағдарламалық код терезесі бос болады.

Жобаның негізгі терезесіндегі View менюіндегі *View Code* командасының көмегімен немесе *Project Explorer* - жоба сілтеуішінің саймандар тақтасынан *View Code* батырмасын шерту арқылы бағдарламалық код терезесін ашуға болады.

Бағдарламалық код терезесінің тақырып қатарынан кейін, тізімдері бар екі терезе орналасқан. Ол терезелердің әрқайсысы бір қатармен бейнеленеді. Терезенің оң жағында орналасқан бағыттауышы бар батырманы бассақ, тізімді ашуға болады.

Бірінші тізімде *жоба объектілерінің* тізбесі, ал екінші тізімде – бірінші тізімде белгіленген объект *процедураларының* тізбесі беріледі. Екінші тізімдегі процедура аттары – белгіленген объектілермен орындалатын оқиға аттарын білдіреді.

Терезе астында - *жазық бағытта айналдыру сызғышы*, ал оң жағында – *тік бағытта айналдыру сызғышы* орналасқан. Олардың атқаратын қызметі – *ұзындығы және биіктігі* бойынша өте үлкен болуы мүмкін бағдарламалық код мәтінін қарастыру.

Тік бағытта айналдыру сызғышының үстінде (*Split*) кішкентай батырма орналасқан, оның көмегімен әрқайсысына код мәтінін толығымен орналастыруға болатындай етіп терезені екі бөлікке бөледі. Мұндай терезені бағдарлама мәтіні өте үлкен болғанда, оны өңдеу үшін қолданған ыңғайлы.

Жазық бағытта айналдыру сызғышының сол жағында екі батырма орналасқан. Олар арқылы тек берілген процедураны (*Procedure View*) модульді (*Full Module View*) толығымен қарастыруға болады. Модульдің өзі бірнеше процедуралардан тұруы мүмкін.

Properties объект қасиеттерінің терезесі

Объект қасиеттерінің терезесін екі тәсілмен: негізгі панельдегі *View* менюінен *Properties Window* командасының көмегі немесе *негізгі панельдегі* саймандар тақтасынан *Properties Window* батырмасын шертіп немесе *F4* батырмасын басу арқылы ашуға болады. Қасиеттер терезесі екі тізімнен тұрады: жоғарғысы – берілген экран пішінінде орналасқан барлық басқару элементтерінің аттарының тізімі; төменгісі – жоғарғы тізімде белгіленген элемент

қасиеттерінің алфавит бойынша орналасқан тізбесі. Терезенің оң жақ жақтауындағы айналдыру сызғышының көмегімен барлық тізімді қарастыруға болады.

Form Layout пішінің орналастыру терезесі

Бағдарламаны орындау барысында (қосымшаны жобалау процесінде емес) монитор экранына пішіннің экрандық көрінісін бейнелеу мүмкіндігін, пішінді экранға орналастыру терезесінен орнатамыз. Экранда пішінді орналастыру терезесін екі тәсілмен ашуға болады: жобаның негізгі панелінен View менюінен, Form Layout Window командасының көмегімен не негізгі панельдегі саймандар тақтасынан Form Layout Window батырмасын шерту арқылы. Монитор экранының өрісіндегі *ақ төртбұрыш* - жұмыс кезіндегі қосымша (экрандық пішін) терезе. Бұл төртбұрышты экранда өзіңізге ыңғайлы жерге мауспен «тасымалдауға» болады. Сонымен, біз қосымша пішіндерін экранға орналастыру жұмысы кезінде пішіндердің бір-бірінен басылып қалмауын, оларды экран бетінде көзге көрнекті қылып орналастыру мүмкіндігін аламыз. Пішінді «тасымалдау» кезінде, оның *Left* және *Top* қасиеттерінің мәні автоматты түрде өзгереді.

3.3. Visual Basic-тен шығу

Visual Basic-тен шығып, Windows-қа оралу үшін әдеттегі жұмысты аяқтау тәсілдерін қолданамыз:

- File/Exit таңдау;
- VB негізгі терезесін жабу батырмасын шерту;
- Alt + F4 пернелерін басу;
- VB-тің Control Menu белгісін екі рет шерту (экран пішінінің жоғарғы сол жақ бұрышындағы жүйелік меню).

Бақылау сұрақтары:

1. *Visual Basic* жүйесінің тағайындалуын атаңыз.
2. Жұмысты бастау режимдерін атаңыз.
3. *Visual Basic*-тің саймандар панелін атаңыз.
4. *Visual Basic*-ті іске қосу тәсілдерін атаңыз.
5. *Visual Basic*-тің жұмысын аяқтау.

4. VISUAL BASIC БАСҚАРУ ЭЛЕМЕНТТЕРІ

Visual Basic-те WINDOWS-қосымшамаларын іс-тәжірибеде басқару элементтерін қолданбай құру мүмкін емес. Себебі, бұл элементтер қолданушыға арналған интерфейсті қамтамасыз етеді. Қолданушы мен қосымша арасындағы әрекет, басқару элементтеріне мәтін енгізу арқылы атқарылады.

4.1. Басқару элементтерін орналастыру

Басқару элементтерін пішінге орналастырудың екі тәсілі бар:

1) *Tool Box* (элементтерді басқару панелі) терезедегі кез келген басқару элементін екі рет шертсек, онда ол элемент *Form* (Пішін) терезесінің ортасына орналасады. Егер пішін ортасында алдынала бір басқару элементі орналасқан болса, жаңа элемент бұрынғы элемент үстіне орналасып, *Form* (Пішін) терезесінде біртекті нүктелермен қалыптасқан тормен байланысады. Тор қадамын өзгерту үшін *Tools* (Сервис) менюіндегі *Option* (Баптау) командасын шертіп, оның ішіндегі, *General* (Жалпы) командасын таңдаймыз. Тор қадамы арнайы өлшем бірлігімен – $\text{twin.ltwin} = 1/20$ пункт немесе 0,0176 м өлшенеді;

2) егер басқару элементін (БЭ) шертсек *Tool Box* (элементтерді басқару панелінде), БЭ пернесі түске боялып белгіленеді. Одан кейін маус меңзері айқастырылған сызықтар түріне айналады. Белгіленген БЭ - тек айқастырылған меңзерді шерткенде ғана пішінде бейнеленеді. БЭ - маус пернесін босатқанда ғана түпкілікті бейнеленеді.

4.2. Өлшемдерді орнату және басқару элементінің орнын ауыстыру

Жұмыс барысында тек таңдап алған элементтің өлшемін өзгертуге болады әрі оны өзгертудің негізі ретінде - өлшемді баптаудың сегіз маркері қолданылады. Осы сегіз маркердің кез келгенін әртүрлі бағытта сүйреп, БЭ өлшемін үлкейтуге не кішірейтуге болады.

Маус көмегімен БЭ пішін терезесінің кез келген аумағына сүйреп апарып орналастыруға болады. Ол үшін БЭ-тің маустың сол

жақ батырмасымен шертіп, батырманы баса отырып, БЭ Form терезесіндегі қалаған жерге апарып қоямыз.

Егер бірнеше БЭ бір топ ретінде басқа жерге ауыстыру керек болса, онда Ctrl пернесін баса отырып, оларды біртіндеп белгілеп немесе БЭ “құрық құру” тәсілі арқылы төртбұрышты таңдауды бір топқа бірігетін элементтер айналасына келтіріп белгілеу керек. Маус батырмасын босатқаннан кейін, БЭ таңдалған бөлікше ішінде белгіленеді.

4.3 Қасиеттерді орнату

Қасиеттер негізінен, сыртқы түрі мен басқару элементтерінің функциялануын анықтайды. VB басқару элементтерінің қасиеттерін қосымшаны құру кезінде ғана емес, бағдарламаны орындау кезінде де орнатуға мүмкіндік береді.

Form терезесіне басқару элементін қосқан кезде Properties терезесі жаңарып, ағымдағы таңдалған басқару элементтерінің қасиетін бейнелейді. Таңдалған басқару элементі болып, негізінен пішінде соңғы орналасқан басқару элементі саналады. Visual Basic-те басқару элементінің қасиетін Properties терезесінде көру үшін бұл элементті шертіп, немесе Properties терезесінің ашылған тізімінен бұл басқару элементін таңдау керек.

Left, Top, Height және Width қасиеттері Properties терезесіне қатынамай-ақ орнатуға болатын қасиеттерге жатады. Басқару элементтерінің өлшемдерін баптап, оны орнынан жылжытқанда, Visual Basic Left, Top, Height және Width қасиеттерін бұл басқару элементін Form терезесінде орнауы мен оның өлшеміне сәйкес өзгертеді. Орналасуы мен пішін өлшемінің өзгеруіне байланысты болғандай, бұл қасиеттер де типтермен өлшенеді (егер Skale Mode қасиетінде басқа бір мәні анықталмаған болса). Left басқару элементі пішіннің сол жақ шетінен қандай ара қашықта орналасқанын, Top – бұл басқару элементі пішіннің жоғарғы шетінен қандай ара қашықта орналасқанын, ал Height және Width оның өлшемін білдіреді.

Қосымшаны орындау кезіндегі басқару элементтерінің қасиеттерін орнату кезінде қасиеттерге қатынау синтаксисі мына түрде беріледі.

[Пішін.] Объект. Қасиет=Мәні

Егер процедурада жазылған пішінге жататын басқару элементімен қатынас орнату керек болса, онда пішін атын жазбай-

ақ қоюға болады. Басқару элементінің қасиет мәні де осы сияқты саналады. Әр қасиет - элемент айнаымалысының ішкі мәні болып табылады. Оның мәнін тек орнатуға ғана емес, оқуға да болады.

Жалпы жағдайда қасиеттер мәні мына түрде оқылады:
Айнаымалы = [Пішін.] Объект. Қасиеті

4.4 Басқару элементінің негізгі қасиеттері

4.4.1 *Позиция*

Басқару элементінің позициясын төрт қасиет: Left, Top, Height және Width анықтайды. Үнсіз жариялау бойынша бұл мәндер твип (twip) өлшеу бірлігі ретінде қолданылады. Твип - бұл экрандық-тәуелсіз өлшеу бірлігі, принтердің 1\20 нүктесіне тең және экран мүмкіндігін ескермей қосымша элементін бейнелеу кепілдігін береді.

Top және Left қасиеттері - басқару элементінің жоғарғы сол жақ бұрышының координатын, ал Height, Width - оның ені мен биіктігін орнатады. Координат жүйесінде санау жоғарыдан төмен (Y) және солдан оңға қарай (X) жүргізіледі. Осы қасиеттері арқылы пішінде басқару элементінің орнын ауыстырып, оның өлшемдерін өзгертуге болады.

4.4.2 *Түсі*

Элементтердің түсін безендіруді басқару - Back Color, Fill Color және Fore Color қасиеттері арқылы орындалады. Үнсіз жариялау кезінде, оларға Windows-тың стандартты түстері тағайындалады.

4.4.2.1 *BackColor қасиеті*

Back Color қасиеті арқылы фон түсі орнатылады. Жобалау кезінде түсті бастау диалогті терезесінен қажетті түсті таңдайды, ал қосымшамен жұмыс істегенде, түс RGB түстік схемасын қолдану немесе VB RUN кітапханасының константалары арқылы орнатылады.

4.4.2.2. *ForeColor қасиеті*

ForeColor қасиетінің көмегімен басқару элементтерінде қолданылатын мәтін мен графиканы бейнелейтін түсті анықтауға немесе орнатуға болады.

4.4.2.3. *FillColor қасиеті*

FillColor қасиетінің көмегімен shapes-толтыру түсін орнатуға (сурет ретінде салынған объектілер) болады.

4.4.3 *Қаріп параметрлері*

Басқару элементтеріндегі қаріп түрі Font (4.1-кесте) - қасиетінің мәнін орнату жолымен таңдалады.

4.1- кесте

Қаріп параметрі

Қасиеті	Мәні
Font.Name	Қаріп аты
Font.Size	Қаріп өлшемі
Font.Bold	Жартылай қалың
Font.Italic	Курсив
Font.Underline	Астын сызу
Font.StrikeThrough	Үстінен сызу
Font.Weight	Белгі қалыңдығы

4.4.4. *Басқару элементтің көрінісі және оны қолдану мүмкінділігі*

Қосымшамен жұмыс істеу кезінде кейбір басқару элементтерін, қолданушы өзгерте алмайтындай қылу керек. Ол үшін Enabled және Visible екі қасиеті қолданылады.

4.4.4.1. *Enabled қасиеті*

Enabled қасиеті оқиға басқару элементіне әсерін тигізе ме жоқ па, соны анықтайды. Егер қасиет мәні False болса, басқару элементі

жетімсіз болып, қолданушы оны жұмыста қолдана алмайды. Бұл жағдайда элемент - менюді таңдауға болмайтын басқа элементтердей сұр түске боялады.

4.4.4.2. *Visible қасиеті*

Visible қасиеті - басқару элементін көрінбейтіндей етіп орналастырады. Егер оның мәні False болса, онда ол көрінбейді, оған қатынау мүмкін емес. Мысалы:

```
Private Sub Command1_Click ()  
Command1.Enabled = False  
End Sub  
Private Sub Command2_Click()  
4 COMNRAND2.VISIBLE = FALSE  
End Sub
```

4.4.5. *Сыртқы түрі*

4.4.5.1. *Apperance қасиеті*

Басқару элементтерінің көбінде Apperance қасиеті бар, ол басқару элементінің бейнесіне жауап береді.

4.4.5.2. *ToolTipText қасиеті*

Басқару элементтерінің көбіне ToolTipText қасиетін орнатуға болады. Енгізілген мәтін, қолданушы маус нұсқаушысын пішіндегі басқару элементіне орнатқан кезде, әзір жауап ретінде бейнеленіп шығады.

4.4.6. *Name қасиеті*

Name қасиеті ерекше рөл атқарады. Оны орнату кезіндегі қателер, күрделі салдар туғызады. Аты - басқару элементінің идентификаторы.

4.5. Негізгі оқиғалар

4.5.1. Маусты шертту оқиғасы

4.5.1.1. Click оқиғасы

Click оқиғасы, қолданушы басқару элементін шерткен кезде шақырылады. Бұл оқиғаны өңдеу процедураларының аргументі берілмейді.

4.5.1.2. DblClick оқиғасы

DblClick оқиғасы басқару элементін маус батырмасымен екі рет шерткен кезде шақырылады. Мұның - оқиғаны өңдеу процедураларына аргументі берілмейді.

4.5.1.3. MouseDown оқиғасы

MouseDown оқиғасы маус батырмасын басқан кезде шақырылады және де оқиғаны өңдеу процедураларына бірнеше аргументі беріледі:

```
Private Sub Command1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
End Sub
```

Берілетін аргументтер маус батырмасының (Button), басқару пернелерінің (Shift), меңзер позициясының күйін анықтайды. X және Y аргументтері басқару элементтерінің жоғарғы сол жақ нүктесіне қатысты экрандағы маус меңзерінің позициясын анықтайды.

4.5.1.4. MouseUp оқиғасы

MouseUp оқиғасы маус батырмасын босатқан кезде шақырылады. Оның аргументтері 4.2-кестеде келтірілген.

4.5.1.5. Mouse Move оқиғасы

Бұл оқиға қолданушы маус батырмасын жылжытқанда шақырылады. Бұл оқиғаның өңдеу процедурасының синтаксисі төменде қарастырылған.

Private Sub Control_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)

Бұл оқиғаларға берілетін мәндер 4.2-кестеде келтірілген.

4.2-кесте

MouseDown және MouseUp оқиғаларының аргументі

Параметрі	Мәні
Button	Маус батырмасы басылған: 1 = сол жақ, 2 = оң жақ, 4 = ортанғы
Shift	Батырма басылған: 0 = ешқайсысы, 1 = [Shift], 2 = [Ctrl], 4 = [Alt], және олардың үйлесімі
X	X координатасы
Y	Y координатасы

4.5.2. Пернелік оқиғалар

4.5.2.1. KeyPreview оқиғасы

Егер KeyPreview пішін қасиетіне True мәні берілсе, онда оқиға пернелер тақтасынан алдымен пішінге, содан соң ағымдағы басқару элементіне беріледі.

4.5.2.2. Keypress оқиғасы

Keypress оқиғасы басылған перненің ASCII кодын қайтарып [PrintScreen] немесе [Alt] арнайы пернелер емес, тек [Enter], [Esc] және [Backspace] пернелері ғана басылады. Процедура басылған перненің ASCII кодын тұратын Key ASCII параметрін береді. Бұл параметр мән ретінде беріледі, яғни оны өзгертуге болады. Бұл мысалға, қолданушы енгізетін белгіні сүзгіден өткізу үшін қолданылады. Егер белгі жеткіліксіз болса, оның мәнін Key ASCII-ті нөлге теңеп, оны ары қарай өңдеуге (бейнелеуге, т.б.) тыйым салуға болады.

Control_Key Press (Key Ascil AsInteger)

4.5.2.3. *Key Down, KeyUp* оқиғалары

Бұл оқиғалар пернеліктің пернесін басып (KeyDown) және босатқанда (KeyUp) шақырылады. Бұл оқиға арнайы басқару пернелерін, мысалы функционалдық пернелерді басқанда да орнайды. Бұл жерде KeyCode және Shift екі параметрі беріледі.

Key Code басылған перненің пернелік кодына (ASCII коды емес) тұрады, мысалы vbKeyF1, ал Shift параметрі перне күйін хабарлайды [Shift], [Ctrl] және [Alt].

```
Control_KeyUp( Key Code As Integer, Shift As Integer)  
Control_KeyDown( KeyCode As Integer, Shift As Integer)
```

Пернені басқаннан кейін, оқиғалар келесі тізбекте орындалады: Key-Down, KeyPress, Keyllp .

4.5.2.4. *Фокус оқиғасы*

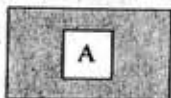
Фокус - Windows басқару элементтеріне қатынаудағы негізгі түсініктің бірі. Жоғарғыда айтылғандай, Windows жүйесі қай қосымша батырманы басатынын шешеді. Мұнда басқаруды екпінді элемент алады, яғни фокусы бар элемент. Егер элемент фокусқа ие болса, онда оған сәйкес түрде экранда бейнеленеді - мәтіндік өріс жыпылықтаған маркермен бейнеленіп, командалық батырма, жазба айналасындағы пунктирлі жақтаумен ерекшелінеді.

Visual Basic фокусті жөнелтуге байланысты: LostFocus және GotFocus екі оқиғаның бірін өңдеу мүмкіндігін береді.

Бір басқару элементінен екіншісіне көшкен кезде, алдыңғы элемент үшін Lost Focus оқиғасы, ал жаңа элемент үшін - Got Focus оқиғасы орнайды.

4.6. Стандартты басқару элементтері

4.6.1. Label (белгі) – басқару элементі



Белгі кез келген мәтіндік ақпарат пен жазбаларды жазу үшін қолданылады.

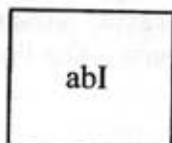
Белгі қасиеттері:

- 1) *Name (аты)* – Label 1 аты, префиксі «lbl»;
- 2) *Caption (белгі аты)* – белгі өрісіне шығарылатын мәтін, мәтін ұзындығы ≤ 1024 байт;
- 3) *Alignment (тегістеу)* – мәтінді тегістеу: 0 – солға қарай, 1 – оңға қарай, 2 – ортасы бойынша тегістеу;
- 4) *Font (шрифт, қарпін)* – шрифт түрі, оның өлшемі және жазылу тәсілі;
- 5) *Back Color / Palette (фон)* – фон түсі;
- 6) *Fore Color / Palette (шрифт түсі)* – мәтін әріптерінің түсі;
- 7) *Auto Size (жазық бағыт)* – True мәніндей белгі, Caption қасиетінің құрамын сыйғызу үшін жазық бағыт бойынша кеңейеді;
- 8) *Word Wrap (тік бағыт)* – True мәніндей белгі, мәтінді бірнеше қатарға орнату үшін тік бағыт бойынша кеңейеді.

Белгі оқиғалары:

- 1) *Click* – басқару элементін мауспен шерту;
- 2) *DoubleClick* – белгі аумағында маус батырмасын екі рет шерткенде орындалатын оқиға.

4.6.2. Text Box (мәтіндік терезе) басқару элементі



Мәтіндік терезе – алғашқы деректермен қорытындыға арналған тікбұрышты терезе. Мәтіндік терезедегі деректер, символдар (белгілер) қатары ретінде енгізіліп қорытындыланады.

Негізгі қасиеттері:

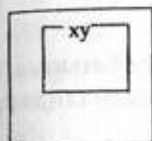
- 1) *Name (имя)* – Text 1, префиксі Txt. Мысалы, Txt терендік;
- 2) *Text 2 (мәтін)* – мәтіндік терезедегі мәтін мәні;
- 3) *Tool Tip Text* – әзір жауап ретінде бейнеленетін мәтіннен тұрады;

4) Visible (көрініс) – қолданушы үшін мәтіндік терезе көрінетінін немесе жасырылғанын анықтайды;

5) Fond (қаріп) – мәтін бейнеленетін қаріп. Үнсіз жариялау бойынша System қаріпі орнатылған;

6) Border Style =0 – мәтін өрісі жақтауланбайды;
Border Style =1 – мәтін өрісі жақтауланады.

4.6.3. Frame (кадр) - басқару элементі



Frame (кадр) – стандартты элементтер тобымен бейнеленеді. Бірнеше элементтердің интерфейсін жиектейтін төртбұрышты элементтер тобы (негізінен жалауша немесе ауыстырып-қосқыш).

Негізгі қасиеттері:

- 1) Name (аты) – Frame 1 аты, префиксі «fга»;
- 2) Caption (кадр аты) – топ тақырыбы.

4.6.4. Command Button басқару элементі (командалық батырма)



Кез келген батырманы басқан кезде батырма орындайтын әрекеттер туралы хабар беруші мәтін шығуы мүмкін. Батырманы басқан кезде, міндетті түрде сол батырма атқаратын арнайы функция орындалады.

Сонымен, бұл батырма қандайда бір операцияны белсенді етеді. Негізінен батырмалар диалогті панельдегі басқарушы элементтер түрінде қолданылады.

Батырманың кейбір қасиеттеріне тоқталайық:

- 1) Name (аты) - Command 1, префиксі «cmb»;
- 2) Caption (перне аты) – пернемен бірге бейнеленетін мәтін;
- 3) Cancel (доғару) – жұмысты доғару «болдырмау» батырмасы, егер мәні True болса, бұл батырманы шерткен кезде ESC – пернесін басқандай іс-әрекет атқарады;
- 4) Default (үнсіз) – егер мәні True болса, бұл батырманы шерткен кезде Enter пернесін басқандай іс-әрекет атқарады.
- 5) Value (мәні) – бұл командалық батырманы таңдаған кезде Value қасиеті True болады және Click (шерту) оқиғасы

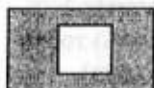
инициализацияланады, False мәні (үнсiз жариялау) батырманың таңдалмағанын бiлдiредi;

6) *Enabled (қатынау)* – егер мәні True болса – онда батырма активтi, егер False болса – керiсiнше.

Басқару батырмасының оқиғасы:

Click – командалық батырма аумағында маус батырмасын басқан кезде орнайды.

4.6.5. *Check Box (жалауша) – басқару элементi*



Check Box (жалауша) – тәуелсiз бекiту батырмасы, қолданушыға анықталған опцияны таңдау/таңдамау мүмкiндiгiн бередi.

Негiзгi қасиеттерi:

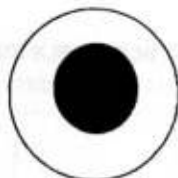
1) *Name (аты)* – (Check 1) аты, префиксi «Chk»;

2) *Caption (жалауша аты)* – жалауша өрiсiнде орналасатын мәтiн;

3) *Value (мәні)* – мән: 0 – таңдалмаған, 1 – таңдалған, 2 – пайдаланбайтын;

4) *Allow Great* – жалауша үшiншi қалып-күйде. Егер қасиет мәні False болса, онда жалауша 0 немесе 1 қалып-күйдiң бiрiнде болады.

4.6.6. *Option Button (ауыстырып-қосқыш) – басқару элементi*



Option Button (ауыстырып-қосқыш) тәуелдi бекiту батырмасы, ол мүмкiн болатын бiрнеше опциялардың бiрiн таңдауға арналған. Сондықтан, тәуелдi бекiту батырмалары бiрнеше элементтен тұратын топқа бiрiктiрiледi.

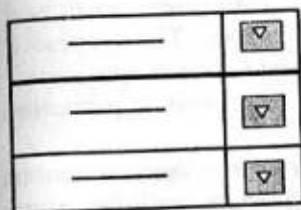
Ауыстырып-қосқыштың негiзгi қасиеттерi:

1) *Name (аты)* – Option 1 аты, префиксi «opt»;

2) *Caption (аты)* – ауыстырып-қосқыш өрiсiнде шығатын мәтiн;

3) *Value (мәні)* – ауыстырып-қосқыш таңдалған/таңдалмағанын бiлдiредi: True – таңдалған, False – таңдалмаған.

4.6.7. Combo Box басқару элементі (тізімді өріс)



Тізімді өріс немесе Combo Box - бұл мәні бойынша аралас тізім, ол басқару элементтерінің екі комбинациясынан тұратын көрсетілім: мәнді тізім және мәтінді енгізу өрісі (мәтіндік өріс). Егер тізімге енгізілетін мәнді алдын ала анықтауға

болмайтын немесе тізімде өте көп элементтер орналасқан жағдайда, тізімді өріс қолданылады. Мұндай жағдайда мәнді тек таңдап қана емес, оны тікелей өріске енгізуге болады. Жана мән енгізілгеннен кейін автоматты түрде тізімде орналасады.

Негізгі оқиғалар:

- 1) Click - тізімдегі элементті таңдау үшін қолданылады;
- 2) Change - мәтінді енгізу өрісінде жазбаны өзгерту үшін.

Негізгі қасиеттері:

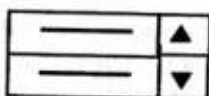
Тізімді өрісте Text Box мәтіндік өріспен List Box тізім қасиеттерінің бәрі бар десе болады (Multi Line қасиетінен басқа). Дегенмен, Style (4.3-кесте) қасиетін ерекшелеуге болады, ол сыртқы көрініспен тізімді өрістің жұмысын анықтайды.

4.3-кесте

Style қасиеті

Константа	Мәні	Сипаттамасы
VbComboDrop Down	0	Үнсіз жариялаудағы мәні. Combo Box ашылатын тізім мен түзетулер енгізуге арналған мәтіндік өрістің көрсетілімін бейнелейді
VbComboSimple	1	ComboBox үнемі мәтіндік өрістің ашық тізіммен көрсетілімі
VbComboDrop DownUst	2	VbComboDropDown мәні бар тізімінен айырмашылығы-қолданушының мәтіндік өріске мәтін енгізе алмайтындығында

4.6.8. List Box басқару элементтері (тізім терезесі)



List Box сайманның көмегімен экрандық пішінде жоба құруға болады. Тізім терезесі бір мәнді емес, көп мәнді сақтауға арналған.

Бір өлшемді массив сияқты, мәндер тізімі дегеніміз - реттелген жиын. Оның элементтерінің индекстері бар.

Индекстің ең кіші мәні - 0, ал ең үлкен мәні - орнатылмайды (бұл - массивтен айырмашылығы). Тізімдегі мәндер String типті қатарлар болады.

Негізгі қасиеттері:

- 1) Name (аты) - List, Lst префиксі;
- 2) Back Color – тізім терезесінің фондық түсін анықтайды;
- 3) Columise – бағана саны беріледі:
0 - тізім терезесі бір вертикальды бағанада айналымды;
1 және одан да көп – тізім элементтері бағананың берілген санында орналасады;
- 4) Fore Color – тізім терезесіндегі мәтін түсін анықтайды;
- 5) List бұл қасиет ашылған тізім элементтерінен тұрады;
- 6) Style тізім элементтерінің алдында жәй тізім терезесінің пішінін немесе жалаушаны анықтайды;
- 7) List Index – тізім терезесінің екінші элементі индексінің мәні; мысалы, I = Lst Орналасу. Lst Index.

Әдістері:

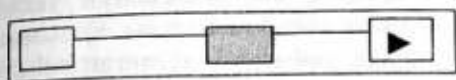
- 1) ADD Item – Тізім терезесіне элементті қосу, мысалы:
Lst3. Горизонттан жоғары (1);
- 2) Clear – тізімді тазалау:
Lst3.Clear – барлық элементтер жоғалады;
- 3) Remove Item – жеке элементті жою:
Объект. Remove Item [индекс];
Мысалы: Lst3.Remove Item [1].

4.6.9. Горизонтальді (HScrollBar) және вертикальды (VScrollBar) айналдыру жолақтарының элементі

Екі объект те ToolBox стандартты жиынында орналасқан.

HScrollBar және VScrollBar элементтері өздігінше орналасып, ешқандай тереземен байланысы болмайды. Олар пернелер

тақтасының көмегінсіз қандайда бір шаманың сандық мәнін орнату үшін қолданылады.



-экрандық пішіндегі орналасқан HScrollBar элементінің түрі.

Бұл элемент арқылы мәнді орнатудың үш тәсілі бар: екі шеткі батырманың бірін шерту арқылы; маус пен жүгірткіні сүйреу арқылы (батырмалар арасындағы объект); жүгірткі мен батырманың арасындағы аймақтың бірін шерту арқылы;

VScrollBar көмегімен мәндер тура осылай орнатылады (вертикальды айналдыру жолағы).

HScrollBar және VScrollBar объектілерінің негізгі қасиеттері:

- 1) Value – айналдыру жолағындағы орнын анықтайтын мән;
- 2) Min – жүгірткінің сол жақ шеткі не жоғарғы шеткі орналасуымен анықталатын ең кіші мәні;
- 3) Max – жүгірткінің оң жақ шеткі не төменгі шеткі орналасуымен анықталатын ең үлкен мәні;
- 4) Small Change – жолақ шеттеріндегі бір батырманы шерткен кездегі Value қасиеті өзгертін мән шамасы;
- 5) Large Change – жүгірткінің аумағындағы ◀ немесе ▶ жылжу батырмасының бірімен шерткендегі Value қасиетінің өзгертін мән шамасы.

Оқиғалар:

1) Scroll – айналдыру – мауспен сызғыш бетінде қозғалту кезінде үзіліссіз генерацияланады:

```
Sub H ScrollBar_Scroll ()  
Text1.Text =Str(HScroll1. Value)  
End Sub;
```

2) Change – өзгертулер – сырғытпақ сызғышының ішінде немесе сыртында шерту кезінде орнайды:

```
Sub H ScrollBar_Change ()  
Text1.Text =Str(HScroll1. Value)  
End Sub.
```

4.6.10. Timer (Таймер) басқару элементі

Таймер көмегімен қосымша процестерін, анықталған уақыт аралығында іске қосуға немесе жұмысты аяқтауға болады. Қосымша фондық режимде орындалған жағдайда, таймерді қолданған тиімді. Жобалау кезінде таймер пішінде бейнеленіп, бағдарламаның орындалу кезінде көзге көрінбейді.

Таймердің жалғыз ғана оқиғасы бар - Timer орнатылған уақыт интервалы аяқталған кезде шақырылады. Таймердің негізгі қасиеті - Interval болып саналады. Ол мәні миллисекундта орнатылатын уақыт интервалын орнату үшін қолданылады. Мысалы, 250 мәні берілген кезде қай қосымша екпінді болғанына тәуелсіз әрбір 250 миллисекундта Timer оқиғасы шақырылады. Таймерді өшіру үшін Interval қасиетіне 0 мәнін немесе Enabled қасиетіне, False мәнін меншіктеу керек. Максимальды рұқсат берілетін интервал 64757 миллисекундтан тұрады. Егер таймер оқиғасын өңдеу Interval-да берілген мәннен ұзақ болса, онда жаңа Timer оқиғасы Visual Basic алдыңғы оқиғаны өңдеп болмай, шақырылмайды.

4.6.11. Shape басқару элементі (фигура)



Shape басқару элементі пішіндегі тік төрт бұрыштарды, квадраттарды, эллипстерді және шеңберлерді құру үшін қолданылады. Керекті бейне Shape басқару құрылғысының көмегімен құрылып, содан кейін Properties (қасиеттері) терезесінде бейнелер сипаттамасы орнатылады.

Shape қасиеті бейне түрін бақылайды; тік төртбұрышты бұрыштары дөңгеленіп келетін төртбұрышты, квадратты бұрыштары дөңгеленіп келетін квадратты, овалды немесе шеңберді таңдауға болады. Күрделі бейнелерді бірнеше фигуралармен кесінділер салу жолымен құруға болады.

Фигура объектісінің басқа маңызды қасиетіне Fill Color (бояу түсі) қасиеттері жатады. Оның атқаратын қызметі: объект түсін орнату мүмкіндігі; Fill Style (бояу типі), толтыру өрнегінің текстурасын орнату мүмкіндігі; және Border Color (контур түсі), жеке фигура контурының түсін орнату мүмкіндігі.

Фигура объектісінің де Visible (көрініс) қасиеті бар, ол қажет болған жағдайда бейнені көрсету немесе жасыру мүмкіндігін береді.

4.6.12. Line басқару элементі (кесінді)



Line (кесінді) басқару элементі пішінде түзу сызық құру үшін қолданылады. Содан соң құрылған түзудің сыртқы көрінісіне әсер ететін қасиеттер тізбегі бекітіледі.

“Кесінді” объектісінің маңызды қасиеттеріне: `BorderWidth` (қалыңдығы), `BorderStyle` (сызық типі), `BorderColor` (түсі) және `Visible` (көрінісі) жатады. `BorderWidth` (қалыңдығы) қасиеті сызықтың қалыңдығын реттеуге мүмкіндік береді. Бұл астын сызу немесе бір объектіні екіншісінен бөлетін сызық құру кезінде аса пайдалы. `BorderStyle` (сызық типі) қасиетті сызықты біртұтас, нүктелі немесе пунктирлі салуға мүмкіндік береді. `Border Color` (түсі) қасиеті – сызыққа Visual Basic-те қолданылатын кез келген стандартты түсті бекітеді. Соңғы `Visible` (көрінісі) қасиеті қажетіне қарай сызықты көрсетуге немесе жасыруға мүмкіндік береді.

4.6.13. Image басқару элементі

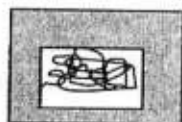
Image басқару элементі, суреттерді бейнелеуге арналған. Бірақ, Picture Box-қа қарағанда, ол *контейнер-элемент* болмайды. Ол сурет салуға және объектілерді топтастыруға рұқсат бермейді. Дегенмен, Picture Box-қа қарағанда Image аз ресурстарды қолданып, суретті қайта салуды жылдамдатады. Сондықтан суреттерді бейнелеу үшін Image ең жақсы вариант болуы мүмкін. Image-нің негізгі тағайындалуы – суреттерді бейнелеу болғандықтан, оның оқиғалары анализденбейді.

Негізгі қасиеттері:

1) Picture - оның көмегімен суретті анықтауға болады; бейнеленетін басқару элементі жобалау кезеңінде не бағдарламаны орындау кезінде болады.

2) Stretch - Stretch қасиеті суреттің қалай бейнеленгенін анықтайды. Егер қасиет мәні True болса, онда сурет өлшемі Image басқару элементінің өлшеміне дейін Ал керісінше болғанда, басқару элементі сурет өлшеміне дейін өзгереді.

4.6.14. Picture Box



Picture Box – суреттермен басқа да графикалық объектілерді бейнелеуге арналған. Бұл басқару элементі де *контейнер-элементі* болып табылатындықтан, оны басқа элементтерді қосу үшін қолдануға болады.

Бейнелеудің негізгі қасиеті.

Name (аты) – Picture 1 аты, “pic” префиксі. Пішіндегі PictureBox жағдайы Align қасиетімен орналастырылады. Ол Picture Box пішінің бір жақ шетінде орналасатынын немесе құрушы орнатқан жағдайды сақтайтынын анықтайды. Егер басқару элементі пішінің бір жақ шетінде бекітілген болса, онда оның өлшемі (ені мен биіктігі) пішін өлшеміне сәйкес орнатылады. AutoSize қасиеті әртүрлі өлшемді суреттерді бейнелеуге арналған басқару элементінің өлшемі автоматты түрде өзгертетінін анықтайды.

PictureBox-тың ең маңызды қасиеті — Picture, ол бейнеленетін графикалық объектіден тұрады. Олар растілік бейне (*.BMP), пиктограмма (*.ICO), метафайл (*.WMF) немесе кеңейтілген метафайл (*.EMF), сонымен қатар GIF және JPEG-файлдары болуы мүмкін.

Қасиеттерді өзгерту үшін қосымшаларды орындағанда, LoadPicture функциясы қолданылады:

```
Picture1.Picture = LoadPicture("C:\WINDOWS\AUTOS.BMP").
```

SavePicture функциясының көмегімен бейнені сақтауға болады:

```
SavePicture Picture1.Picture, "BUILD.BMP"
```

PictureBox оқиғалары әдетте өңделмейді, бірақ қажет болған жағдайда оны орындауға болады.

PictureBox әдісі нүкте, сызық және шеңбер салуға мүмкіндік береді, сонымен қатар мәтінді қорытындылайды (Print әдісі):

```
Picture1.Print "Әлем сәлем бердім!"
```

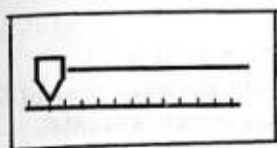
```
Picture1.Line (0, 0)-(100, 500), vbRed Picture1.Circle (300, 300),  
250, vbBlue
```

```
Picture1.PSet (X, Y), vbGreen.
```

PictureBox элементінің әртүрлі пішінді суреттерді бейнелейтін қабілетін икондарды (*.ICO) растілік бейнеге (*.BMP) түрлендіру үшін қолдануға болады. Ол үшін иконды жүктеп, оны BMP

тіркеуімен сақтау керек. Бірақ, растылық бейнені пиктограммаға түрлендіруге болмайды.

4.6.15. *Slider басқару элементі (Сырғытпақ)*



Slider басқару элементі ToolBox стандартты жиынында орналасқан. Slider басқару элементі, пернелік тақтаның көмегімен сандық мәндерді бекітуге арналған.

Негізгі қасиеттері:

- Orientation – сырғытпақтың (Slider) жылжу ориентациясын (горизонтальды немесе вертикальды) анықтайды;
- Tick Style – сырғытпақ (Slider) қозғалысы бойындағы белгілердің (штрихтардың) барын анықтайды;
- Tick Frequency – сырғытпақ (Slider) қозғалысы бойындағы белгілердің (штрихтардың) жоғын анықтайды;
- Min – сырғытпақтың шеткі сол жақ не шеткі жоғарыда орналасқанын анықтайтын ең кіші мән;
- Max – сырғытпақтың шеткі оң жақтағы немесе шеткі төменгі жақтағы орналасқанын анықтайтын ең үлкен мән;
- Small Change – жылжу кезінде сырғытпақпен реттелген аймақ мәні өзгертін шама;
- Large Change – сырғытпақтың жылжу жолағында мауспен шерткенде, сырғытпақпен реттелетін аймақ мәні өзгертін шама;
- Text - бағдарламалық кодпен құрылатын, сырғытпақтың жылжуын сүйемелдейтін жазбаның мәнін анықтайды;
- Text Position – сырғытпақтың жылжуын сүйемелдейтін жазба орнын анықтайды;
- Value – жылжу жолағындағы сырғытпақтың ағымдағы позициясын нұсқайды.

Оқиғалар:

- Change – өзгертулер – сырғытпақтың жылжу жолағында мауспен шерткен кезде орындалады:

```
Sub Slider_Change ()  
Slider1.Text = Str(Slider1/ Value)  
End Sub;
```

- Scroll – айналдыру – маусты жылжу жолағымен қозғалтқан кезде сырғытпақтың орны ауысқанда үздіксіз генерацияланады:

```
Sub HScroll_Change ( )  
Text 1. Text = Str(HScroll1. Value)  
End Sub.
```

Бақылау сұрақтары:

1. Басқару элементінің негізгі қасиеттерін атаңыз.
2. Command Button басқару элементтерін атаңыз.
3. *Image* басқару элементінің негізгі қасиеттерін атаңыз.
4. *Timer (Таймер)* басқару элементінің қасиеттерін атаңыз.

5. VISUAL BASIC ТІЛІНДЕ БАҒДАРЛАМАЛАУ

Тілдің негізгі конструкциялары алфавит көмегімен жазылады. Оларға тұрақтылар, айнымалылар, функциялар мен өрнектер жатады.

5.1. Тіл алфавиті

Тіл алфавитінің құрамы:

- Латын алфавитінің А-дан Z-ке дейінгі бас және кіші әріптері;
- 0-ден 9-ға дейінгі сандар;
- арнайы белгілер:
 - а) тыныс белгілері (. , () ; ; “ және ?);
 - ә) арифметикалық операция белгілері (+ , - , * , / , \);
 - б) басқа белгілер " (= % & < > ^ ! \$).

Мұнымен қатар, коментарийлер мен мәтіндік константаларды жазу үшін орыс алфавитінің әріптерін қолдануға болады;

5.2. Айнымалы

Айнымалы - деректерді сақтауға арналған жадтағы аталымды аймақ.

Айнымалы аты ерекше және оны өзгертуге болмайды, ал мәні - алгоритм орындалу процесінде өзгеруі мүмкін.

Айнымалылар аты - бұл айнымалыны бағдарламаның басқа объектілерінен ажырататын белгілер қатары.

Айнымалылардың аттары анықталған ереже бойынша құрылады:

- 1) Айнымалы атының бірінші белгісі *әріп* болуы керек;
- 2) Қалған белгілері - әріп пен сандар (бас әріп пен кіші әріп ажыратылады);
- 3) “_” белгісін қолдануға болады. Нүктені қолдануға болмайды;
- 4) Белгі саны 255 аспауы керек;
- 5) Айнымалы аты Visual Basic-тің түйінді сөзі болмауы тиіс.

Айнымалы мәні - Visual Basic жүйесінде сақталатын және өңделетін деректер. Деректер әр түрде сақталып өңделеді. Ол - деректердің қай типке жататынына байланысты. Компьютерлік жүйедегі деректердің сақталуы мен көрсетілім тәсілі - *деректер түні* деп аталады.

Visual Basic-6 тілінде жиі қолданылатын деректер типі :

- теріс емес, қысқа бүтін сан (Byte). Сан (бұл типі айнымалының мәні) жадтың 1 байтында орналасады;
 - бүтін сан (Integer). Бұл типті айнымалының мәні жадта 2 байт орын алады;
 - ұзын бүтін сан (Long). Бұл типті айнымалы мәнін жадта 4 байт орын алады;
 - жай дәлділікті ондық сан (Single). Мән жадта 4 байт орын алады;
 - екі есе дәлдікті ондық сан (Double). Мән жадта 8 байт орын алады;
 - қатар (String). Жадтағы алатын орны қатардағы белгілер санына сызықты тәуелді;
 - ерікті мән (Variant). Жалпы, Variant типі жаңа тип болып саналмайды. Бұл “типті” айнымалы кез келген типтен тұруы мүмкін. Бірақ бұл типті мән өте қымбатқа түседі, себебі бұл мәнің жадтағы алатын орны әртүрлі, бірақ ең кемі 16 байт;
 - ұзындықты ондық сан (Currency). Бұл типтің мәні ретінде ондық нүктенің сол жағында 15 белгі, оң жағында 4 белгісі бар айнымалы алынады. Негізінен финанстық ақпарат көрсетілімінде қолданылады.
 - логикалық шама (Boolean). Мұндай типті айнымалының тек екі мәні болады: Ақиқат (True) және Жалған (False). Мән жадта 2 байт орын алады;
 - дата (Date). Мұндай типті айнымалының мәні, мерзім жайлы ақпараттан тұрады және 8 байт орын алады;
 - объект (Object). Мұндай типті айнымалы ретінде объект сілтемесі алынады. Ол 4 байт орын алады.
- Сандық айнымалының өзгеру аумағы:
- *Byte* типі үшін : 0-ден 255-ке дейін;
 - *Integer* типі үшін :-32768-ден 32767 ге дейін;
 - *Long* типі үшін :-2147483648-ден 2147483647-ге дейін;
 - *Single* типі үшін : 1401298E-45-пен 3,402823E³⁸-ке дейін (модуль бойынша);

Microsoft фирмасы айнымалы атының сол жағынан ерекше қосымша (прификс) жазуды ұсынады, ол айнымалы типін нұсқайды. 5.1-кестеде әртүрлі типті айнымалы аттарына арналған қосымшалар келтірілген:

Айнымалы аттарына ұсынылатын префикстер

Айнымалы типі	Жазылу үлгісі
<i>Byte</i>	<i>Byt</i>
<i>Integer</i>	<i>int</i>
<i>Long</i>	<i>Ing</i>
<i>Single</i>	<i>Sng</i>
<i>Double</i>	<i>Dbl</i>
<i>String</i>	<i>Str</i>
<i>Variant</i>	<i>Vnt</i>
<i>Currency</i>	<i>Cur</i>
<i>Boolean</i>	<i>Bin</i>
<i>Date</i>	<i>Dtm</i>
<i>Object</i>	<i>Obj</i>

Айнымалы атынан оның қай тип екенін білу үшін, жиі айнымалы атына қосып *суффикс* ретінде мына белгілердің бірін қолдануға болады: %, 7, !, #, \$, @.

Суффикстер тек алты типті айнымалыларда болады. (5.2-кесте) және суффикс айнымалы алғаш рет кестеде пайда болғанда, тек бір рет қана көрсетіледі.

5.2-кесте

Айнымалы суффикстері

Суффикс	%	&	!	#	\$	@
Типі	Integ	Long	Singl	Doub	Strin	Currency

Visual Basic-те бағдарлама жазған кезде, айнымалыны жариялау қажеттігін, қолданушы өзі шешеді. Айнымалыны айқын жариялау үшін, келесі синтаксистен тұратын айнымалыны анықтайтын оператор қолданылады:

Dim айнымалы аты [*As* айнымалы типі];

мұндағы Dim, As Visual Basic-тің түйінді сөзі, оператор осылардың көмегімен жазылады. Бұл оператордың тағайындалуы – айнымалыны жариялау, яғни оның аты мен типін беру.

Айнымалы аты (суффиксіз) алдында айтылған ереже бойынша құрылады;

Айнымалы типі - жоғарыда айтылған түйінді сөздердің бірі – тип аттары (Integer, Long, және т.б.).

Тік жақшалар [] - конструкцияларды жиектеу үшін қолданылады, конструкциялар болмауы да мүмкін.

Dim сөзінен кейін, үтір арқылы бірнеше айнымалыларды жазуға болады:

Dim X As Single, Құс саны As Integer, Пу As Double

Dim Алма, КР, БжФМ

String типті айнымалының мәні ерікті немесе бекітілген ұзындықты қатар болуы мүмкін:

Dim Айнымалы Аты As String Қатар ұзындығы

Бекітілген ұзындықта String типті айнымалыларын жиі қолданады. Себебі, мұндай айнымалы мәнінің жад көлемі, байтпен өлшенетін бекітілген ұзындыққа тең болады. Ерікті ұзындықты қатардың алатын жад көлемі - оның байт ұзындығына 10 байт қосқанға тең.

Көптеген айнымалыларға бір типті меншіктеу үшін, арнайы оператор – типті жариялау операторы қолданылады. Оның синтаксисі:

Def Типтің Алғашқы әрпі – Соңғы әрпі,

мұнда Тип орнына анықталған деректер типі (5.3-кесте) қолданылады.

5.3-кесте

Типті жариялау операторы

Айнымалы типі	Типті жариялау операторы
Integer	DefInt
Long	DefLng
Single	DefSng
Double	DefDbf
Currency	DefCur
String	DefStr
Byte	DefByte
Boolean	DefBool
Date	DefDate

Object	DefObj
Variant	DefVar

Егер қарапайым жариялау бөлімінде Dim түйінді сөзі қолданылатын болса, онда ол айнымалының локальді (Private) екенін білдіреді.

Локальді (Private) айнымалысы - берілген бір модульдің барлық процедураларына қатысты, бірақ оны жобаның басқа модульдерінен "көруге" болмайды. Ал ауқымды (Public) айнымалы, жобаның әртүрлі модульдерінде қолданылады.

Егер айнымалы тек бір процедурада қолданылатын болса, онда оны жариялау қатары, осы процедура ішінде бірінші болып жазылады. Егерде айнымалы бірнеше процедурада қолданылатын болса, онда оны жариялау қатары - бірінші процедураның бірінші қатарына дейін (General) аты бар бағдарламаның жалпы бөлімінде жазылады.

Айнымалы типін жариялаудан бас тарту үшін Visual Basic ортасындағы айнымалыларды жариялау қажеттігіне кепілдік беретін механизм бар.

Жобаның негізгі панелінің Tools менюінде Options командасы бар. Оны қолданған кезде диалогты терезе ашылады. Require Variable Declaration режиміне жалауша орнатса, бағдарлама айнымалыларды жариялау қажеттілігіне кепілдік береді. Айнымалы жарияланған кезде оның типі де жарияланады.

Айнымалыға мән меншіктеу үшін меншіктеу операторы қолданылады:

[Let] Айнымалы аты = Айнымалы мәні

Visual Basic бірнеше операторларды бір қатарда жазуға мүмкіндік береді. Бұл жағдайда оператор ретінде ":" (кос нүкте) белгісі тұруы керек. Оны бөліктегіш деп атайды, мысалы:

```
Dim Var1 As Integer, Var2 As Long, Var3 As Single, Var4 As Double
Var4 = 9E-300: Var1 = 33000: Var2 = -3000000000: Var3 = Var4
Dim Var1 As Integer, Var2 As Long, Var3 As Single, _ Var4 As Double
Var4 = 9E-300: Var1 = 33000: Var2 = -3000000000: Var3 = Var4
```

5.3 Константа

Константа (тұрақтылар) - мәні өзгермейтін шамалар. Айнымалылар сияқты, константалар да бағдарламалық кодтың басында жарияланады. Константаны жариялау синтаксисі:

`Const Константа аты [As Tun] = Константа мәні`

Константа аты мен *константа мәні* де айнымалы аты мен мәні сияқты беріледі. Типін көрсетпеуге де болады (үнсіз жариялау бойынша Variant типі беріледі), мысалы:

```
Const pi = 3.1415
```

```
Private Sub Form_Load()
```

```
HScrollBarF.Show
```

```
Picture1.Scale (-11, 1.1) - (11 -1.1)
```

```
MsgBox «Параметр мәндерін бекітетін» _
```

```
& «графикалық терезені шертіңіз», 64, «Жұмысты бастау туралы»
```

```
Text1.Text = HScroll1.Value
```

```
Text2.Text = HScroll2.Value
```

```
Text3.Text = HScroll3.Value
```

```
End Sub
```

```
Private Sub HScroll1_Change()
```

```
Text1.Text = HScroll1.Value
```

```
End Sub
```

5.4 Арифметикалық өрнек

Арифметикалық өрнек дегеніміз - аралары арифметикалық операциялардың белгілерімен жалғасқан сандар, тұрақтылар, айнымалылар, функциялар және дөңгелек жақшаға алынған арифметикалық өрнектер тізбегі.

Арифметикалық өрнектің мәндері, жалпы ережелер бойынша есептеледі. Visual Basic тілінде қолданылатын арифметикалық

операциялар (мұндай операциялар сегіз) тізбегі 5.4-кестеде келтірілген.

5.4-кесте

Арифметикалық операциялар тізбегі

Операция	Операция сипаттамасы
$A \wedge B$	A-ны B-ға дәрежелену
-A	A-ның таңбасын ауыстыру
$A * B$	A-ны B-ға көбейту
A / B	A-ны B-ға бөлу
$A \setminus B$	A-ны B-ға бүтін сандық бөлу
$A \text{ Mod } B$	A-ны B-ға модуль бойынша бөлу
A+B	A мен B-ны қосу
A-B	A-дан B-ны алу

5.5. Функциялар

Әртүрлі есептерді бағдарламалау кезінде квадрат түбір астын, санның логарифмін, бұрыштың синусы мен басқаларды есептеу қажеттілігі жиі туады. Осы және басқа да шамаларды есептеу функция деп аталатын бағыныңқы бағдарлама арқылы орындалады. 5.5-кестеде Visual Basic-те жиі қолданылатын математикалық функциялар қарастырылған.

5.5-кесте

Жиі қолданылатын құрамдас функциялар

Функция аты	Функция аргументі	Қайтарылатын мән
Abs	Сан	Санның абсолютті шамасы
Cos	Сан (радиан)	Санның косинусы
Fix	Сан	Бөлшек бөлігі жоқ сан
Int	Сан	Берілген саннан аспайтын ең үлкен сан
Rnd	Аргументі жоқ	N псевдо-кездейсоқ сан ($0 < N < 1$)
Sin	Сан (радиан)	Санның синусі
Sqr	Теріс емес сан	Санның квадрат түбірі

5.6. Басқарушы операторлар

Басқарушы оператор дегеніміз - процедура операторларының орындалу ретін өзгерте алатын операторлар. Visual Basic-те бұл

үшін функциялар мен операторлар қатары қолданылады. Жиі қолданылатын оператор If...Then, оның құрамы қарапайым бір қатарлы немесе блокты болады.

Бір қатарлы оператор синтаксисі:

If шарт Then оператор [Else оператор]

Бұл құрылымды оператордың орындалуы айтарлықтай жеңіл. Егер If-тен кейінгі шарт ақиқат болса, нәтиже True-ге тең болып, одан кейінгі оператор орындалады. Егер нәтиже False-ге мәніне тең болса, онда Else түйінді сөзінен кейінгі оператор орындалады. Егер Else операторы жоқ болса, келесі оператор орындалады:

If A = 7 Then Beep

'операторлар

If X < 9 Then Print "False!" Else Print "True!"

Бірінші жағдайда A=7-ге болса, дыбысты сигнал беріледі. Екінші жағдайда X мәні 9-дан кіші болса экранға "False", үлкен болса, "True" мәтіні шығады.

Блок синтаксисі:

If Шарт Then

1-оператор

[Else If Шарт Then

2- оператор]

Else

[3- оператор]

End If

Жалпы блокты жазбаның да, бір қатарлы сияқты мүмкіндіктері бар, тек End If операторын міндетті түрде қосу керек. Бірақ шартқа байланысты жай команданы емес, операторлар тобын қолдану керек болса, онда блокты синтаксисті қолдану керек. Бұл Else тармағына да қатысты. Мұнымен қатар, Else If блокты құрылым бірнеше шарттарды анализдеуге мүмкіндік береді:


```

If A = 5 Then Print "Күтіңіз" ' Печать
If Name = "Иванов" Then
Print "Сіздің карточкаңызды ұстап қаламыз!"
Else
Print "Ақшаңызды алыңыз!"
End If

```

```

If Хабар = 1 Then
Print "Құрметті мырза"
ElseIf Хабар = 2 Then
Print "Құрметті ханым"
ElseIf Хабар = 3 Then
Print "Құрметті ханымдар мен мырзалар"
Else
Print "Сәлеметсіздер ме?"
End If

```

Бұл жағдайда блок синтаксисін қолдану, бағдарламаның оқылуын жеңілдетеді.

Visual Basic-те тағы бір тармақталу Select Case операторы бар. Ол шарттың мәніне байланысты бірнеше операторлар тобынан, біреуін орындайды.

Select Case нұсқауы келесі синтаксистен тұрады:

```

Select Case мәтін_өрнек
Case тізім_1 өрнек
{{1 Операторлар }}
[Case тізім_2 өрнек
[2 Операторлар]]
.....

[Case Else
[Операторы3] ]
End Select

```

Мысалы:

```
Private Sub Command_Click()
```

```
Айным = Int (Rnd * 9) + 1
```

'1-ден 10-ға дейінгі кездейсоқ сан

```
Select Case Айным ,
```

```
Case 1
```

```
Print "1-ге тең"
```

```
Case 2, 3
```

```
Print "2 немесе 3 тең"
```

```
Case 4 To 6
```

```
Print "4-тен үлкен немесе тең және 6-дан кіші немесе тең"
```

```
Case Is >= 9
```

```
Print "9-дан үлкен немесе тең"
```

```
Case Else
```

```
Print "Алдыңғылардың бірі де емес"
```

```
End Select
```

```
End Sub
```

Тізім-өрнек ретінде бір ғана мән емес, үтірмен ажыратылған бірнеше мәндерді нұсқауға болады. Сонымен қатар, салыстыру аймағын (4 To 6) анықтауға немесе салыстырылымды (is >= 9) салыстырмамен қолдануға болады. Тікелей тексерілетін өрнектің орнына Is түйінді сөз қолданылады. Егер алдыңғы шарттардың бірде-бірі ақиқат болмаса, онда Case Else блогы орындалады.

Егер Select Case шартына сан блогының бірнешеуі сәйкес болса, онда оның біріншісі орындалады:

```
Private Sub Command1_Click()
```

```
Select Case Айнымалы
```

```
Case 0
```

```
Print "0-ге тең"
```

```
Case -10 To 10
```

```
Print "-10 мен +10 аралығында, 0 басқа"
```

```
End Select
```

```
End Sub
```

Case екінші блогында - 10-нан +10-ға дейінгі мәндер өңделеді, бірақ 0 мәні бірінші Case блогымен қабылданады. Сондықтан, екінші блоктағы Case операторлары, егер шарт мәні -10-нан үлкен немесе тең және 0-ден кіші, сонымен бірге 0-ден үлкен және +10-нан кіші немесе тең болғанда орындалады.

Төмендегі циклдер бір немесе бірнеше операторларды бірнеше рет орындауға арналған:

- параметрлі цикл: For To Next
- “Дейін” циклі;
- Do...Loop вариацияларымен
- “Әзір” циклі

For To Next цикл операторының синтаксисі:

For Санауыш = Алғашқы мәні To Соңғы мәні [Step қадам]

- Цикл денесі

Next [Санауыш]

Цикл басында санауыш мәні алғашқы мәнге орнатылады. Әр орындалған сайын санауыш айнымалы 1-ге немесе қадам өлшеміне өседі. Егер санауыш соңғы мәнге жетсе немесе одан үлкен (кері қадамда, кіші) болса, онда келесі операторлар орындалады:

For I = 1 To 10

*Print I * 100*

Next I

For L = 100 To 5 Step -0.5

*X = Y * L*

Next

For i = 1 To 5

For j = 10 To 20

Print i+j

Next j

Next i 'Немесе ..(Next j,i)

Бұл мысалдарда For To Next циклінің әртүрлі конструкциялары көрсетілген. Цикл ішінде есептеулерді жүргізу үшін, жиі сандық айнымалылар қолданылады:

```

'(General)(Declaration)
Dim aArray(1997 To 2050)
Private Sub Commmand1_Click ()
For I = LBound(aArray) To UBound(aArray)
Print aArray (I)
Next I
End Sub

```

Бұл мысалда массив құрамы түгелдей қорытындыланады. Циклден шартсыз шығу үшін Exit For операторы қолданылады:

Do..Loop цикл операторы. Егер өту саны шартқа тәуелді болса, онда Do..Loop циклін қолданады. Шарт позициясына байланысты Do..Loop циклінің үш варианты ажыратады:

Әзір циклі:

(While – шарт ақиқат болғаныша; Until - шарт ақиқат болғанға дейін)

Do [(While | Until) Шарт]

[Операторлар]

[Exit Do]

[Операторлар]

Loop

ДЕЙІН ЦИКЛІ:

Do

[Операторлар]

[Exit Do]

[Операторлар]

Loop [(While | Until) Шарт]

Шартсыз цикл:

Do

[Операторлар]

[Exit Do]

[Операторлар]

Loop

Егер шарт цикл басында тексерілсе, онда шарт орындалмаған жағдайда, ол бірде-бір рет орындалмайды. Егер тексеру цикл соңында болса, цикл шарттың орындалуына тәуелсіз, ең болмаса бір рет орындалады. Шартсыз цикл денесінде циклден шығуды атқаратын Exit Do операторы болуы керек.

Цикл денесі шарт циклден шығуды шақырғанға дейін немесе Exit Do операторы орындалғанға дейін анықталмаған санды орындалады.

```
Do Until EOF(1)
```

```
Input #1, Буфер
```

```
Loop
```

```
Do
```

```
X = X + 1
```

```
Print "Hello"
```

```
Loop While X < 9
```

While/Wend цикл операторы. Visual Basic-те While/Wend циклі Basic-тің басқа диалектерімен, сонымен қатар Do..Loop операторы қолданбаған Visual Basic-тің алғашқы версияларымен үйлесімді болуы үшін қолданылады.

```
While Sharp
```

```
[Операторлар]
```

```
Wend
```

Оның атқаратын амалының принципі - Do While... Loop циклындай. Сондықтан оның орнына Do..Loop циклін қолданған ыңғайлы, себебі While.. Wend циклінде, циклден мерзімінен бұрын шығу Exit типті операторы жоқ.

Бақылау сұрақтары:

1. Visual Basic-те қандай айнымалылар қолданылады?
2. Айнымалылар қандай түрде жарияланады?
3. Қолданылатын арифметикалық операциялар түрлерін атаңыз.
4. Цикл денесі дегеніміз не?

6. VISUAL BASIC-те ЖОБА ҚҰРУ

Visual Basic қосымшасын құру кезінде әртүрлі пішінді файл коллекциясы болып келетін жоба құрылады. Ол модульдер мен пішіндер файлы немесе осы қосымшаға арнайы құрылған ActiveX басқару элементтінің файлы болуы мүмкін. Сонымен қатар, жобада бұрын басқа жобада құрылған компоненттер қолдануы мүмкін. Осы аталған барлық объектілерді жұмысқа қосқаннан кейін және оларға бағдарламалық кодты қосқанда, жоба соңғы нәтижеде -қосымшаның жалғыз орындалатын модуліне компиляцияланады.

6.1. Жоба құрылымы

Жоба құрылымы, қосымшаны құрайтын әртүрлі файлдарды басқару жұмысын атқарады. Жоба төмендегі файлдардан тұрады:

- жоба файлы (.vbp);
- жобаның жұмыс аймағының файлы (.vbw);
- жобаның әр пішініне арналған файл (.frm);
- элементтерді басқару пішіні қасиеттерінің мәнінен тұратын әр пішінге арналған деректердің екілік файлы(.frx). Мұндай файлдарды түзетуге мүмкіндік берілмейді және олар .frm тіркеуі бар екілік қасиеттерден тұратын Picture немесе Icon әр файл үшін автоматты түрде генерацияланады.

Жобаның типі мен күрделілігіне қарай, оның құрамында төменде аталған файлдар болуы мүмкін:

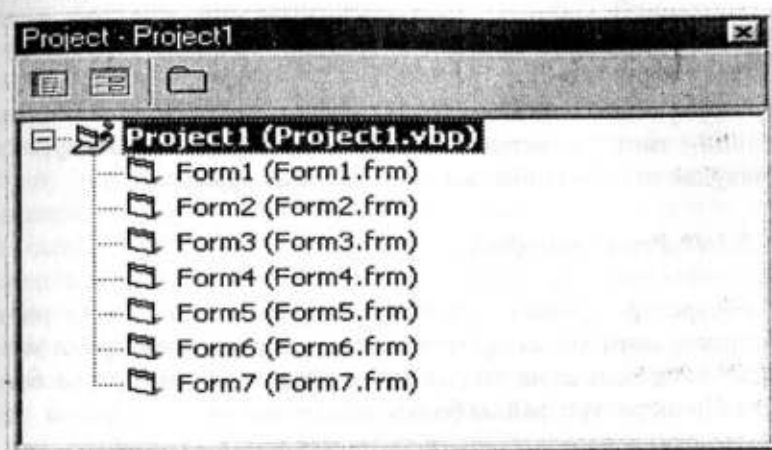
- әр класс модуліне арналған файл (.cls);
- әр стандартты модульге арналған файл (.bas);
- Active X басқару элементтерінен тұратын файлдар (.ocx);
- ресурстар файлы (.res).

Сонымен қатар, Visual Basic-тің әртүрлі процестері мен функцияларын құру және орындау кезеңдерінде төменде аталған файлдарды құрады:

- объект қасиеттерінің парак файлы (.pag және .pgx);
- топтық жоба файлы (.vbg);
- қолдануының басқару элементтерінің файлы (.ctl және .ctx).

Барлық файлдар пішімделгеннен кейін, жоба қосымшаны орындалатын файл (.exe) түріне келтіріледі: File менюінен Make projectexe командасын таңдау.

Жобадағы түзетуге болатын файлдарды құру, қосу немесе жою кезінде, Visual Basic бұл өзгерістерді, жобадағы ағымдағы файлдар тізімінен тұратынын (6.1-сурет) Project Explorer терезесінде бейнелейді.



6.1-сурет. Project Explorer терезесі

6.1.1. Жоба файлы

Visual Basic жобаны сақтау кездерінде жоба файлын (.vbp) модификациялайды. Жоба файлы дегеніміз - жобамен қауымдастырылған Project Explorer терезесінде көрсетілген барлық файлдар мен объектілердің тізімі. Сонымен қатар, жоба файлы құру ортасының опциясына қатысты ақпарат пен жоба файлы ActiveX басқару элементіне арналған сілтемелер мен ендірілу объектілерінен тұрады. Жобаның барлық файлдары мен объектілерін басқа жобаларда да қолдануға болады.

Жобаның құрылған файлын ашу үшін Project саймандар көмегімен Open батырмасын шертіп немесе менюден File/Open Project командасын таңдау керек.

6.1.2. Пішін модулі

Пішін модулі (файл тіркеуінің аты .frm) – пішін мен оларды басқару элементтерінің мәтіндік сипаттамасы мен олардың

қасиеттерін орнатудан тұруы мүмкін. Сонымен қатар, олар константаның, айнымалылардың және сыртқы процедуралардың оқиғаны өңдеу процедуралары мен негізгі процедуралардың жергілікті жариялануынан (пішін деңгейінен) тұруы мүмкін.

6.1.3. Стандартты модульдер

Стандартты модульдер (файл тіркеуінің аты .bas) модуль деңгейінің типі, константасы, айнымалы мен процедураларын жариялаудан немесе public жарияламасынан тұрады.

6.1.4. Ресурстар файлы

Ресурстар файлы (файл тіркеуінің аты .res) растрлік суреттерден, мәтіндік қатар мен басқа да деректерден тұруы мүмкін. Оларды бағдарламалық кодты қайта түзетпей-ақ өзгертуге болады. Жобада бір-ақ ресурс файлы болуы тиіс.

6.1.5. ActiveX құжаттары

ActiveX құжаттары (.dob) пішін тәріздес, бірақ олармен Internet терезесінде, атап айтқанда Internet Explorer терезесінде жұмыс істеуге болады. Visual Basic Professional және Enterprise редакциялары ActiveX құжаттарын құруға мүмкіндік береді.

6.1.6. Қолданушы басқару элементтерінің модулі мен парақ қасиеттері

Қолданушы басқару элементтерінің модульдері (.ctl) және парақ қасиеттері де (.pag) пішін тәріздес, бірақ олар ActiveX басқару элементтерін құруға және құру кезеңінде қасиеттердің мәндерін орнатудағы олардың қауымдастырылған парақтарын құруға қолданылады. ActiveX басқару элементтерін Visual Basic-гі Professional және Enterprise редакцияларында құруға болады.

6.1.7. Компоненттер

Модульдер мен файлдарға қосымша, жобаға компоненттердің басқа типтерінің қатары қосылуы мүмкін:

1) **ActiveX басқару элементтері.** ActiveX басқару элементтері (файл тіркеуінің аты .ocx) - қосымша басқару элементтері, оларды объект палитраларына қосып, пішіндерде қолдануға болады. ActiveX басқару элементтері мен басқа құрушылардың басқару элементтерімен қатар, VB Professional және Enterprise редакциясында ActiveX қолданушы басқару элементтерін де құру мүмкіндігін береді.

2) **Ендірілетін объектілер** - Microsoft Excel-ден ендірілетін Worksheet сияқты объектілер - жинақталған шешімдерді қалыптастыру үшін құрылыс блоктары ретінде қолдануға болатын компоненттерден тұрады. Жинақталған шешім дегеніміз - әртүрлі пішімдегі деректерден тұратын қосымша. Мысалы, әртүрлі қосымшада құрылған электрондық кестелер, растрлік суреттер және мәтіндер.

3) **Сілтемелер.** Қосымшада ActiveX сыртқы компоненттеріне сілтемелер болуы мүмкін. Сілтемелер References (Project/References меню командасы) диалогты терезесі арқылы орнатылады.

4) **ActiveX дизайнерлері.** ActiveX дизайнерлері - IDE Visual Basic-тегі терезелер. Бұл класстарды құруға арналған инструментальды құрылғылар. Бұл класстардан объектілер құруға болады. Visual Basic-те құрамдас дизайнерлер бар. Олар (Professional және Enterprise редакциясында) басқару элементтерін, ActiveX құжаттарын және пішіндерін құру үшін керек. Сонымен қатар, басқа түптектерден қосымша дизайнерлер қосылуы мүмкін.

ActiveX дизайнерлері - өте иілгіш саймандар. Олардың кейбіреулері, мысалға User Connection дизайнері, даналары бағдарламаланатын, бірақ орындалу барысында көрінбейтін класстарды құрады. Басқаларында, Microsoft Office-де қолданылатын Microsoft Forms-та Visual Basic пішіндеріне ұқсас көзге көрнекті объектілер құрылады.

6.2. Жобаларды құру, ашу және сақтау

Жобаларды құру, ашу және сақтау жұмыстары File менюіндегі төрт команда арқылы орындалады:

New Project – файлдағы барлық өзгертулерді сақтауға шақырып, ағымдағы жобаны жабады да, қолданушыға New Project диалогы терезесінен жоба типін таңдауға мүмкіндік береді. Одан кейін Visual Basic жалғыз файлы бар жаңа жобаны құрады.

Open Project — барлық өзгертулерді сақтауға шақырып, ағымдағы жобаны жабады. Одан кейін Visual Basic құрылған жоба мен оған қоса жобаның (.vbp) файлындағы пішіндерді, модельдерді және ActiveX басқару элементтерін де ашады.

Save Project - ағымдағы жоба файлы мен оның барлық пішіндерін, стандартты модульдерін және класстар модулін модификациялайды.

Save Project As – жоба файлы арнайы атпен сақтай отырып, ағымдағы жоба файлы модификациялайды. Visual Basic бұл жерде де пішіндер мен модульдердің барлық өзгертулерін сақтауға ұсыныс жасайды.

Файлды бірнеше жобаларда қолдануға болады. Пішін сияқты жалғыз файл бір жобаның ғана бөлігі бола алады. Бір жобадағы пішінде немесе модульде жасалған өзгертулер, қолданылатын модульдері бар барлық жоба арасында тираждалатынын есте сақтау керек.

6.3. Бірнеше жобалармен жұмыс істеу

Құрылатын қосымша күрделілігінің өсуіне байланысты бағдарламалау ортасының бір сеансында бірнеше жобаны ашуға тура келеді. Мысалы, бір жобаны қосымшаның орындалатын файлын қалыптастыру үшін қолданып, екіншісін - негізгі қосымшаға аудару кезінде бағдарламалық кодті тестіден өткізу кезінде бастапқы мәлімет ретінде қарастырады.

Ағымдағы түзету сеансына жаңа немесе құрылған жобаны қосу үшін, оны жоба тобына енгізу керек. Жоба тобын сақтағаннан кейін, онымен келесі сеанстарда жұмыс істеуге болады. Жобалар тобын не жоба тобындағы жеке жобаны ашуға немесе жоба тобын не жекеленген жобаларды басқа жоба тобына қосуға болады.

Жоба тобындағы орындалатын бір жоба - іске қосу жобасы деп саналады. Егер жоба тобы ашық тұрғанда Run менюінен Start командасын тандап, саймандар панелінен Start батырмасын басып немесе F5 функционалды пернесін бассақ, онда Visual Basic жобаны іске қосады.

6.3.1. Жоба тобына жаңа жобаны қосу

Жоба тобына жаңа жобаны қосу үшін мына әрекеттерді атқару керек:

- саймандар панелінен Add Project батырмасын шертіп немесе File менюінен Add Project командасын таңдау керек;
- жобаның қажетті типін не қалқыма менюден Add Project батырмасын, не Add Project диалогті терезесіндегі New қосымшасынан таңдауға болады.

6.3.2. Құрылған жобаны жоба тобына қосу

Құрылған жобаны жоба тобына қосу үшін:

- File менюінен Add Project таңдаймыз. Visual Basic Add Project диалогты терезесін ашады;
- Existing (стандартты файл терезесі) қосымшасын ашамыз;
- Жоба файлын таңдап, содан соң Open-ді басамыз.

Visual Basic таңдалған жобаны жоба тобына қосады. Project Explorer терезесінде иерархияның жоғарғы деңгейінде жоба тобының көрсетілімі, оның төменгі деңгейінде - топталған пішіндері, модульдері, басқару элементтері және қасиеттер парағы немесе құжаттары иерархиялық түрде шығарылады.

6.3.3. Жоба тобынан жобаны жою

Жоба тобынан жобаны жою үшін төмендегі әрекеттерді орындау керек:

- Project Explorer терезесінен жоба атын таңдау керек;
- File менюінен Remove Project командасын таңдау керек.

Visual Basic таңдалған жобаны жоба тобынан жояды.

6.4. Жоба тобын іске қосу

Жоба тобында бірнеше жоба болатындықтан, үнсіз жариялауда Visual Basic жоба тобына қосылған бірінші орындалатын жобаны (.exe) іске қосады. Бірақ, басқа да іске қосу компонентін спецификациялауға болады:

- Project Explorer терезесінен жобаны таңдап;
- Маустың оң жақ пернесін шертіп, контексті менюден Set as Start Up таңдаймыз.

Visual Basic жоба атын Project Explorer терезесінде қалың қаріппен ерекшелейді.

6.5. Жобаға файлды қосу, жою және сақтау

6.5.1. Жобаға құрылған немесе жаңа файлды қосу

Жобаға файлды қосу үшін:

- Project Add файл_типi командасын таңдау керек (мұндағы файл_типi - Project менюіндегі файлдың бір типі). Add файл_типi диалогті терезесі ашылады (мысалы, Add Form).

- құрылған (Existing қосымшасы) немесе жаңа файлды (New қосымшасы) таңдап, Open-ді басамыз.

Жобаға файл қосылған кезде, бұл файлға арналған сілтеме қосылады; файл көшірмесі құрылмайды. Сондықтан, егер файлда өзгертулер енгізіліп, олар сақталса, онда ол құрамында осы файл бар кез келген жобаға әсер етеді. Басқа жобаларға әсерін тигізбейтіндей, файлдарға өзгертулер енгізу керек болса, онда Project Explorer терезесінде файлды таңдап алып, File менюінен Save файл_аты As командасын таңдап, файлды жаңа атпен сақтау керек.

6.5.2. Жобадан файлды жою

Жобадан файлды жою үшін:

- Project Explorer терезесінен файлды таңдап;

- Project менюінен Remove файл_аты командасын таңдау керек.

Файл дискіден емес, тек жобадан ғана жойылады. Егер файл жобадан жойылса, Visual Basic жобаны кезекті сақтау кезінде, жоба файлында бұл ақпаратты модификациялайды. Бірақ, файл Visual Basic ортасыннан тыс жойылса, жоба файлы модификацияланбайды. Демек жобаны келесі ашқан кезде Visual Basic экранға оған сәйкес, қате туралы хабар шығарады.

6.5.3. Жобаны сақтамай жеке файлдың өзін сақтау

Файл өзін, жобасыз сақтау үшін:

- Project Explorer терезесінен файлды таңдаймыз;

- File менюінен Save файл_аты As командасын таңдаймыз.

6.6. Мәтін енгізу

Басқа файлда бар мәтінді модульдердің біріне қоюға болады. Бұл - жеке мәтіндік файлда сақталатын код фрагменттері немесе константа тізімін қосу кезінде пайдалы.

Бағдарламалық кодқа мәтіндік файлды қою үшін:

- Project Explorer терезесінен кодты қоюды қажет ететін пішін немесе модульді таңдау керек;
- View Code батырмасын басып, мензерді код редакторының мәтін қойылатын жеріне орнату керек;
- Edit менюінен Inset File таңдау керек;
- Файлдың терезеде қажетті мәтіндік файл атын таңдап, Open-ді басу керек.

6.7. ActiveX басқару элементтерін қосу және жою

Төменде көрсетілген басқару элементтерінің жиыны, әр жобаға бапталған болуы мүмкін. Басқару элементінің кез келгені, оны жобада қолданғаннан бұрын, объектілер панелінде орналасады.

6.7.1. Объект панеліне басқару элементтерін қосу

ActiveX басқару элементін объект панеліне қосу үшін:

- Project менюінен Components командасын таңдау керек. Components диалогті терезесі ашылады;
- .ocx басқару элементі атының жанына жалауша орнатып, ОК пернесін таңдау керек. Диалог терезесі жабылып, ал таңдалған объект объект панелінде пайда болады.

Басқару элементтерін таңдаған кезде терезені жаппас үшін, ОК-дің орнына Apply (Windows 95 желілік версиясы үшін - қолдану) батырмасын басу керек. Басқару элементі объект палитрасына орналасқаннан кейін, оны құрамдас басқару элементі ретінде пішімге қосуға болады.

6.8. Басқа қосымшаның объектілерін қолдану

Басқа қосымша объектілері, мысалы Microsoft Excel объект модульдерінің кітапханасының құрамына кіретін немесе объект палитралары арқылы басқару элементтері немесе мәтін кодында

бағдарламалайтын объектілер ретінде қолданылады. Объектілерді объект палитрасына қосу - Components терезесінде басқару элементтерін қосқанға ұқсас.

Бағдарламалық кодта басқа қосымша объектілеріне қатынауды туғызу үшін, бұл қосымшаның объект модульдерінің кітапханасына сілтеме орнату керек.

Ол үшін:

- Project менюінен References-ті таңдау керек. Referens диалогті терезесі ашылады;

- жобаға қосатын әр сілтеменің жалаушасын орнату керек.

References диалогті терезесінде көрсетілмеген қосымшаларға сілтеме орнату үшін, Browse батырмасын басып, файлдық терезеде қосымшаны таңдау керек.

- ОК-ді басу керек. Таңдалған сілтемелер жобада тіркеледі.

Егер қандай да бір адресі берілген кітапхана объект жобада қолданылмаса, онда сілтеме жолаушысын алып тастау керек. Себебі, ол Visual Basic трассировка жасайтын объект сілтемесінің санын азайту үшін қажет. Бұл - жобаны компиляциялау уақытын азайтады.

Объектілі кітапханаға сілтемелерді орнатқаннан кейін, оның объектілерін, әдістерін және қасиеттерін Object Browser терезесінен таңдауға және қарауға болады. Object Browser-де көрсетілген кез келген объекті қосымшаның бағдарламалық кодында қолдануға болады.

6.9. Ресурс файлдарын қолдану

Ресурс файлдары - қосымшаға қажетті барлық мәтіндік және графикалық материалдарды бір жерде жинауға мүмкіндік береді. Ол константа жарнамасын, пиктограммаларды, экрандық мәтінді, т.б. деректерді қоса алады. Ресурс файлдары негізінен қосымша версияларын тарату және сүйемелдеу мақсатында қолданылады.

Ресурс файлы мәтіндік редактор мен ресурс компиляторлары (мысалға, Microsoft Visual C++-нен берілетін) көмегімен құрылады. Компиляцияланған ресурс файлының .res тіркеуінен, растілік кейіптер, басқа да элементтерден тұратын қатар жиыны. Оның әрқайсысының ресурс деректерінің типіне сәйкес Long немесе String типті ерекше идентификаторы бар. Visual Basic ресурс файлынан деректерді іріктеуді арнайы қолданылатын функциялар атқарады (6.1-кесте):

Деректерді іріктеу функциялары

<i>Функциялар</i>	<i>Сипаттамасы</i>
Load Res String	Мәтіндік қатарды қайтарады
Load Res Picture	Раст, пиктограмма немесе меңзер болатын Picture объектіні қайтарады
Load Res Data	Байттық массивті қайтарады. Мысалға, .wav файлдарында қолданылады

Ресурс файлын жобаға қосу үшін:

- Project менюінен Add File таңдау керек. Add File диалогті терезесі ашылады.

- Құрылған ресурс файлын (.res) таңдап, Open-ді басу керек.

Жобада тек бір ғана ресурс файлы болуы мүмкін, егер .res тіркеуі бар екінші файл қосылса, онда қате болады.

6.10. Орындалатын файлды құру және іске қосу

Қосымшаның орындалатын файлын (.exe) құру процедурасы төмендегідей:

- File менюінен Make жоба_аты .exe таңдау керек. Мұндағы жоба_аты жобаға арналған қосымша ат.

- Файл атын енгізу немесе файлды жаңа версиясымен қайта жазу керек болады. Каталогтарды қарап шығып, құрылған файл атын таңдау керек;

- Option... батырмасын шертіп, Project Properties диалогті терезесінде орындалатын файл версиясына қатысты деректер қатарын спецификациялауға болады;

- Егер жоба версиясынан нөмірін өзгерту қажет болса, Major, Minor және Revision – сәйкес мәнін енгізу керек. Егер Auto Increment-ті таңдасақ, Revision мәні - берілген жобада Make.exe командасын таңдаған сайын, автоматты түрде өсімшені алады;

- Сонымен қатар, ақпаратты енгізе отырып, қосымшаға жаңа атты спецификациялауға болады. (Application бөлімінде) және тізімнен тақырыпты таңдап, мәтін өрісіне, версияға спецификалық деректер қатарын енгізуге болады. (Version Information - бөлімі-

коментарийлері, компания аттары, дайындаушының сауда таңбасы, авторлық құқық ақпараты, т.б.);

- Project Properties диалогті терезесін жабу үшін, ОК-ді басамыз;

- Компиляцияны іске қосып, орындалатын файлды құру үшін, Make.exe диалогті терезеде ОК-ді басамыз. Қосымша құрылғаннан кейін Windows-тің басқа қосымшасындай оны іске қосуға болады: оның пиктограммасын екі рет шерту керек.

6.11. Жоба параметрлерін орнату

Visual Basic әр жобаны, оның қасиеттер қатарын орнату арқылы баптау мүмкіндігін береді. Төменде Project Properties диалогті терезесінің General қосымшасы (6.2-кесте) арқылы орнатуға болатын кейбір опциялары келтірілген:

6.2-кесте

Опция сипаттамасы

Опция	Сипаттамасы
Startup Object	Орындау кезінде Visual Basic-тің экранға шығаратын бірінші пішіні немесе Sub Main Q
Project Name	Жоба аты. Бағдарламалық кодта жобаны идентификациялайды. Атында нүкте (.), пробел болмауы керек және тек әріптен басталуы тиіс public класының атында, жоба аты мен класс атының қосындысы 37 белгіден аспауы керек
Help File	Жобамен қалыптастырылған көмекші файл аты
Project Help Context ID	Көмек тақырыбының идентификаторы қолданушы "?" белгісін басқан кезде іске қосылуы тиіс
Project Description	Жобаның сипаттама аты. Object Browser және References диалогті терезелері экранға шығарылады

Бақылау сұрақтары:

1. Жоба құрылымын атаңыз.
2. Жоба құру тәсілдерін атаңыз.
3. Жобаны сақтау түрлерін атаңыз.
4. Бірнеше жобамен жұмыс істеу тәсілін атаңыз.
5. Жобаға файл қосу тәсілдерін атаңыз.

7. ҚОСЫМША ИНТЕРФЕЙСІН ҚҰРУ

Қосымшаның негізгі бөлігінің бірі - қолданушы интерфейсi, себебі қолданушы үнемі сол ортада жұмыс істейді. Қолданушы үшін интерфейс қосымшаны бейнелейді, оны “сахна” сыртында орындалып жатқан бағдарламалық код ынталандырмайды.

Бағдарламалық кодты жазып, оны оптимизациялауға құрушының қаншалықты күш жігерін және қанша уақытын бөлгеніне тәуелсіз болып, соңғы қолданушыға - қосымша сапасы оның сыртқы интерфейсiмен анықталады.

Қосымшаны жобалау кезінде интерфейске қатысты бірқатар шешім қабылдануы керек. Интерфейсте қай стил қолданылады – бір құжат (BDI) па, көп құжат па (MDI)? Қанша әртүрлі пішін қажет болады? Қосымша менюін қандай командалар қосады және саймандар панелі қолданыла ма? Қолданушымен қарым-қатынас құру үшін диалогті терезелер қолданыла ма?

Қолданушы интерфейсін құру кезінде қосымшаның тағайындалуын ескеру керек. Үнемі қолданылатын және экранда бейнеленетін қосымша дизайны басқа жиі немесе қысқа уақытта қолданатын бағдарламалардан өзгеше болуы керек. Тағайындалуы экранға ақпаратты қорытындылау болып келетін қосымшаға қойылатын негізгі талап - енгізуге, ақпаратты жинақтауға немесе деректер базасымен интенсивті айырбас жасауға арналған қосымшалардан өзгеше болуы керек.

Біріккен қолданушылар тобы жобаны құруда үлкен ықпалын тигізеді. Себебі, бастапқы қолданушыға бағытталған жоба қосымша құрастырудың жай түрін талап етеді. Ал тәжірибелі қолданушыға арналған дизайнға күрделі элементтерді қосуға болады.

Дәстүрлі есептерді шығаруға арналған - бухгалтерлік процедуралар, тауарларды жүктеу және тарату сұрақтары, қоймалық есептер, т.б. қосымша интерфейсінде дәстүрлігі үлкен рөл атқарады.

7.1. Қолданушы интерфейсін құру принциптері

Қолданушы интерфейсiтерiнiң тиiмдi, жеңiл басқарылатын және интуитивтi-түсiнiктi түрлерiн құру ережелерiн қарастырайық.

Дизайнның базалық принциптері (композициясы, түсі) компьютер мониторияндағы бейнелерге де қолданылады. Тиімді интерфейсi құру үшін суретшi болу қажет емес; негiзi – базалық

принциптерді сақтау, сонда интерфейссті қолдану жеңіл болады. Оны қолдайтын кодтың қандай болатыны, интерфейсстің сырқты пішініне және оған салынған концепцияға тура тәуелді болады.

Visual Basic-те құрылатын кез келген қосымшаның интерфейсстің негізгі элементі - пішін болып табылады. Оған қосымшаның функционалдығына қатынауды қамтамасыз етуші меню мен басқару элементтері жатады. VB-тің интерфейс құруды өте жылдам орындауға мүмкіндік беретініне қарамастан, алдымен қағазға қандай басқару элементтері қажет болатынын, оның құрылымы мен байланыстарын ойластырып, сызып алу керек.

Интерфейс элементтерін қосымшаны қолданудағы оның көзге көріктілігі мен қолдануда максималды ыңғайлылықты құру - композиция деп аталады. Композицияда төменде аталған факторларды ескеру керек:

- қарапайымдылық;
- элементтерді белгілеу;
- біртектілік ;
- танымалдылық;
- қабылдау жеңілділігі.

Қарапайымдылық - композицияның ең маңызды принципі. Күнделікті өмірдегі объектілердің аналогі мен интерфейсін модельдеуді көздеу - интерфейссті жобалаудағы ең кең таралған қате. Айталық, студенттің жеке іс-қағазын толтыруға арналған қосымшасын құруымыз керек. Әрине, біздің алғашқы ойымыз, қағаз пішінді іс-қағазының дубликатындай интерфейссті жобалау болар еді. Бұл қолма-қол бірнеше проблеманы туғызады: қағаз пішіні мен өлшемі экран өлшемімен сәйкес келмейді; қағаз пішінін дублеу интерфейссті тек мәтіндік өрістер және жалаушылармен шектейді; қолданушыға ешқандай жұмыс артықшылығын бермейді. Сондықтан, өз интерфейсін құру тиімді. Терілетін мәтін көлемін қысқарту үшін мүмкін болатын жауап вариантының жиынынан тұратын тізімді қолдануға болады. Бұл - деректерді енгізу кезінде жіберілетін қателерді азайтуға мүмкіндік береді.

Кейде үнсіз жариялау варианттарын ұсынған қолайлы. Егер туған жері деген мәліметке 9/10 “ҚР” деп енгізілетін болса, онда үнсіз жариялауда “ҚР” вариантын ұсыну керек. Сирек кездесетін және күрделі операцияларды орындау кезінде, сәйкес келетін әзір жауаптарды қарастырып, көмектесуге болады.

Элементтерді белгілеу. Көпшілік интерфейстерде барлық элементтердің маңыздылығы бірдей болмайды. Жиі қолданылатын элементтер ең тиімді позицияда және көзге көрнекті орналасуы керек. Оңша маңыздылығы жоқ элементтерді, көзге көрнекті қылмай-ақ орналастыруға болады.

Көптеген елдерде сөзді оқу - солдан оңға қарай және жоғарыдан төмен қабылданатынын ескеру керек. Бұл - қағаздағы мәтінге ғана емес, экран көрінісіне де қатысты. Қолданушы экранды көзбен шолу кезінде ең алдымен сол жақ жоғарғы бұрышқа назар аударады, сондықтан, ең маңызды элементті сол жерге орналастыру керек. Мысалы, пішін Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық университетінің студентімен құрылса, онда бұл өріс көрнекті болуы үшін, ол бірінші болып сол жақ жоғарғы бұрышта орналасуы керек. Ок және Next батырмалары, пішінмен жұмыс істеуді аяқтаған кезде қолданылатындықтан, әдетте экранның төменгі оң жағында орналасады. Интерфейс элементтерін логикалық топтау үшін көп жағдайда Frame басқару элементімен, жақтауларды құруды қолдануға болады.

Біртектілік. Интерфейсті жобалау кезінде біртектілік пен келісімділікті ескеру керек. Бүкіл қосымшадағы интерфейс дизайнының біртектілігі - оны қолдану ыңғайлылығын айтарлықтай жоғарылатады. Біртектіліктің жоқтығы, кімді болса да жаңылыстырады. Интерфейс жүйесіз және былықты болып көрінеді, сонымен қатар қолданушы қосымшаның сенімділігінде күмәнденуі де мүмкін.

Сонымен қатар, әртүрлі қосымша интерфейсінде біртектілікті сақтау ұсынылады, ондағы құрылған клиент қосымшамалары “әріп және рухқа” жауапты болуы керек. Қосымшаны құрғанға дейін оның қолданылатын басқару элементінің типі, өлшемдері, топтау критерийі және қолданылатын қорының стандарттарын бекіту керек. VB-қолданылатын басқару элементінің көптігі олардың бәрін қолдануға еліктіреді. Бірақ, бұл еліктіруге берілмей, олардың арасынан нақты қосымшаға оңтайлы кейбір жиынын таңдап алу керек.

Мұнымен қатар, элементтерді атқаратын қызметі бойынша қолдану керек. Мысалы, тақырыптар мен атауларды мәтіндік өріс бойынша шығаруға болғанмен, оны орындау үшін Label (белгі) элементін қолдану керек.

Танымалдылық. Танымалдылық қолданушы интерфейсі компонент-терінің атқаратын қызметін бейнелейтін арнайы элементтермен анықталады. Мысалы, командалық батырмаларды үш өлшемді бейнелеуде берілетін түрі арқылы, оларды басу керектігі анықталады. Егер жазық батырма құрылған болса, онда қолданушы бұл батырма екенін түсінбеуі де мүмкін. Жазық батырмалар тек ойындарда немесе арнайы мультимедиялық қосымшаларда ғана қолданылады.

Мәтіндік өрісте де кейбір кезде танымалдылық болуы мүмкін. Ақ фонды жақтауланған өрісті көріп, қолданушы оған мәтін енгізуге болады деп есептейді. Ал егер жақтаусыз мәтіндік өріс құрылған болса ($\text{Border Style}=0$), ол белгіге ұқсас болып, оның не үшін қажет екенін анықтау қиындайды.

Қабылдау жеңілділігі. Қолданушы интерфейсі элементін қабылдау жеңіл болу үшін бір-бірінен жеткілікті аймаққа бөліп, бірінің үстіне бірі орналаспайтындай болуы керек. Егер пішінде өте көп элемент болса, онда олардың қажеттісін табу қиындайды.

Элементтер арасындағы интервалдың сәйкестігі және оларды горизонталь мен вертикаль бойы туралау - дизайнды функционалды түрде сипаттайды. VB-де басқару элементтерін, оның өлшемі мен аралығын туралайтын құрылғылары бар командалар *Format* менюінде орналасқан. Түс интерфейсті көркемдейді, сонымен қатар күшті әсер туғызуы да мүмкін. Қолданылатын салаға байланысты кейбір түстер ерекше мәнді болуы да ықтимал.

Кейбір адамдар дальтонизмнің кейбір пішінімен қайғыланады. Мысалы, негізгі түстердің әртүрлі үйлестірілділігін қабылдай алмайды, атап айтсақ, қызыл мен жасылды. Олар жасыл фондағы қызыл түсті оқи алмайды.

Қолданылатын түстер санын шектеп, бүкіл қосымшада анықталған түс сұлбесін ғана қолданған жөн.

Бейнелер мен белгілер. Қосымшаны суреттер мен белгілерде көркемдейді, бірақ негізгі шама – сезімі мен орындылығы. Бейнелер ақпаратты ешқандай мәтінсіз бере алады, бірақ оны кісілер әртүрлі қабылдайды.

Саймандар панеліне орналасқан белгілер, жасырын функционалдықтың визуалды көсетілімін бере алады. Мұндай белгілерді құру кезінде, басқа қандай стандарттар қабылданғанын

түсіну үшін, басқа қосымшалармен алдын ала танысу қажет. Мысалы, New File - командасы үшін шеті бүгілген парақ белгісі, Print үшін - принтердің суреті салынған бейне, ал Save үшін -дискет бейнесі қолданылады. Жеке белгілерді құрғанда, оларды қарапайым түрде құруға ыңғайлау керек.

Көптеген түспен салынған күрделі суреттер нашар масштабталады.

Қаріп те интерфейстің маңызды бөлігі. Кейбір қаріптер экранның әртүрлі кеңеюімен монитордың әртүрлі типтерінде оңай оқылады – мысалы, Arial және Times New Roman қаріптері. Декоративті қаріптер қағаз бетінде әдемі көрінгенмен, оларды экранда оқу қиын.

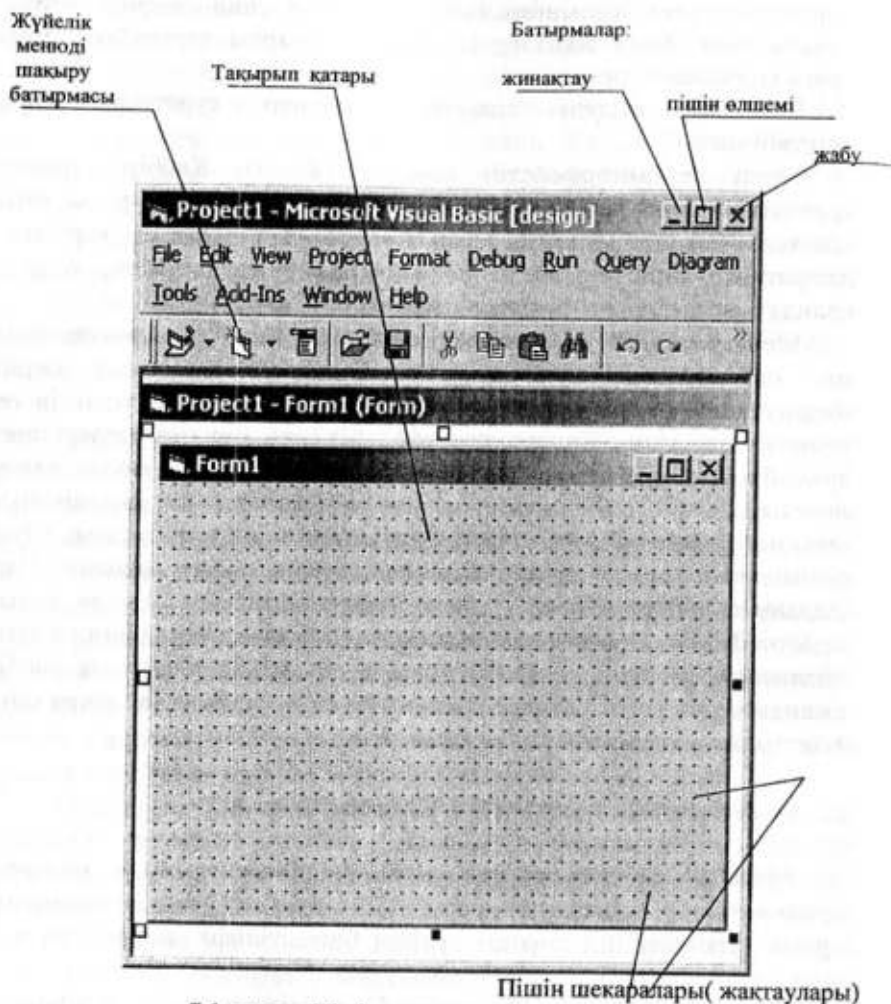
Меню мен саймандар панелі қосымшада құрылған командалар мен саймандарға қатынауды жеңілдету мүмкіндігін беріп, қолданушыға қосымша мүмкіндігі мен оның атқаратын қызметін тез түсінуге арналған. Қолданушы жиі бірінші кезекте (құжаттарға қарамай) меню құрамын қарастырады. Егер меню жақсы ойластырылған болса, онда қолданушы қосымшаның функционалдығы мен тиімділігі туралы өз әсерін құра алады. VB-гі қосымшаның әр бөлігінің өз жүйесі мен меню жиыны бар. Қолданушы интерфейсіндегі біртектілік қажеттілігі, менюге толық қатысты. Microsoft Word қосымшасының типінде қабылданған стилді қолданыңыз, менюді динамикалық жоя немесе қоса отырып, кейбір командаларды қоса немесе өшіре отырып, қосымша контексіне қатысты меню құрылымын бақылаңыз.

7.2 Пішінді жобалау

Пішін - қолданушы қосымшаның бағдарламалық кодымен қарым-қатынас жасайтын терезе (7.1-сурет). Олар қосымшаның барлық терезелерінің сыртқы түрінің біртектілігін қамтамасыз етуі керек.

Әртүрлі қосымшаларда немесе бір қосымша ішінде дизайндағы айырмашылық пен интерфейс сипаттамасы қолданушыны оқытудағы шығын мен техникалық қолданудың өсуіне әкеледі. Бұл фактор - корпоративтік ортада өте маңызды. Бүкіл қосымша интерфейсін басқару элементтері орналасқан бір пішіннен тұруы мүмкін. Пішіндердің өздерінің қасиеттері, оқиғалары мен әдістері бар,

олардың көмегімен жұмыс барысында пішіннің түрі мен сипаттамасын басқаруға болады.



7.1-сурет. Пішін элементтері

7.2.1 Пішін қасиеттерін орнату

Caption, BackColor және Font стандартты қасиеттерінен басқа, пішіндердің тек өзіне қатысты жеке қасиеттері бар. Пішін қасиеттерін

көру үшін қасиеттер терезесіндегі пішіннің бос жерінен шертү керек немесе қасиеттер терезесіндегі объект тізімінен пішінді таңдау керек.

Пішін қасиеттері:

- **Name қасиеті**, бағдарламалық кодта пішінге қатынау үшін қолданылатын жоба атынан тұрады. Үнсіз жариялау барысында, пішін жобаға қосылғанда, оның аты Form 1, Form 2, т.б. болады;

- **Caption қасиеті** - пішіннің титулдық қатарында шығарылатын мәтінді анықтайды;

- **Icon қасиеті** - кішірейтілген пішіннің пиктограммасын орнатады;

- **Border Style қасиеті** - стандартты терезенің жақтауы (Border) бар. Ол үшін жүйелік менюде сәйкес команда бар. Жақтау түрін Border Style қасиетінің көмегімен өзгертуге болады, оның қабылдай алатын мәндері 7.1-кестеде келтірілген. Border Style қасиетін өзгерте отырып, пішін өлшемін өзгерту режимін басқаруға болады;

- **ControlBox қасиеті**. ControlBox қасиеті - қолданушы бағдарламадан шыға алатын (Alt+F4), жүйелік менюді шақырушы батырма бейнеленгенін анықтайды. Егер жүйелік меню жойылған болса, қолданушыға бағдарламадан шығудың басқа тәсілін қамтамасыз ету керек;

- **MaxButton қасиеті** - үлкейту батырмасының көмегімен қолданушы терезені экран өлшеміне дейін үлкейтуге алады. Оның бар екені пішінінің MaxButton қасиетімен анықталады. Егер бұл қасиетке False мәнін меншіктесек, онда сәйкес батырма болмайды, ал команда (өрбіту) жүйелік менюден жойылады;

- **MinButton қасиеті**. Егер MinButton мәнін False деп орнатсақ, онда батырма қараңғыланып, ал жүйелік менюден Minimize (жинақтау) қатары жойылады;

- **Height және Width қасиеттері** - пішіннің алғашқы өлшемін анықтайды;

- **Left and Top қасиеті** - экранның сол жақ жоғарғы бұрышына қатысты пішін позициясын анықтайды;

- **WindowState қасиеті** - пішіннің іске қосылған – үлкейтілген, кішірейтілген немесе қалыпты күйінің бірін нұсқайды.

7.3. Пішіннің негізгі оқиғалары

Пішін модулі – оқиғаларды өңдейтін процедуралардан (Event produce) тұратын немесе оқиғаларды өңдейтін (event landlers)

модуль коды. Оқиғалар қолданушының немесе жүйелік оқиғалардың әрекеттеріне жауап ретінде генерацияланады. Әр оқиға өзінің процедурасымен өңделеді. Бұл процедураларға арналған кодты бағдарламашы жазады.

Пішіннің әртүрлі типті көптеген оқиғалары бар. Пішін қалай қолданылатынына байланысты кейбірі жетімсіз болуы мүмкін, бірақ басқалары ылғый жұмысқа қосылады. Пішіннің негізгі жүйелік оқиғалары төмендегі тәртіпте өңделеді:

- Initialize;
- Load;
- Activate;
- Deactivate;
- Query Unload;
- Unload;
- Terminate.

Initialize оқиғасы. Initialize оқиғасы әдетте қосымшаны жұмысқа дайындау кезінде қолданылады. Бұл оқиғаны өңдеу кезінде айнымалыларға алғашқы мәндері меншіктеліп, пішінге басқару элементтері қосылып, олар масштабталады. Initialize оқиғасы пішін данасын құру кезінде орнайды (оны бейнелеуге немесе жүктеуге дейін). Бірақ, бұл оқиға қосымшаның жұмыс сеансының кезінде тек бір рет генерацияланады. Initialize оқиғасын қайта шақыру үшін бағдарлама жұмысын аяқтап, оны қайта іске қосу керек. Сондықтан, егер қандайда бір код бірнеше рет орындалатын болса, оны басқа оқиғаны өңдеушіге көшіру керек.

Пішін деңгейінің айнымалылары, пішіннің берілген модулінің ішіндегі барлық процедураларға қатынай алады.

Инициализациядан кейін де, сәйкес пішін жүктеуден алып тасталғанда, қосымша жұмыс істеп тұрғанда пішін болады.

Пішінді қайта жүктегенде Initialize оқиғасы генерацияланбайды.

Load оқиғасы. Load оқиғасы пішінді экранға шығару алдында кейбір әрекеттерді орындау үшін қолданады. Мұнымен қатар, пішін басқару жүйесі қасиетінің алғашқы мәнін меншіктеуге мүмкіндік береді.

Бұл оқиға пішінді жадқа жүктеген сайын орнайды. Бірінші жүктеу кезінде Load оқиғасы Initialize оқиғасынан кейін атқарылады. Load оқиғасы Load және Slow оқиғаларын қолдану нәтижесінде, сонымен қатар жүктелмеген пішіннің әдістері мен басқару элементтерінің қасиеттері не берілген сілтемелерден кейін

генерацияланады. Төменде көрсетілген мысал, Load оқиғасын өңдеуде Combo Box инициализацияланып, үнсіз таңдалатын элементі бекітудің иллюстрациясы келтірілген:

```
Private Subform_Load()
```

```
Rem тізімге элемент қосамыз
```

```
Combo.AddItem "электровозы"
```

```
Combo.AddItem "гировозы"
```

```
Rem үнсіз жариялау бойынша таңдалатын элементті бекітеміз
```

```
Combo.ListIndex = 2
```

```
End Sub.
```

Activate/Deactivate оқиғасы. Қолданушы қосымшасы екі және одан да көп пішіндер арасында ауыстырылуы үшін Activate және Deactivate оқиғаларын осы пішін тәртібін анықтау үшін жұмысқа қосу керек. Енгізу фокусы берілген қосымшаның бір пішінінен берілген пішініне ауысу кезінде Activate оқиғасы орнайды. Бұл жерде пішін көрінетін болуы керек. Мысалы, Load операторы мен жүктелген пішін Show әдісі қолданғанға дейін немесе пішіннің Visible қасиетінің оқиғасы GotFocus оқиғасына дейін генерацияланады.

Deactivate оқиғасы бір қосымшадағы енгізу фокусы берілген пішіннен, басқа пішінге ауысқанда орындалады. Ол Lost Focus оқиғасынан кейін генерацияланады. Екі оқиға да (Activate және Deactivate) егер енгізу фокусы бір қосымшаның пішіндерінің арасында атқарған болса ғана генерацияланады. Егер сіз басқа қосымшаға ауысып, одан кейін Visual Basic бағдарламасына оралатын болсаңыз, онда бұл оқиғаның ешқайсысы орындалмайды.

QueryUnload оқиғасы. Бұл оқиға қолданушы пішінді қалай жабатынын білу үшін пайдалы.

QueryUnload оқиғасы - Unload оқиғасының алдында генерацияланып, Unload-қа қарағанда қосымша мүмкіндіктерді береді: Unload-Mode аргументі пішінінің алып тастау мүмкіндігін береді. QueryUnload оқиғасы орнайтын кейбір жағдайлар:

- қолданушы пішіннің жүйелік менюінен Close командасын таңдайды;

- бағдарламада Unload операторы орындалады;

- Windows операциялық жүйесінің ағымдағы жұмыс сеансы аяқталады;

Task Manager Windows операциялық жүйесі берілген жобаны жабады;

- MDI негізгі пішіні жабылатындықтан, MDI туынды пішіні де жабылады.

Пішінді жүктеуден алып тастау үшін Cancel аргументіне тиіс мәнін меншіктеу керек. Мысалы:

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)

' пішін қалай жабылатынын анықтаймыз ;

If UnloadMode < > vbFormCode Then

MsgBox "Use the Exit button to close the form"

Cancel = True

' пішін сшық қалады !"

End If

End Sub

Unload оқиғасы. Unload оқиғасы - Terminate алдында генерацияланады. Unload оқиғасын өңдеуші пішінді жүктеуден алып тастау керектігін тексеру немесе пішінді жүктеуден алып тастау кезіндегі орындалатын операцияларды анықтау үшін қолданылады. Сонымен қатар, пішінді жабу үшін немесе файлдағы деректерді сақтау үшін қажетті пішін деңгейін тексеруші кодты қосуға болады. Unload оқиғасын өңдеушіге End операторын қосуға болады. Ол бағдарлама аяқталғанға дейін, барлық пішінді жүктеуден алып тастаудың кепілдігін береді.

Cancel аргументіне нөлден басқа мән меншіктеу - пішінді жойдырмайды, бірақ Windows ортасынан шығу сияқты оқиғаларды орындауға тыйым салмайды. Windows-тан шығармау үшін QueryUnLoad оқиғасын қолдану керек.

Terminate оқиғасы. Бұл оқиға пішін данасына берілген барлық сілтемелерді жадтан жойған кезде генерацияланады. Бұл пішіннің айнымаларын жадтан алып тастау және онымен алынған жүйелік ресурстарды босату үшін, пішіннің объектілік айнымалысына Nothing мәнін меншіктеу керек. Мысалы:

Set frmMyForm = Nothing.

Кластан басқа, барлық объектілердің Terminate оқиғасы UnLoad оқиғасынан кейін генерацияланады. Бірақ, бағдарлама жұмысы дұрыс аяқталмаған кезде, жадтан класс немесе пішін данасы жойылса, бұл оқиға орындалмайды.

Resize оқиғасы. Resize оқиғасы - қолданушы немесе код арқылы пішін өлшемі өзгерген сайын, пішін инициализациялады. Бұл оқиғаның әсері ретінде пішінде орналасқан басқару элементінің өлшемін өзгерту немесе орнын ауыстыру сияқты әрекеттерді атқаруға болады.

7.4. Пішіннің негізгі әдістері

Әдіс объектімен қандай да бір операция орындайды. Пішін әдістерін білу - компьютердің жүйелік ресурстарын қосымшаны құруда тиімді қолдануға мүмкіндік береді.

Visual Basic бағдарламасында пішіндерді басқару үшін төменде аталған әдістер қолданылады:

- Load;
- Un Load;
- Hide;
- Show.

Пішіндерді құрудың алғашқы кезеңдерінде, олардың қай компьютерде жұмыс істейтінін ескерген өте маңызды. Себебі, компьютердің жүйелік ресурстарының шектелуі - пішіндерді өңдеу сипаттамасына әсерін тигізуі мүмкін. Сонымен, жәй жұмыс істейтін машиналарда пішіннің бір бөліктерін қосымшаны жүктеген кезде іске қосу керек. Одан кейін, қажеттілігіне қарай оларды Show және Hide тәсілдерін қолданып, орнатуға әрі жасыруға болады.

Load әдісі. Load операторы пішінді экранға шығармай, жадқа жүктеп инициализациялайды. Пішінге жасалған кез-келген сілтеме, ол жадқа әлі жүктелмеген болса, бұл пішін ресурстарын автоматты түрде жүктеуді орындайды. Төмендегі мысалда My Form пішіні жадқа жүктеледі, бірақ экранға шығарылмайды:

Load frm My Form.

UnLoad әдісі. UnLoad операторы пішінді жадтан жояды. Пішінді жүктеуді алып тастау - жүйелік ресурстарды босату немесе оның қасиеттерін ағымдағы күйде жинақтау үшін керек болады. Ағымдағы пішінге сілтеме орнату үшін Me операторын қолдануға болады. Келесі мысалда пішінді MyForm жадтан алып тастаудың екі тәсілі көрсетілген:

UnLoad. My Form

немесе

Un Load Me.

Hide әдісі. Hide тәсілі пішінді жадтан жоймай, экраннан алып тастайды. Пішіннің жасырын басқару элементтеріне қолданушы қол жеткізе алмағанмен, оларға бағдарлама арқылы қатынауға болады. Пішін жасырын орналасқан кезде, қолданушы қосымшаның сәйкес бөлігімен қарым-қатынас жасай алмайды. Төмендегі мысалда *My Form* пішінді барлық мәндерін сақтай отырып, жасырудың екі тәсілі келтірілген:

My Form Hide

немесе

Hide My

Show әдісі. Show тәсілі пішінді экранға шығарады. Егер шақыру кезінде пішін жадқа жүктелмеген болса, Visual Basic алдымен *Load* тәсілін шақырады. Show тәсілі пішіндерді модельді, не модельді емес түрде көрсетуге мүмкіндік береді. Модельді пішін қосымшаның басқа пішініне көшуге болғанға дейін, онымен жұмыс істеуді аяқтауды талап етеді. Ал модельді емес пішін жұмысты аяқтауды талап етпейді. Үнсіз жариялау кезінде Show әдісі пішінді, модельді емес түрде көрсетеді.

Егер Show тәсілі модельді емес пішінді шығарса, бағдарламаның келесі коды ретпен орындалады. Ал егер ол модельді пішінді көрсетсе, пішін жасырылғанша немесе жадтан жүктеуден алып тасталғанша келесі кодтың орындалуын тоқтатады. Соңғы жағдайда, барлық енгізулерді (пернелер тақтасынан немесе тышқан көмегімен) қосымша тек модельді пішінде қабылдайды және осы пішіннің модульдерінің процедуралары жұмыс істейді. Бірақ, бұл берілген қосымшаға ғана байланысты, қалған бағдарламалар өз жұмыстарын әдеттегідей атқарады. Show тәсілін қолдану мысалы:

frmResults. Show vbModal

7.4.1. Жобаға жаңа пішіндерді қосу

Visual Basic бағдарламаға бір немесе бірнеше пішінді қосуға мүмкіндік береді. Әр жаңа пішін - *объект* деп саналады және ол өз объектілерінен тұрады.

Объектідегі бірінші пішін аты: Form 1, ал келесілері - Form 2, Form 3, т.б. болады. Бағдарламадағы пішіндердің іс-тәжірибеде қолдануының мысалдары 7.1-кестеде берілген.

7.1-кесте

Бағдарламадағы пішіндердің іс-тәжірибеде қолданылу мысалдары

Пішін немесе пішіндер	Сипаттамасы
Экран – қыстырма	Сәлемдеме-хабарды, қыстырма немесе құрастырушы құқығын қорғау туралы ақпаратты бейнелейтін экран
Бағдарлама нұсқаулары	Бағдарламамен жұмыс істегенде, ақпарат пен ұсыныстарды бейнелейтін экран
Диалогты терезелер	Алғашқы мәндерді енгізуге және қорытындыланатын деректерді бейнелеуге арналған қолданушының диалогты терезелері
Құжат құрамы және графикалық нәтиже	Бағдарламада қолданылатын бір немесе бірнеше файлдардың құрамын бейнелейтін экран

7.4.2. Пішін бланктері мен үлгілері

Жаңа пішінді Project (жоба) менюіндегі Add Form (пішінді қосу) командасын шерту арқылы құруға болады. Құрылатын пішін типін таңдап алуға болатын, диалогты терезе пайда болады. Жаңа, бос пішін немесе арнайы есепке алынған кейбір аяқталған пішінді құруға болады.

7.4.3. Бұрын құрылған пішіндерді жобаға қосу

Visual Basic жаңа бағдарламалық жобаларда пішіндерді қайта қолдануға мүмкіндік береді. Ол үшін пішіндерді жеке .frm файлында сақтау керек. Бұрын құрылған пішінді жобаға қосу үшін Project (жоба) менюіндегі Add Form (пішінді қосу) командасын шерту керек. Шыққан диалогты терезеде Existing (бар жобалар) қосымшасын шертіңіз. Барлық пішіндер тізімінен тұратын диалогты терезе бейнеленеді. Осы тізімдегі бар пішінді қосу үшін диалогты терезедегі пішін файлының атын екі рет шерту керек. Visual Basic белгіленген пішінді жобаға қосады, одан кейін ол пішінді

қарастырып, оның *үзуді өңдеу* процедурасының мәтінін Project (жоба) терезесіндегі Code (бағдарламалар) мен View Object (жоба) батырмаларын шерту арқылы өзгертуге болады.

7.4.4. Іске қосу пішінін таңдау

Қосымша пішінің негізгі параметрлерін орнатумен қатар, іске қосу сұрақтарын шешіп және қосымша жұмысын, бұл кезде орындалатын процестерді ескере отырып, дұрыс аяқтай білу керек.

Үнсіз жариялау бойынша іске қосылу пішіні болып, қосымшадағы бірінші пішін алынады. Қосымша орындала бастағанда, бұл пішін экранға шығарылады (орындалатын бірінші код - бұл пішінің Form-Initialize оқиғасының процедурасының коды).

Егер іске қосу пішінін өзгерту керек болса:

- 1) Project менюінен Project Properties таңдау;
- 2) оның ішінен General-ды таңдау;
- 3) Startup Object тізімінен іске қосу пішіні болатын пішінді таңдау;
- 4) ОК басу керек.

7.4.5. Қосымшаны пішінсіз іске қосу

Кейде, қосымшаны алғашқы қатаң орнатылған пішінсіз іске қосу қажеттілігі туады. Мысалға, алдымен деректер файлын жүктейтін кодты орындап, деректер файлында бар нақты ақпаратқа байланысты, бірнеше пішіндердің ішінен біреуін экранға шығару керек делік. Бұл жұмысты төмендегі мысалда көрсетілгендей, стандартты модульден Main атымен берілген Sub процедурасын құру арқылы атқаруға болады (Get Status процедурасы бар деп есептелінеді):

```
SubMainO
```

```
Dim intStatus As Integer
```

```
' статусты тексеру функциясын шақыру intStatus = GetStatus
```

```
' алынған статусқа сүйене отырып пішінді ашу
```

```
If intStatus = 1 Then
```

```
frmMain.Show Else
```

```
frmPassword.Show
```

```
End If
```

Бұл процедура Sub процедурасы болып, ол пішін модулінде орналаспауы керек. Sub Main процедурасын іске қосу объектісі ретінде орнату үшін, Project менюінен Project Properties таңдап, General қосымшасын ашып, Startup Object тізімінен Sub Main таңдау керек.

7.4.6. Іске қосуда қыстырманы экранға шығару

Кейде іске қосу кезінде ұзақ процедураны орындау қажеттігі туады. Деректер базасынан үлкен көлемді ақпаратты немесе үлкен растілік суреттерде жүктеу керек болады. Осы уақытта қолданушыны сергіту үшін, экранға қандай да болсын қыстырманы шығаруға болады. Қыстырма экраны дегеніміз - әдетте, қосымша атынан, автордың құқықтық ақпаратынынан және жай суреттерден тұратын пішін.

Қыстырма-экранды шығару үшін, іске қосу объектісі ретінде Sub Main процедурасы мен пішінді шығару үшін Show әдісі қолданылады:

Private Sub MainO

' Қыстырма-экранды ашу

frmSplash.Show

' Бұл жерде іске қосу процедуралары орындалады

' Негізгі пішінді жүктеп, қыстырманы жабу

frmMain.Show

Unload frmSplash

End Sub

Іске қосу бағыныңқы бағдарламалар қосымшаның тез жүктелу елесін құрып, өз жұмысын жүргізіп жатқанда қыстырма-экран қолданушы назарын өзіне аударады. Іске қосу бағыныңқы бағдарламалары жұмысын аяқтаған кезде қыстырманы жауып, қосымшаның бірінші пішінін іске қосу керек.

Жобалау кезінде қыстырма-экранды қарапайымдылау безендендіру керек. Егер үлкен суреттер мен көптеген басқару элементтерін қолдансақ, онда қыстырманың өзі ұзақ уақыт жүктеледі.

7.5. Қосымша жұмысын аяқтау

Басқару-оқиғалы қосымша, барлық пішіндер жабылып, ешқандай код орындалмайтын кезде өз жұмысын аяқтайды. Егер

барлық көзге көрінетін пішіндер жабылғаннан кейін, қандай да бір жасырын пішін бар болса (бірақ жабылмаған болса), қосымша жұмысы аяқталғандай әсер аламыз. Бірақ, қосымша барлық жасырын пішіндер жабылғанша өз жұмысын жүргізеді. Бұл жағдайлар жиі кездеседі. Себебі, пішін немесе басқару элементінің қасиеттеріне қатынау кезде пішін экранға шығарылмай жүктеледі.

Бұл мәселені туғызбаудың ең оңай тәсілі - қосымшаны жабу кезінде барлық пішіндердің жүктелуден алынғанына көз жеткізу. Егер қосымшада бірнеше пішіндер болса, онда Forms коллекциясымен Unload операторын қолдануға болады. Мысалы, негізгі пішінде бағдарлама жұмысын дұрыс аяқтайтын cmd Quit командалық батырмасын орнату керек. Егер қосымшада тек бір пішін болса, онда пішіннің Click оқиғасының процедурасы тым қарапайым болады:

```
PrivateSub cmdQuit.Click ()  
Unload Me ,  
End Sub
```

Бұл жерде параметр ретінде ME - Visual Basic-тің түйінді сөзі қолданылады. ME - айқын жарияланбаған айнымалы, оны модульдің кез келген процедурасында қолдануға болады және ол кластың екпінді данасының көрсетілімі болып табылады (бұл жағдайда Form класы).

Егер пішінде бірнеше пішіндерді қолданса, онда негізгі пішіннің Unload оқиғасының процедурасына кодты орналастырып, олардың бәрін жүктеуден алып тастауға болады.

Қосымшаның барлық пішіндерін жүктеуден алып тастауда қолданылатын Forms коллекциясы төменде келтірілген:

```
private Sub Form_Unload  
Dim i as integer  
' Циклде пішіннің Forms коллекциясын жүктеуден алып тастау  
For I = 0 to Forms.Count - 1  
Unload Forms(I)  
Next  
End Sub
```


Кейде құрылған пішіндер мен объектілердің күйін ескермей, шұғыл жағдайлармен қосымша жұмысын аяқтау керек болады. Бұл жұмысты Visual Basic-те End операторы атқарады.

End операторы қосымша жұмысын қолма қол орындайды, одан кейін ешқандай код орындалмайды және ешқандай оқиғалар орнамайды. Атап айтқанда, Visual Basic пішінінің Query Unload, Unload немесе Terminate оқиға процедураларын орындамайды. Объектілік сілтемелер босатылады және егер қосымшада қандай да бір қолданушы кластары анықталған болса, Visual Basic бұл кластардан құрылған Terminate оқиғасын орындамайды.

End операторына қосымша Stop операторы бар, ол қосымша жұмысын аяқтамайды, оны тек тоқтатады. Алайда, Stop операторын жөндеу кезінде қолдану керек. Себебі ол, мысалға, объектілердің сілтемелерін босатпайды, айнымалыларды тазартпайды және ашылған файлдарды жаппайды.

7.6. Енгізуді бақылау

Қандай да қосымшаны құру кезінде – қарапайым калькулятор немесе деректер базасын басқаруға арналған күрделі жүйе болсын, қолданушы енгізетін ақпаратты бақылау керек. Енгізуді бақылаудың бірнеше тәсілдері бар: әр басылған пернені қабылдауға және пішінде енгізілген ақпаратты толығымен тексеруге болады.

Көптеген қосымшаларда қолданушы ақпаратты - жұмыс барысында өңдеу үшін енгізеді. Бақылау, қосымшаға енгізілетін әрбір мән күтілетін типке сәйкес болуына және берілген диапазонға сиятынына кепілдік береді. Деректерді тексеруді орындау үшін бірнеше қатынастарды қолдануға болады:

- қолданушы интерфейсінің кодында верификациялау;
- деректер базасының ішінде шектеулер қолдану;
- қолданбалы логика арқылы бақылау.

Деректерді тексерудегі ең кең тараған мақсат – оның типін бақылау. Мұнда “Қатарда әріптік белгілер бар ма?” және “Бұл сан екені анық па?” - деген жәй сұрақтар қойылады. Бақылаудың мұндай қарапайым түрі, қолданушының интерфейс кодында қолданылады.

Жақсы ойластырылған қосымша енгізу кезінде кездесетін көптеген қателерді болдырмауға себепкер болады. Мысалы, тек “сандар” енгізу керек болса, сандық деректерді қабылдайтын өріс құрылады, қолданушы бұл өріске басқа типті дерек енгізе алмайды.

7.6.1. Деректерді бақылау түрлері

Visual Basic-те енгізілетін деректерді бақылаудың екі тәсілі бар:

- пішін деңгейінде - деректер пішіндегі барлық өрістер енгізілгеннен кейін тексеріледі. Мысалы, пішінге тапсырыс берушінің атын, оның мекен-жайын, телефон нөмірін, тұрған қаласын, елін және пошталық индексін енгізу керек. Қолданушы бар өрістерді енгізіп болғаннан кейін, ОК батырмасын шертуі керек, одан кейін бағдарлама әр өрісте деректерді верификациялайды.

- өріс деңгейінде - деректер әр өріс толтырылғаннан кейін тексеріледі. Мысалы, қолданушы пошталық индексті енгізеді. Верификациялау - индексті теру процесінде, не енгізу фокусы келесі өріске көшкенде орындалады. Енгізілген мән, сандық екенін тексеру үшін Text Box басқару элементінің Change оқиғасын қолдануға болады.

7.6.2. Қолданушымен кері байланыс

Қолданушы өріске енгізуге болмайтын деректерді енгізгенде, оған дыбыстың немесе көрсетілімді сигналмен хабар беруге болады. Мұны қалай жасау керегі таңдалған бақылау тәсіліне байланысты болады. Егер пішін деңгейіндегі бақылау жүргізіліп онда қате кетсе, экранға мүмкін емес деректері бар өрістер тізбектелініп берілетін қосымша пішін шығаруға болады. Айталық, қолданушы пошталық индексті дұрыс енгізбей, тапсырыс беруші қай қаладан екенін көрсетпеді делік. Онда экранға екі өрістегі қателер көрсетілген жеке пішін шығаруға болады.

Өріс деңгейінде енгізуді бақылау кезінде қателерді дыбыс сигналымен хабарлауға болады. Мысалы, қолданушы пошталық индексті енгізу кезінде әріптік белгіні де енгізді делік. Бұл қатені тапқан кезде Visual Basic-те Veep операторын қолдануға болады, ол жүйеге стандартты дыбыс сигналын (default sound) бергізеді. Мұнымен қатар, экранға жіберілген қате - сипаттамасы бар терезені шығаруға болады немесе осы мақсатқа Status Bar басқару элементтік, төменде көрсетілгендей қолдануға болады:

```
If Not IsNumeric(txtZipCode.Text) Then  
    ' дыбыс сигналы
```

BEEP

' көрініс сигналы

StatusBar1.Panels("ErrorDescription").Text = _

"Zip code must be numeric."

End If

7.6.3. SetFocus тәсілін қолдану

Объект фокуста орналасқан кезде, ол қолданушының пернелер тақтасынан немесе маустан берген енгізулерін қабылдай алады. Microsoft Windows операциялық жүйелерінде бір мезгілде бірнеше қосымшалар жұмыс істей алады, бірақ тек біреуі енгізу фокусын алады; бұл қосымша екпінді деп саналады, ал оның терезесінің тақырып қатары - көк түспен ерекшеленеді. Тура осы айтылғандай, бірнеше мәтіндік өрістері бар пішінінің енгізу фокусында тек бір өріс орналасады. Басқару элементі енгізу фокусын алғаннан кейін, ол үшін GotFocus оқиғасы генерацияланады. Енгізу фокусын жоғалтқан кезде, LostFocus оқиғасы орнайды. Бұл оқиғалар пішіндер мен көптеген басқару элементтерінде қолдау табады. Кейбір басқару элементтері (мысалы; Frame, Line, Image і Label) енгізу фокусын қабылдамайды. Бұл орындау кезінде көрінбейтін (Time, т.б.) элементтерге де қатысты.

Енгізілетін деректерді бақылау үшін қай тәсіл таңдалса да, фокусты қате деректер табылған өріске қайтару керек. Сонымен, егер қолданушы мәтіндік өрісте пошталық индексті қате енгізсе, курсорды SetFocus тәсілін қолданып сол өріске қайтару керек. Сонда қолданушы өз қатесін түзете алады.

IF NOT ISNUMERIC(TXTZIPCODE.TEXT) THEN

BEEP

StatusBar1.Panels("ErrorDescription").Text = _

"Zip code must be numeric."

' енгізу фокусын пошталық индекс өрісіне қайтарады
txtZipCode.SetFocus

End If.

7.6.4. *LostFocus* оқиғасын қолдану

Бұл оқиға - объект фокусты жоғалтқан кезде орнайды. Оны жоғалту – қолданушының басқа өріске көшу немесе басқа объектіңі шерту әрекетін орындау нәтижесінде болады. Енгізу фокусын басқару элементінің арасында орналастыруды, бағдарлама арқылы - *SetFocus* тәсілімен атқаруға болады. *LostFocus* оқиғаны өңдеу процедурасы, өріс деңгейінде деректерді бақылау үшін қолайлы. Бірақ, онда *SetFocus* әдісі шақырылған болса, ол шексіз циклден шықпай қалуы мүмкін.

```
Private Sub txtZipCode_LostFocus()  
If Not IsNumS'ricdxtZipCode.Text) Then  
'дыбыс сигналы  
Beep  
'көрініс сигналы  
StatusBar1.Panels("ErrorDescription").Text = _  
"Zip code must be numeric."  
txtZipCode.SetFocus  
End If  
End Sub
```

Бақылау сұрақтары:

1. Қосымшаны құру принциптерін атаңыз.
2. Пішін қасиеттерін атаңыз.
3. Қолданылатын негізгі оқиғаларды атаңыз.
4. Қолданылатын негізгі әдістерді атаңыз.
5. Қосымша жұмысын аяқтау тәсілдерін атаңыз.

8. БАҒДАРЛАМАНЫ ЖӨНДЕУ

8.1. Бағдарлама кодын редакциялау

Бағдарлама кодын редакциялау дегеніміз компьютерге (анығырақ айтсақ Visual Basic ортасына) жаңа бағдарлама кодын, немесе өзгертілген ескі кодты енгізу.

Редакциялау үшін *Бағдарлама кодының терезесі* қолданылады. Бағдарлама қағаз бетіне жазылғаннан кейін оның қатесі жоқ екеніне ешкім кепілдік бере алмайды. Ондағы қателер екі түрлі болады – *синтаксистік және логикалық*.

Синтаксистік қателерді Visual Basic жүйесі өзі тауып, оны бағдарламаны енгізуші адамға хабарлайды.

Логикалық қателерді Visual Basic жүйесі бағдарлама іске қосылған кезде таба алады. Нақтылы айтатын болсақ Visual Basic жүйесі қатені тапқанымен бағдарлама жұмысының нәтижесін сіздің қалауыңыздай шығармайды, бірақ оны әртүрлі түрде сипаттайды:

- бағдарлама аяғына дейін жұмысын аяқтамай күрт тоқтауы мүмкін;
- бағдарлама *циклден* шықпай қалуы мүмкін;
- бағдарлама сіздің күткен нәтижеңізден басқа нәтиже беруі мүмкін.

Сонымен, Visual Basic нақты қатені таба алмайды. Сіздің міндетіңіз – бағдарламаның дұрыс жұмыс істемей тұрған себебімен табу. Бағдарламаны *өңдеу* негізі осында.

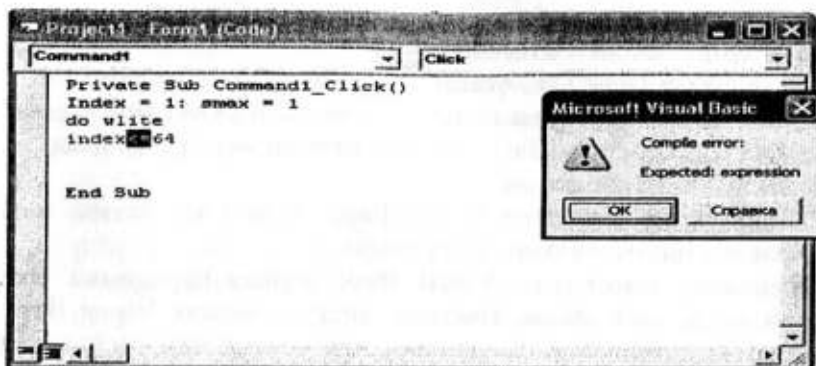
8.2. Бағдарламаны редакциялау

Бағдарламаны редакциялау жұмысын Visual Basic-тің *ішкі редакторы* атқарады. Оның көмегімен келесі жұмыстар атқарылады:

- бағдарлама синтаксисін оперативті түрде тексереді;
- Visual Basic тілі бойынша оперативті анықтама береді;
- экранға қажетті әзір-жауаптарды шығарады;
- кілттік сөздерді және комментарийді жеке түспен белгілейді.

Бағдарлама коды пернелер тақтасының көмегімен қатар тізбегін теру арқылы *енгізіледі*. Visual Basic бұл процессті қатаң қадағалайды. Егер сіз оның синтаксисінің бір ережесін бұзған болсаңыз, ол қолма қол қате жазбаны қызыл түске бояп, кішкене хабар терезесіне (8.1. сурет) қате туралы мәлімет шағарады.

Бұл хабарда *Do While ... Loop* операторының синтаксисі бүлінген — *index <= 64 шарты* While кілттік сөзі орналасқан қатарда орналасуы керек. Хабар туралы қате шығару терезесінде екі перне орналасқан — *OK* және *Анықтама*.



8.1. сурет - Синтаксисті қате туралы хабар шығару терезесі

Егер қате өзіңізге түсінікті болса, онда *OK* пернесін шерте беріңіз, ал қате түсініксіз болса *Анықтама* пернесін бассаңыз біраз уақыттан кейін Visual Basic-тің қате туралы анықтама ақпараты бар терезесі пайда болады.

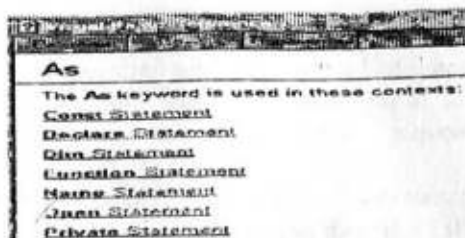
Оперативті анықтаманы Visual Basic-тің кез келген кілтті сөзі бойынша алуға болады. Ол үшін курсорды қажетті сөзден кейін орнатып *F1* пернесін басу керек.

Мысалы, бағдарламамызда *As* сөзінен кейін бірінші қатарға курсорды орнатып, *F1* пернесін басамыз. Нәтижесінде 8.2. суретте келтірілген мәлімет пайда болады.

Айталық, Сіз экрандық форманың кез келген объект атын, мысалы мәтіндік *Text1* өрісінің атын таңдап, нүкте қойсаңыз, онда Visual Basic объектінің *Мәтіндік* класс қасиеттерінің толық тізімін ашады.

Бұл тізімдегі қасиеттер алфавит бойынша реттелген. Tab пернесін басқан кезде, сөз тез арада нүктеден кейін орналасады. Егер бағдарламаны енгізу барысында сіз жаңа қатарға көшетін болсаңыз, ал Visual Basic жүйесі ешқандай қателерді тіркемеген болса, онда ішкі редактор тез арада арнайы түспен келесі сөздерді белгілейді:

- *кілттік сөздерді* — көк түспен;
- *комментарийді* — жасыл түспен.



8.2. сурет - As кілттік сөзі бойынша берілген Visual Basic-тің анықтамасы

Кодты редакциялау терезесінде редакциялаудың қосымша саймандар панелі бар (8.3. сурет).

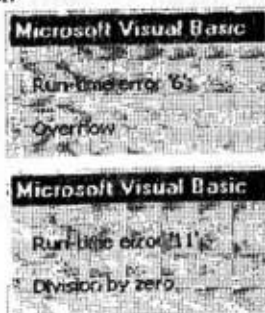


8.3. сурет - Кодты редакциялау терезесі

8.3. Бағдарламаны өңдеу

Бағдарламаны өңдеу жұмысын атқару:

- бағдарламаны уақытынан бұрын тоқтату. Мұндай тоқтаудың себебі көп болуы мүмкін. Мысалы, айнымалыға мүмкін болатын мәндер (*Overflow*) шекарасынан асқан мән меншіктеу; санды нөлге бөлу (*Division by zero*) кезінде. Бұл себептердің салдарынан Visual Basic-тің қате туралы мәліметі бар *хабар терезесі* шығады. Оны *Орындау кезіндегі қателер* — *Run-time error* (8.4. сурет) категориясына жатады.



8.4. сурет - Visual Basic-тің *Орындау кезіндегі қателер* туралы хабары

Бұл хабарға жауап ретінде терезенің соңғы қатарында орналасқан төмендегі перненің бірін басы керек.

Жұмысты доғаруыңызға (*End*), бағдарлама *өңдеушіні* (*Debug*) қосуыңызға немесе Visual Basic-тен анықтама (*Help*) алуыңызға болады;

- бағдарламаның циклденуі. Атқаратын қызметі — пернелер тақтасынан **Ctrl + Break** пернелерін басы.

Жұмыс аяқталған кезде Visual Basic бағдарламаның бір қатарын *сары түске* бояйды — бұл соңғы жұмыс істеген оператор. Бұл ақпарат бойынша бағдарламаның циклдену қатесін анықтауға болады.

ӨНДЕУШІ. *Өңдеушіні* жұмысқа қосу үшін *Бас панелдің* саймандар сызғышын маустың *оң жақ* пернесін басып, ашылған менюден *Debug* командасын таңдау керек. Бұл команданы шерткен кезде, жобаның *Негізгі панелінің* сызғышында *Debug* — Бағдарлама өңдеушісі пайда болады. Бұл сызғышты (8.5. сурет) мауспен екі рет шертіп, терезе түріне айналдыра аламыз.



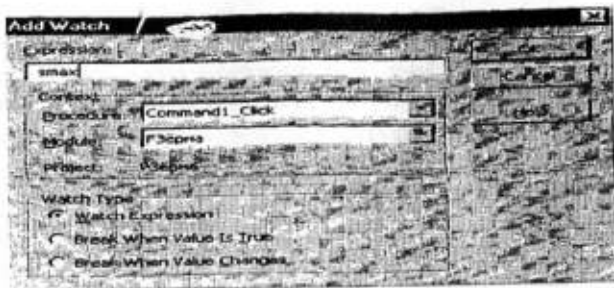
8.5. сурет - Бағдарлама өңдеуші саймандар терезесі

Алғашқы үш сайман— бағдарламаны *Іске қосу* (*Run*), *Жұмысты тоқтату* (*Break*), және *жұмысты аяқтау* (*End*) батырмалары.

Өңдеуші. *Өңдеуші* саймандарына қол жеткізу үшін *Негізгі панель* менюінен **Debug** опциясын қолдануға болады, онда *Өңдеуші саймандары* менюі ашылады.

Бағдарламаны тоқтату оның қателерін іздеуде тиімді болу үшін *Бақылау терезесін* орнатамыз. Онда бағдарлама тоқтаған кездегі біз таңдаған айнымалының мәні беріледі. Біздің бағдарламамыздағы айнымалылар: *index* және *smax*.

Өңдеушінің басқа да саймандары бар, оларды қажетіне қарай оқып-үйренуге болады.



8.6. сурет - Бақылау терезесін орнату панелі

Біздің жобаға *Бақылау терезесін* қосу үшін *Debug* опциясынан *Add Watch* командасын шертеміз. Ашылған *Add Watch* терезесінің көмегімен *Бақылау терезесінде* өзімізге қажетті айнымалылардың атын енгізуге мүмкіндік береді (8.6. сурет).

Expression	Value	Type	Context
index	2748	Integer	F3.epna Cont...
smax	1	Double	F3.epna Cont...

8.7. сурет - Айнымалылардың мәні туралы ақпараты бар *Бақылау терезесі*

8.7. суретте *Бақылау терезесінің* жұмысы келтірілген бұл айнымалылардың мәні біздің бағдарламадағы қатені табу үшін өте пайдалы.

Immediate терезесін қолдану. Бағдарламаны өңдеу кезінде кейбір оператордың жұмысын көру қажеттілігі туады. Бұл жағдайда не істейміз?

Сіз саймандар сызғышындағы *Пауза* пернесінің және *Immediate (Тез арада орындау)* терезесінің көмегімен бағдарлама жұмысын тоқтатып, әр қатардағы оператор жұмысын тексере аласыз. Сонымен қатар, сіз бұл терезеге тез арада орындалатын бір қатарлық операторды енгізе аласыз.

Бақылау сұрақтары:

1. Бағдарлама кодын енгізу.
2. Бағдарламаны өңдеу тәсілдерін атаңыз.
3. Өзір жауапты қолдану тәсілдерін атаңыз.

9. ӨЗІНДІК БАҚЫЛАУҒА АРНАЛҒАН ТЕСТ СҰРАҚТАРЫ

1. Visual Basic бағдарламалау стилінің қайсына жатады?
 - A) визуалды
 - B) соқыр
 - C) жүйелі
 - D) базалық
 - E) нүктелі
2. Visual Basic-ті іске қосқаннан кейін жаңа проект қалай ашылады?
 - A) New Projects диалогты терезесі арқылы
 - B) Tool Box панелі арқылы
 - C) Әске қосу перені арқылы
 - D) жарлық арқылы
 - E) негізгі меню арқылы
3. Қолданушы ортасының *Негізгі терезесінде* неше терезе ашуға болады?
 - A) 13
 - B) 3
 - C) 5
 - D) 15
 - E) 1
4. Объектілер қай панелде орналасады?
 - A) элементтерді басқару панелінде
 - B) саймандар панелінде
 - C) Бас меню панелінде
 - D) тақырып панелінде
 - E) уақыт панелінде
5. Visual Basic өңдеу ортасының тақырып қатары қалай орналасады?
 - A) бірінші болып
 - B) саймандар панелінен кейін
 - C) элементтерді басқару панелінде
 - D) Бас меню панелінде
 - E) форма терезесінде
6. Объект қасиеттері қай терезеде орналасқан?
 - A) қасиеттер терезесі
 - B) жоба терезесі
 - C) код терезесі

- D) форма терезесі
 - E) дұрыс жауап жоқ
7. Әр объект қандай параметрлермен сипатталады?
- A) уақиғалар, әдістер, қасиеттер
 - B) класс, уақиғалар, әдістер
 - C) туындылау, класс, полиморфизм
 - D) инкапсуляция, туындылау, полиморфизм
 - C) уақиғалар
8. VB іске қосылғаннан кейін New Object диалогты терезесі арқылы қай жоба ашылады?
- A) жаңа
 - B) ескі
 - C) өшірілген
 - D) жақындатылған
 - E) бұзылған
9. Басқару элементінің панелінде не орналасады?
- A) басқару элементтері
 - B) форма батырмалары
 - C) диалогты терезе түрлері
 - D) маус сілтеуіші
 - E) бас меню
10. Орындау кезінде формада объект қай қасиетті жасырады?
- A) Visible
 - B) Name
 - C) Font
 - D) Alignment
 - E) Text
11. Әр объект атына қай қасиет енгізіледі?
- A) Name
 - B) Caption
 - C) Visible
 - D) Font
 - E) Alignment
12. Жазба объектісі формада қандай басқару элементін құрады?
- A) Label
 - B) Text
 - C) Caption
 - D) Visible
 - E) Alignment

13. Топтан бір ғана опцияны орнату үшін қай басқару элементі қолданылады?
- A) Option Button (ауыстырып-қосқыш)
 - B) Check Box (жалауша)
 - C) Command Button (батырма)
 - D) Label (жазба)
 - E) Text
14. Опция қатарынан біреуін немесе бірнешеуін таңдау үшін қай басқару элементін қолданамыз?
- A) Check Box (жалауша)
 - B) Menu (меню)
 - C) Option Button (ауыстырып-қосқыш)
 - D) Command Button (батырма)
 - E) Text
15. Бағдарламалық код терезесін қай команда арқылы ашуға болады?
- A) View Code
 - B) View Project
 - C) View Form
 - D) Toggle Folder
 - E) View Cinema
16. Форма терезесін қай команда арқылы ашуға болады?
- A) View Form
 - B) View Code
 - C) View Project
 - D) Toggle Folder
 - E) View Print
17. Жоба терезесін қай команда арқылы ашуға болады?
- A) View / Project Explorer
 - B) File / Open
 - C) File / Print
 - D) File / Exit
 - E) View / Save Project
18. БЭ қасиеттері терезесін қай команда арқылы ашуға болады?
- A) View / Properties
 - B) View / Code
 - C) View / Form
 - D) дұрыс жауап жоқ
 - E) File / Exit

19. Маркерлер қай объект аймағында орналасады?
- A) активті
 - B) көрінбейтін
 - C) қол жетпейтін
 - D) өшірілген
 - E) дұрыс жауап жоқ
20. Басқару элементерінің терезесі қай команда арқылы ашылады?
- A) View / Toolbox
 - B) View / Code
 - C) View / Object
 - D) View / Debug
 - E) View / File
21. Батырмаға мәтін орнататын қасиетті атаңыз.
- A) Caption
 - B) Name
 - C) Value
 - D) Click
 - E) File
22. Белгі өрісіне мәтін орнататын қасиетті атаңыз.
- A) Caption
 - B) Name
 - C) Value
 - D) Click
 - E) File
23. Қай қасиет арқылы шрифт, оның түсі, өлшемі және жазылу үлгісі орнатылады?
- A) Font
 - B) File
 - C) Name
 - D) Alignment
 - E) Click
24. Фон түсін қай қасиет арқылы орнатамыз?
- A) Back Color
 - B) File
 - C) Font
 - D) Code
 - E) Caption

25. Мәтін әріптерінің түсін қай қасиет арқылы орнатамыз?
- A) Font Color
 - B) Name
 - C) Disk
 - D) Click
 - E) Home
26. Белгі аймағында мауспен екі рет шерткен кезде қандай оқиға атқарылады?
- A) Dbl Click
 - B) Mouse Down
 - C) Key Preview
 - D) дұрыс жауап жоқ
 - E) Click
27. Басқару элементін шерткен кезде қай оқиға орнатылады?
- A) Click
 - B) Mouse Up
 - C) Focus
 - D) дұрыс жауап жоқ
 - E) Dbl Click
28. Маус пернесін басқан кезде қай оқиға шақырылады?
- A) Mouse Down
 - B) Click
 - C) Dbl Click
 - D) Focus
 - E) End
29. Маус пернесін босатқан кезде қай оқиға шақырылады?
- A) Mouse Up
 - B) Click
 - C) Dbl Click
 - D) Focus
 - E) End
30. Формада түзу сызық құру үшін қай басқару элементі қолданылады?
- A) Line
 - B) Mouse
 - C) Focus
 - D) Key
 - E) Circle

Дұрыс жауаптар кестесі

Сұрақ нөмірі	Дұрыс жауап	Сұрақ нөмірі	Дұрыс жауап	Сұрақ нөмірі	Дұрыс жауап
1	A	11	A	21	A
2	A	12	A	22	A
3	A	13	A	23	A
4	A	14	A	24	A
5	A	15	A	25	A
6	A	16	A	26	A
7	A	17	A	27	A
8	A	18	A	28	A
9	A	19	A	29	A
10	A	20	A	30	A

ГЛОССАРИЙ

File (*Файл*) менюі Microsoft фирмасының басқа қосымшасындағыдай стандартты командалардан тұрады.

Edit (*Түзету*) менюі түзетудегі кеңінен танымал командалардан тұрады.

View (*Түр*) менюі жұмысты жалғастыру үшін *жобаның негізгі панелінде саймандар терезесін* ашу мүмкіндігі.

Project (*Жоба*) менюі жұмысты жалғастыруға арналған командалар.

Format (*Формат*) менюі - экран пішінінде орналасқан объектілерді форматтау мүмкіндігі.

Debug (*Жөндеу*) менюі бағдарламаны *жөндеу* мүмкіндігі.

Run (*Орындау*) менюі – бағдарламаны орындау режимін басқару мүмкіндігі.

Tools (*Саймандар*) менюі жаңа процедуралар мен функцияларды құру, *түзету менюінің терезесін* ашу, жобалау ортасының параметрлерін баптау командалары.

Add-Ins (*утилиттерді қосу*) менюі – қосымша мүмкіндіктерін қосу арқылы жобалау ортасының мүмкіндіктері.

Window (Терезе) менюі негізгі панельдегі терезенің орналасу ретін (жазық, тік немесе каскадты түрін) өзгерту мүмкіндігі.

Help (Анықтама) менюі Visual Basic-тің анықтама жүйесімен жұмыс істеу мүмкіндігі.

АЖО - белгілі бір пән аймағында дербес кәсіби ЭЕМ-ді пайдаланатын қолданушылардың жұмысын қамтамасыздандыруға арналған әдістемелік, тілдік және бағдарламалық құрылғылар жиыны.

Ақиқат, дұрыс ақпарат (Информация достоверная) – істің шынайы нақты жағдайын көрсететін ақпарат.

Анықтама немесе көмек сұрау (Справочник или помощь) – көмек алу мүмкіндігін гипермәтін тәсілімен ұйымдастырған анықтамалық мәлімет.

Ауыстырып-қосқыш (Переключатель) – " Іске қосылған" немесе "қосылмаған" тәрізді мәндердің бірін автоматты түрде немесе тапсырма бойынша қабылдай алатын бағдарламадағы айнымалы, ол – бірнеше мәселені шешу тармақтарының бірін таңдауға мүмкіндік береді.

Әдістер - объектімен байланысты, кейбір өндеулерді орындайтын бағдарламалық процедуралар.

Бағалы ақпарат (Информация ценная) – ақпарат бағалылығы оның көмегімен қандай мәселелер шешуге болатынына байланысы анықталады.

Бағдарлама – өңделуге тиісті реттелген командалар тізбегі; есепті шешу алгоритмін сипаттайтын бағдарламалау тілінің сөйлемдер тізбегі.

Бастапқы жүктеу (Начальная загрузка) – 1) Құрылғыға жұмыс істеуге керекті қалыпты өздігінен орнататын процедура. 2. Операциялық жүйенің бастапқы бөлігі жүктелгенде, кез келген құрылғы өздігінен басқарылады.

Деректер (мәліметтер) қоймасы немесе базасы (База данных) – қолданбалы бағдарламаға тәуелсіз, белгілі бір ережелер бойынша ұйымдастырылған деректерді сақтау, сипаттау, олармен әртүрлі әрекеттер жасау принциптері қарастырылған деректер жинағы.

Деректерді сұрыптау (Сортировка данных) – белгілі бір ережелер бойынша жиындар элементін топтарға жіктеу.

Диск – файл түрінде берілген ақпаратты магнитті тәсілмен жазып алатын құрылғы.

Дисплей - ақпаратты бейнелеудің негізгі құрылғысы болып саналады.

ДЭЕМ-ның негізгі құрылғысы – микропроцессор, ол бағдарлама құрамындағы әртүрлі операциялардың орындалуын қамтамасыз етеді.

ДЭЕМ-нің жады екі компоненттен (бөліктен) тұрады – тұрақты сақтау құрылғысы (ПЗУ) және оперативті еске сақтау құрылғысы (ОЗУ).

Екпінді терезе (Окно активное) – дәл осы мезетте жұмыс істеп тұрған екпінді терезе. Басқа терезелердің үстінде орналасып, маус пен пернелер әсер ететін терезе.

Енгізу-шығару базалық жүйесінің кеңейту модулі (Модуль расширения базовой системы ввода-вывода) – сыртқы құрылғылармен ақпарат алмасу әрекеттерін орындайтын операциялық жүйенің негізгі модулі.

Енгізу-шығарудың базалық жүйесі (Базовая система ввода-вывода) – MS DOS операциялық жүйесінің қызмет ету функцияларын атқаратын модульдерінің бірі.

Жад (Память) – деректерді қабылдауға, сақтауға және оларды басқаларға беруге арналған ЭЕМ-нің функционалдық бөлігі.

Жедел жад (Память оперативная) – осы мезетте орындалып жатқан бағдарламалар мен оларға дер кезінде қажет деректерді сақтайтын жад.

Жол аралық интервал (Межстрочный интервал) – дисплей экранында немесе қағаз бетіндегі қатар орналасқан жолдар арасындағы қашықтық.

Жүйелік дискет (Дискета системная) – операциялық жүйенің өз файлдары сақталатын дискет.

Жұмыс аймағы (Рабочая область) – енгізілген ақпарат орналасатын терезе аймағы.

Жүйелік ақпарат (Информация системная) – компьютердің құрылғылары, сипаттамалары туралы толық ақпарат: процессордың жұмыс өнімділігі мен разрядтылығы, сыртқы және жедел жад көлемі, адаптер немесе пернелер тақтасының типі т.с.с. параметрлер.

Жүйелік меню (Windows ортасында) (Меню системное) – терезелер арқылы орындалатын негізгі іс-әрекеттер мен командалардың тізімі.

Жүйелік менюдегі батырма (Кнопка системного меню) – жүйелік менюді шақыруға арналған батырма.

Инкапсуляция – ОББ-ның барлық версиясында объектімен жұмыс істеудің негізгі концепциясы болып табылады. Формалды

түрде инкапсуляция дегеніміз - деректер мен объект тәртібінің түрлерін бір пакетке біріктіру, оны жүзеге асырудың анық-қанығын қолданушыдан жасыра білу. Объект деректері айнымалы-мүше, айнымалы түрі немесе түр өрісі.

Келісім (Соглашение) – модуль аралық байланысты ұйымдастыру және сақтау аймағының стандартты құрылымын анықтау үшін қолданылатын регистрлерді пайдалану ережелері.

Кодтау (Кодировка) – 1). Деректерді олардың кодтық комбинация-ларымен бейнелеу немесе деректер элементі мен символдар жиынының кодтық комбинацияларын сәйкестендіру. 2). Нақты сипаттамаларды бағдарламаға түрлендіру.

Компьютерлік сауаттылық (Компьютерная грамотность) – ЭЕМ-ді пайдаланып, мәліметтер оқу, жазу, есептеу, сурет салу және ақпарат іздеу жолдары. Белгілі бір саладағы мәселені шешу жолында ЭЕМ-ді қолдану үшін керекті мүмкіндіктер мен білімдер жиыны.

Конфигурация – белгілі бір жүйені оның нақты сипаттарын, құрылғылар құрамын, өзара байланыстарын және олардың функционалдық элементтерінің негізгі сипаттамаларын анықтай отырып орналастырушы; аппараттық жабдықтар мен олардың бір-бірімен байланысу мүмкіндіктерінің жиыны және көрсетілген кешенге немесе жүйеге қосылатын қондырғылар жиыны.

Көп есептілік (Многозадачность) – операциялық жүйенің бір мезетте бірнеше есептерді шығару мүмкіндігі.

Курсорды басқаратын перне (Клавиша управления курсором) – дисплей экранындағы курсорды белгілі бір бағытта бір орынға жылжытатын перне.

Қалқыма терезе (Окно всплывающее) – таңдап алынған түйінді сөзді түсіндіретін қосымша мәліметі бар терезе.

Масштабтау (Масштабирование) – 1) Берілген шамалар мен оларды түрлендіру нәтижелерін белгілі бір диапазоннан шығармай көрсету тәсілі. 2) Бейнені немесе оның бөлігін үлкейтіп немесе кішірейтіп көрсету.

Математикалық модель (Математическая модель) – 1). Объектінің қызметі мен құрылымын сипаттайтын математикалық байланыстар жүйесі. 2). Математикалық формула және теңдеулермен өрнектеу арқылы кез келген объектін математикалық түрде сипаттау.

Маус(Мышь) – дисплей экранындағы сілтеме көрсеткіш қалпын өңдеуге арналған құрылғы.

Меню жолы (Строка меню) – менюдің құрамына кіретін командаларды көрсететін экранның екінші жолы.

Меню – экран бетінде көрсетіліп тұрған бірнеше мүмкіндікті тізім. Оның кез келген біреуін таңдап алуға болады. Таңдап алынған мүмкіндік (вариант) – жүйенің келесі орындалатын іс-әрекетін сипаттайды.

Микропроцессор – орталық процессордың қызметін атқаратын интегралдық схема.

Объект - біртекті технологияны қолдана отырып, онымен ұқсастыруға және басқаруға болатын жүйенің әр компоненті.

Операциялық жүйе жүктеушісі (Загрузчик операционной системы) – операциялық жүйенің модульдерін жедел жадқа оқитын бағдарлама.

Өзекті ақпарат (көкейтесті) (Информация актуальная) – өзгерген жұмыс жағдайларында қажетті маңызын жоғалтпайтын ақпарат, яғни зерттелу аймағында объектінің қалып-күйін кез келген сәтте дәлме-дәл бейнелей алатын ақпарат.

Полиморфизм (грек тілінен “көпжақтылық”) дегеніміз туындалған объектілердің, туындау тізбегінде орналасқан орнына байланысты, қандай тәсілді қолдану.

Протоколдар – хабарларды жіберу формалары мен тәсілдерін анықтайтын стандарттар. Протоколдар әртүрлі құрал-жабдықтардың желілерде бірге жұмыс істеу тәртіптерін орнатады.

Редакциялау (түзету, толықтыру) – деректер мазмұнын түзету.

Сандар форматы (Формат чисел) – сандарды бейнелеу түрі.

Сызық (Линия) – 1). Бейнелеу, сызу элементі. 2). Графикалық редактордағы ең қарапайым геометриялық элемент.

Сілтеме (Указатель) – 1). Сілтеу, байланысу адресі. 2). Курсор.

Сілтеуіш (Проводник) – Windows жүйесінің файлдармен жұмыс істейтін бағдарламасы. Оның көмегімен каталогтар мен файлдар көшірмесін алуға, орнын ауыстыруға, өшіруге, компьютерлік желілерде байланыс жасауға, бағдарламаларды іске қосуға, дискеттерді форматтауға, т.с.с. орындауға болады.

Тақырып жолы (Строка заголовка) – құжаттың немесе қолданбалы бағдарламаның аты көрсетілетін жатық жол (қатар).

Текст немесе мәтін – кез келген символдар жиыны, құжаттың, бағдарламалардың және хабарлардың ақпараттық мазмұны.

Терезе (Окно) – бөлек экран тәрізді жұмыс істейтін дисплей экранының бөлігі; кез келген хабарды қабылдап және өңдейтін объект.

Терезе шекарасы (Граница окна) – терезенің периметрі бойынша жүргізілген тік және көлденең сызықтар.

Толықтыру (Модификация) – объектінің мазмұнын (мағынасын) өзгертпейтін кез келген түрлендірулер.

Тұрақты жад (Память постоянная) – ЭЕМ жұмысы барысында мазмұны өзгертілмейтін жад. Әдетте бұл жадқа мәлімет оны құрып жатқанда жазылады.

Түпкі каталог (Корневой каталог) – ең жоғарғы деңгейдегі негізгі каталог.

Түсінікті ақпарат (Информация понятная) – ақпаратты алушыға түсінікті тілде жазылған мәлімет.

Тізімдер (Списки) – келтірілген тізімнен бір немесе бірнеше вариантты таңдап алу мүмкіндігін беретін сұқбат терезелерінің стандартты элементі.

Ұяшық сілтемесі (Указатель ячейки) – ағымдағы ұяшықты анықтайтын, теңбіл түсті тік төрт бұрыш (теңбілтор).

Ұяшықтарға тосқауыл қою (Блокировка ячеек) – ұяшықтарда мәлімет жазу, өзгертуден сақтау.

Файлдық жүйе (Система файловая) – ат қойылған деректер мен бағдарламалар жиынтығы.

Хабарлар жолы (Строка сообщений) – осы мезетте орындауға болатын командалар туралы анықтама берілетін жол.

Шекара (Граница) – белгілі бір шаманың өзгеру шегі.

ӘДЕБИЕТТЕР ТІЗІМІ

1. Шевякова Д.А., Степанов А.М., Карпов Р.Г. Самоучитель Visual Basic 2005.– С.П.: БХВ – С.-Пб. 2006.
2. Нұрпейісова Т.Б. КТМ пайдаланудағы және жерасты кенішті БЖ жұмыс орнын автоматтандырудағы жобаны құру. – Алматы: КазНТУ, 2004.
3. Нұрпейісова Т.Б. КТМ мен жерасты кенішті БЖ ЖОА арналған қосымша интерфейсін құру. - Алматы: КазНТУ, 2004.
4. Маркова Л.Н. Автоматизированные рабочие места при эксплуатации ГМ и ЭМО. Алматы: - КазНТУ, 2006.
5. Алиев В. - Visual Basic: – М. СОЛОН-Р, 2002.

КІРІСПЕ	3
1 ТАУ-КЕН МАШИНАЛАРЫН ПАЙДАЛАНУ КЕЗІНДЕГІ ИНЖЕНЕР ҚЫЗМЕТКЕРЛЕРІНІҢ ЖҰМЫСЫН АВТОМАТТАНДЫРУ	4
1.1 Автоматтандырылған жұмыс орнының (АЖО) тағайындалуы.....	4
1.2 АЖО-ның құрылымы.....	6
1.3 АЖО-ның классификациясы.....	
2 ОБЪЕКТЛІ БАҒЫТТАЛҒАН БАҒДАРЛАМАЛАУҒА КІРІСПЕ	8
2.1 Қазіргі қоғамдағы ЭЕМ жаңа маңыздылығы.....	
2.2 Бағдарламалаудың даму тарихы.....	9
2.3 Объектілі бағытталған бағдарламалау.....	11
3 VISUAL BASIC ЖОБАЛАУ ЖҮЙЕСІНДЕ ЖҰМЫСТЫ АТҚАРУ	19
3.1 Visual Basic-ті іске қосу.....	22
3.2 VB жобалау ортасының негізгі панелі.....	29
3.3 Visual Basic-тен шығу.....	
4 VISUAL BASIC БАСҚАРУ ЭЛЕМЕНТТЕРІ	30
4.1 Басқару элементтерін орналастыру.....	30
4.2 Өлшемдерді орнату және басқару элементінің орнын ауыстыру.....	30
4.3 Қасиеттерді орнату.....	31
4.4 Басқару элементінің негізгі қасиеттері.....	32
4.5 Негізгі оқиғалар.....	35
4.6 Стандартты басқару элементтері.....	38
5 VISUAL BASIC-6 ТІЛІНДЕ БАҒДАРЛАМАЛАУ	49
5.1 Тіл алфавиті.....	49
5.2 Айнымалы.....	49
5.3 Константа.....	54
5.4 Арифметикалық өрнек.....	54
5.5 Функциялар.....	55
5.6 Басқарушы операторлар.....	55
6 VISUAL BASIC-6 ЖОБА ҚҰРУ	62
6.1 Жоба құрылымы.....	62
6.2 Жобаларды құру, ашу және сақтау.....	65
6.3 Бірнеше жобалармен жұмыс істеу.....	66
6.4 Жоба тобын іске қосу.....	67
6.5 Жобаға файлды қосу, жою және сақтау.....	68
6.6 Мәтін енгізу.....	69

6.7	ActiveX басқару элементтерін қосу және жою.....	69
6.8	Басқа қосымшаның объектілерін қолдану.....	69
6.9	Ресурс файлдарын қолдану.....	70
6.10	Орындалатын файлды құру және іске қосу.....	71
6.11	Жоба параметрлерін орнату.....	72
7	ҚОСЫМША ИНТЕРФЕЙСІН ҚҰРУ	73
7.1	Қолданушы интерфейсін құру принциптері.....	73
7.2	Пішінді жобалау.....	77
7.3	Пішіннің негізгі оқиғалары.....	79
7.4	Пішіннің негізгі әдістері.....	83
7.5	Қосымша жұмысын аяқтау.....	87
7.6	Енгізуді бақылау.....	89
8	БАҒДАРЛАМАНЫ ЖӨНДЕУ	93
8.1.	Бағдарлама кодын редакциялау.....	93
8.2.	Бағдарламаны редакциялау.....	93
8.3.	Бағдарламаны өңдеу.....	95
9	ӨЗІНДІК БАҚЫЛАУҒА АРНАЛҒАН ТЕСТ СҰРАҚТАРЫ.....	98
	ГЛОССАРИЙ	103
	ӘДЕБИЕТ ТІЗІМІ	109

Нұрпейісова Т.Б.

ТАУ-КЕН МАШИНАЛАРЫН ПАЙДАЛАНУ КЕЗІНДЕГІ ЖҰМЫС ОРНЫН АВТОМАТТАНДЫРУ

Оқу құралы

Республикалық оқу-әдістемелік бірлестігі
оқу құралы ретінде ұсынған

Пішімі 60x100 1/16
Тығыздығы 80 гр./м². Қағаздың ақтығы 95%.
Қағазы офсеттік. РИЗО басылымы.
Көлемі 112 бет. Шартты баспа табағы 7



«Эверо» баспасында басылымға
дайындалды және басып шығарылды
ҚР, Алматы, Байтұрсынұлы к., 22.
тел.: 8 (727) 233 83 89, 233 83 43,
233 80 45, 233 80 42
e-mail: evero08@mail.ru
