ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
«АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ»
коммерциялық емес акционерлік қоғамы
IT-инжиниринг кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ
Кафедра меңгерушісі
PhD, доцент
_____ Т.С. Картбаев
«____» _____ 2019 ж.

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: Компьютер техникасын сату бойынша интернет магазинның программалық қамтамасын әзірлеу

Мамандығы: 5В060200 – «Информатика»

Орындаған: Садулла Н. Тобы: ИНФк-15-1
Ғылыми жетекші: аға оқытушы Айткулов Ж.С.

Кеңесшілер:
Экономикалық бөлім: аға оқытушы _____ С.К. Тулегенова
«22» 05 2019 ж.

Өміртіршілік қауіпсіздігі: т.ғ.д., аға оқытушы _____ Ш.Ш. Бекбасаров
«07» 05 2019 ж.

Есептеу техникасын қолдану: аға оқытушы _____ Ж.С. Айткулов
«22» 05 2019 ж.

Норма бақылаушы: аға оқытушы _____ К. Мукапил
«27» 05 2019 ж.

Сын-пікір беруші: т.ғ.к., асс. профессор _____ Ф.Н. Абдолдина
«____» _____ 2019 ж.

Алматы 2019

Ұсынылатын негізгі әдебиеттер:

1. Автоматизированные информационные технологии в банковской деятельности. //под ред. Титоренко Г.А. – М.: Финстатинформ. 2000.

2. Культин Н.Б. Основы программирования в C#. – Санкт- Петербург, “БХВ–Петербург”, 2003.

3. Введение в Microsoft Windows 95. – издательство Microsoft 1995.

4. Дантеманн Д. Программирование в среде C#. – Киев, DiaSoft Ltd., 1995.

Дипломдық жобаның бөлімдеріне қатысты белгіленген кеңес берушілер

Бөлімдер	Кеңесшілер	Мерзімі	Қолы
Экономикалық бөлім	Тулегенова С.К.	22.05.2019	
Өміртіршілік қауіпсіздігі	Бекбасаров Ш.Ш.	24.04.	
Программалық қамтама	Айткулов Ж.С.	02.04.2019-22.05.2019	
Норма бақылау	Мукапил К.	04.04.2019 10.05.2019	

Дипломдық жобаны дайындау
КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекшіге ұсыну мерзімдері	Ескерту
Талдау бөлімі	29.10.2018-14.01.2019	орындалды
Жобалау бөлімі	14.01.2019-4.03.2019	орындалды
Жүзеге асыру және тестілеу бөлімі	4.03.2019-22.05.2019	орындалуда

Тапсырманың берілген күні «30» қазан 2018 ж.

Кафедра меңгерушісі _____ Т.С. Картбаев

Жобаның ғылыми жетекшісі _____ Ж.С. Айткулов

Тапсырманы орындауға алған
білім алушы _____ Н. С. Садулла

Аңдатпа

Дипломдық жобада интернет магазинның компьютер техникасын сатушысының жұмысын автоматтандырудың комплекстік жүйесі қарастырылған. Жұмыста сатып алушы мен компьютер техникасы түрлері туралы деректер қоры құрылымы көрсетілген.

Бұл бағдарламаның тиімділігі сатушының жұмыс орнын автоматтандырады соның арқасында жұмысты жеңілдетеді және уақытты үнемдейді. Бұл бағдарлама Visual Studio ортасында жасалып, деректер қоры MySQL көмегімен жеке файл болып жасалған.

Бұл дипломдық жоба кіріспеден, пәндік аймақты зерттеу, өңдеу құралдары, жобалау, қолданбалы, экономикалық есептеулер және еңбек қорғау бөлімдерінен тұрады.

Аннотация

В дипломной работе предусмотрена комплексная система автоматизации работы продавца компьютерной техники интернет магазина. В работе представлена структура базы данных о покупателях и типах компьютерной техники.

Эффективность этой программы автоматизирует рабочее место продавца, благодаря чему облегчает работу и сэкономит время. Это приложение создается в среде Visual Studio и создан отдельный файл с помощью базы данных MySQL.

В данном дипломном проекте предусматривается введение, изучение предметной области, средства разработки, проектирование, прикладные, экономические расчеты и охрана труда.

Annotation

The thesis provides a comprehensive system of automation of the seller of computer equipment online store. The paper presents the structure of the database of buyers and types of computer equipment.

The effectiveness of this program automates the workplace of the seller, making it easier and save time. This application is created in Visual Studio and a separate file is created using the MySQL database.

This diploma project provides for the introduction, study of the subject area, development tools, design, applied, economic calculations and labor protection.

Мазмұны

Кіріспе

Компьютерлендіру қазіргі уақытта кеңінен таралғандықтан, қажетті ақпаратты табу қиын мәселе. Бұл тұрғыда коммуникациялық бизнестегі қажетті компьютерлік технологиялар туралы ақпаратты жылдам іздестіру мәселесі бар. Деректерді жедел дайындау және іздеу үшін автоматтандырылған ақпараттық жүйелер қажет.

Бағдарламалық қамтамасыз етудің ақпараттық жүйесінде клиенттік деректер базасына негізделген серверлік өнім болуы керек. Осылайша сіз файлдық серверде және клиент серверінің ортасында жұмыс істей аласыз.

Бұл кешен әртүрлі байланыс кәсіпорындарында сатушылардың жұмысын автоматтандыру үшін әзірленген.

Енгізілген деректер жүйеге қолмен енгізілген және деректер кестелері бастапқы құжаттар ретінде ұсынылуы керек. Байланыс саласындағы бизнесте компьютерлік технологиялар базасы құрылды. Бұл бағдарлама компьютерлік технологиялар туралы ақпаратты және оқырмандар туралы мәліметтерді, сондай-ақ енгізілген деректерді басқаруға негізделген. Шығыс құжаттары бастапқы деректер негізінде жасалады. Менің бағдарламам қоршаған ортада C # болып табылады және MySQL дерекқоры – бұл файл.

Бұл әрекетті орындау үшін пайдаланушыға тақырыпты жақсы біліп, орта есеппен дербес компьютермен Windows ортасында жұмыс істеуі керек екенін есте ұстау қажет.

Бағдарламаны орнату кезінде күрделі тестілеу жүргізіледі. Бағдарламаның дұрыс жұмысы, орнату барысында техникалық және бағдарламалық құралдарды қалпына келтіру. Қолданбалар мен қажетті өзгертулер жүйенің жұмысы жарияланған сәтте анықталады.

Бұл бағдарлама әрбір сатып алушының таңдауына байланысты тиісті компьютерлік техникадағы қажеттіліктеріңізге жауап береді. Бағдарлама компьютерлік техниканы, деректер базасынан провайдерлердің деректерін алады. Ақпараттың тиімді пайдаланылуы ұйымның жалпы жұмысын қамтамасыз ету үшін маңызды.

Ақпаратты өңдеу саласында технологияларды енгізу өнеркәсіптік революция басында стандарттар мен жинау конвейерінің нәтижелерімен үйлесімді өнімділіктің артуына әкелді. Ақпараттық қоғамда бәсекеге қабілетті емес ескі технологияларды қолданатын ұйымдар секілді ақпараттық технологияларды пайдаланбайтын ұйымдар бәсекелестік артықшылыққа ие емес.

Ақпараттық технологиялар технологиялардың тізбесіне біріктірілген әдістердің, технологиялық процестердің және бағдарламалық технологиялардың жиынтығы болып табылады. Олар ақпаратты жинау, өңдеу, сақтау және таратуды қамтамасыз етеді. Олар сондай-ақ ақпараттық ресурстарын пайдалану процестерінің жүктемесін азайтады және олардың сенімділігі мен жылдамдығын арттырады.

Жоғары сапалы ақпарат, яғни маңызды, уақтылы және нақты ақпарат, сапалы шешім қабылдаудың міндетті шарты болып табылады. Ұйымның жұмысына әсер ететін барлық параметрлер бір-бірімен байланысты екендігі белгілі. Компьютерлерді пайдалану ұйымдық құрылымды өзгертуге әкелуі мүмкін. Бұл жұмыс басқарушы-клиенттік-серверлік технологияларды компьютерлендірудегі жаңа технологияларды қолдануға әкеледі. Жұмыстың мақсаты коммуникациялық дүкен үшін автоматтандыру жүйесін құрастыру болып табылады.

Автоматтандыру процесі – белгілі бір салада түпкі пайдаланушылардың жұмысын ұйымдастыруға мүмкіндік беретін әдістемелік, лингвистикалық, техникалық және бағдарламалық құралдар жинағы.

Жаңа автоматтандырудың және ақпаратты өңдеудің негізгі артықшылықтары бағдарламаланған шешімдердің орындалуына немесе үлкен көлемді міндеттердің мақсатына немесе таза механикалық жұмысқа қайта оралуға болатын қайталанатын есептеулерде. Мұндай есептерді көбісі шығармашылық немесе ерекше деп санайды. Компьютерлік технология әрбір шығармашылық үдерісті жеделдетуге мүмкіндік береді.

Компьютерлік технологиялар мен жаңа ақпараттық технологияларды қолданатын адамдардың дамуы әдетте шығармашылықты дамытудың нәтижесі болып табылады, өйткені ол әртүрлі нұсқаларды қарастырады.

Кәсіпорынды автоматтандыру процесі техникалық жұмыс орындарын өңдеу, бизнес процестерді жобалау және анықтау, деректер қорын құру, пайдаланушылық интерфейсті кодтау, автоматтандыру жүйелерін сынау және қайта құру секілді кезеңдерден тұрады. Автоматтандыру жүйесін одан әрі дамыту жүйені талдау мен қолдауды қамтамасыз етеді. Жүйенің жұмыс істеуі барысында кәсіпорынның бизнес-құрылымының өзгеруіне байланысты автоматтандыру жүйесін қайта құру қажет болуы мүмкін.

Клиент-серверлік технологияларды қолдану шағын және орта кәсіпорындарда автоматтандыру жүйесін құрудың әмбебап әдісі болып табылады. Қазіргі уақытта біз компьютерлік революцияның куәсі болдық. Миллион долларға толы әмбебап компьютерлер компьютерлік желілермен алмастырылады, олардың құны мыңға жуық. Бұл компьютерді ұлғайтудың нәтижесі. Кәсіпкерлікті қайта құрылымдаумен айналысатын компаниялар ұйымдық секторда кеңеюде. Олар орта деңгейдегі әкімшілікті осы өзгерістерге бейімдеп, алдыңғы деңгейге жақындататын шешімдерді жүзеге асырады. Кем емес адамдармен іскерлік қарым-қатынас жасайтын ұйымдар өкілеттіктерді бөлуді ұтымды етуге мәжбүр.

Жеке тұлғалар мен командалар арасындағы өкілеттіктердің таралуына жәрдемдесу құралы – ұжымды және әрбір адамды басқара алатын компьютерлік жүйе. Дегенмен, үлкен компьютерлер арзан кішкентай компьютерлерді орталықтандырылған басқарудың орнына кеңейту, үлкен комьютерлерді жергілікті командалардың және белгілі бір адамдардың қажеттіліктерін қанағаттандыру үшін өзара әрекеттесетін шағын жүйелермен ауыстыруға арналған. Бұл процесс жергілікті үдерістер мен өзін-өзі

басқаратын командаларды осы процесс орталығынан басқаратын ұйымның мәдениетінің өсуі. Нәтижесінде, орталықтандырылмаған шешімдерді қолдайтын және өнімнің сапасына және пайдаланушының қажеттіліктеріне тез арада көрсетуге қабілетті мамандарды басқаратын компьютерлік жүйелерді пайдаланамыз. Бұл 90-шы жылдардағы клиент-сервер технологиясының төңкерісі.

Компьютерлендіру арқылы пішіндерді толтыру және қызметкерлердің контингентіне байланысты есептеу нәтижелерін жасау кітапхана іс-әрекетін мүмкіндігінше жеңілдетуге мүмкіндік береді. Нәтижесінде сіз аналитикалық жұмыстың бос уақытын көбейтесіз, тұтастай алғанда жұмыстың сапасын жақсартасыз және сіздің жауапкершілігіңізді жоғарылатасыз.

1 Талдау бөлімі

1.1 Байланыс магазиндерге қойылатын талаптарды анықтау

Компьютерлік технологияға деген сұраныстың артуына байланысты коммуникациялық бизнестегі компьютерлік техникаға тапсырыс беру туралы ақпарат жылдам ізденуде. Есеп келесі жағдайларда жасалады:

– іздестіру нәтижелерін есеп формасында көрсету;

– компьютерлік технологияны қолдану арқылы мұрағат құру;

– конфигурация провайдері арқылы мұрағат жасаңыз;

– сұратылған ақпаратты тапсырыс берушіге жеткізу.

Компьютерлік техника мұрағаты туралы барлық ақпарат (компьютерлік технологиялар моделі, әртүрлі қасиеттер туралы ақпарат клиентке есеп (есеп) бар.

– интерактивті түрде жұмыс істеу үшін келесі шарттарды орындаңыз:

– файлдағы сұрауға жауап 1-2 секунд ішінде қайтарылуы керек;

– ақпаратты енгізуді азайту; реттелетін процестер және жұмыс істеу қадамдары;

– динамикалық ақпаратпен барынша автоматтандыру;

– қолданбалы қателер туралы ақпаратпен жұмыс қадағалау.

Бағдарламалық қамтамасыз етудің ақпараттық жүйесінде клиенттің жазбасына негізделген серверлік өнім болуы керек. Осылайша сіз файлдық серверде және клиент серверінің ортасында жұмыс істей аласыз. Клиенттің интуитивті графикалық пайдаланушылық интерфейсіне және контекстке сезімтал қолданушыларға арнайы IT білімі жоқ шиыршық бағдарламаларына қол жеткізуге мүмкіндік береді. Кешенді қарым-қатынас дүкен ассистентін пайдалануға бағытталған.

– кешеннің тұрақты жұмысын қамтамасыз ету үшін:

– енгізілген деректерді ықтимал мәндер ауқымында тексеру қатесі;

– кесте құрылымына өзгерістер жиынтығын тексереді;

– компьютерде істен шыққан жағдайда ақпаратты қалпына келтіру;

– қателер туындаған жағдайда бағдарламаны тоқтатпаңыз;

– файлда қате пайда болса, деректерді дұрыс қайта қосу үшін сұрауды аяқтауыңыз керек.

Жүйе әр қолданба нұсқасында тексерілуі керек. Бұдан басқа, әрбір нұсқасы кіріс ақпаратының әр түрлі жиынтығын тексереді. Жүйе өнімділікке жылдам тексеруден өту керек, ал арнайы сынақтар дұрыс бағдарлама нәтижесі болуы керек.

1.2 Кіріс мәліметтерін ұйымдастыру

Дерекқорда енгізілген деректер жүйеде қолмен енгізілген және деректер кестелері сияқты бастапқы құжаттар ретінде ұсынылуы керек. Бұл құжаттарда мынадай ақпарат болуы керек:

а) компьютерлік технологиялар туралы ақпарат:

1) компьютерлік техника коды;
2) компьютерлік техниканың атауы;
3) компьютерлік модель;
4) сипаттама;
5) баға;

ә) жеткізушілер туралы ақпарат:

1) жеткізушінің коды;
2) провайдердің елі;
3) жеткізушінің мекен-жайы;
4) ұйымның атауы;
5) қандай модельдер жеткізіледі;
6) баға.

Ортақ ақпарат бастапқы құжаттардың электронды нұсқалары, деректер кестелері болуы керек. Кітапхана төменгі жүйе үшін кестелер ретінде ұсынылуы керек.

Пайдалану шарттары: осы бағдарламалық жасақтаманың Windows 8 операциялық жүйесін және одан да көп табысты пайдалану үшін, RAM 8 Мбайт, принтерге қажет. Пайдаланушылар тақырыпты жақсы білуі керек және дербес ортада Windows ортасында жұмыс істей білуі керек.

1.3 Ақпараттық және бағдарламалық сәйкестілік

Кешен жағдайға байланысты салынады. Болашақта басқа блоктардағы ең аз өзгерістері бар барлық бағдарлама блоктары оңай болуы керек.

Бастапқы және аралық файлдар, басқа бағдарламалық өнімдермен бірге, деректер қоймасының әдепкі деректер кестелерін пайдалану керек.

1.4 Бағдарламалық құжаттарға қойылатын талаптар

Тиісті бағдарламалық өнім келесі құжаттармен бірге жіберіледі:

- спецификациялар;
- техникалық жоба;
- бағдарлама мәтіні;
- бағдарламаның сипаттамасы;
- пайдаланушы нұсқаулығы.

1.5 Мәліметтер қорын ұйымдастыру

Бағдарламалаушы тоқтау нүктесінің сипаттамасын өзгерте алады. Мұны орындау үшін, көрініс мәзірінің келесі деңгейіндегі BreakPoint пәрменін таңдаңыз. BreakPoint тізімінің тілқатысу терезесінде жолда тінтуірдің оң жағын басып, мәтінмәндік мәзірде сипаттар пәрменін таңдаңыз. Ақырында, тілқатысу терезесі Сілтеме Вакиппе сипаттары болады.

Бұл диалог тоқтау нүктесінің қасиеттерін өзгертуге мүмкіндік береді. Осы сәтте, мысалы, бағдарламаны тоқтату үшін жағдайды өзгертуге болады (диалог күйінің мазмұны). Тоқтау нүктесінің сипаттарын өзгерткеннен кейін ОК түймешігін басыңыз.

Тоқтату нүктесін жою үшін, BreakPoint тізімінің тілқатысу терезесінде нүкте туралы ақпарат жолына тінтуірдің оң жағын басып, шығу контекстің мәзірінен Жою (Жою) пәрменін таңдаңыз.

Сонымен қатар, редактор тоқтау нүктесі код терезесінде тұрған қызыл нүктеге басу керек.

Әсіресе жөндеу кезінде, айнымалы мәннің бағдарламаның кезеңдік орындалуында неге бірдей екенін білуіңіз керек. Бағдарлама ауыспалы механиканың мәндерін бақылауға мүмкіндік береді.

Көрініс мәзірінде бағдарламаның орындалуының орындалу параметрін көру үшін, Windows пәрменін Debug пәрменін және содан кейін мәзірдің келесі деңгейіндегі сағат пәрменін таңдаңыз. Содан кейін, Watch List тілқатысу терезесінде терезеде тінтуірдің оң жақ түймешігімен нұқыңыз және пайда болатын контекстік мәзірде Watch add командасын таңдаңыз.

Соңында, Контроллаған элементтер сипаттары диалогтық терезесі пайда болады, онда Expretion өрісінің мәнін бақылайтын айнымалы мәнді енгізуге және ОК түймешігін басыңыз.

Соңында, айнымалы аты бар жол Watch List тілқатысу терезесіне қосылады. Бағдарлама айнымалы мәндері бағдарлама орындалғанда ғана көрінетін болса, хабар алмасу процесі айнымалы атаудан кейін енгізіледі (жақшаларда).

Редактордың код терезесі бағдарламаны нұсқаулықпен орындаудың келесі қадамында орындалған мәлімдеме көрсетеді. (Егер сіз <F8> пернесін бассаңыз немесе «Іске қосу» мәзірінде «Қадам» пәрменін таңдасаңыз, «Көру тізімі» тілқатысу терезесі бағдарлама параметрінің мәнін көрсетеді.

Әрбір бағдарлама нақты өмірдің бөлігін бейнелейді, ол сол элементтің деректері ретінде ресімделеді. Үлкен деректер массиві, әдетте, бағдарлама кодын, дерекқорды, ұйымдастырылған және орналастырылған сияқты орындалады. Бағдарламалық қамтамасыз етуді пайдалана отырып деректермен жұмыс істеуге арналған деректер базасын басқару жүйелерінен (ДББЖ) бастап 60-шы жылдары. ДББЖ әрбір мұрагерге жауапты:

- деректер сипаттары және сіздің физикалық жағдайыңыз;
- деректерді іздеу;
- деректер базасының мазмұны жаңартылды;

– деректерді заңсыз пайдаланудан және рұқсатсыз кіруден қорғау;

– бір уақытта бірнеше пайдаланушыдан реттелетін қызмет сұраулары.

Клиент-сервердің сәулетінде ДББЖ пайдаланушыларға желідегі сұрауларды қабылдайды және нәтижені желі арқылы қайтарады. DBMS дерекқор сервері деп аталады.

Тақырыпты талдау. Оңтайлы есепке алу келесі параметрлермен анықталады:

– компьютерлік технологиялардың саны, компьютерлік технологиялар бір жерде болады;

– бір шкафта бірнеше полка болуы мүмкін, кез келген сандар болуы мүмкін, мысалы: «2» немесе «5»;

– компьютерлік инжиниринг шифры, барлық төлемдер, мысалы «25» немесе «39»;

– компьютерлік техниканың моделі. Мән барлық символикалық ақпаратты қамтиды;

– үлгі деректер.

МҚ үлгі деректерінің құрылымы оның базасында байланыстырылғандығын анықтайды. Бірнеше МҚ модельдері бар, мысалы: графикалық модель, желілік модель, және т.б. Қазіргі кезде ең көп таралған реляциялық деректер моделі. Реляциялық әдісдегі деректер екі өлшемді кестелер түрінде көрсетіледі.

Иерархиялық модель нысан иерархиясының принципіне сәйкес қалыптасады, яғни бір нысан түрі ең маңызды, ал екіншісі төмендегілерге бағынады. ол «жиынтығында» қатынасқа қойылады, яғни негізгі түрдегі объект бірнеше бағынатын түрлерге ие болады. Балалардың түрлері балалардың түрлері болуы мүмкін. Иерархиядағы (атрибут жиынтығы) ең жоғары түйін түбір деп аталады. Деректер ағаш сияқты ағашқа ұқсайды. Бұл деректер үлгісі өрістер сериясы болып табылатын сегментке негізделген және бұл өрістер осы сегментті сипаттайды. Сегменттер түрі бойынша бөлінеді, және әрбір түрі деректердің өрістерінде көрсетілген ұзындық пен бөліну арқылы анықталады. Іргелес деңгейде орналасқан екі байланысты сегмент нәтиже (жоғарғы деңгей) және туған (төменгі деңгей) деп аталады. Иерархиялық құрылымда мастер немесе түбір деп аталатын соңғы сегменті жоқ сегмент бар. Бұл сегмент әдетте иерархияның екінші немесе төменгі деңгейлерінің сегменттерінде ашылатын сипаты бар нысан идентификаторы болып табылады.

Осы үлгіні физикалық деңгейде жақсарту үшін дәйекті, индекс тізбегін, тікелей, индекстік-тікелей қатынаудың осы иерархиялық әдістерін қолдана отырып сегменттері бар сақтау құрылғыларына деректерді орнатудың бірнеше стандартты әдістерін пайдаланыңыз. Сегментті орнату әдістеріне сәйкес, олар сізге қол жетімді болатын тәртібін анықтайды. Қазіргі кезде қатысушылардың санын көбейту есебінен қатысушылардың санын көбейту арқасында Интернетке қосылған қатысушылардың саны, қатысушылар санының 50%

астамын құрайды. Бұл модельдің кемшіліктерінің бірі болып саналады. Дене деңгейінде деректердің көп мөлшерін талап ететін «ғасырлық» кезеңді енгізудің күрделілігі ДБ-нің өсуіне әкеледі.

«Бала және негізгі тип» сызығы әрбір деректер түрі бірнеше бала балалар болуы мүмкін және негізгі түрге бірнеше типтерді шақыруға болатын принцип бойынша жасалады.

Желілік деректер моделі - иерархиямен біріктірілген құрылымдық құрылым. Желі түйіндерінің бөлек жазу көшірмелері бар. Жазу түйіндері - DB қатынауының бірлігі. Бұл әр түйінде бірнеше бала түйіндері бар екенін білдіреді, мысалы бірнеше бала түйіндері, бұл құрылым «көп жылдық» хабардың тікелей бейнесін береді. Түйін енгізілімдері арасында байланыс үшін, барлық көшірмелер тізбекте болатын сілтеме көшірмелерінің жазбасы бар. Желілік деректер үлгісінің негізгі құрылымы – бұл сөйлем. Схемада анықталған әрбір жазбаның түрі үшін нақты жазба иесінің жазбасының түрі және жазба мүшесінің жазбаларының кез келген санын көрсету керек. Үкімнің әрбір данасы жазбалардың мүшелерінің бір данасы немесе бірнеше данасынан тұрады.

Әрбір жазба тиісті жазба мүшелерінің көшірмелері арасында әр жазбаны қамтитын көшірмелерімен иерархиялық қатынастарды анықтайды. Бұл сөйлем тек қана бір мүше елтаңбасы жазбада бір данасы болуы мүмкін шектеудің нәтижесі болып табылады. Жазбаның әрбір көшірмесі тиісті мүше жазбасымен және көшірмені әдетте желі схемасында анықталады. Осындай қарым-қатынастарды ұйымдастырудың бір жолы – жазба иесінің көшірмелерінің тізбегін жасау, ол жазбаның барлық мүшелеріне және жазбаның иесінің көшірілген көшірме деректеріне жіберетін сұраулар үшін жоғары өңдеу жылдамдығын қамтамасыз етеді.

Желілік үлгісінің негізгі кемшілігі жад құрылымының күрделілігі болып табылады. Пайдаланушы тізбектердің болуын және болмауын білуі керек.

Нәтижесінде сұрау тілі рәсімдік болып табылады және бағдарламалық білім қажет.

Байланысты деректер моделі. Деректер түрі - бағдарламалау тілінде маңызды болып табылатын тұжырымдама. Барлық қазіргі заманғы дерекқорлар барлық түрлері, ерекше түрлері, символдар және сызықтар, арнайы күнтізбелік деректер түрлері болып табылады.

Домен – бұл қарапайым деректер түріндегі деректердің жиынтығы. домен пайдаланылатын деректер түрі мен деректер үшін екі элементтерін-булева сөз анықталады. Егер осы сөздің нәтижесі шын болса, деректер данасы домен болып табылады.

Әрекет – бұл нақты тақырыптан және денеден тұратын екі өлшемді кесте.

Тақырып – тіркелген атрибуттардың жиынтығы. Бір доменде анықталған тіркелгілердің әрбір, мен атрибуттарды және домен анықтау арасындағы қарым-қатынас бар. Тақырыбы: аяқтау «компьютерлік технологиялар саны», «компьютерлік технологиялар белгілеу,» алған

«қалалық компьютерлік технологиялар», «компьютерлік технологиялар полкі», «күй», «баға», «әдіс төлеу», «күні», «күні». Ол өз домендегі төлсипаты арқылы анықталды. мәселесі органның өзгеріссіз тәуелді болып қалады. хабары орган жинақтарын жиынтығы, әр бір атрибуты-мән жұбы болып табылады. Қатынау 1.1-кестеде көрсетілген болады.

1.1-кесте – Қатынас кестесі

Компьютер техника	Сатушы аты	Компьютер техника орны	Тапсырыс	Коды	Сатушы коды
1	Нурсултан	3/5		25	1
2	Адайбек	5/5		30	22
1	Шыңғысхан	4/9		15	11

Хабардың орындалуы - бұл сандардың саны және хабардың дәрежесі - атрибуттардың саны.

Байланыстың дәрежесі - бұл хабардың тұрақты мәні, ал катаракта өнімділігі өзгереді. Байланыстың күші тағы бір рет түбегейлі санға жатады.

Басқа да кемшіліктер ретінде төмендегілерді айтуға болады:

– бұл пішін жоюға жоғары дәлдікті талап етеді, себебі шығыс сегменттің соңғы сегментін жою нәтижесінде туындаған сегменттерді жояды;

– талап етілген сегментке қатынау нәтижесінде алынған сегменттен ғана пайда болады және МҚ сұрауы үшін жауап беру уақытын көбейтеді;

– желі деректері үшін үлгі.

Жоғарыда келтірілген кестеде ара қатынасы 4, қатынасы дәрежесі - 3.

Күнделіктер осы тұжырымдардың орнына сіздің баламаңызды пайдаланады:

– жинақтау кестесі;

– атрибут бағаны немесе өріс;

– тупловый жазба немесе жол.

Арақатынасы кестедегі графиктердің саны және негізгі сандардың саны – бұл жолдардың саны.

Көптеген сөйлемдердің арқасында классикалық теория бойынша сөйлемдер бірдей элементтер бола алмайды, пропорцияда екі бірдей теңдеу болмауы керек. Сондықтан, бұл қол жеткізу үшін әртүрлі атрибуттар әрдайым бар, олар бірегейлендіреді. Осындай атрибуттардың жиынтығы кілт деп аталады. Кілт келесі талаптарға сай болуы керек:

– бірегей болуы керек;

– бұл функция минималды болуы керек, яғни ол атрибуттың кілтінен жойылуын тоқтатады.

Әдетте, әр пернеге арналған атрибуттардың саны katarash деңгейінен аз, бірақ төтенше жағдайларда барлық атрибуттар болуы мүмкін, себебі барлық атрибуттардың тіркесімі бірегей шартқа сәйкес келеді. Әдетте контактілерде бірнеше кілт бар. Барлық коммуникациялық кілттердің («мүмкін кілттер» деп

аталатын) біреуі басқа кілттерге қарағанда бірінші кілт ретінде таңдалады, бірінші кілт – бұл атрибуттардың ең аз саны бар кілт және ұзақ сызықтық кілттері бар кілттерге ие болу мүмкіндігі аз пайдаланыңыз.

Іс жүзінде, бірінші негізгі арнайы сандық атрибуты, оның құны ДҚБЖ механизмі арқылы анықталады арнайы агенттер немесе триггер қалыптасады авто-қадамдық дала болып табылады.

Қатынастарды қалыпқа келтіру. ТТ концептуалды, логикалық дизайн. Жоғарыда аталған деректерді жаңарту ауытқуларының алдын алу үшін, DB үлгісі тәуелсіз қатынастардан қалыптастырылуы тиіс.

Бұл мәселені шешу үшін, кодталған қарым-қатынасты қалпына келтіру құрылғысын ұсынды. Қалпына келтіру – бұл жад аномалиясын жою үшін хабарларды және атрибуттарды тексеру мен өзгерту процесі. Қалпына келтіру процесі деректер құрылымын стандартты пішіндерге тұрақты түрде қалпына келтіруге мүмкіндік береді. 6 қалыпты түрі бар:

- бірінші қалыпты пішін (1NF);
- екінші қалыпты түрі (2 NF);
- үшінші қалыпты нысаны (Z NF);
- қалыпты түрде Воусе коды;
- төртінші қалыпты үлгі (4 NF);
- бесінші қалыпты пішін (5 NF).

Қалыпты нысандар функционалдық мақсаттылықты түсінуге негізделген. Функционалдық тәуелділік (FT) байланыс байланыс құны болып табылады, және атрибут арасындағы қарым-қатынастар, әрбір құнынан бір ғана осы атрибут туралы айтты. А атрибуттары детерминант деп аталады. Детерминанттарды құруға болады, яғни жеке төлсипаттар емес, екі немесе одан да көп атрибуттардан тұратын топтар (топтар).

Функционалды тәуелділіктің 3 түрі бар: толық, оқылатын және өтпелі.

А элементіндегі атрибут бірнеше бөліктерге қарамастан, анықтамалық атрибуттарға толық функционалды тәуелділік деп аталады.

Егер В компонентінің детерминанты тек қана бөлігіне байланысты болса, ол бөлінген функционалдық тәуелділік деп аталады. А-В және В-С жағдайы атрибуттары А, В және С жүзеге асырылады және ешқандай кері тәуелділік бар болса, мұндай функциясы транзит деп аталады. Transitive функция тәуелділігі - толық тәуелділіктің үйлесімі.

Аномалиялар бөлшектелген және уақытша функционалдық тәуелділіктерге әкеледі. Сондықтан, кез келген нәтижесінде қатынасы ғана толық, функционалдық тәуелділік, сіздің біріктіру және сіздің біріктіру өзара қарым-қатынастарда өзгеруіне осы қарым-қатынас өзгерісті бар қарым-қатынастардың өзгеруі, сіздің біріктіру осы қарым-қатынас өзгеруі, өйткені.

Теория тұрғысынан алғанда қалпына келтіру процесі тақырыптың барлық атрибуттарын қамтитын қарым-қатынастан басталуы керек. Ол үлкен кесте ретінде көрсетілуі мүмкін. Әлбетте, осы кестедегі деректер қайталанады.

Кез-келген қалыпты пішін бұрынғы қалыпты пішін талаптарына толығымен сай келеді және қосымша талаптар (ұсыныс) ұсынады.

1-қалыпты пішін. Алғашқы қалыпты пішіндегі қатынас бір мезгілде, егер оның барлық атрибуттары атомдық мәндері болса.

2-қалыпты пішін. Кәдімгі 2-ші формаға сәйкес келу үшін, функция тұрғысынан тәуелділіктердің жоғарылауы болмауы керек, яғни әрбір негізгі емес атрибут дайын кілттің бөліктеріне емес, өзіне тәуелді болуы керек.

Қалыпты 3-ші пішін транзиттік тәуелділікті жоюды талап етеді. Іс жүзінде, бұл қарым-қатынастың негізгі сипаты арасындағы тәуелділік болмауы керек. Бала кодексінің қалыпты формасы детерминант ықтимал (ықтимал) кілт болуды талап етеді.

4-қалыпты пішін. Атрибут арасында көптеген тәуелділіктер болмауы керек.

Реляциялық модель МҚ-ны жобалау үшін өте қолайсыз. Оның орнына, әртүрлі семантикалық үлгілер пайдаланылады, олардың ішінде ең көп таралған ER үлгісі немесе «нысанға байланған» үлгі. ЕП моделінің негізгі компоненттері зат, байланыстырушы және атрибут болып табылады.

Нысан – бұл шындық немесе деректер базасында ол туралы ақпаратты сақтайтын нысан. ER үлгісінің диаграммасында нысан тіктөртбұрыш түрінде көрсетіледі және объектінің атауы жазылады.

Атауы – семантикалық мән. Нысанның түрін және объектінің көшірмесін ажырату қажет. Нысанның атауы нысанның данасына емес, оның түріне тағайындалмайды. Атаудың данасы – бұл түбірлік сөздер, заттар, ойлар және т.б. Байланыс дәрежесіне қосымша, ол одан да көп сипаттамаларға ие, сізде өмір бойы (тұрақты, көп және қысқа мерзімді) және таңдауға (міндетті, міндетті емес, мүмкін шартты) ие боласыз.

Атрибут (меншік) – бұл нысан сипаты. Бұл анықтаманы, сәйкестендіруді, жіктеуді, сандық сипатты, заттардың күйін білдіретін құрылым элементі. Мысалы, «компьютерлік техниканы сатып алу» тақырыбының атрибуттары, компьютерлік жабдықтардың саны, сатушының атауы және т.б. Жобалау үдерісі кезінде жобалау үдерісі үш негізгі фазаға бөлінген: тұжырымдамалық, логикалық және физикалық.

Техникалық іске асыру шарттарына тәуелсіз бизнес-ақпараттық модель құру үшін ДБ әдістерінің тұжырымдамалық жобасы. ДБ дизайнының тұжырымдамалық кезеңі іске асырудың әрбір бөлігінен толықтай тәуелсіз компания арқылы тұжырымдамалық ақпараттық модельді құрудан басталады.

Деректер моделіне негізделген корпоративтік ақпараттық модельді жобалау үрдісі, қолданылатын ДҚБЖ мен басқа физикалық іске асыру шарттарына қарамастан.

Осыған орай факт фактісі фактісі себебінен қазіргі кезде фактіге байланысты факт болып табылатындығына байланысты бұл адам өмірінің қауіпсіздігін қамтамасыз ету үшін адам өмірінің қауіпсіздігін қамтамасыз ету қажет. Бұл дизайнерге деректермен жұмыс істеудің әртүрлі аспектілерін жан-жақты талдауға мүмкіндік береді, бұл ең тиімді жобалық шешімді таңдауда

өте маңызды. Физикалық дизайн МҚ – екілік жадқа орналастырылған МҚ нақты іске имиджін жасау процесі. Ақпаратқа барынша тиімді қолжетімділікті жасау үшін, деректерді сақтау құрылымын және қол жетімділік әдістерін көрсетуді қарастырыңыз. Сондықтан, қолданылатын ДҚБЖ-нің барлық ерекшеліктерін ескере отырып, физикалық дизайн міндетті болып табылады. Кері байланыс әрқашан физикалық және логикалық дизайн фазалары арасында болады, себебі жүйенің жұмысын жақсарту үшін жобалау кезеңінде шешім физикалық, логикалық деректер модификациялары арқылы түзетіледі, бұл қайта қарауға әкелуі мүмкін. Концептуалды ДБ жобасы, мәміле талаптары:

– компьютерлік технология туралы ақпаратпен жазбаларды жасау және редакциялау;

– компьютерлік модельді іздеу;

– кодты іздеу;

– жыл бойынша басылым бойынша іздеу;

– компьютерлік нұсқаулық туралы ақпаратты қамтитын жазбаларды жасау;

– тапсырысқа енгізілген компьютерлік техника туралы ақпаратты қамтитын жазбаларды жасау;

– алынған компьютер технологиясы туралы ақпаратты қамтитын жазбаларды жасаңыз.

1-қадам. Жергілікті тұжырымдамалық деректер үлгісін жасаңыз. Атап айтқанда, жергілікті деректер тұжырымдамалық модельді құрастырған кезде, қолданыстағы техникалық сипаттамалармен осы үлгінің әр түрлі құрамдастарын жасауыңыз қажет. Жасалған әрбір деректер үлгісі келесі құрамдастардан тұрады:

– зат түрі;

– Binding түрі;

– пропан;

– домендерін төлсипат;

– мүмкін кілттер;

– Kernel бірінші кілті.

2-қадам. Заттардың түрлерін анықтау. Қолданыстағы спецификацияларға негізделген заттардың маңызды түрлерін анықтау арқылы бастайық.

3-қадам. Сілтеме түрлерін анықтаңыз. Жеке нысандар арасындағы қатынастардың түрлерін анықтайды. Жалпы алғанда, бұл қатынас этишпен бірге етістік немесе етістік ретінде анықталады. «1» (1: 1), «pro» (1: M) немесе «көп» (M: M) болып табылатын байланыстардың 3 негізгі түрі бар.

Қосылыс флоат нөмірлері бар. Ерекшелікте бұл қатынас келесі сөзбен анықталады: «Әрбір шкафта бірнеше сөрелер бар». Бұл қосылыстың түрі 1: M.

Енді біз Float Has State комбинациясына немесе «Бөлмелер мен мемлекет» деп аталатын бөлмеге бөлмелер мен мемлекет арасындағы

байланыс ретінде сене аламыз. Бірақ, мен «Тақырыптық сілтеме» диаграммасында байланыс деп ойлаған кезде, оның түскі жәшігі үлкен болғандықтан, мен 1-түрді таңдадым.

Сондай-ақ, басқа қосылымдарды тексеріңіз. 1-қадам. Атрибуттарды анықтап, оларды заттардың түрлерімен байланыстырыңыз.

Таңдалған атрибуттар және олардың тиісті заттармен байланысы туралы мәліметтерді 1.2-кестеде табуға болады.

1.2-кесте – Атрибуттардың домендері туралы ақпарат

Доменнің аты	Доменнің сипаттамасы	Мүмкін мәдерінің мысалы
	Кез-келген бүтін типті сандар	1
	Ұзындығы 10 символ болатын	2/5
	Жол	
	Ұзындығы 20 символ болатын	
	Жол	
	Ұзындығы 20 символ болатын	
	Жол	
	Ұзындығы 20 символ болатын	

2-қадам: атрибут домендерін анықтау домен – бір немесе бірнеше атрибуттар үшін ықтимал мәндердің жинағы.

Қадам 3. Ресми ережелерді қолданып деректер үлгісін қарап шығыңыз. Бұл қадамда қалыптасқан қарым-қатынасты формальдылық рәсімінің барлық талаптарын қанағаттандыру үшін қайта қарау қажет. 1 (1NF) стандартты пішінді келтіріңіз, ол қайталанатын атрибуттар жиынтығын IBM Cognos Conference қатынастарынан жоюға мүмкіндік береді:

– бірінші төлсипат кілтіндегі толық емес тәуелділікті жою үшін 2-пішін үлгісін (2NF) анықтаңыз;

– бірінші кілтке өтпелі тәуелділікті жою үшін атрибуттарды 3 қалыпты пішінге (ZNF) қойыңыз;

– IBM® Cognos аномалитикасы функционалды тәуелділіктердің ауытқуларын Boys кодын қалыпты пішініне шығаруға мүмкіндік береді.

Әрбір қарым-қатынастың кем дегенде кәдімгі ұлдар кодексі (NFBC) түрінде болуын қамтамасыз ету үшін осы қатынастар арасындағы функционалдық тәуелділіктерді тексеру қажет. Егер NFBC-те қатынастар анықталмаса, жасалған логикалық құрылымның құрылымы қате немесе толық реляциялық сөйлемді анықтаған кезде дұрыс жасалмаған.

4-қадам. «Объектілі қатынас» диаграммасын жасаңыз. Деректер объектілер (нақты аймақтар), атрибуттар және қарым-қатынастар ретінде ұсынылатын қатынастың қатынас моделі жоғары деңгейлі тұжырымдамалық үлгі болып табылады. Деректер көрінісін шектейді, бірақ нақты емес. МҚ мен

қосымшаларды жасау кезінде арнайы өңдеуді талап ететін байланыстарға ерекше назар аударылады және тақырыптардың өзара әрекетін көрсетеді.

ER симуляциясын есептеу, деректер қасиеттерінің ерекшеліктерін құрастыру. Сондықтан, ER модельдеуде ұсынылған жүйенің бөлігі болып табылатын МҚ бар барлық деректер түрлерінің және атауларының толық сипаттамасы бар.

Тақырып-сілтеме моделінің негізгі түсініктері зат түрлерін, байланыстыру түрлерін және атрибуттарды білдіреді.

ER-модельдеудің негізгі тұжырымдамасы көптеген объектілерді көрсететін нақты әлемде бірдей қасиеттерге ие заттың (субъект түрінің) түрі болып табылады.

Заттың табиғаты тәуелсіз өмірмен сипатталады және физикалық тірі зат немесе физикалық өмір объектісі бола алады. Заттай заттың қатаң формальды айқындамасы жоқ, оны тек қана жұмысқа анықтай аламыз. Бұл дегеніміз, әзірлеушілер түрлі элементтерді бөле алады.

Элемент бірегей пішінде анықталатын зат түрінің данасы болып табылады. Нысанның әр бірегей идентификацияланатын данасы қарапайым зат деп аталады. Кейбір басқа авторлар оны атын (субъекттің пайда болуы немесе мекеме данасы) деп атады.

Әрбір зат түрі атаумен және қасиеттер тізімімен анықталады. Әдетте, МҚ әр түрлі заттарға ие. Маталардың түрлері атрибуттардың бірегей жиынтығына ие болса да, әр элементтің әрбір төлсипаты үшін өз мәндері бар. Заттардың түрлері күшті және әлсіз болып саналады.

Заттардың табиғаты табиғаты басқа субстанцияның табиғатына байланысты заттардың түрі болып табылады.

Күшті заттар - басқа заттың түріне қарамастан оның бар болуы.

Әлсіз заттар кейде үстем үстем (үстем) немесе мастер ретінде бағынышты және күшті деп аталады.

Әр күшті зат зат есім жазылған тікбұрыш түрінде және әлсіз зат түрі – екі жақты контуры бар тіктөртбұрыш түрінде көрсетіледі. Деректер базасының моделін жобалау. Физикалық жобаны жасамас бұрын логикалық дерекқор үлгісін жасау үшін қадамда жиналған қарым-қатынас ақпаратын жақсы меңгеруіңіз және талдауыңыз керек. Бұл ақпарат деректер сөздігіне және DBDL-те қатынастардың анықтамасына қосылуы мүмкін.

Логикалық деректер түріндегі әрбір көрсетілген коэффициент келесі қадамдарды қамтиды:

– хабарлама атауы;

– қапшықтардағы қарапайым атрибуттардың тізімі;

– бірінші кілт және балама (АК) Кесте анықтамасы және сыртқы (FK) пернесі (егер бар болса);

– мінез-құлыққа арналған әрбір шетелдік кілт сілтемесі үлгі талаптарын анықтаңыз;

– деректер сөздігі әрбір атрибут үшін келесі ақпаратты қамтуы тиіс;

– атрибут доменін анықтау, деректер түрі, ұзындығы атрибут, және рұқсат етілген мәндер үшін қажетті шектеулерді көрсетеді;

– үнсіз атрибуттан алынған мәндер (қосымша);

– тізімдегі көрсетілген атрибут үшін нөлге рұқсат етіңіз.

2 Жобалау бөлімі

Бұл дипломдық жобада байланыс бизнесі жұмысының техникалық және экономикалық көрсеткіштерін анықтау жүйесі құрылады. Бұл жүйені құру үшін сіз алдымен дұрыс жобалауыңыз керек. Қазіргі уақытта жасалған жүйені жобалау үшін UML тілін (Unifed Mideling Language) пайдалану өте тиімді. Бұл UML құралдары сізге тұжырымдамаларға негізделген объектілі-бағытталған және стандартты негізделген жүйелерді жобалауға және құруға мүмкіндік береді, себебі сіздің басты мақсатыңыз дизайнды жобалау стандартымен қамтамасыз ету болып табылады. Жобаның басында алдымен объективті аймақты анықтау керек. Біз пәндік сала ретінде коммуникациялық бизнестің деректерді өңдеу аймағын аламыз.

2.1 Прецеденттер диаграммасын құру

Алғашқы фазада (алғышарттың фазасы) бастапқы басымдық диаграммасы (пайдалану жағдайлары) көрсетіледі. Бұл схема жүйеге қойылған тапсырмаларды анықтайды (олардың функциялары жұмыс барысында жүзеге асырылады).

Басымдылық диаграммаларын жасаған кезде алдымен барлық негізгі таңбалардың тізімі (жеке немесе сыртқы жүйе) жасалады. Олар келесі сұрақтар бойынша анықталады:

- жүйені тікелей пайдаланатын;
- ол жүйенің жұмысына жауапты;
- IBM жүйесінде қолданылатын сыртқы құралдар;
- осы жүйемен өзара әрекеттесетін басқа жүйелер бар ма?

Қолдану жағдайын анықтау үшін келесі анықтамалар қабылданады: әр қолданыс жағдайында жүйенің жауабы ретінде орындалатын белгілі бір функция көрсетіледі және жүйенің нақты пайдаланылуы орындалатын адам мен жүйенің арасындағы диалогты сипаттайды.

Бұған қоса, жүйе талаптарына сәйкес жүйе аудитін, соңғы пайдаланушы әрекеттесуін және пәндік сараптаманы сақтау керек.

Басымдылық диаграммасы қолданбалы нұсқамен бірге актер деп аталатын басқа элементті пайдаланады. Актерлер орындалатын рөлдерді береді, яғни белгілі бір атауды немесе жұмысын көрсетпейді. Басымдылық диаграммалары адам түрінде актерлерді көрсетсе де, сол жүйеден белгілі бір ақпаратты (мысалы, компьютер жүйесі) шығарғысы келетін сыртқы жүйе болуы мүмкін. Орындаушы нақты орындау нұсқасына қажет болған жағдайда, оны диаграммада көрсету керек.

Қазіргі кезде жүйе 5 мыңнан астам абонентті басқарады. Қолдану жағдайын жүйенің актерлерімен бірге пайдаланушының нақты тапсырмаларын анықтау және шешіміңізге баламалы нұсқаларды қарастыру арқылы талдау қажет.

Актерлер қолданыстағы жағдайларда түрлі рөлдерді ойната алады. Сіз жүйенің нәтижелерін пайдалана аласыз немесе сіз оған өзіңіз қатыса аласыз. Актердің рөлі оның қарым-қатынасы қалай пайдаланылатынынан тәуелді.

Сыртқы оқиғалар сәйкестік көзі ретінде пайдаланылады. Ең алдымен, белсенді сыртқы әлемде орын алатын оқиғалар тізбегінен бастау керек. Кейбір нақты оқиғалар пайдаланушыға тигізбестен жүйе әрекетін тудыруы мүмкін немесе пайдаланушы бұл әрекеттерді іске асыра алады. Соққы талап ететін оқиғаларды анықтау бағдарлама параметрін анықтауға көмектеседі.Актерлер мен бағдарлама опциялары арасындағы байланысқа қосымша, «қақпақ» және «кеңейту» қосылымдарының екі түрі қосылады. Қолдану жағдайларының біреуі біріне ұқсас болған жағдайда, бірақ жүктемені көп алады, кеңейтім қолданылады. Пайдалану жағдайларын анықтау.

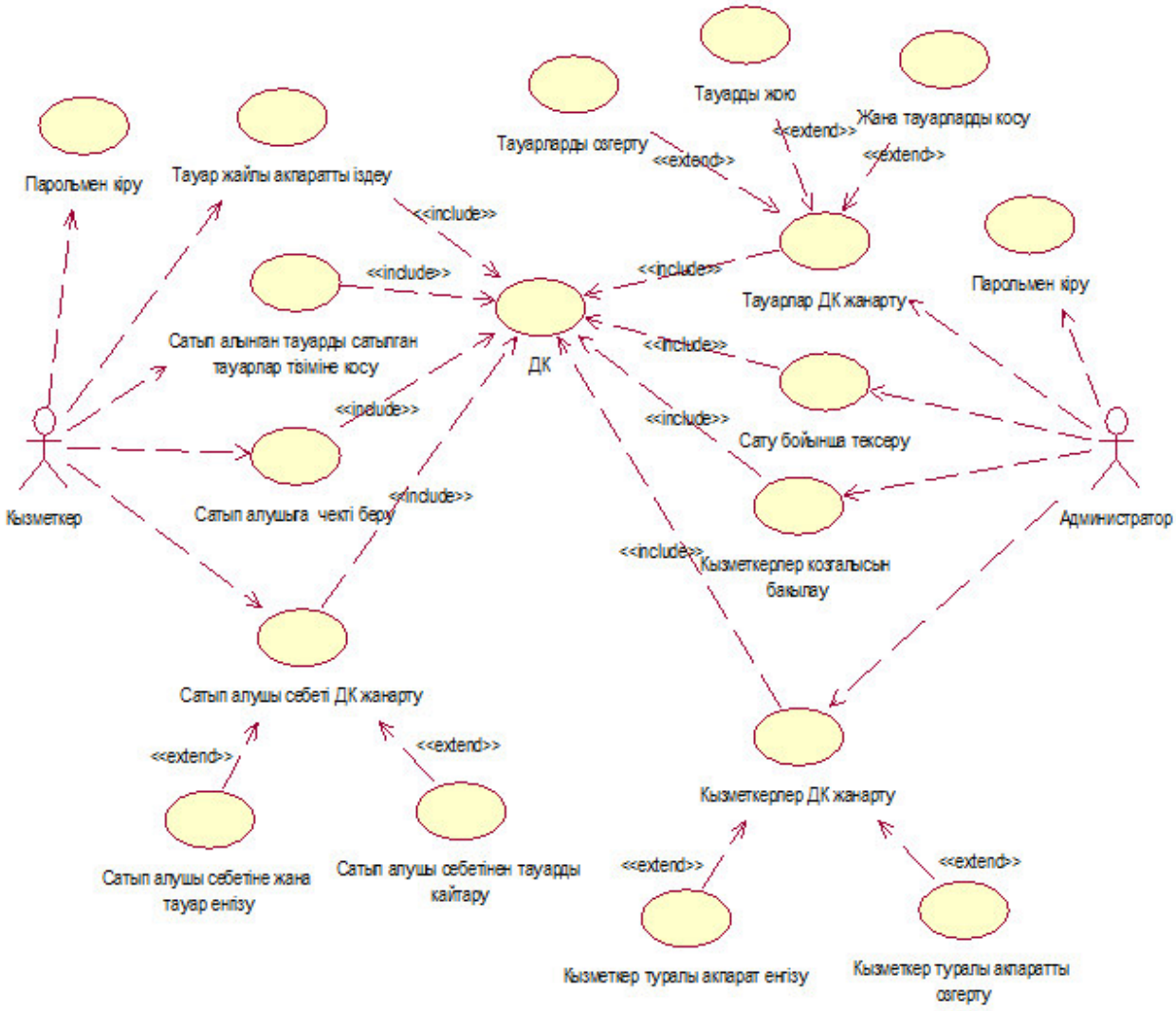
«Компьютерлік техникаға тапсырыс беру», «Компьютерлік техника үшін төлем ақпараттары», «Компьютерлік техниканың түрі», «Модель», «Көлем», «Монитор» компьютерлік техникасын сатып алуға арналған өрістер бар. Өрістер орындалатын рөлдерді көрсетеді, яғни адам немесе жұмыс үшін нақты бір атау жоқ. Қолдану диаграммасында орындаушылар адамдық сандар болып табылады, сол жүйеден нақты ақпаратты алуды қалайтын сыртқы жүйелер болуы мүмкін. Егер орындаушылар нақты орындау нұсқасына қажет болса, оны графикте көрсету керек.

Байланыс жүйесінің жұмыс істеуі арқылы жеткізілетін сыртқы жағдайларды қолданудың кез-келген нұсқасы. Мысалы, егер компьютер техникаға тапсырыс беру файлын қажет етсе, бұл талап орындалуы керек. Опциялардың қолданылуы бір мезгілде пайдаланушының белгілі бір тапсырмаларын анықтауы және шешуге баламалы әдістер беру арқылы жүйе сатыларымен бірге талдануы керек.

Егер суретшінің қосымша параметріне қатысты түрлі рөлдерде ойнасаңыз, нәтижелеріңізді пайдаланыңыз:

- төлем ақпарат;
- тапсырыс берушіге ақпарат беру;
- модельдік компьютерлік технологиялар;
- көлемі;
- салмағы;
- электр қуатымен қамтамасыз етуді бақылау;
- сеанстары жолақ;
- тінтуір;
- нарықтық бағасы.

Басымдылық диаграммасы 2.1-суретте көрсетілген.



2.1-сурет – Басымдылық диаграммасы

2.2 Тізбек диаграммасын модельдеу

Тізбек диаграммасы жаңалықтар бағдарламасының кезектілігін өте жақсы көрсетеді. Графикалық көрініште кесте бойынша дәйектілік диаграммасы көрсетіледі. Уақытты реттейтін хабарларды көрсетеді, олар х бойымен объектілер бойымен және уақыт бойынша реттеледі.

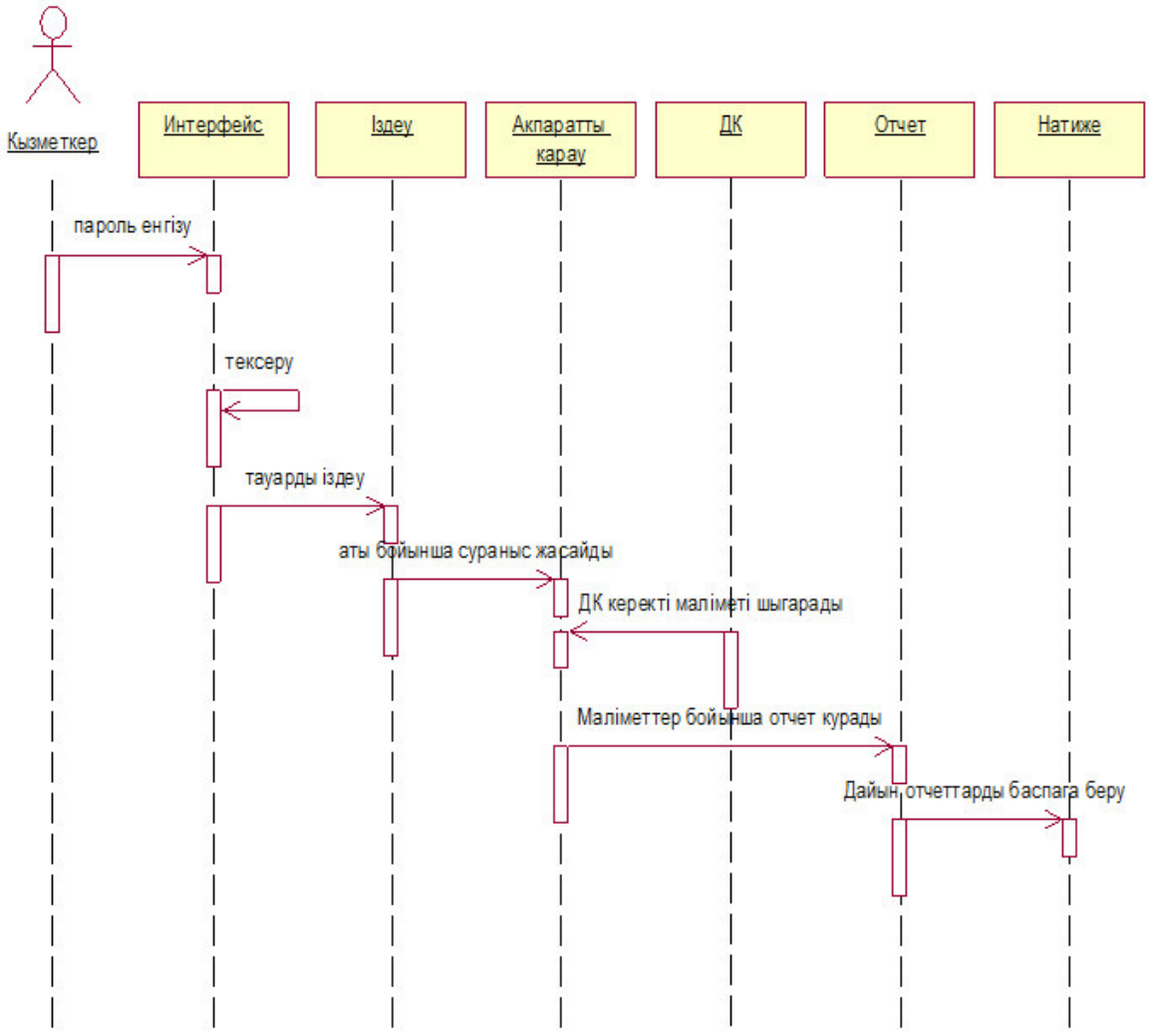
Диаграммаларда объекттің төменгі бөлігіндегі тізбектің тік сызықпен төртбұрыш түрінде бейнеленеді. Бұл сызық объектінің тіршілік ету сызбасы деп аталады және нысанның өзара әрекеттесуі кезінде өмірлік циклдің фрагментін көрсетеді. Бұл нысанды алғаш рет Иван Джекобсон таныстырды.

Екі нысанның арасындағы әр хабарлама сіздің тірі қалжыңдарыңыздың арасындағы нұсқаулар ретінде көрсетіледі. Хабарлар жоғарыдан төменге қарай көрсетіледі. Әрбір хабар кем дегенде хабардың атауы бойынша анықталады. Қажет болған жағдайда, аргументтерді, арнайы басқару ақпаратын қосуға және өзін-өзі қозғалысқа енгізуге болады, яғни нысанның

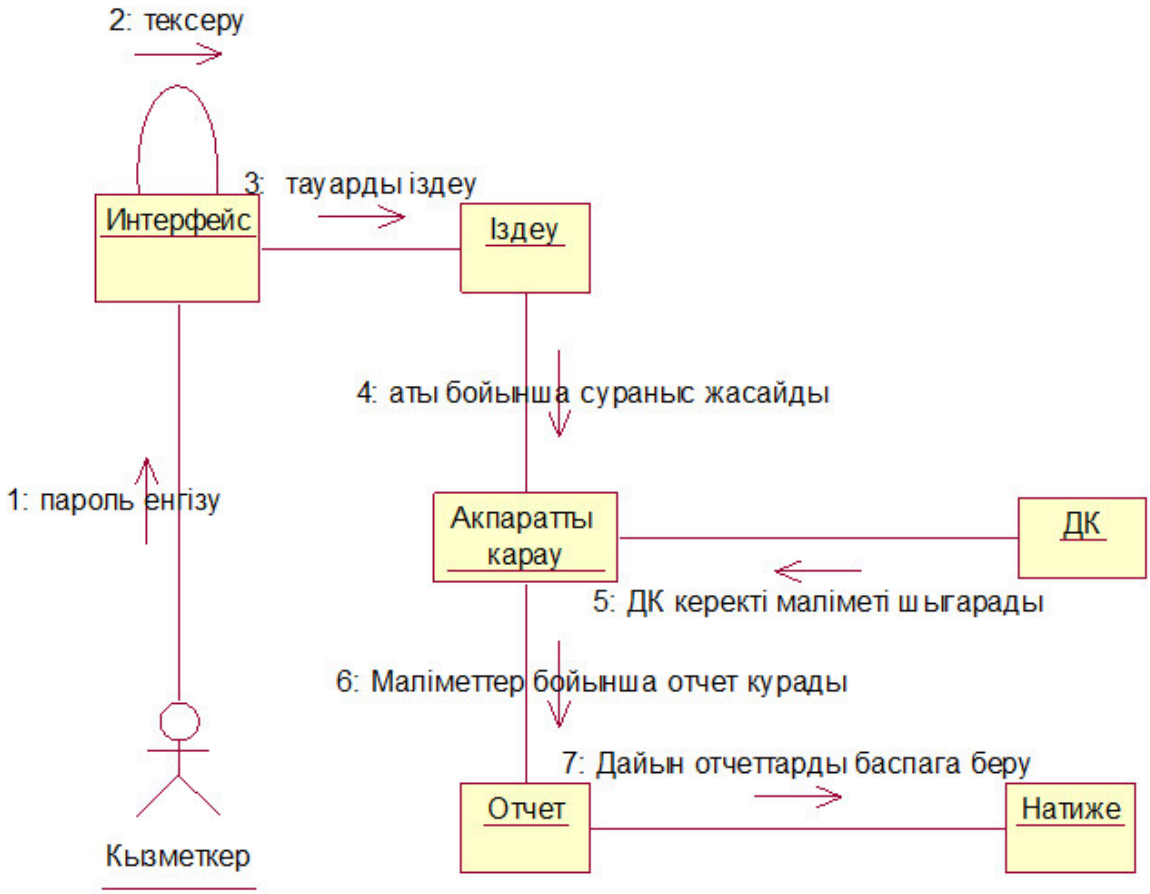
өзі хабар жіберетінін білдіреді, бұл маршрутизатор хабарды тек өмір жолына береді.

Барлық мүмкін болатын әкімшілік деректер келесі екі түрі үшін өте маңызды. Хабардың жіберілу уақытын көрсететін бірінші шарт. Егер бұл шарт орындалса, хабарлама жіберіледі. Екінші пайдалы бақылау – бұл хабардың мақсатты аудиторияға бірнеше рет жіберілгенін көрсетеді иерация белгісі.

Цилиндрдің диаграммалары өте қарапайым және көрнекі (бұл сіздің басты артықшылығыңыз) және төрт-полюс жүйенің процесін жақсы түсінуге көмектеседі. Параллельді процестерді тасымалдау үшін схема схемасын қолдануға болады. Схемалардағы диаграммада оқиғаның дамуы үшін өрістердің бірі пайдалану жағдайының бөлігі ретінде көрсетіледі. Диаграмма өзара іс-қимыл диаграммасында пайдаланылатын диаграмманы нақтылау үшін пайдаланылады. 2.2-суретте тізбектік диаграмма, 2.3-суретте ынтымақтастық диаграммасы көрсетілген.



2.2-сурет – Тізбектік диаграмма



2.3-сурет – Ынтымақтастық диаграммасы

2.3 Кластар диаграммасын құру

Класс диаграммасы (Class Diagram) объектілі-бағдарланған бағдарламалау сыныптарының терминологиясында құрылатын жүйенің моделінің статикалық құрылымын анықтау үшін пайдаланылады. Атап айтқанда, ол жеке субъектілердің (субъектілердің) өзара байланысын көрсетеді, олардың ішкі құрылымын және тартылу сипатын анықтайды.

2.6-суретте сынып диаграммасы шыңдар «жіктеуіш» элементтері болып табылатын және әртүрлі сындарлы қарым-қатынастарға қосылатын графалар ретінде орындалады. Сынып диаграммасының негізінде жүйе архитектурасы жасалады.

Бұл сабақтар тұжырымдамалық болып табылады және «не?» Термині жауап береді. Түпнұсқалық тізім келесідей құрылады:

– сынып, кандидаттар бастапқы деректердің ленеди сипаттамасында;

– зат атауының сипаттамасында ең бірдей сыныптар болуыңыз керек (ескерту: нысан, объект, қауымдастық немесе мән сыныптың кейіпкерлері болуы мүмкін).

Жүйеде кандидаттардың рөлі талданды. Әрбір класс басқа класстармен өзара әрекеттеседі. Әр сыныпқа абстракцияның мазмұнын көрсететін арнайы ат берілді.

Қысқа және семантикалық деп айту қиын болса, бұл сыныптың бөлінбеуінің белгісі. Кез-келген жұптар сіздердің арасындағы байланыстың болуымен анықталады. Қауымдастықтың рөлі сізге тиесілі және сіздің тұтастықпен анықталады. Содан кейін әр сыныптың кейіпкерлерінің тізімі жасалады. Таңбаларды тану процесі ұзақ уақытқа созылмайды, себебі семантикалық кейіпкерлерді кейінірек қосуға болады. Бастапқы ақпаратқа әкелетін семантикалық қасиеттер байқалмайды. Әр сыныпта орындалған әрекеттер (операциялар) анықталады. Міндеттерді анықтаған кезде, төмендегілерді есте ұстаңыз: әр операция қарапайым функцияны орындау керек; тапсырманың атауы функциялардың нәтижесін көрсетуі керек. Мысал операциясы: функционалдық мәнді алу, функционалдық мәнді тағайындау, басқа объектімен қарым-қатынас орнату немесе жою, белгілі бір нысанды жою және т.б.

Класстік диаграмма – нысанға бағытталған көзқарастың орталық бөлігі. Сынып диаграммасы жүйелік нысандардың түрлерін және олардың арасындағы әртүрлі статикалық қатынастарды анықтайды.Статикалық байланыстың екі негізгі түрі бар:

– бірлестіктер;

– жалпылау.

Сынып схемасы сыныптың атрибуттарын, сынып әрекеттерін және нысандар арасындағы шектеулерді көрсетеді.

Класс диаграммасын жасау түрлі көріністермен қаралуы мүмкін:

– сабақтардың диаграммасы зерттелетін территорияның широкомпаниясы өлшемінің тұжырымдамасын көрсетеді. Бұл ұғым оны жүзеге асыратын сыныптарға сәйкес келеді, бірақ мұндай тікелей сәйкестік жиі орын алмайды. Іс жүзінде, тұжырымдамалық модель әлсіз болуы мүмкін немесе іске асырылатын бағдарламалық қамтамасыз етумен байланысты емес, оны іске асыру құралдарынан (бағдарламалық жасақтама тілдерінен) бөлек қаралуы үшін.

– көрініс спецификация деңгейіне түседі, бірақ сыныптарды бағдарламалық жасақтаманы іске асыру емес, тек интерфейстер (интерфейс – сыртынан көрінетін сыныптық операциялар жиынтығы);

– іске асыру моделін іске асыру тұрғысында анықталған сабақ бағдарламалары. Бұл тәсіл бағдарламашылар үшін өте маңызды.

Көзқарас тұжырымдамасы класс диаграммасының құрылуы, сондай-ақ зерделеу үшін өте маңызды. Алайда, осы көріністің арасындағы айырмашылық айқын емес, көптеген әзірлеушілер оларды диаграммада орналастыру мүмкіндігіне ие.

Диаграмма жасаған кезде, сіз бір нүктені таңдауыңыз керек. Егер біз диаграмманы оқып шығатын болсақ, диаграмманы сіздің көзқарасыңыз бойынша дұрыс көрсетуіңіз керек. UML классикалық диаграмма – ресми егін жинау кәсіпорындары корпустың бөлігін, үлгілерді, құрылысты және өте пайдалыдығын талдауда. UML дизайны барлық үш жоспарланған пункттерде қолданылуы мүмкін. Актерлерді тапқаннан кейін және жағдайларды

қолданғаннан кейін жүйе кластарын анықтап, негізгі функцияларды анықтаңыз. Бұл кластардың құрылымы мен функциялары класс диаграммасын жақсы көрсетеді.

Әр сыныптың атрибуттары мен функциялары бар. Төлсипатты анықтаған кезде, төлсипаттың атауы, түрі және мәні диаграмманың механикаландыру деңгейіне байланысты дәйекті сәйкестіктерді пайдалана отырып көрсетіледі. UML синтаксисінде бұл төменде көрсетілген: <visibility> <name>: <type> = <silence>, онда көрінетін-таңба мәні төменде сипатталған әрекеттерге сәйкес келеді. Атрибуттар әрдайым бір мәнге ие. Жалпы алғанда, графика атрибут міндетті немесе міндетті емес екендігін көрсетпейді.

Операциялар сыныптарды іске асыратын процестер. Операциялар мен сабақтарға қолданылатын әдістер арасында матч болып табылады. Бөлшектер тізімінің деңгейінде ол операция түрлеріне қолданылатын жалпы әдістерге сәйкес келеді. Әдетте манипуляция атрибуттарға қол жеткізуге болатын манипуляциялық операцияларды көрсетпейді, себебі олар да қатысады. Кейде бұл атрибут оқуға керек немесе оның өнімділігі өте тұрақты. Іске асыру үлгісі сондай-ақ құпиялылық пен операцияларды қорғау деңгейін көрсете алады. Операцияға арналған толық процессордың синтаксисі келесідей: <visibility> <name> (<parameter list>):<мәнін қайтаратын өрнек түрі> {<property-string>} / мұнда көріну белгісі үш мәннің бірін қабылдай алады:

– «+» (жалпы), «#» (қорғалған) немесе «-» (құпия);

– файлдың атауы символдық серия береді;

– IBM Cognos байланыс параметрлері тізімінде синтаксисті та балар синтаксисіне сәйкес келетін осымша аргументтер бар;

– кестеге мән қайтаратын өрнектің түрі қосымша ерекшеліктерге және нақты бағдарламалау тіліне байланысты;

– сипаттар жолы осы операция үшін пайдаланылатын сипаттардың мәнін көрсетеді.

Сыныптардың мәртебесін өзгертпейтін сыныптар мен операциялар мәртебесін өзгертетін әрекеттер арасында шекараларды орнату жақсырақ. Сыныптан алынған белгілі бір мәнге әкелетін сыныптың күйін өзгертпейтін әрекет сұрау деп аталады. Нысан күйімен байланыстырылған сұрауларды қолдануды анықтай алатын мемлекет басқарылатын нысан күйі деп аталуы мүмкін. Сұраныс пен модификатор арасындағы айырмашылықты ажырату қажет.

Концептуалды тұрғыдан, қауымдастық сабақтардың концептуалдық қатынастарын береді. Әр қауымдастықтың рөлі болуы мүмкін. Рөлге таңбамен бірегей ат тағайындалады. Рөлде қоңырау сипаттары болады және бұл қатынасқа қанша нысан қатыса алатынын көрсетеді.

Жалпы алғанда, жалпы сипат қарым-қатынас жасайтын нысандардың санын төменгі және жоғарғы деңгейден анықтайды. «*» белгісі «нөлдік шексіздік» интервалын көрсетеді.

Практикада жиі қолданылатын таңдау: «1», «*», «0..1» (нөл жоқ, ешқандай). Жалпы алғанда, бір санды қолдануға болады, диапазондардан

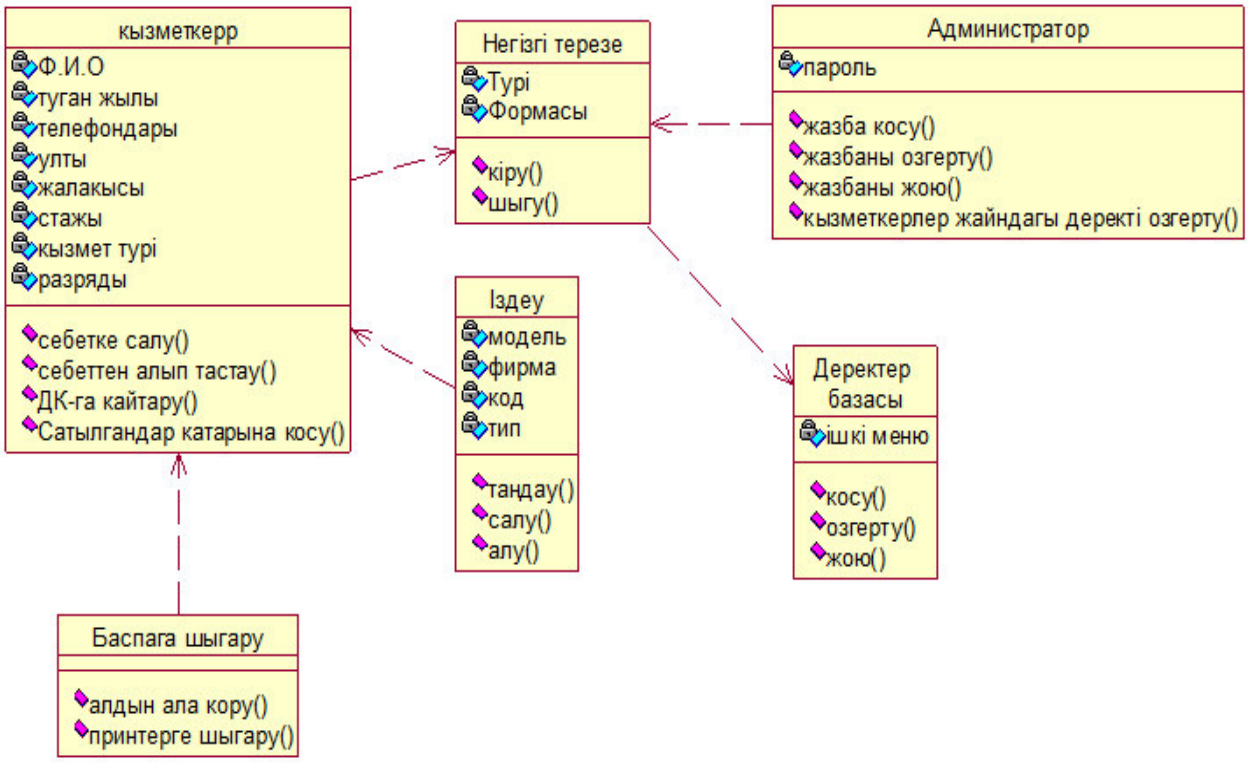
немесе сандардан тұратын дискретті құрам. Қауымдастық спецификациялау тәсіліндегі сабақтардың жауапкершілігін көрсетеді.

Сұрау әдістерінің атауы стандартты болып табылатын болса, осы әдістердің атауын диаграммадан шығаруға болады. Мысалы, сіз әдісімен бірге бірегей қарым-қатынас байланыстырылған нысан мен тобында байланысты объектілерді тізбегі бар көпжақты қарым-қатынас көрсетіледі қайтарады, ол жүзеге асырылатын осы конвенцияның қабылдай алады.

Спецификацияның үлгісі шеңберінде жалпыландырудың мәні ішкі түрдегі интерфейс супертүрлі интерфейстің барлық элементтерін қабылдауға тиіс. Ішкі типтегі интерфейс супер интерфейсімен анықталған.

Іске асыру тұрғысында жалпылама бағдарламалық қамтамасыз ету тілдерінде мұрагерлік тұжырымдамасымен байланысты. Сыныптас мұраланған әдістерді және суперкласстың барлық өрістерін асыра алады.

Біріктіру – бүтін-бөлік қатынасы және ең көп қолданылатын модельдеу әдістерінің бірі. Қарапайым агрегациядан басқа, UML композиция деп аталатын күшті біріктіруге ие. Ол онымен өмір бұзады және бірге бола яғни құрамы, нысаны бөлігі біртұтас қамтуы мүмкін, бөлшектерді өмірлік циклі, циклына сәйкес қосылады. Барлық бөліктер де жойылады. Сынып диаграммасы 2.4-суретте көрсетілген.



2.4-сурет – Сынып диаграммасы

3 Жүзеге асыру және тестілеу бөлімі

3.1 Бағдарламалау тілін таңдауды негіздеу

Компьютерде жоғарыда аталған мәселелерді шешу үшін оларды өңдеуге арналған бағдарламалық жасақтама қажет. Проблеманың негізгі мәселесі – белгіленген жобаның жоқтығынан техникалық жобаны бастауға жол берілмейді.

Бірінші жобалық кезең – компанияның талаптарын талдау. Мұны сарапшылардың сұраулары негізінде жасау үшін сіз барлық ағымдағы және әлеуетті қажеттіліктерді анықтауыңыз керек. Негізгі талдау филиалдың жұмысы кезінде анықталады, Д.Х., байланыс салалық құжаттармен анықталады. Барлық құжаттар компьютерлік технологиялар арқылы жеке деректерге негізделген. Алғашқы кіріс деректерін анықтағаннан кейін деректер базасы сол құжаттар негізінде жасалады.

Екінші дизайн фазасы – ақпараттық жүйенің негізін құрайтын тұжырымдамалық база кестелерін жасау, яғни кестелер енгізу құжаттарының негізінде жасалады. Деректерді сақтауға арналған құрылымды құру кезінде міндеттерді шешкен кезде, ең ертегі нысан – қызметкерлердің тізімі.

Дерекқордың сұрауы деректерді басқару жүйесін өңдейді, әр кестені бөлек файлда сақтайды.

Бұл жағдайда дерекқор сізде кесте файлдары бар каталог болып табылады. Деректерді өзгерткен кезде, жаңа жазбаны қосыңыз, бар жазбаларды жойыңыз, барлық өзгерістер дерекқорда жасалады және тек кестеде жасалынған ақпараттың жарамдылық тексерілгеннен кейін ғана кез-келген өзгерістер жазылады.

Жасалған кестелерді пайдаланып, пайдаланушы енгізілген деректерді өңдейді және теңшей алады, яғни деректер ағынын автоматтандырады және қажетті ақпаратты басып шығаруға мүмкіндік беретін реттелген, сұрыпталған, электрондық кірістерді жасай алады.

Дизайнның үшінші фазасы - деректерді тікелей сұрыптау, бағдарлама, экрандық нысандар, деректер сұрауларын орындауға мүмкіндік беретін есептер. Бұл кезеңде бағдарламаның визуалды және түсінікті интерфейсін жасау керек, яғни пайдаланушыға достық интерфейспен оңай шарлауға болатын интерфейсті құру қажет. Оған қол жеткізу үшін, сіз мәртебе жолағын, ашық мәлімдемені немесе жүгіргі оған жақындаған сайын әр түйменің функциясын қысқаша сипаттайтын орынды жасауыңыз керек.

Ақпараттық жүйелерді құру жобаның мақсатын анықтаудан басталады. Сәтті жобаның басты мәселесі мынада:

– жүйе іске қосылғанда және операция кезінде:жүйенің қажетті функционалдығы мен өзгерген еңбек жағдайларын сараптау дәрежесі;

– қажетті жүйелік өткізу қабілетін есептеу;

– сұратуы бойынша жүйенің қажетті реакция уақыты;

– жүйелік үлгісін қалаған режимде, басқаша айтқанда;

– IBM® Cognos Express бағдарламасында пайдаланушы сұрауын өңдеу үшін келесі қадамдарды орындаңыз;

– жүйені оңай пайдалану және қызмет көрсету;

– қажетті қауіпсіздік.

Жүйенің тиімділігін анықтайтын негізгі фактор – оның өнімділігі. Жақсы жобалық шешім қуатты жүйенің негізі болып табылады.Қазіргі уақытта нысанға бағытталған Visual Studio бағдарламалау ортасы C# тіліне негізделген кеңінен қолданылады. Бұл бағдарламаларды тез арада түрлі деңгейдегі қиындықтармен қарапайым бағдарламалаудан дерекқорлармен жұмыс істеу үшін кәсіби бағдарламаларға дейін жасауға мүмкіндік береді. Біз дипломдық жобаны әзірлеу үшін Visual Studio бағдарламалау ортасын пайдаландық.

3.2 Visual Studio ортасын сипаттау

Кез келген көрнекі бағдарламалау қоршаған ортаға сияқты үйлесімді тілдерді қолдану жазу, жөндеу, салу, жөндеу және іске қосу қосымшалар үшін Visual Studio ортасы құралдарын ұсынады. Visual Studio платформасының құрастыру ортасы төрт тілмен жұмыс істеуге арналған: C #, VB.NET C ++ және J #, бірақ қазіргі уақытта платформаға басқа бағдарламалау тілдерін қосу белгілі.

Visual Studio көмегімен әр түрлі жобалар түрлерін жасауға болады, мысалы:

– IBM Cognos Analytics дәстүрлі Windows басқару элементтерін пайдаланады;

– консольді Command Processor терезесінде көрсетеді;

– басқа қосымшаларда пайдаланылатын сыныптық кесте (dlll модульдері);

– қолданбалы веб-қосымшалар, яғни интернеттен шақыруға болатын қосымшалар және т.б.

Visual Studio ортасының көп бөлігі Windows технологиясын және веб-қосымшаларды жасау болып табылады, бірақ қазіргі заманғы әзірлеушілер консолі қосымшалармен жұмыс істейді.

Егер сіз консольдық қосымшамен жұмыс істесеңіз, пайдаланушы Windows пәрмен жолы терезесіне немесе мәтіндік режим терезесіне ұқсас мәтіндік терезеде жұмыс істейді, бұл Turbo Pascal ортасында.

«Консольдық қосымшалар тіл үйрену үшін ең қолайлы, себебі сіз көптеген стандартты GUI әзірлеу нысандарын пайдаланбайсыз.»

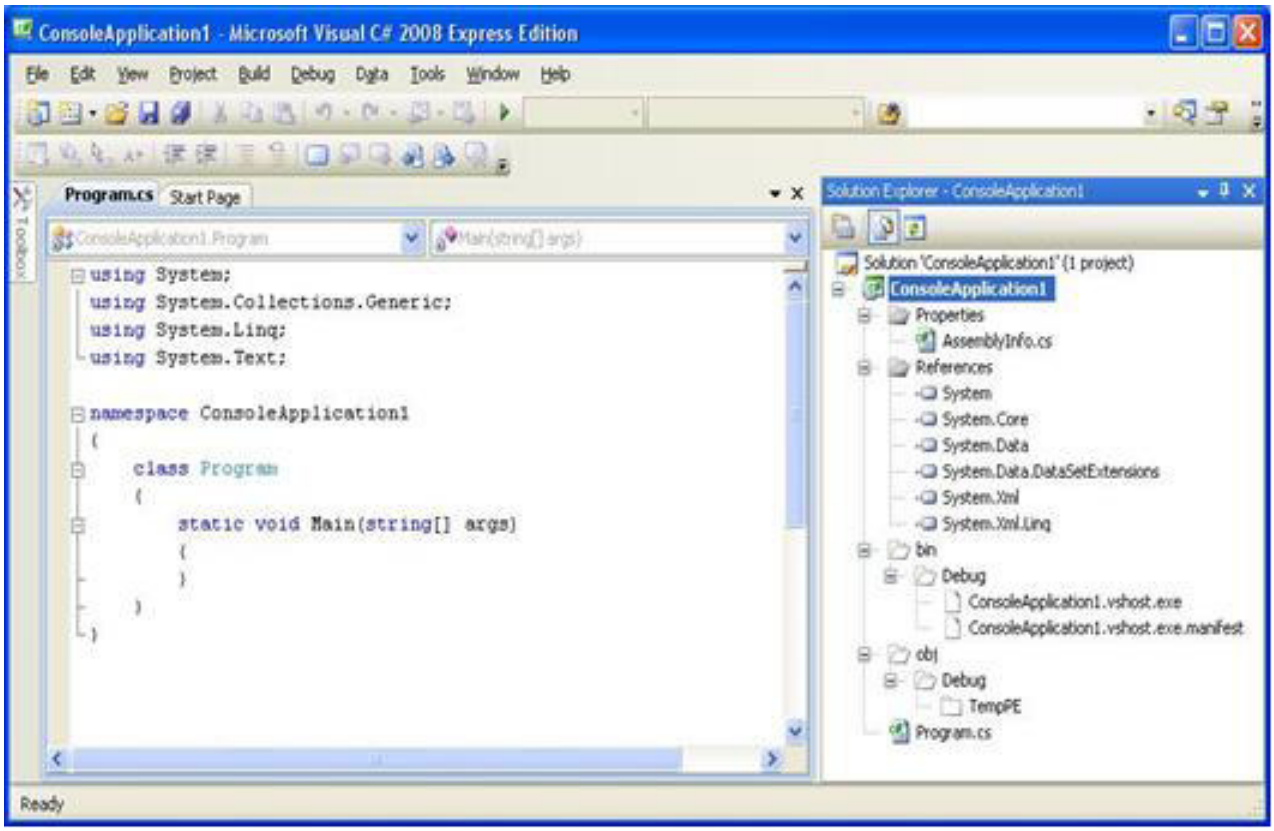
Visual Studio бағдарламасында жұмыс істеуге арналған жобаларды құрастырыңыз, оларды магниттік дискке сақтаңыз, жобаны бастаңыз және т.б. Сонымен, Visual Studio ортасын үйреніп, C # тілін үйренейік.

Негізгі мәзірден ортаны қосудан кейін Studio.NET жобасын жасау үшін Visual, File -> New -> Project пәрменін таңдауыңыз керек.

Ашық тілқатысу терезесінің сол жағында Visual C # Windows түймешігін нұқыңыз және тінтуірдің оң жақ түймешігімен консольдық бағдарламаны нұқыңыз.

Атау өрісінде жоба атауын жазуға болады, бірақ бұл жоба атауын сақтау ұсынылады. Компьютердің жұмыс орнын Орын өрістегі жобаның орны ретінде енгізіңіз (дискіде жоба сақтау орны).

Осы мәндерді ортада растағаннан кейін, жоба осы атаумен құрылады. Бағдарламаның үлгі коды 3.1-суретте көрсетілген.



3.1-сурет – Консольді қосымша жобасы құрылғанынан кейінгі Microsoft Visual Studio ортасының үлгісі

3.1-суреттің жоғарғы жағында негізгі мәзір көрсетіледі (Файлдар режимдері, Өңдеу, Көрініс және т.б.). Қоршаған орта режимі мен қоршаған ортаны басқару пәрмендерін жылдам таңдау үшін.

Экранның жоғарғы оң жақ бұрышында Project Management Solution Explorer терезесі (егер ол көрінбесе, сіз басты мәзір көрінісі пәрменін пайдалануыңыз керек).

Терезе System, System.Data, System.Volume сияқты Xml аттар кеңістігінде және т.б. сияқты жобамен байланысты барлық ресурстарды жазады.

Экранның негізгі бөлігі редакторлық терезе болып табылады, онда автоматты түрде құрылған бағдарлама коды (мәтін) ортасында тұр.

Бағдарлама коды қажет болған жағдайда өз кодын жазуға болатын үлгі ретінде жасалады.

Бағдарлама кодын көрсетсеңіз, түрлі компоненттердің түстері пайдаланылады. Қызмет сөздері үшін (резервтік кодтар) көгілдір түс пайдаланылады. Түсініктеме сұр немесе қара-жасыл, ал қалған бөлігі қара түспен көрсетіледі.

Мәтіннің сол жағында тік сызықтар мен квадраттар минус немесе плюс қосылған құрылымдық таңбалар болып табылады. Кез-келген квадратты минутпен бассаңыз, тиісті кодтың фрагменті (блокты) жинақтай аласыз, мұнда «плюс» минус болады. Егер шаршы плюс итерілсе, кодтың тиісті фрагменті (блокты) экранның тиісті бөлігінде «таралады» болады. Бұл ақпарат Windows пакеттерін қарап шыққанда пайдаланылады және қажетті бағдарлама кодының фрагментіне назар аударуға мүмкіндік береді.

Магниттік дискідегі жобаны жазу үшін Файл -> Барлығын сақтау (бірнеше дискідегі құралдар тақтасында) пәрмен жолы тізбегін пайдалану керек. Файл -> Бағдарламаны сақтаңыз, тек редакцияланған файлдағы өзгерістерді сақтау үшін құралдар тақталарын басыңыз немесе cs енгізілген бір дискета дискісін таңдаңыз.

Жобаны ашу үшін File -> Open Project командасын пайдаланыңыз.

Сабақтардың пайда болуы бағдарламалау технологиясын өзгертті. Егер бұрын құрылымдалған бағдарламалаудың негізгі бөлігі функция мен рәсімдер болса, сыныптардың пайда болуы деректер мен әдістерді біріктіретін функционалды түрде аяқталған бағдарламалық модульдерді құруға мүмкіндік берді. Мұндай бағдарламалардың негізгі бірлігі – объектілерге бағдарланған технология деп аталатын сыныптар (нысандар), сыныптар арқылы программалау технологиясы. Windows қосымшаларын құрастыру (күрделі бағдарламалық жасақтама жүйелер) объектілі-бағытталған технологияларды пайдалану мәселелерін шешеді. Object-Oriented Programming (ООВ) технологиясындағы сабақтарды пайдалану екі функцияны орындауға болатындығын көрсетеді: бағдарлама модулі немесе деректер түрі.

Модульдік құрылым – Windows қосымшаларының басты ерекшеліктері. Уақытша модульдерге бөлінбестен үлкен бағдарламалық жасақтама жүйесін әзірлеу модульдік бағдарламалау технологиясы арқылы дамыған жүйеден әлдеқайда көп жұмсайды. Windows CBS-де қосымшалар модульдік принцип бойынша жасалады, ол модульдің негізін құрайтын сыныптардан тұрады. Құрылымның модульділігі күрделі бағдарламалық жасақтама жүйелерін дамыту үдерісін жеделдетудің маңызды құралы болып табылады.

Екінші жағынан, сынып – деректер түрі. Windows бағдарламаларын объектілі-бағдарлы түрде жасау деректер негізіндегі дизайн деп аталатын стильге негізделген. «Жүйенің дизайны белгілі бір есепке сәйкес деректерді абстракцияның іздестірілуіне негізделеді, әрқайсысы мұндай абстракция класс ретінде құрылады, сынып – бұл бағдарламалық жасақтама құрылымының сәулет бірлігі болып табылатын модуль.

Көптеген Windows бағдарламаларында сыныптар екі функцияны да орындайды, сондықтан әрбір бағдарлама модулінің өздерінің нақты тапсырмасы бар. C # деректер түрлері және модуль функциялары бар сыныптарды пайдаланады. Модульдер сыныптары, мысалы, Консольдық сыныптар, модуль ретінде математикалық сыныптар объектілерді жасай алмайды, бірақ мұндай сыныптың нысаны болады. Осы модульдің өрістері мен әдістері басқа кластар әдістеріне қол жетімді.

Ірі бағдарламалық жасақтама жүйелерін әзірлеуде маңызды рөл атқарады. Бағдарламалау орталықтарымен ұсынылатын бағдарламалау технологиялары күрделі бағдарламалық жасақтама жүйелерінің даму уақытын айтарлықтай төмендетеді.

Консольдық бағдарламаларда Main () әдісінің нұсқаулары бағдарлама басталғаннан кейін басталады. Windows үшін жазылған қосымшалардың ерекшелігі, бағдарламаны іске қосқаннан кейін, Windows-ден келген хабарларды күтудің шексіз циклына өтеді.

Хабарлама Windows жүйесіндегі операциялық жүйенің жүйеде болған оқиғаларға жауап болып табылады. Оқиға ретінде, компьютердегі кез келген «стандартты емес» жағдайды, мысалы, пернетақтадағы пернені басып, меңзерді жылжытады, нөлден бөліп, т.б. есептеуіңізге болады.

Windows жүйесі жауап беретін барлық іс-шаралар нөмірленіп, әрбір бағдарлама бойынша «жауап беру векторы» бойынша оқиғаға дұрыс жауап беретін арнайы бағдарламамен анықталады. Мысалы, перифериялық компьютердің драйвері (пернетақта, тінтуір, таймер).

Оқиға болған кезде Windows оқиғалардың «нөмірін» анықтайды және тиісті драйверді бастайды. Драйвер іс-шараны «өңдейді», Windows жүйесіне хабар жібереді.

Windows жүйесі оқиғаларды басқару принципіне негізделген. Жүйеге және Windows жүйесіне жазылған барлық қосымшаларды белсендіргеннен кейін, пайдаланушы операциялық жүйенің әрекеттерін немесе оқиғаларын күтеді және оларға көрсетілген тәртіпте жауап береді.

Windows хабары оқиға туралы жазба болып табылады. Мысалы, кейбір хабарлар құрылымында бағдарлама терезесінің дескрипторы, хабарламаның коды (id) және параметрлер (мысалы, x және y тышқандар координаттары) хабар жасау уақытын анықтайды.

Windows алған барлық хабарлар жүйе кезегіне бірдей данаға арналған хабарларға орналастырылады. Содан кейін, кезектен тыс жүйелік хабарлар әр Windows қолданбасының кезектерімен синхрондалады. Сонымен қатар, әрбір бағдарлама үшін бөлек хабарлар кезегі жасалады. Хабарлама кезегі тек жүйелік хабарлармен толтырылмайды. Әрбір қосымша хабарлама әр басқа хабарға, сондай-ақ өздігінен жіберілуі мүмкін. Әрбір Windows бағдарламасында Windows жүйесінен келетін хабарларды өңдеу циклы бар. Бұл цикл арқылы қосымшалар «сіздің» хабарламаңызды алады және оларды қолданбаның тиісті хабар өңдеушісіне жібереді. Әрбір бағдарлама терезесінде

өзінің хабарламаларды өңдеу циклі және терезе функциялары бар (ол қосымша кезектен алынған хабарларды жібереді).

Бұл, әдетте, негізгі элементтер – мәзірлер, түймелер, құсбелгілер және т.б. қамтитын Windows бағдарламасының басты терезесі болып табылады В. Қолданбада жұмыс істеген кезде пайдаланушылар мәзірді таңдап, түймешіктерді басады немесе басқа басқару элементтерін қолданады.

Әр басқарудың өз идентификаторы бар. Әлеуметтік емес: салықты қызметкер демалыста демалыссыз немесе аударымсыз болғандықтан есептен шығармайды. Пайдаланылған басқару элементінен келетін хабар Windows операциялық жүйесін осы басқару элементінің кезегіне жібереді.

Windows жүйесінде жасалынған қолданбалар (Файл -> Жаңа -> Жоба -> Windows Forms Application) екі негізгі түрді қолданады: пішін, бағдарлама.

Қолданба сыныбы қосымшаны басқарады: хабарламаларды өңдеу циклі Application.Run (); ағымдағы Main () әдісін іске қосады, хабарлама алған кезде тиісті әрекеттерді орындайды және қосымша жұмыс (Бағдарлама) дұрыс аяқтайды.

Пішін сыныбы пайдаланушы интерфейсін анықтайды: пішін терезесін инициализациялайды, жұмысқа арналған қосымшаны дайындайды (Form1.cs файлы).

3.3 Формаға компоненттерді орналастыру

Windows қолданбаларын жасау процесі екі қадамға байланысты.

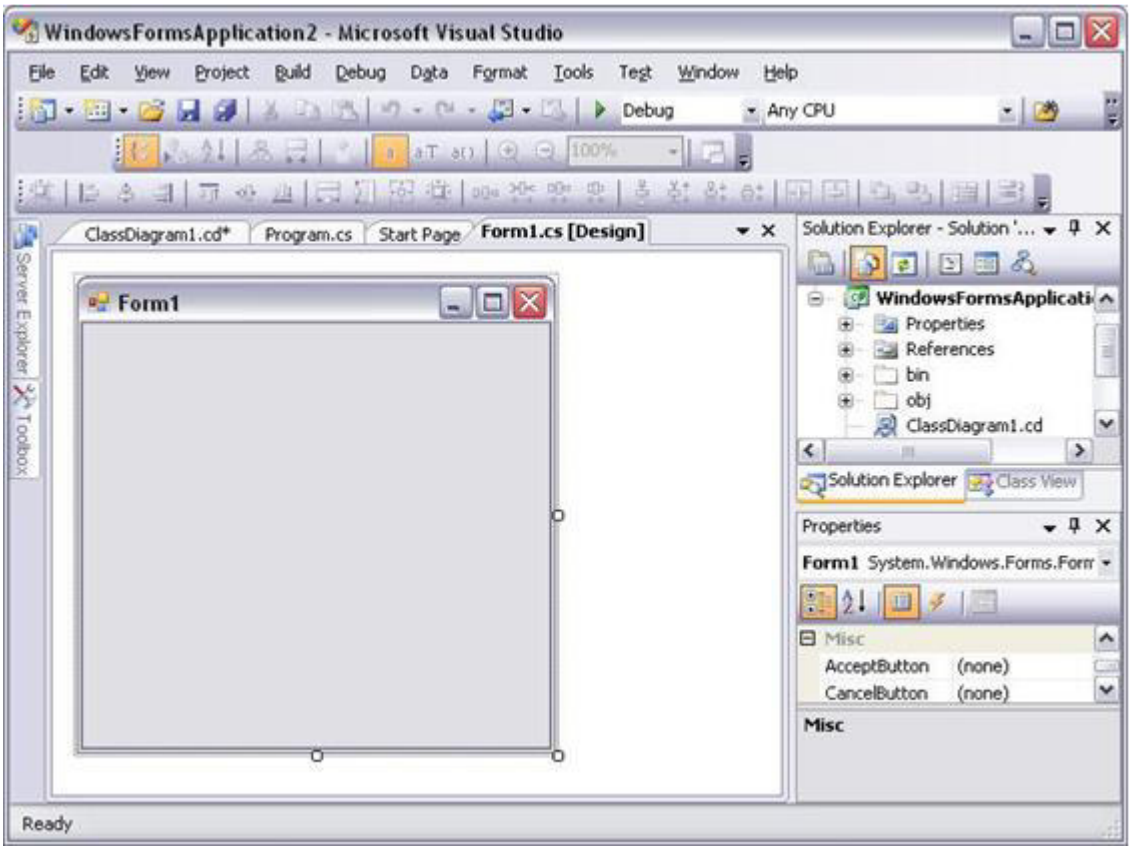
Бірінші кезең – пайдаланушы интерфейсі үшін визуалды бағдарламалау кезеңі.

Екінші кезеңде, бағдарламаның хабар өңдегішінің кодын жасауыңыз керек, және бағдарлама Windows жіберген хабарламаны алған кезде оны анықтайды.

Visual Studio Windows бағдарламаларын жасаудың алғашқы қадамын жасау үшін визуалды ортаны ашу қажет (File -> New -> Project -> Windows Forms Application), 2.2-сурет. Жұмыс үстелінде жобаны дайындағанда, барлық файлдар сақталатын буманың атауын көрсетуіңіз керек.

System.Windows пішіндер аты өрісі ат кеңістігіне сәйкес келетін Form1 терезесін қамтиды. Жобаны іске қосу үшін F5 пернесін басыңыз немесе Debug ортасының Бастау режимі пәрменін таңдаңыз. 3.3-сурет іске қосылған қосымшаның терезесін көрсетеді.

Visual Studio ортасында жеке Windows әрекеттерін егжей-тегжейлі қарастырыңыз.



3.2-сурет – Visual Studio ортасы

Form1 терезесі жобаның әртүрлі өңдеу режимдерінде пайдаланылатын 4 беттің тұрады, мысалы, Form1.cs [Design] пішінде элементтерді орналастыру үшін пайдаланылады және Program.cs терезесі бағдарлама кодын өңдеу үшін пайдаланылады.

Сипаттар терезесі басқару элементтерінің сипаттарын анықтайды. Бұл терезенің беті санат, сипат немесе оқиға бойынша топтастырылған.

Сервер жетектеушісі терезесі (3.3-суретте көрсетілген, сол жағында, жиналған пішінде) – сіздің компьютеріңіздегі, сервердегі деректер көздеріне қол жеткізу үшін қолданылған.

3.3-сурет – Программаның жұмыс терезесі

Toolbox терезесі (3.2-сурет, сол, жиналған) түрлі басқару элементтерін қамтиды.

Solution Explorer терезесі (3.2-сурет, оң жақта) жоба файлдарын көруге және өңдеуге мүмкіндік береді. Ол файлды және сыныптар бойынша жобаны көруге мүмкіндік береді.

Әдетте, бағдарлама басталғаннан кейін қатенің сызық нөмірі бағдарлама кодында пайда болатын қате терезесі пайда болады.

3.4 Формаларды визуалды жобалау технологиясы

Windows көрнекі бағдарламалау ортасында. Forms.Designer – бұл пішінді көрнекі түрде интерактивті режимде орналастыруды басқару арқылы жасауға мүмкіндік беретін құрал. Дизайнердің артықшылығы мынада, басқару элементтерін пішінде дұрыс жерде орналастыруға болады және оның қасиеттерінің дәл мәндерін анықтау қажет емес, мысалы, өлшемнің сандық мәні, орналасқан жері. 1-кестеде келтірілген талаптарға сәйкес 1-кестеде алынған деректер көзі ретінде пайдаланылуы мүмкін екендігін ескеру керек.

Windows бағдарламасында (консольдық қосымшаға қарағанда), автоматты түрде бірнеше .cs құралы Form1.cs және Program.cs сияқты кеңейтіммен сыныптар жасайды. C # жүйесінде сыныптар бірнеше бөліктен тұрады, олардың әрқайсысы кілт сөзді ішінара бастайды. Бір сыныптың сипаттамаларын бөлікке бөлу қабілеті үлкен сыныппен жұмыс істеуді жеңілдетеді. Класстың әрбір бөлігі жеке файлдарда өз атымен сақталады. Мысалы, алдыңғы бөлімде үлгі автоматты түрде Form1 файлын пайдаланады. Cs-Form1.cs кеңейтімімен екі файлды және Form1.Designer.CS құрылды. Form1. «Cs» файлында сақталған Form1 әзірлеуші сыныбының бірінші бөлігі автоматты түрде жасалатын басқару элементтері үшін оқиға өңдегішті орналастырады.Барлық, басқарылатын деп аталатын пішіндермен жұмыс істеу негізінде осындай бағдарламалау әдістері оқиға бағдарламалау технологиялары Мысалы, алдыңғы бөлімде берілген Form1.extract cs файлынан:

```
namespace WindowsFormsApplication1 {
public partial class Form1 : Form
{
public Form1()
{
InitializeComponent();
}
private void button1_Click(object sender, EventArgs e){
int a,b,c,p;
a = Convert.ToInt32(textBox1.Text);
b = Convert.ToInt32(textBox2.Text);
c = Convert.ToInt32(textBox3.Text);
p = a + b + c;
```

Form1 сыныптың екінші бөлімі, «Form1.Designer.located» cs «файлында.

Сабының бұл бөлігі автоматты түрде пішінді компиляторға толтырады.

Көрнекі конструкция, әртүрлі басқару түріндегі келісім, олардың қасиеттерін өзгертеді және оқиғалардың редакторларын анықтау, пішінді құрастырушы, осы әрекеттерді тиісті класс объектілерінің іс-әрекеттеріне аударады тиісті кодты құрайды және кодты Form1 сыныпта орындаңыз.

Form1 сынып кодын түзету үшін пішін әзірлеушінің жұмысына кедергі жасаудың қажеті жоқ. Дегенмен, сіз оның жұмысы туралы және оның жасаған коды туралы білуіңіз керек.

Cs файлынан Form1.Designer.extract төменде берілген:

```
namespace WindowsFormsApplication1 {
partial class Form1
{
<summary>
Required designer variable.
</summary>
private System.ComponentModel.IContainer components =
null;
. . .
private void InitializeComponent()
{
this.label1 = new System.Windows.Forms.Label(); this.label2 = new
System.Windows.Forms.Label(); this.label3 = new
System.Windows.Forms.Label(); this.label4 = new
System.Windows.Forms.Label(); this.button1 = new
System.Windows.Forms.Button();
. . .
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(12, 9); this.label1.Name =
"label1";
this.label1.Size = new System.Drawing.Size(174, 13); this.label1.TabIndex =
0;
this.label1.Text = "А үшбұрышының қабырғаларын енгізіңіз";
. . . және т.б.
```

Бағдарламада автоматты түрде жасалатын бағдарлама. Cs сынында статикалық Main () әдісі болады. Бағдарлама басталса, Windows негізгі () әдісін іздейді және нұсқауларды орындауға кіріседі. Негізгі () әдісі әдетте бағдарламаға кіру нүктесі деп аталады.

Төменде cs бағдарламаның үлгісі берілген файл:

```
namespace WindowsFormsApplication1 {
static class Program
{
<summary>
The main entry point for the application.
</summary>
[STAThread]
static void Main()
{
Application.EnableVisualStyles();
Application.SetCompatibleTextRenderingDefault(false);
Application.Run(new Form1());
}}}
```

Консольдық қосымшаларда Main () әдісінің негізгі бөлігі бос болады, онда кодты жоба әзірлеушісі жазуы тиіс. Windows бағдарламаларында Main () әдісі тиісті нұсқаулармен толықтырылады және әдетте әзірлеуші оны өзгертпейді. Автоматты түрде жасалатын Main () әдісі не істейді? FCL кітапханасында өз кезегінде тиісті қосымша сыныптың үш статикалық әдісін шақыру арқылы жұмыс істейді.Application.

The EnableVisualStyles () әдісі Windows XP стилінде жасалған қосымша құрамдастар ұсынады.

Application.SetCompatibleTextRenderingDefault (жалған); әдіс шынайы – GDI +, false – GDI экранына мәтін шығару механизмін анықтайды.

Application.Run (жаңа Form1 ()); әдіс – бағдарлама сыныбының негізгі әдісі және Main () әдісіндегі жалғыз әдіс.

Негізгі жұмыс Run әдісі арқылы жүзеге асырылады, қоңырау кезінде Form1 сынып нысаны жасалады және оның нысаны - визуалды нысан терезесі ашылады. Барлық басқару элементтерін инициализациялауды аяқтаған соң, Windows хабарларының циклын қамтитын пішінде орналасқан, Windows жүйесінен хабарлама күтіп тұр. Ал егер пайдаланушы, мысалы, келесі әрекеттерді орындай алады: пішін өрістерінде деректерді енгізіңіз.

3.5 Файлды ашуға арналған оқиғалардың өңдеуіштері

Құралдар тақтасында Файлды ашу пәрмені түймешігін басыңыз. Осы оқиға редакторын жазу қадамдарының кезегін қарастырайық.

Бағдарламада файлды ашу үшін оқиға өңдегішін жазу үшін бізге OpenFileDialog элементі қажет. Бұл элементтің белгішесін құралдар жинағы терезесінен көшіріңіз. OpenFileDialog элементінің белгішесі пішіннің астындағы панельде көрсетіледі.

3.4-сурет – OpenFileDialog элементін қосу

OpenFileDialog элементін пайдалану - компьютердегі қалталарды, дискілерді, файлдарды пайдалануға мүмкіндік беретін толық бағдарламалау технологиясы. Бұл элемент OpenFileDialog1.The ShowDialog әдісі сияқты құрал терезесінің терезесіндегі кез-келген элемент сияқты көптеген файлды манипуляциялау әдістерін қамтитын сынып ретінде ұсынылған. Экрандағы стандартты файлды таңдау тілқатысу терезесі көрсетіледі.

OpenFileDialog1 элементінің сипаттарын теңшеу үшін ашылатын файлдардың аттарына сүзгілерді анықтауға болады. Ол үшін openFileDialog1 элементінің сипаттар терезесінде сүзгі сипаттарын өзгертуіңіз керек. Мына сипаттарға келесі мәтін жолын тағайындаңыз:

Text files|*.txt|RTF files|*.rtf| All files|*.*

Сүзгі жолы «|» таңбасымен бөлінген бөліктерден тұрады. Бірінші бөлім файл түрінің атауын мәтіндік файлдар ретінде көрсетеді, ал қалған бөліктер қосқыш (маска) файл атауларын білдіреді. Мәтіндік файлдар үшін *.used экран (маска) txt.

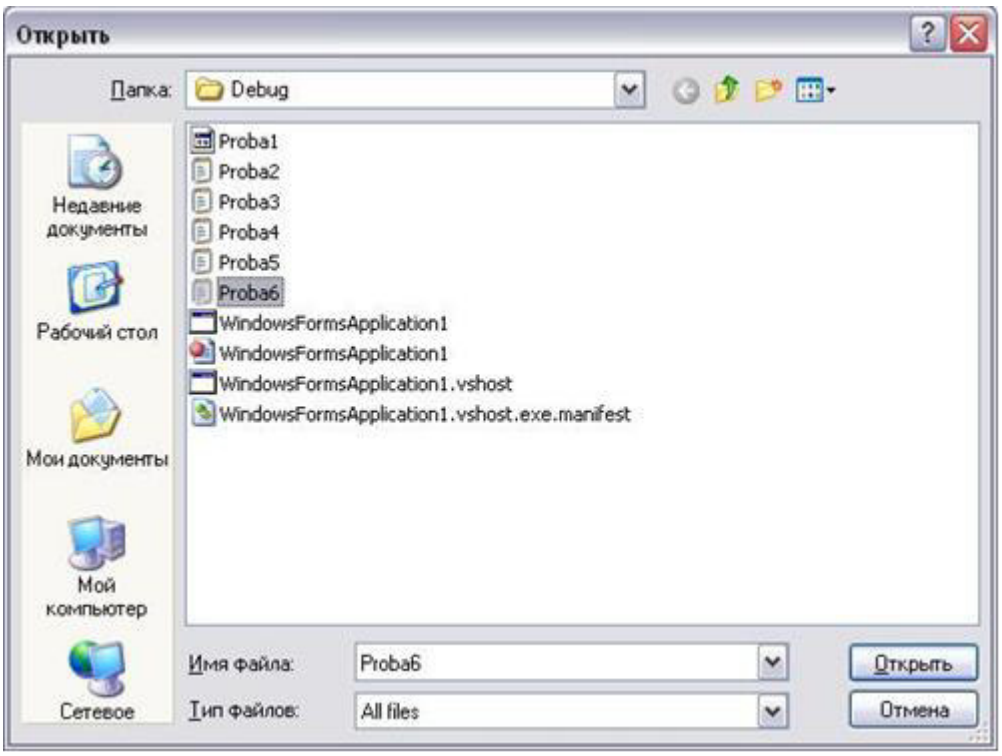
Содан кейін rtf файлдарының форматы табылды. RTF файлдары *.bulkhead (маска) rtf пайдаланылады.

Қолданбаның барлық файл түрлерін ашу үшін (Барлық файлдар) *.* маска (бөлім) қолданылады.

Есепке сәйкес біз мәтіндік файлдармен жұмыс істейміз, бірақ басқа файл түрлерін де ашуға болады.

Файлдың қосымша коды кеңейтіледі:

```
private void loadTToolStripMenuItem_Click(object sender, EventArgs e)
{
if (openFileDialog1.ShowDialog() == DialogResult.OK &&
openFileDialog1.FileName.Length > 0) {
try
{
richTextBox1.LoadFile(openFileDialog1.FileName,
RichTextBoxStreamType.PlainText);
}
catch (System.ArgumentException ex)
{
richTextBox1.LoadFile(openFileDialog1.FileName,
RichTextBoxStreamType.RichText);
}
this.Text="Файл ["+openFileDialog1.FileName+"]";
```



3.5-сурет – Диалогтық терезені ашу бойынша бағдарлама жұмысы

Есепке сәйкес біз мәтіндік файлдармен жұмыс істейміз, бірақ басқа файл түрлерін де ашуға болады.

Егер пайдаланушы файлды таңдау терезесіндегі «Ашу» түймесін бассаңыз, ShowDialog DialogResult әдісі. OK мәнін қайтарады. Келісімде файлды таңдаудың қосымша тексеруі (таңдалған файлға арналған жолдың ұзындығы openFileDialog1.FileName.Length ұзындығы 0-ден артық болуға тиіс).

File richTextBox1.When LoadFile әдісі ашылса, екі параметр рұқсат етіледі. Бірінші параметр - файлдың жолы, ал екінші параметр – файл түрі.

Мәтіндік файлдармен жұмыс істеуіңіз керек болғандықтан, негізгі түрі – RichTextBoxStreamType.PlainText.Таңдалған файлдың пішімі LoadFile әдісіндегі PlainText пішімінен өзгеше болса, System арнайы ArgumentException күйі (қоспағанда) шығарылады. Редактор файлды RichText файлы ретінде қайта ашуға тырысады. Бұл түрі RTF форматына сәйкес келеді. Қосымша ақпарат үшін ашылған файлдың толық жолағы негізгі бағдарлама терезесінде көрсетіледі.

Құралдар тақтасы бағдарлама терезесінде (Save пәрмені) жазу файлына жазу үшін кез-келген түймешікті қамтиды. Осы оқиға редакторын жазу қадамдарының кезегін қарастырайық.

Бағдарламада файлға жазатын редакторды жасау үшін бізге SaveFileDialog элементі қажет. Осы элементтің белгішесін құралдар терезесі терезесінен пішін терезесіне көшіріңіз. Элементке арналған SaveFileDialog1

белгішесі пішім астындағы тақтаға шығады.SaveFileDialog1 элементінің сипаттарын теңшеу үшін, сүзгі және файл атауы сипаттарын өңдеуіңіз керек.

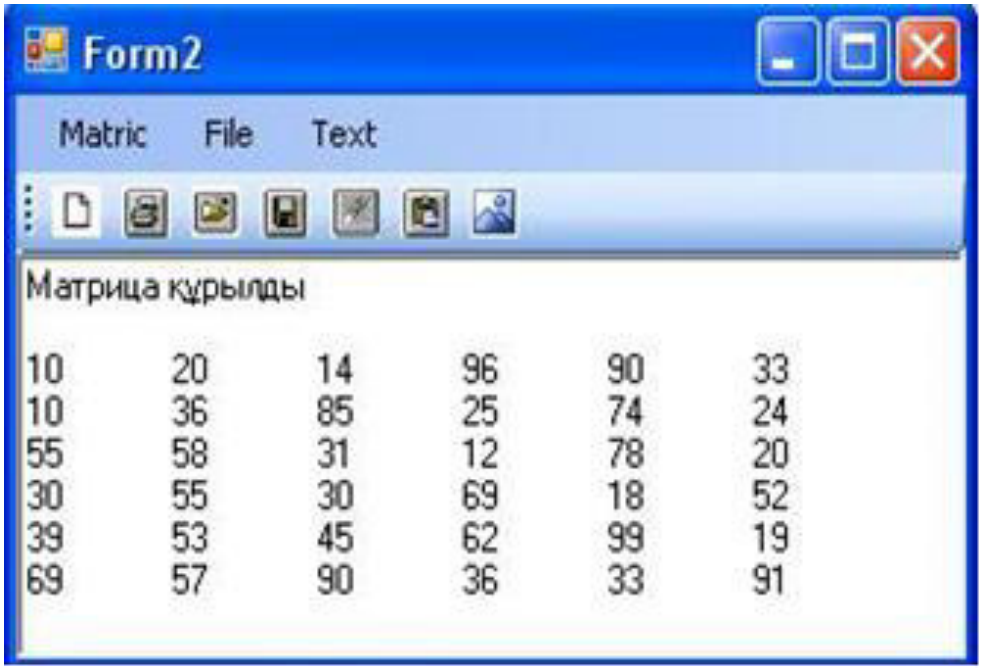
Сүзгі сипаты үшін мәтіндік файл түрін көрсету керек – *.txt мәтіндік файлдары. FileName сипатын document-doc1.you атауына txt мәнін орнату керек. Файлдар осы атаумен сақталады.

Файлдың қосымша коды кеңейтіледі:

```
private void saveTToolStripMenuItem_Click(object sender, EventArgs e)
{
if (saveFileDialog1.ShowDialog() == DialogResult.OK &&
saveFileDialog1.FileName.Length > 0) {
richTextBox1.SaveFile(saveFileDialog1.FileName,
RichTextBoxStreamType.PlainText);
this.Text = "Файл [" + saveFileDialog1.FileName + "]";
}
}
```

Файлдың толық жолағы негізгі бағдарлама терезесінде көрсетіледі.

RichTextBox1.SaveFile әдісінің екінші параметрі пайдаланылатын файлдың түрін анықтайды. Бұл мысалда бағдарлама тек мәтіндік файлдармен жұмыс істеуге бағытталған. RichTextBoxStreamType.If параметрі RichText мәнін жібергенде, құжат RTF пішімінде сақталады.



3.6-сурет – Қосымшада мәтіндік файлмен жұмыс жасау мысалы

Қолданба аймағындағы мәтінмен жұмыс істеу үшін, мәзірді тазалау және ауыстыру пәрмендеріне сәйкес екі оқиға өңдеушісі бар.

Мәтінді тазалауға редактор болған жағдайда, барлық жолдарды жоятын richtextbox1 мәтіндік редакторының әдістерінің бірі жазылған.

```
private void clearTToolStripMenuItem_Click(object sender, EventArgs e)
{
richTextBox1.Clear();
}
```

Мәтінді түрлендіруді жүзеге асыратын оқиға редакторы пайдаланушыға арналған аумақта орналасқан мәтіннен 6 * 6 матрицасының (әдетте файлдан алынған) мәндерін таңдау үшін пайдаланылады.

Мәтін өңдеу әдісінің коды төменде көрсетілген:

```
private void obraboToolStripMenuItem_Click(object sender, EventArgs e)
{
string ss;
string[] clova;
int k, n;
int[] masi = new int[100];
n = 0;
= richTextBox1.Text; clova = ss.Split('\t', ' ');
for (int i = 0; i < clova.Length; i++)
{
k = clova[i].Length; if (k == 2 || k == 3)
{
masi[n] = Int32.Parse(clova[i]); n++; }
richTextBox1.AppendText(i.ToString() + " = " + clova[i] + " k= " + k.ToString() +
"\n");
}
int j1, j2; j1 = j2 = 0; for (int i = 0; i < 36; i++)
{
a[j1, j2] = masi[i]; j2++;
if (j2 == 6) { j1++; j2 = 0; } if (j1 == 6 && j2 == 6) break;
}
}
```

Мұнда кейбір түсініктемелер беру қажет.

Ұяшық, бос орын немесе қайтару таңбасын пайдаланып, сплит әдісімен пайдаланушыға арналған аумақта орналасқан мәтіннен таңдалған кез келген таңбалар тіркесімдерін жазатын String типтік массиві жарияланды:

```
ss = richTextBox1.Text;
clova = ss.Split('\t',' ','\n');
```

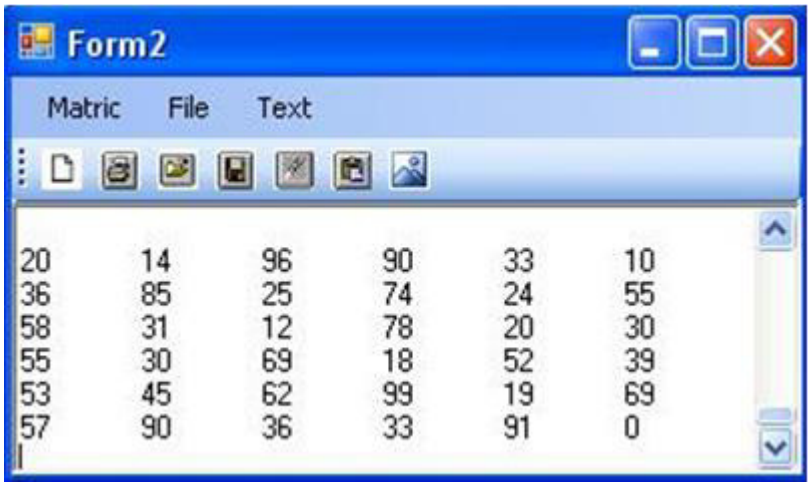
clova.Length – мәтіндегі барлық сөздердің санын анықтайды.

Циклда 2 немесе 3 таңбаға тең ұзындықтағы барлық сөз бүтін санға айналады және Masi бүтін сандарға жазылады. Барлық сөздер мен олардың ұзақтығы бір уақытта экранда пайда болады (басқару үшін).

Оқиға редакторының соңғы бөлігінде массивте таңдалған бүтін сан 6 * 6 матрицасына жазылады.

Пайдаланушы аймағын тексергеннен кейін оны тазалауға болады.

Содан кейін матрицаның шығу режимін экранға шығарыңыз.



3.7-сурет – Пайдаланушы аумағындағы мәтінді түрлендіргеннен кейін матрицаны эканға шығару

3.6 Protected және Private директивтері

Әрбір өрісте қолжетімділік модификаторы бар, мүмкін мағыналары: public, private, protected, internal. қорғалатын және ішкі атрибуттарды бірге пайдалануға болады жеке модификатор. Жеке модификатор әдетте модификаторлар көрсетілмесе, пайдаланылады. Ол өрістерді басқа барлық сыныптардан бөлек сақтайды және сынып әдістеріне (оқу, жазу) тікелей қол жеткізуге мүмкіндік береді.

Барлық өрістер сыныптың барлық әдістері үшін қол жетімді екенін ескеріңіз. Олар класс әдістеріне арналған көптеген ақпарат болып табылады,барлық әдістер олармен жұмыс жасайды, яғни өрістерден қажетті ақпаратты алуға және оларды өзгертуге мүмкіндік береді.

Қорғалған модификатор. Бұл модификатор мұрагерлердің сыныптарында өрістерді сақтау үшін ашылады. Егер А класының модификаторы қорғалған өрісті жарияласа, онда сынып В әдістеріне А класының мұрагері А сыныбын иеленіп, тікелей осы өрістерде жұмыс істей алады.Ішкі модификатор. Өрістер осы модификатор бойынша бірлескен сабақтарға ашық болады. Егер А және В сыныптары сол құрылымға (Ассамблеяға) тиесілі болса, онда сол сыныптар өзара әрекеттесетін сыныптар деп аталады. Егер А сыныбының белгілі бір өрісі ішкі модификатормен жарияланса, сынып А клиенті болып табылатын В сыныпты жұмыс істеу әдістері осы өріске тікелей жұмыс істей алады.

Қорғалған және ішкі атрибуттардың (комбинациясы) жіктелуі. Бұл компонент мұрагерлерге немесе бірлесіп жұмыс істейтін сабақтарға арналған алаңды ашады. Егер өрістерге қол жеткізуді қатаң шектейтін болсаңыз, тек сыныптармен жұмыс істейтін мұрагерлерге ғана қол жеткізуге болады, ішкі сыныптың өзі модификатор ретінде және сәйкес өрісті қорғалған модификатор ретінде жариялауыңыз керек. Егер өрістер тек сынып әдістері үшін қол жетімді болса, олар жеке қатынас модификаторымен жарияланады.

Бұл өрістер жабық өрістер деп аталады, бірақ әдетте олардың кейбіреулері басқа сыныптарға ашық болуы керек.

Егер А класының кейбір өрістері А сыныпты мұрагері В әдістеріне қол жетімді болса, онда бұл өрістер қорғалған өзгерткішпен бірге жариялануы керек. Мұндай өрістер қорғалған өрістер деп аталады. Егер кейбір өрістер А, В1, В2, ТВ кластерлерімен серіктес болса, сыныптық өзара әрекеттесу әдістеріне қолжетімді болған жағдайда, осы өрістерде ішкі модификаторды қолданыңыз, барлық сыныптардағы ынтымақтастық бір жобаға (жинауға) қойылуы керек. Бұл өрістер өзара әрекеттесетін өрістер деп аталады. Сондықтан, егер кейбір өрістер В сыныбының кез келген әдістеріне қол жетімді болса (В класы), сол өрістерде қоғамдық модификатор болуы керек. Бұл өрістер ашық және үлкен қолмен өрістер деп аталады.

3.7 Объекті-бағытталған программалаудың принциптері

3.7.1 Инкапсуляция ұғымы

Объектілі-бағытталған бағдарламалау технологиясы үш негізгі принципке негізделген: инкапсуляция, тұқым қуалаушылық және полиморфизм.

Инкапсуляция қағидаты, яғни бір құрылымға деректерді өңдеу және өңдеу әдістерін біріктіру класс ұйымының негізі болып табылады.

Ресми инкапсуляция – деректерді тікелей қолжетімділіктен қорғау үшін сыныптардың бірлестігі мен сынып әдістері.

Нысан өрістері интерфейс арқылы – қасиеттер жиынтығы немесе қатынау ережелерімен пайдаланылуы керек. Нысанның өрістерін жасыру – оларды инкапсуляциялау («капсула» сөзінен) қасиеттер арқылы жүзеге асырылады.

3.7.2 Мұрагерлік ұғымы

Мұрагерлік принципі объекті-бағытталған бағдарламалау тұжырымдамасында іргелі принцип болып табылады. Мұрагерліктің мақсаты – құрылып қойған кластарды қайталап қолдану.

Кейбір авторлар кластардың мұрагерлік идеясын түсіндіргенде жалпыдан жекеге қарай жалғасатын иерархиялық байланыстың мысалын келтіреді:

Жануарлар – мысық тектестер – жолбарыс.

Басқа авторлар кластардың мұрагерлік идеясын кіші объекттен үлкен объектке қарай жалғасатын иерархиялық байланыстың мысалы арқылы түсіндіреді:

Нүкте – кесінді – тіктөртбұрыш.

Материалды түсіндіруде авторлардың осы екі көзқарастары қолданылады.

«Кіші объекттен үлкен объектке қарай» тәсілі мұрагерлік идеясын түсіндіруге мүмкіндік береді – нүкте, одан кейін кесінді, т.б. Осы технологияны мұраланатын кластардың тізбегін «нөлден» дайындаған кезде қолдануға болады.

«Жалпыдан жекеге қарай» немесе «жалпыдан нақтыға қарай» екінші тәсілі бағдарламалауда құрылып қойған кластарды пайдалану арқылы қолданылады, мысалы, DELPHI-де VCL, VISUAL C++ ортасында MFC және басқа да стандартты кітапханалар, C# тілінде атаулар кеңістігі.

Деректері немесе әдістері мұраланатын класты базалық класс деп атайды.

Деректерді немесе әдістерді базалық кластан мұраға алатын класты туынды класс деп атайды.

Мұрагерлік туынды класқа өз қасиеттерін, деректерін, әдістерімен қатар базалық кластың қасиеттерін, деректерін, әдістерін қолдануға мүмкіндігін туғызады.

Мұрагерлік идеясын түсіндіру үшін келесі мысалды қарастырайық. Қосымшада құрылып қойған класқа қарағанда қосымша жаңа қасиеттер мен деректерге ие белгілі бір класты жазу керек болсын.

Оны жүзеге асырудың екі жолы бар.

Бірінші әдіс бойынша бір қосымшадағы класты жаңа қосымшадағы ға көшіріп, оған керекті өзгерістерді жазу керек.

Екінші әдіс бойынша алдынғы класты мұраға алатын (алдыңғы кластан туындайтын) екінші класты құру керек.

Бағдарламаны жазу бойынша қарастырылған екі әдістің ішінен ОББ тұрғысынан екінші әдіс қолайлы болып келеді. Кластарды мұраға алу бағдарлама кодын айтарлықтай қысқартады және қосымшаны құру уақытын азайтып, оның сенімділігін жоғарлатады.

Мұрагерлік иерархиялық құрылымды құруға мүмкіндік береді, иерархиялық құрылымда туынды кластар базалық кластың өрістерін, қасиетерін, әдістерін өз мүмкіндігіне алады және оларды толықтыра немесе өзгерте алады. Сонымен мұрагерлік кодты бірнеше рет қолдануға мүмкіндік береді. Базалық кластың кодын жазып, оны дұрыстағаннан кейін, осы класты мұралану арқылы оның кодын өзгертпей-ақ түрлі жағдайлар үшін қолдануға болады.

Иерархиялық құрылымның басына жақын орналасқан кластарда құрылымның төменгі жағындағы кластардың жалпы сипаттары біріккен. Иерархиялық құрылымның бойымен төмен жылжыған сайын кластардың айқын ерекшеліктері ұлғаяды.

C# тілінде кластардың иерархиялық тізбегінің базалық класы – System.Object.

Мұрагерлік принципі объектілі-бағдарланған бағдарламалау тұжырымдамасындағы іргелі қағида болып табылады. Мұраның мақсаты – жасаған сабақтарды қайта пайдалану.

Кейбір авторлар сыныптардан мұрагерлік идеясын түсіндіріп, жалпыға ортақ жалғастыра отырып, иерархиялық қатынастардың мысалдарын келтіреді: жануарларға арналған мысықтар - жолбарыстың түрі.

Басқа авторлар шағын нысаннан үлкен объектке дейін жететін иерархиялық қарым-қатынас үлгісі арқылы сынып мұрасын идеясын түсіндіреді: Point-segment – тікбұрышты.

Авторлардың осы екі көрінісі материалды түсіндіруде қолданылады.

«Кішкентай нысаннан үлкен объектке дейін» әдісі мұрагерлік ұғымның идеясын түсіндіруге, одан кейін қиып алуға және т.б. мүмкіндік береді. Бұл технология оны құрастыратын «нөлдік» сыныптық тізбекті дамыту үшін пайдаланылуы мүмкін.

Бұл жағдайда, мысалы, DELPHI, VCL, VISUAL C ++, MFC және басқа стандартты кітапханаларда қолданылатын түріне қарай, аттар кеңістігі C # -де болады.

Деректер немесе әдістер тасымалданған сынып базалық класс деп аталады.

Деректер немесе әдістер негіздік сыныптан мұраланған алынатын сынып деп аталады.

Мұра туынды сыныпқа оның қасиеттерін, деректерін, әдістерін және негізгі сынып қасиеттерін пайдалануға мүмкіндік береді.

Мұрагерлік идеясын түсіндіру үшін келесі мысалды қарастырайық. Қосымша қосымша қасиеттері мен деректерін қолданбада жасалған сыныпқа қарағанда белгілі бір сыныпты жазуыңыз қажет.

Оны жүзеге асырудың екі жолы бар.

Бірінші әдіс сыныпты бір қолданбада жаңа қосымшаға көшіру және оған қажетті өзгерістерді жазу болып табылады.

Екінші әдіске сәйкес сіз алдыңғы класс (алдыңғы сыныптан алынған) екінші сыныпты құруға тиіссіз.

Бағдарламаны жазу үшін ұсынылған екі әдіс бойынша, ЦРУ тұрғысынан екінші әдіс қолайлы. Сыныптар мен мұра кодын жасау, бағдарлама, бағдарлама уақытты едәуір қысқартады, оның сенімділігін арттырады.

Мұрагерлік иерархиялық құрылымды құруға мүмкіндік береді, иерархиялық құрылымдағы алынған сыныптар өздерінің мүмкіндіктеріне ие болуы мүмкін және базалық класс өрістерін, қасиеттерін, әдістерін толықтырады немесе өзгертеді. Бұл жағдайда мұраға кодты бірнеше рет пайдалануға мүмкіндік береді. Негізгі сынып кодын енгізіп, оны түзеткеннен кейін, негізгі кодын мұраға қарай өзгертпестен әртүрлі жағдайларда пайдалануға болады.

Иерархиялық құрылымның басында орналасқан сыныптар құрылымның төменгі бөлігіндегі сыныптың жалпы сипаттамаларын біріктіреді. Иерархиялық құрылымды төмендету сыныптардың нақты мүмкіндіктерін арттырады.

C# – System.Object ішіндегі иерархиялық сынып тізбегінің негізгі сыныбы.

Сондықтан мұра келесі өзара байланысты мақсаттарда қолданылады:

– бағдарламадағы код үзінділерін қайталамауға жол бермеу;

– бағдарламаны модификациялауды жеңілдету;

– үлгіде жасалған бағдарламаны пайдаланып жаңа бағдарламаны жасауды жеңілдету үшін.

Дегенмен, бұрынғы пайдалану коды қол жетімді емес, бірақ өзгертулерді қажет ететін нысандарды пайдаланудың бір жолы.

Мұра аудиториясының жазба пішімінде әдетте сынып кластың атауы қос нүктемен жазылған әдебиет шығарылады:

[атрибуттар] [спецификаторлар] class класс_атауы [: базалық класс]

{класс денесі}

Егер базалық кластың атауы көрсетілмесе, онда базалық класс болып System.Object класы есептеледі.

Егер бізде базалық класс бар болса, онда оны қос нүкте арқылы көрсету керек, мысалы:

public clas otr : tka

{класс денесі}

Бұл мысалда сегменттік класс нүктелік класспен игеріледі. Әлбетте, егер tka сыныбы туралы деректер жекешелендіруге қол жетімділік белгісімен жабылса, ол сондай-ақ алынған сабаққа жабық болады.

Кейбір базалық сыныптар қорғалған қатынас сипаттамасынан кейін деректерді орналастырады. Мысалға:

protected int x;

protected int y;

Қорғалған қатынау сипаттамасына деген қажеттілік туынды класс негізгі сыныптың (public specifier) қоғамдық элементтерін пайдалана алады.

Екінші жағынан, туынды класс бастапқы сыныптың жеке элементтерін тікелей пайдалана алмайды, олар үшін бастапқы сынып негізгі әдістерді қолдануы тиіс. Сондықтан, қорғалған сынып мүшелерін анықтайтын қорғалған қатынау спецификасы енгізілді. Қорғалған элементтер жабық және ашық элементтер арасындағы аралық бос орынды алады. Егер элемент қорғалған болса, онда алынған сыныптың нысандары оны ашық элемент ретінде пайдалана алады. Қолданбаның қалған бөлігінде қорғалған элементтер жабылады. Егер сіз қолданбаңызда жеке элементтерді пайдаланғыңыз келсе, оны тек тиісті сынып әдістерін қолдану арқылы ғана іске қоса аласыз.

Сыныпты мұра еткен негізгі мәселе – объектілердің туынды класты құрастырушыларды шақыру үшін жасалатын кезектілігін қарау.

Бастапқы сыныптың әдістерін деректер мен әдістерді негізгі сыныптың мұра етуі мүмкін болғандықтан, негізгі сынып барлық алынған деректерді және әдістерді туынды сынып объектісін дайындау кезінде қамтуы керек.

Сондықтан туынды класс конструкторының жұмысы негіздік сыныптың конструкторы шақыруымен басталады, оның жұмысы аяқталғаннан кейін туынды класс объектісі құрылады.

Бастапқы сыныптың конструкторы бастапқы сыныптың деректер элементтерін инициализациялауы керек, олардан алынған сыныптың конструкторы мұраға алады.

Бастапқы сыныптың деструкторын шақырғанда, алдымен алынатын сыныптың нысаны жойылуы керек, содан кейін базалық кластың нысаны.

Мұраны қарастырған кезде келесіні қарастырайық: базалық кластың конструкторы мен деструкторы сабақтас сыныпқа мұра етпейді.

3.7.3 Полиморфизм ұғымы

Алдыңғы тақырыптағы мұрагерлік тұжырымдамасы базалық сыныптың деректерін және әдістерін пайдалануға мүмкіндік береді, бірақ мұра екі түрге бөлінеді: статикалық және серпінді мұра (әдістердің статикалық және динамикалық байланыстары).

Статикалық мұра – мұрагерлік, барлық қатынастар құрастырылған кезде жасалады және сипаттама сынып құрылымдарында жазылады.

Динамикалық мұра және байланысқан полиморфизм – кейбір сілтемелер бағдарламаны орындау процесінде тұр.

Полиморфизм – мұраланған сынып тізбегінде бірдей типтегі әдістер.

Полиморфизмнің қасиеттері арнайы виртуалды әдістер мен дерексіз базалық кластар арқылы жүзеге асырылады.

Дерексіз базалық кластардың тұжырымдамасын қарастырайық.

Класс мұрасын пайдалану үшін ODO әзірлеушілері базалық сыныптар құра бастады. Негізгі сыныпта объектілердің белгілі бір жиынтығының деректерін өңдеудің барлық ықтимал әдістері бар, бірақ базалық сыныптарда әдетте ешқандай деректер элементтері жоқ.

Мысалы, сіз геометриялық фигуралар базалық классын жасаған кезде оған аумақ және көлемді іздеу әдістерін қосуға болады. Әлбетте, егер сынып класы немесе сегменті сыныптары алынған болса, жоғарыда аталған әдістер осы нысандар үшін мән болмайды.

Егер базалық класстарда объектілерді құру мүмкін болмаса немесе оның құны жоғалса, мұндай сыныптар дерексіз базалық сынып деп аталады. Дерексіз базалық сыныптар тек ұрпақты құру үшін ғана қолданылады. Әдетте, олар әдістердің жиынтығын ғана жазады, және бұл әдістер әр буынның көмегімен орындалады. Дерексіз базалық сыныптардың мұндай әдістері пайдаланылмайды немесе виртуалды деректер элементтеріне арналған (яғни олар болашақта тұқым қуалау тізбегіндегі кластардың деректер элементтеріне арналған).

Болашақта жасалып жатқан сыныптар мұрагерлік тізбекте қолданылмайды, виртуалды деректер элементтеріне арналған әдістер виртуалды әдістер деп аталады.

C # тілі виртуалды әдістерге сілтеме жасау үшін арнайы термин виртуалды пайдаланады. Мысалға:

```
virtuual publlic doublle ploc() {return 0.0;}
```

Виртуалды сөз ағылшын тілінде «нақты» («факт») білдіреді. Бұл ретте, Қазақстан Республикасының ішкі істер министрінің 2000 жылғы 25 қарашадағы бұйрығымен бекітілген жол қозғалысы қауіпсіздігі туралы, сондай-ақ жол қозғалысы қауіпсіздігі туралы Қазақстан Республикасы заңнамасының талаптарына сәйкес № 648, жол қозғалысы қауіпсіздігінің талаптарын бұзған жағдайда, оның ішінде жол қозғалысы қауіпсіздігі және көлік құралдарын пайдалану ережелері. Бұл механизм динамикалық немесе кейінге қалдыру әдісі деп аталады.

Компилятор виртуалды әдістерді анықтағаннан кейін виртуалды әдістердің атауын және кіру нүктелерінің мекенжайларын жазатын виртуалды әдіс кестесін (VMT) жасай аласыз. Әрбір класс үшін бір виртуалды әдіс кестесі жасалады.

Сонымен қатар, бағдарламаны іске қосқан кезде, меңзер жасақталып, бағдар жасақталған VMT кестесіне жіберілген әрбір нысанға қосылады.

Бұл жағдайда сіз келіспесеңіз, осы мәселе бойынша көмек сұрай аласыз. Осылайша, виртуалды әдістерді пайдаланған кезде бірдей атаудың барлық әдістерінен виртуалды әдіс нысан шақырылған нақты түрге сәйкес таңдалады.

Егер алынған сынып бірдей атаудың виртуальды әдісін өз нұсқасында орындаса, бұл әдіс класта переопределяемость атрибутымен бірге жазылады және орналастыру немесе жабу (жабу) әдісімен жарияланады. Мысалға:

publiic douuble plooc () {. . . }.

Әрбір жалауда сізде виртуалды әдісті анықтаудың қажеті жоқ. Егер әдіс бастапқы сыныпта қажетті әрекеттерді орындайтын болса, әдіс тек ассимиляцияланады.

Бағдарламаны іске қосқан кезде кез-келген негізгі сыныпқа арналған алмастыру әдісін таңдағанда, виртуалды әдістер кестесінде тиісті әдіске сілтеме жазылады және әдіс ата-ана классының қалыпты бөлігі ретінде қызмет етеді. Виртуалды ауыстыру әдісі сол атаудағы негізгі сынып әдісі сияқты параметрлердің жиынтығына ие болуы керек.

Полиморфизм қағидаты дерексіз базалық класс виртуалды әдістерін ауыстыру әдістерін қолдана отырып, «қабаттасу» («қабаттасу») негізделген. Бұл жағдайда әрбір алынған сыныптың мұраланған виртуалды немесе өтпелі әдістері басқаша түрде жүзеге асырылады.

Бұл жағдайда полиморфизм қасиеті әртүрлі мұрагерлік сыныптардың объектілері үшін базалық кластың бірдей виртуалды функциясын қолдану арқылы әртүрлі жолмен орындау мүмкіндігі.

Грек тіліндегі полиморфизм сөзі «әртүрлі жолдар» дегенді білдіреді, ал бұл жағдайда «шақыру үшін бірнеше әдістер». Егер сыныптың жалауы сыныптағы нәрсеге орнатылуға тиіс болса, онда осы әдістерді базалық сыныптарды сипаттағанда виртуал ретінде анықтау ұсынылады. Егер әдіс иерархияның барлық сыныптарында бірдей орындалса, бұл әдіс қарапайым әдіс ретінде анықталуы керек.

3.8 Класс ұғымы

C # – объектілі-бағытталған бағдарламалау тілі. Оның негізгі тұжырымдамасы – сынып. Барлық тілдерде C # тілін үйрену сынып құрылымын қолдандық.

C # бағдарламалау кітаптарындағы анықтамалардан бастайық.

Фаронов В.В анықтамасы бойынша, сынып алдында қызмет көрсету сөздік класы бар код фрагменті анықталады [1]: «Класс – деректер түрі, яғни сыныптың нақты даналары - объектілер жасалатын« схема ». А.П. Павловская Анықтама [2]: «Класс – сыныптың даналары деп аталатын нақты нысандардың жиынтығының сипаттамалары мен әрекеттерін анықтайтын жалпылама түсінік».

Бұл жағдайда, C # тіліне дейін, СВВ бар және класс тұжырымдамасы қазірдің өзінде қолданылған. Біз әдебиетте орын алған ең қысқа анықтаманы береміз: «сыныптар – бағдарламашы анықтаған тип».

Бұл анықтамада сыныптың өте маңызды ерекшелігі – массивтер, белгілер немесе құрылымдарға қарағанда жаңа деректер түрі көрсетілген. Дегенмен, бағдарламашы анықтаған қандай да бір сынып болуы мүмкін емес. Сыныпты анықтаудағы екінші маңызды элемент – сыныптың болуы, қысқаша жазу сабақтары және жеңіл есте сақтау.

Класс – өрістерден, әдістерден және оқиғалардан тұратын деректер түрі.

Деректер түрі – сыныптың данасы деп аталатын көптеген нысандардың қасиеттері мен әрекеттерін сипаттайтын семантикалық құрылғы.

Семантикалық сынып сынып өрістері, әдістер және сынып оқиғалары деп аталатын деректер сипаттамасын білдіреді.

Кейбір авторлар белгілі бір топтың кеңейтімі болып табылатын сыныптарды бөледі, мысалы, басқару класы. Мұндай сыныптарда қосымша қызметтер бар. Олар жоба дизайнының жеке архитектуралық бірлігі.

Класс жазбасының форматын қарастырыңыз. Сынып жазба форматында, сынып қызметі туралы сөзден кейін, оның аты жазылады, ал содан кейін жақшаларда – сыныптың денесі. Бұл сынып сипаттамаларының ең қысқа құрамы. Сыныптың жалпы сипаттамасы келесі форматта беріледі (қосымша элементтер тік жақшаларда берілген):

[атрибуттар] [спецификаторлар]

class кластың_атауы [: түп тегі]

{ кластың_денесі }

Бұл жағдайда атрибуттар – сынып туралы қосымша ақпарат береді; спецификаторлар – класс құрамына кіруге жағдай жасау:

– сөздің шығу тегі (ата-аналар) – базалық сыныптар;

– мән класс элементтерінің құрамын – сыныптың денесін анықтайды.

Класс жариялауға мүмкін ерекшеліктер дерексіз, мөрмен және қорғалған. Олар мұраны қарастырған кезде егжей-тегжейлі талқыланады. Жеке, қоғамдық, статикалық және ішкі спецификациялар бағдарламалар үшін сыныптың қолжетімділігін анықтайды. Жеке сипат сыныптың көрінуін

толығымен жауып тастайды, ал жұртшылық спецификациясы сынып бағдарламасының кез келген бөлігін көрсетеді (кіруді анықтайды). Негізінде, класс ішкі қолжетімділік сипаттамасы болады. Сынып құрылымда анықталады және оған қол жетімді болады. Статикалық спецификатор кластың тиісті айнымалысын (сынып нысанын) жасамай, сыныпты және оның элементтерін пайдалануға мүмкіндік береді.

Барлық спецификацияларды сыныпта немесе жеке мүшелерде, мысалы, өрістерде, әдістерде қолдануға болады.

Сыныптың пішімдеудің кейбір қосымша элементтері келесі бөлімдерде талқыланады.

Класс белгілі бір мәндермен «толтырылған» үлгі болып табылады, яғни класс түрінің айнымалысы – сынып данасын құруға арналған деректер түрі.

Бағдарламада әр түрлі мәндерді беру, сыныптың әртүрлі нысандарын көреміз, бірақ сынып түрі өзгертілмейді.

Сыныптың аттар кеңістігінде немесе басқа класта сипатталуы мүмкін. Соңғы сценарийге сәйкес, класс қабатты класс деп аталады.

C#-те, сынып – анықтамалық түрі және сыныптың нысанын компьютердің жадына орналастыру үшін жаңа операторды пайдалану керек.

3.8.1 Класс құрамы

Сыныптағы органда деректер процессорлары, әдістер және оқиғалар болуы мүмкін. Бұл класс компоненттері әдетте сынып мүшелері деп аталады.

Сыныптың негізгі элементтерін және олардың функцияларын қарастырайық:

– сыныптың тұрақты мәндерінде өзгермейтін мәндер;

– сыныптық өрістер (класс айнымалыларының түрлері мен атаулары));

– класс әдісі, белгілі бір атаумен бағдарламаның код фрагменті, сынып деректерімен жұмыс істеу үшін арналған; сыныптың қасиеттері – олардың өрістерінің мәндерін алмасу (оқу немесе жазу) үшін сыныптарға мүмкіндік беретін әдістер жиынтығы;

– сынып конструкторы класс нысандарын құру және сынып өрістеріне мәндерді тағайындау үшін арнайы сынып әдісі болып табылады;

– сыныптың деструкторы – объектіге бөлінген ресурстарды босату тәртібі;

– тізімдегі сынып оқиғалары – арнайы сынып әрекеттеріне немесе бағдарламадағы нақты өзгерістерге жауап беруге көмектесетін арнайы әдістер;

– тізімдер, құрылымдар, сыныптар, делегаттар, интерфейстер сияқты деректер түрі.

indexer – сыныптың деректер мүшелеріне қол жеткізу құралы; операциялар операциялардың негізінде жүзеге асырылатын сынып объектілеріне арналған арнайы операциялар болып табылады.

Класс деректері айнымалылар, тұрақты немесе сынып (өріс) болуы мүмкін. Деректерді сыныпқа жариялағанда, әдетте, сіз қол жеткізу белгілерін, мысалы:

private int a;

Класс деректерін жариялау кезінде оларды жалпы жазу пішімі мынандай болады:

[атрибуттар] [спецификаторлар]

[const] тип атауы [= бастапқы_мәні].

Әдетте, сынып мәліметтері «бағдарламаға жабылады» – жеке спецификатор қолданылады. Егер FA әлі де амортизацияланбаған болса, онда қалдық құнын жеке адамға сату керек.

Белсенді емес деректер мен әдістер үшін жеке спецификатор қолданылады.

Нысан – класс элементінің мәндері компьютер жадында сақталатын жеке аймақта орнатылған сынып түрінің айнымалысы.

Алайда сыныпта барлық нысандар үшін ортақ болып табылатын бір данаға статикалық элементтер болуы мүмкін. Статистикалық деректер көбінесе сынып деректерімен аталады, ал қалған бөлігі сынып данасының деректері болып табылады, яғни нысандар.

Кейбір элементтер (әдістер) және сынып өрістері нысан жасағаннан кейін ғана қол жеткізуге болады. Егер рұқсат етілген болса, нүкте оларды пайдалану үшін пайдаланылады, мысалы, зерттеу объектісіндегі атау өрісіне қол жеткізу үшін келесідей жазыңыз: stud.name = ivanov; нысан үшін, сіз сыныптың әдісін шақыруыңыз мүмкін, мысалы, stud.poisk (a), мұнда poisk (int a) – сыныптың әдісі, синтаксис бойынша класс сыныбының болуы мүмкін. Бұл жағдай ортақ. Ішкі класс өзін және оның ұрпақтарын құрған сыртқы сыныпта қолданыла алады. Ішкі сыныптарда әдетте жеке немесе қорғалған кіру рұқсаттары болады.

3.8.2 Класс әдістері

Әдіс – сынып деректерімен және әдістерімен жұмыс істеу үшін пайдаланылатын сынып атауы бар функционалды элемент. Әдістемелер оған қолданылуы мүмкін әрекеттер жиынтығын анықтайды (сыныптың жұмыс істеу жолын анықтайды). Әдіс тек бір рет сипатталады және әртүрлі сынып нысандары үшін бірнеше рет аталуы мүмкін.

Класс әдістеріне арналған жалпы жазылу форматы келесідей:

[атрибуттар] [спецификаторлар] әдіс типі әдістің атауы ([параметрлер])

{әдістің денесі}

Мысалы:

static void Main(string[] args)

{ }

Ең кең таралған ерекшеліктер жеке, қоғамдық және статикалық болып табылады.

Жеке сипаттамамен жарияланған сыныптың кез-келген әдісі тек осы сыныптың әдістері үшін қол жетімді болады.

Қоғамдық специфика осы әдісті бағдарламаның кез келген бөлігінде пайдалануға мүмкіндік береді.

Статикалық спецификатормен сыныпты нысан жасамай әдісті «сынып деңгейінде» пайдалануға болады. Бұл өте маңызды, себебі біз статикалық әдістерді жиі қолданамыз.

Өтініште қол жетімді әдіс класс нысандарын жасамай, сынып конструкторын (ол нысанды жасайды) ұсынады.

Басқа әдістерге тек сынып объектісі жасалғаннан кейін ғана қол жеткізуге болады.

Егер сіз квалификаторды көрсетпесеңіз, сыныптың спецификаторы әдісінде жеке болып табылады.

Әдістің түрі кез-келген қолданба түрінде немесе стандартты C# тілі түрінде немесе бос орын түрінде анықталуы мүмкін. Мысалға:

int kol(int a) { … }

public double sym(out float r) { … } public void poisk(ref float s) { … }

public int funkcij(int a, out int b, params int[] c) { … }

Егер бұл жағдайда әдіс түрі көрсетілсе (бос орын түрінен басқа), онда әдіс денесінің соңғы операторы әдісті пайдалану нәтижесін қайтаратын қайтарушы оператор болып табылады. Әдіс айнымалыға тағайындалуы немесе операторлардағы өрнек ретінде қолданылуы керек. Бұл әдістер функция деп аталады.

Егер түрін қабылдамас бұрын көрсетілсе, әдіс қайтарым операторы арқылы оның жұмысының нәтижелерін қайтармауы керек (әдіс денесінде қайтарушы оператор болмайды). Бұл әдіс әдетте рәсім деп аталады. Бағдарлама құрастырған әдіс-идентификатордың атауы. Әдістің аты (аты) мәні оның жұмысына, мысалы, сим, макс, улан және т.б. байланысты болуы керек.

Әдіс параметрлері (ресми параметрлер) әдіс пен бағдарлама арасындағы деректермен алмасу үшін пайдаланылады. Әдетте әдісті шақыру параметрлері «параметрлер» әдісін білдіреді.

C#-де келесі әдіс параметрлері бар:

– мәндерді анықтайтын параметрлер (мәндердің мәндері, яғни әдіс қабылдайтын енгізу параметрлері);

– шығыс параметрлері (сөзден тыс));

– эталондық параметрлердің мәні (ref-термині арқылы белгіленеді);

– массивный параметр («params» функциясы арқылы белгіленеді).

Функция сөздері жоқ мәндерді анықтау параметрлері. Сынып әдісінің параметрлері үтірлермен бөлінеді. Әдісте бір жиым параметрі және параметр тізіміндегі соңғы параметр болуы керек. Егер әдіс мәндерді анықтайтын параметрлерді жариялаған болса, онда бұл әдіс кейбір айнымалылардың

кейбір түрлерінің көшірмелерін алғанын көрсетеді. Әдіс осы көшірмелердің мәнін өзгерте алады, бірақ олардың түпнұсқасы (бағдарлама) өзгеріссіз қалады.

Әдіс аяқталғаннан кейін мәндерді анықтайтын параметрлер компьютердің жадынан жойылады.

Әдістің тұтынылатын параметрлері нәтижелерге қолдану үшін жеткізіледі. Бұл жағдайда, қандай параметрлердің әдіс денесінен келетіндігіне байланысты, кейбір мәндерді тағайындауыңыз керек, әйтпесе қосымша жинақта қате туралы хабар пайда болады.

Егер сілтеме параметрлері әдіспен жарияланса, әдіс тиісті айнымалылардың мекен-жайын қабылдайды және оларды алгоритміне сәйкес пайдалана алады (жаңа мәндерді жазу және оқу).

Айнымалылардың кез-келген санымен жұмыс істеу үшін әдіске жарияланған массив параметрі кез-келген нақты айнымалылар санымен жұмыс істеуге арналған. Сонымен қатар, params қызмет сөзінен кейінгі ресми параметр кез-келген өлшемдегі деректер массивімен теңестіріледі.

Осылайша, әдіс қажетті деректерді параметрлерге (мәндерді және анықтамалық параметрлерді анықтайтын параметрлерді) беруге мүмкіндік береді және әдіс оның жұмысының нәтижелерін (бастапқы параметрлер мен анықтамалық параметрлер) қайтара алады.

Әдістің корпусында кейбір алгоритмдерді орындайтын бағдарлама коды бар. Әдіс формальды параметрлермен қатар әрекеттер моделі ретінде пайдаланылады. Бағдарламада айнымалы мәндерді тазарту үшін ресми параметрлердің орнына қолдануға болады, нақты айнымалы мәндер үшін әдістердің нақты параметрлері мен үлгілері қолданылады.

3.8.3 Объект құрылымы

Бұл сынып айнымалы деп аталады. Нысанның ауыспалы болуына байланысты ол компьютердің жадында бос орын алады. Нысанның жадында нақты сақталғанын қарастырыңыз. Сыныптың барлық деректер өрісі сақталады.

Нысанның дайындалуында автоматты түрде осы параметрдің (сілтеме) арнайы өрісі объектінің мекен-жайын сақтайды.

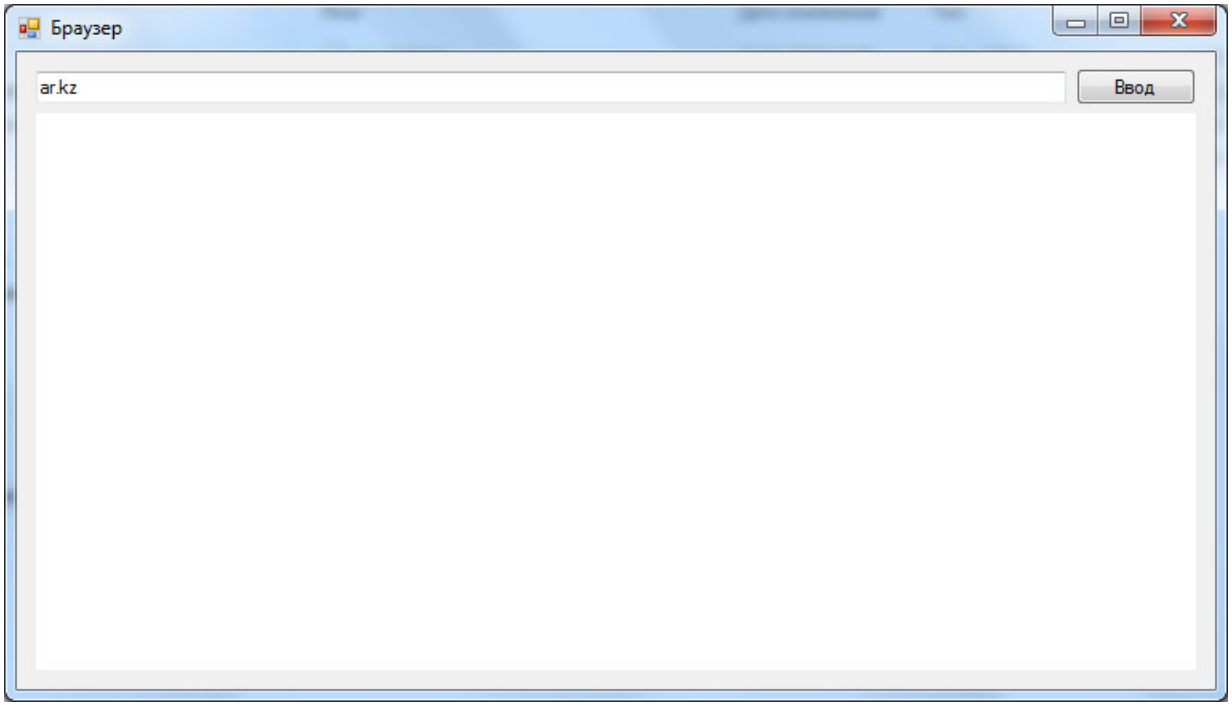
Класс объектілері мен әдістерін қосу осы параметр бойынша жүзеге асырылады. Әрбір сынып әдісі осы параметрді ағымдағы нысанның элементтерімен жұмыс істеу үшін тікелей пайдалана алады. Себебі бұл әрқашан қолданыстағы объектіге (ағымдағы жылы бағдарламаның жұмыс істеу нысанында), ағымдағы объектінің элементтерімен жұмыс істеу әдістеріне сәйкес келеді.

Осы нұсқаулықты пайдалану әрбір нысан үшін класс әдістерінің көшірмелерін жасаудан аулақ болуға мүмкіндік береді. Класс әдісі әр нысан үшін қайталанбайды.

Бағдарламаның сипаттамасы. Бұл бағдарлама компьютерлік техниканы сатып алуға арналған, әр таңдауға қарай, автоматты түрде компьютерлік жабдықты алады, компьютерлік техника тізімінен модельге немесе атауына қарай іздейді. Бағдарлама компьютерлік техниканы, негізгі дерекқорды жеткізушілерден алынған деректерді алады. 3.9-сурет бағдарламаның негізгі бетін көрсетеді.

Бұл бағдарлама әрбір сатып алушының таңдауына байланысты тиісті компьютерлік техникаға қажеттіліктеріңізді қанағаттандыруға арналған. Бағдарлама негізгі компьютерлік жабдықтардың сатушыларынан деректерді алады. Ақпараттың тиімді пайдаланылуы тұтастай алғанда ұйымның өнімділігі үшін маңызды. Компьютерлік техниканы сатып алуға келген клиенттердің құжаттары толық C # бағдарламалау ортасында орындалады. Бұл кешен түрлі коммуникациялық дүкендерде сатушылардың жұмысын автоматтандыруға арналған. Бұл бағдарлама әрбір сатып алушының таңдауына байланысты тиісті компьютерлік технологияға қажеттіліктеріңізді қанағаттандыруға арналған. Бағдарлама негізгі құрал-жабдықтардан компьютерлік жабдық жеткізушілерден деректерді алады. Ақпараттың тиімді пайдаланылуы тұтастай алғанда ұйымның өнімділігі үшін маңызды.

Бағдарламаның жұмысы компьютерлік технологиялар мен оқырмандарға қатысты ақпарат, сондай-ақ енгізілген деректерді басқару негізінде деректерді енгізу нәтижесінде жүзеге асырылады. Автоматтандыру жүйесін одан әрі дамыту жүйені талдауды және қолдауды қамтиды.



3.8-сурет – Бағдарламаның негізгі беті

Интерактивті бетте орнатылған дүкенде компьютерлік техниканы сатып алушының деректерін сақтау үшін дерекқор қажет:

– шарттың санаттары бойынша анықталған объектілердің интерактивті каталогы болуы керек;

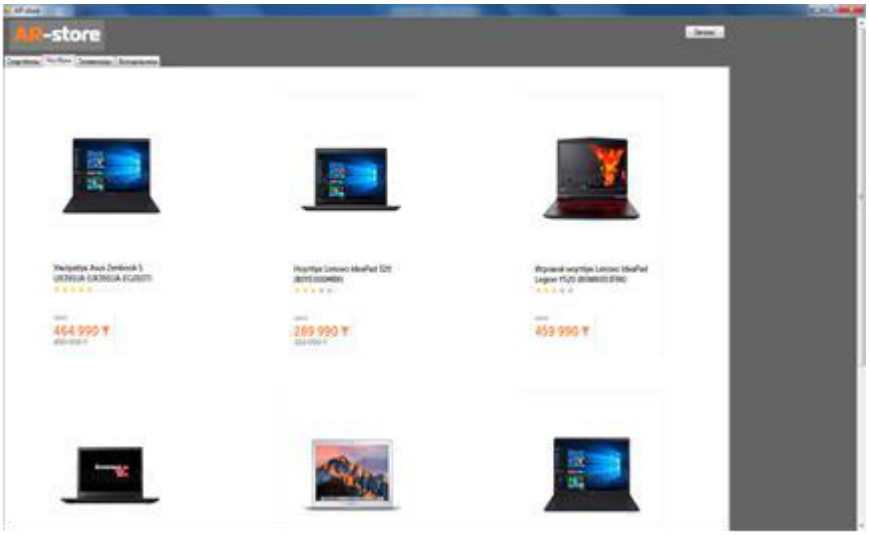
– тұтынушы өтініштерін өңдеу;

– клиенттерге негізделген арнайы бетті ашу;

– компьютерге арнайы бетті ашу мүмкіндігін беру керек, яғни компьютерлік жабдықтың түрі, кепілдік мерзімі, пайыз;

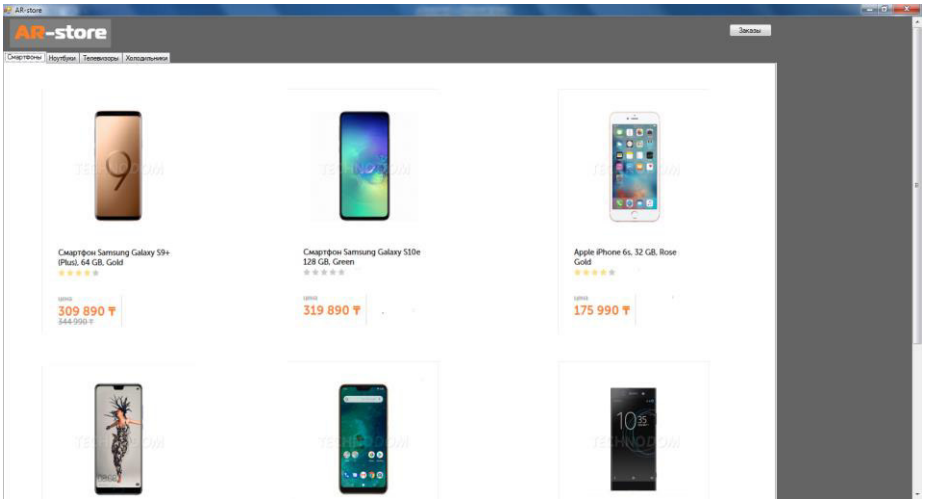
– webadmin интерфейсі.

3.9-сурет жаңа компьютерлік аппараттық модельді енгізу терезесін көрсетеді.

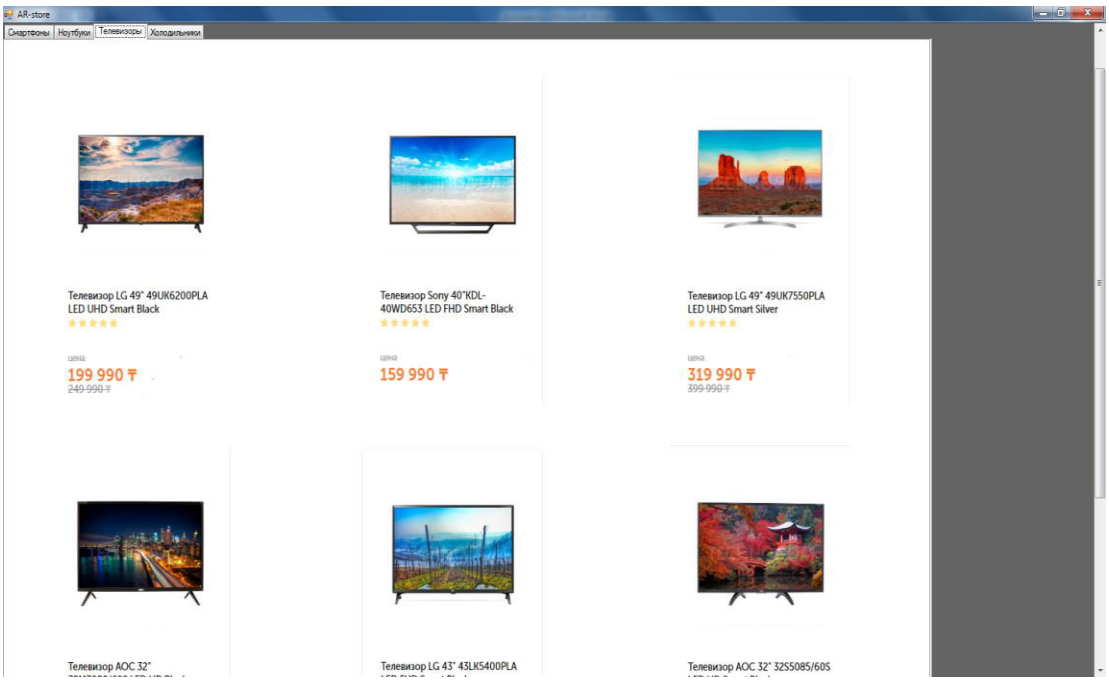


3.9-сурет – Жаңа модель енгізу

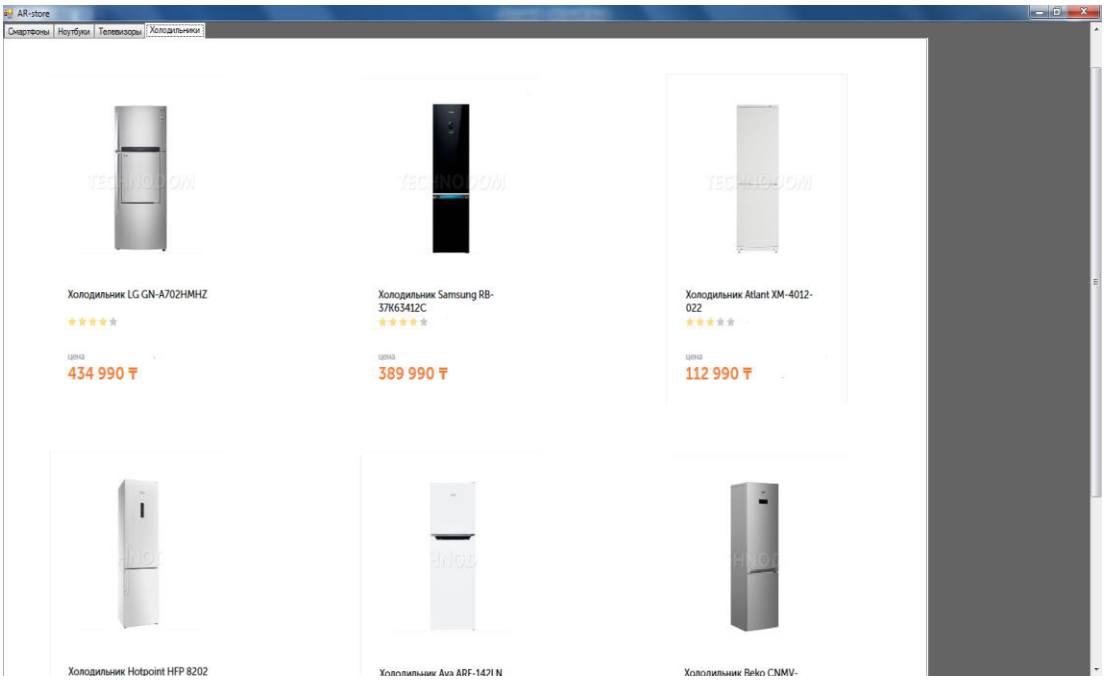
Сонымен қатар басқа тауарларды қарауға болады.



3.10-сурет – Жаңа модель енгізу



3.11-сурет – Жаңа модель енгізу

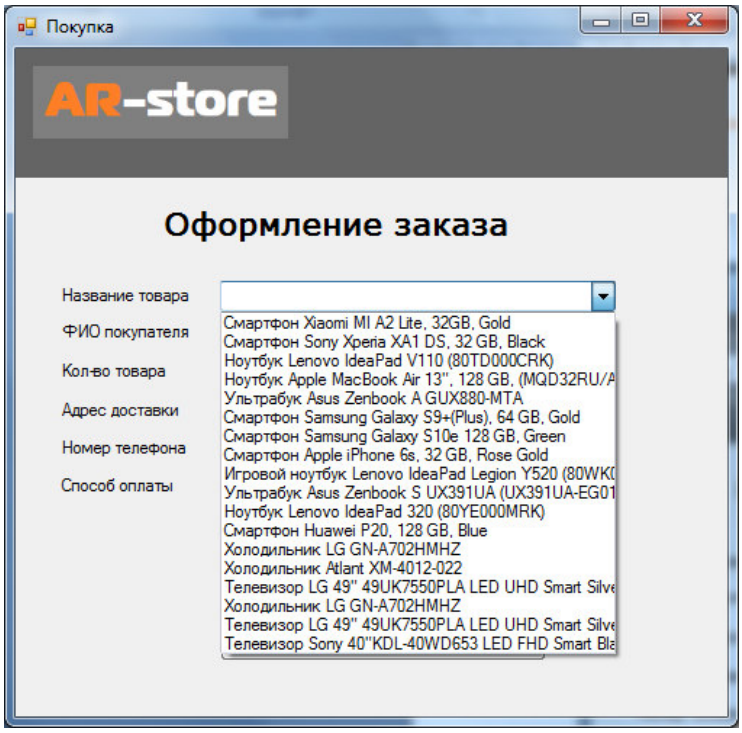


3.12-сурет – Жаңа модель енгізу

«Мінездемесі» батырмасын басқанда анықтама мен мәліметтерді тез шақыру формалары шығады. Ол 3.13-суретте көрсетілген.

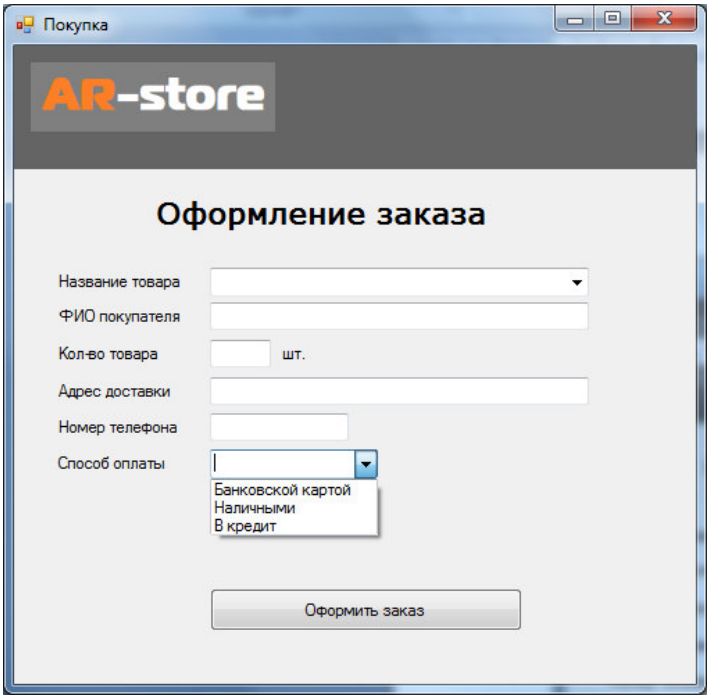
3.13-сурет – «Мінездемесі»

Клиенттің таңдауы бойынша тауарға тапсырыс беріледі.

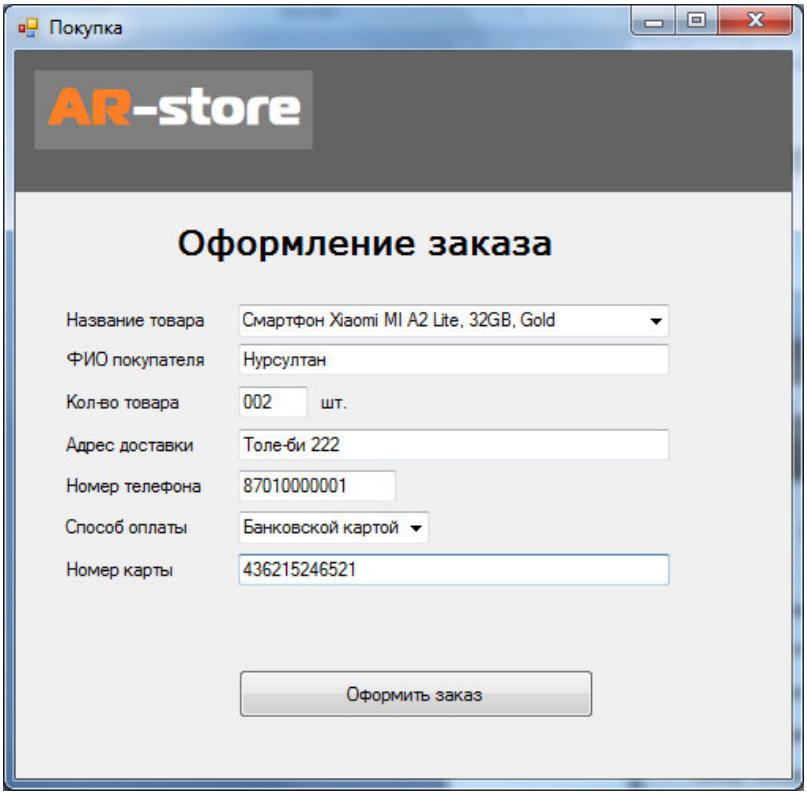


3.14-сурет – «Мінездемесі»

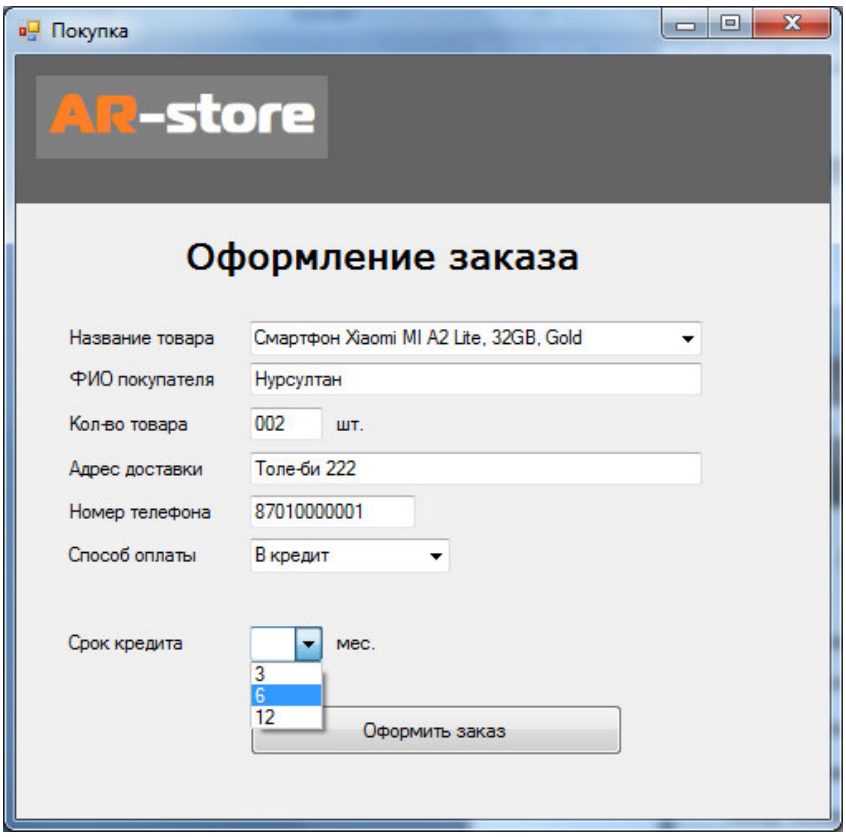
Сонымен қатар компьютер техникаға төлем ақпаратты қарастырылған. 3.15-суретте көрсетілген.



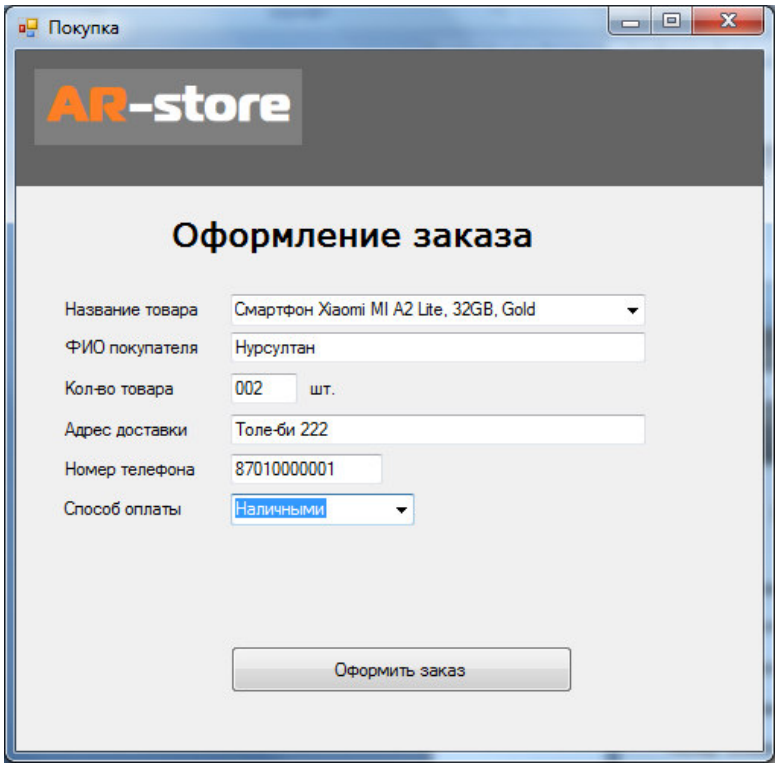
3.15-сурет – Компьютер техникаға төлем ақпараты



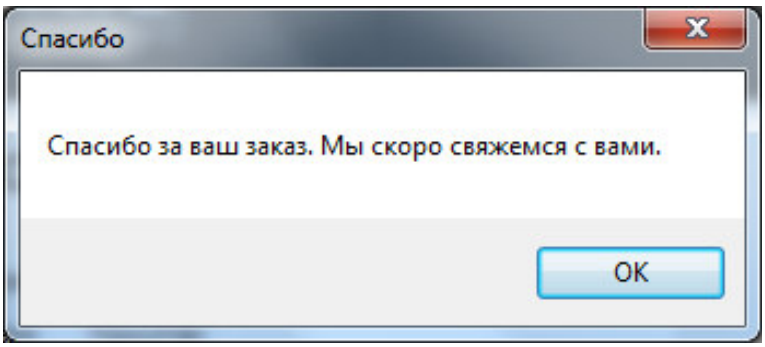
3.16-сурет – Тапсырысты қабылдау



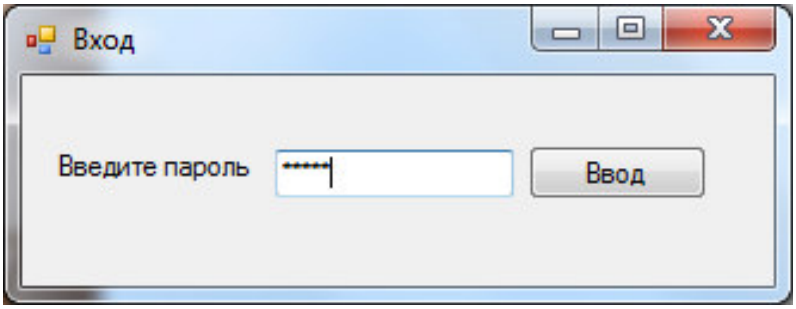
3.17-сурет – Тапсырысты кредит бойынша қабылдау



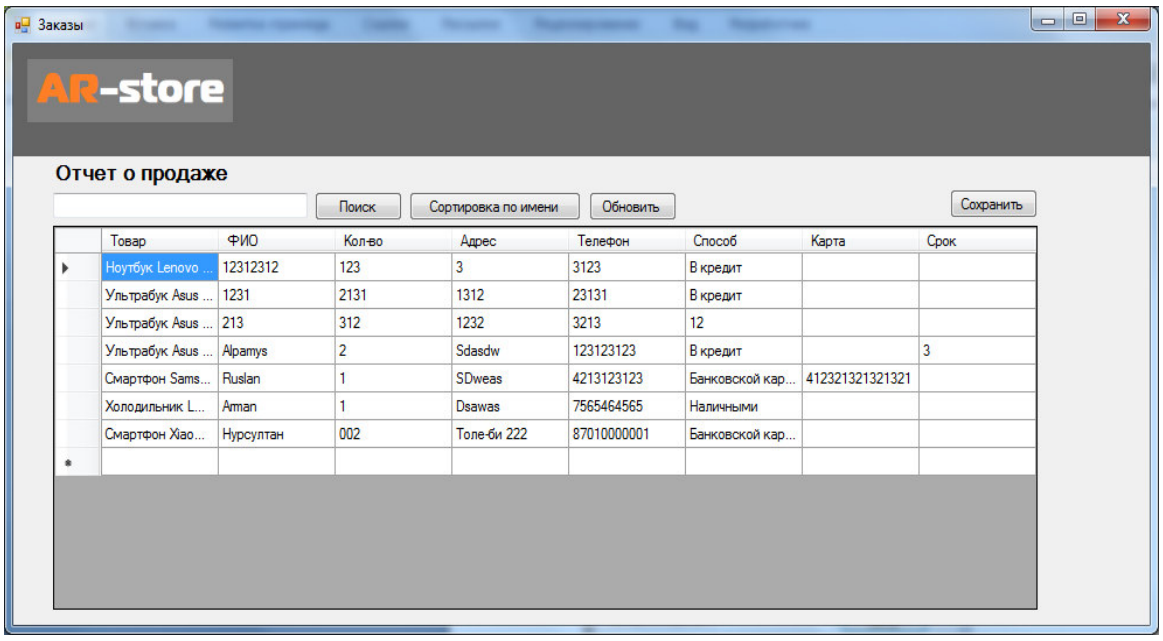
3.18-сурет – Тапсырысты қолма-қол төлем бойынша қабылдау



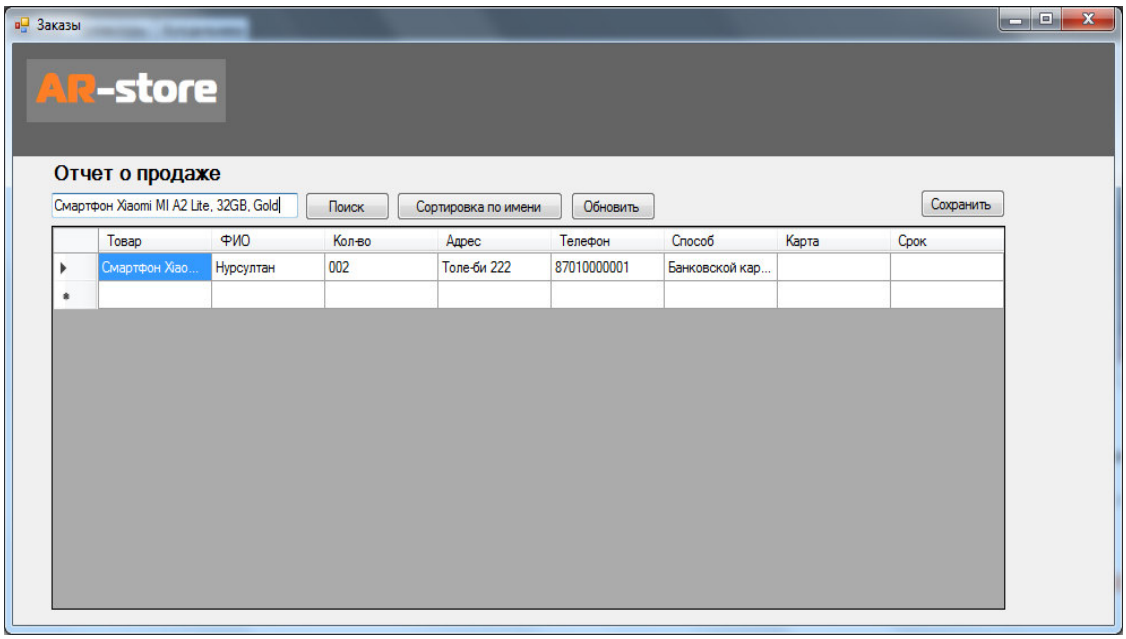
3.19-сурет – Тапсырысты қабылданған туралы ақпарат беру



3.20-сурет – Сату бойынша мәліметтерді Менджердің жеке парақшасына кіру



3.21-сурет –Менджердің жеке парақшасындағы тапсырыстар тізімі



3.22-сурет –Менджердің жеке парақшасындағы тапсырыстар тізімінен іздеу нәтижесі есеп түрінде көрсету

4 Экономикалық бөлім

«Ar store» интернет-дүкені компьютерлік техника және олармен бірге жүретін тауарлар саласындағы сауда қызметін жүзеге асырады. Бұл компакт-диск, mp3, DVD дискілері, бағдарламалық жасақтамасы бар дискілер; дискілерге, бейне / аудиоға арналған префикстердің ауқымы.

Ар дүкен әлемнің ең перспективалы клиенттеріне (30 жасқа толмаған және аға ұрпақ) келуді қамтамасыз ететін Мира көшесіндегі 32 орналасқан. «Славтек», «МЕГАПОЛИС» сияқты әйгілі дүкендерге жақындық, сондай-ақ клиенттердің ағыны қамтамасыз етеді.

«Бюро» дүкені қаладағы «негізгі бәсекелестердің» (шамамен 900 метр) орналасуы мен қаланың негізгі сауда орталықтарынан қашықтығының белгісі. Нәтижесінде, әлеуетті клиенттердің көптеген дүкендердің болуы туралы хабардар болмауы.

Потенциалды тұтынушыларға «AR Store» -тің басылымдарда немесе радиода, теледидарда артықшылықтарын мезгіл-мезгіл ескерту қажет.

Ірі қалаларда көптеген дүкендер «супермаркеттер» қағидаты бойынша жұмыс істейді, онда сатып алушы тауарға бағдарлы позицияны неғұрлым қолайлы жарықпен жүзеге асырды, есептегішті еркін сатып алуға болады, тексереді.

Мұндай тауар нарығын зерттеудің ерекшелігі – белгілі бір қажеттілікті қанағаттандыру үшін түрлі нысандар мен құралдардың өзара байланысын ескеру.

4.1 Бағдарламалық және аппараттық қамтамасыз етуді сатып алуға, ұстауға және пайдалануға арналған шығындар

Шығындар құны желілік дүкендерде баға тізімінен алынған және 3.1 кестеде көрсетілген.

4.1-кесте – Бір жылға арналған бағдарламалық қамтамасыз ету мен аппараттық құралдардың құны

Атауы	Саны	Бағасы, тг
1С-Битрикс «управление сайтом»	1	5400
SSL - сертификат	1	4199
TimeWeb	1	960
Қорытынды		10599

4.2-кесте – Материалдар

Атауы	Саны	Бағасы, тг
USB-флешка KINGSTON DataTraveler SE9 16Gb (DTSE9H)	1	690
Диск DVD-R 4.7Gb SlimCase VS	1	44
Қорытынды		734

Электр энергиясына арналған шығындар кезеңнің ұзақтығына байланысты. Заңды тұлғалар үшін программалық жасақтамаға және Алматы қаласының тарифіне жұмсалатын кВт / сағ саны - 4,09 құрайды. Шығындар 4.3-кестеде көрсетілген.

4.3-кесте – Электр энергиясына шығындар

Жүйе элементі	Берілген қуат, кВт	1 сағ 1 кВт мөлшері	Жұмыс уақыты	Жалпы құны, тг
Acer A315-21G41DY NX.GQ4ER.001	0,057	4,09	398	106,18
Қорытынды				106,18

Жабдықтарға арналған амортизациялық шығындар оларды пайдалану кезеңінде, яғни іске асыру сатысында және программалық қамтамасыз етуді қолдануда. Ақша амортизациясының көрсеткіші - ағымдағы шығындар болып табылатын тозу. Амортизация сомасы амортизация нормаларына негізделген. 1 сағаттан кейін амортизация жүйенің (4.1) формуласына сәйкес есептеледі.

$$Ач = Ф_{перв} \cdot \frac{a}{F_Д}, \quad (4.1)$$

мұндағы, $Ф_{перв}$ – жүйенің немесе жеке элементтердің бастапқы құны;
a – амортизация нормасы (0.2);
$F_Д$ – жыл ішіндегі жұмыс уақытының қоры (2500 сағат)).

4.4-кесте – Амортизациялық аударымдар есебі

п/п	КТС элементі	Ф перв	F д	Ач	Жұмыс уақытының саны	Жалпы құны, тг
1	Acer A315-21G41DY NX.GQ4ER.001	22900	2500	1,9999	398	795,9602
Қорытынды						795,9602

Есептеу нәтижелері бойынша бағдарламалық қамтамасыз ету мен аппаратураны сатып алу, ұстау және пайдалану құны 12235,14 теңгені құрады.

4.2 Жалақы шығындары

Жалақы мөлшерін есептеу үшін орташа жалақыны көтеру керек. Бағдарлама жұмысының күрделілігі бойынша сағаттық жылдамдық, әрбір адамға арналған адам-сағаттық жүйенің даму сатылары.

Орташа сағаттық ставка (4.2) формуласы бойынша есептеледі.

$$З_ч = \frac{З_м}{168},$$ (4.2)

мұндағы, $З_ч$ – бағдарламашының орташа сағаттық ставкасы;

$З_м$ – бастауыш бағдарламашы орташа айлық мөлшерлемесі (20000 тг).

Зч= 20000/168 = 119 теңге

$З_ч$ = 200 теңге – жұмыс басшысы мен консультанттың орташа сағаттық ставкасы тіршілік қауіпсіздігі;

$З_м$ = 250 теңге – экономикалық бөлім бойынша консультанттың орташа сағаттық мөлшерлемесі.

Алынған мәліметтерге сүйене отырып, барлық қызметкерлердің жалақысының даму сатысын есептеуге болады. Нәтиже 4.5-кестеде келтірілген.

4.5-кесте – Негізгі және қосымша жалақы есебі

п/п	Жұмыс мазмұны	Еңбек қарқындылығы, адам/сағ	Негізгі жалақы, тг
1	Пәндік саланы талдау	10	1880
2	Есеп қою	4	1040
3	Әзірлеу, техникалық тапсырмалар	10	2190
4	Интерфейспен жұмыс	20	2380
5	Бағдарламаның модульдерін әзірлеу	86	4420
6	Жүйені тестілеу	94	2900
7	Құжаттау	11	2450
8	ДРК басшылығы (ғылыми жұмыс жетекшісі мен кеңесшілер)	20	3100
Қорытынды			20360

4.3 Әлеуметтік қажеттіліктерге аударымдар

Бірыңғай әлеуметтік салық негізгі және зейнетақы қорына, әлеуметтік қамсыздандыру қорына, сақтандыру және медициналық сақтандыруға жатады. 2019 жылы бұл көрсеткіш (4.3) формула бойынша есептеледі.

$$ЕСН = 30\% \cdot (ЗП_{осн} + ЗП_{доп}),$$ (4.3)

мұндағы, ЕСН – бірыңғай әлеуметтік салық;

ЗП – негізгі жалақы;

$ЗП_{қж}$ – қосымша жалақы.

ЕСН = 0,30*(20360+3054) =7024,2 (тг.)

Жүйені енгізу үшін жалпы шығын сметасы:

4.6-кесте – Жалпы шығындар сметасы

п/п	Шығын элементі	Құны, тг
1	Сатып алу, ұстау және пайдалану	12235,14
2	Негізгі жалақыға арналған шығындар	20360
3	Қосымша жалақыға арналған шығындар	3054
4	Әлеуметтік қажеттіліктерге аударымдар	7024,2
Қорытынды		42673,34

4.4 Экономикалық тиімділікті бағалау

Осылайша, онлайн-дүкеннің даму және іске асыру құны (бұдан әрі – ЭОТ) жобалық нұсқасы 42673,34 теңгені құрайды. Қарастырылған көрсеткіштерден басқа, интернет-дүкеннің аяқталған нұсқасы мен жоба нұсқасының (бұдан әрі – АТ) арасындағы айырмашылықты есептеу қажет. Сатып алынған интернет-дүкеннің құны (сатып алынған) 74800 теңгені құрайды.

Яғни, АТ = 74800 – ЭОТ = 74800 – 42673,34 = 32126,66 тг;

Осылайша, интернет-дүкеннің дизайн нұсқасын жасау осы өнімнің соңғы нұсқасын сатып алу үшін пайдалы.

5 Өміртіршілік қауіпсіздігі

Ғылыми және технологиялық прогрестің дамуы халықтың өз міндеттерін қауіпсіз орындауға қабілеттілігінің маңызды рөлін атқарады. Осыған байланысты еңбек қауіпсіздігі және адамның өмір қауіпсіздігі ғылымы құрылып, дамиды.

Өміртіршілік қауіпсіздігі – қоршаған ортаға адамның қауіпсіздігін қамтамасыз етуге, оның денсаулығын сақтауға, зиянды және қауіпті факторлардың қолайлы құндылықтарға әсерін төмендету жолымен әдістер мен құралдарды әзірлеуге, төтенше жағдайлардың салдарынан зиянды шектеу шараларын әзірлеуге бағытталған шаралар жиынтығы. Бейбітшілік пен соғыс уақытында.

Қызметкерлердің денсаулығын қорғау, еңбек жағдайлары қауіпсіздігі, кәсіби аурулардың және кәсіби жарақаттардың жойылуы адамзат қоғамының басты проблемаларының бірі болып табылады. Жұмыстарды ғылыми ұйымдастырудың прогрессивті формаларын кеңінен қолдану қажеттілігі, еңбек ресурстарын азайту, қолтық, кәсіби жарақаттану, кәсіптік ауруларды және өндірістік жарақаттануды болдырмайтын жағдайларды жасау қажеттілігі айтылмаған.

Осы жобада компьютерлік технологияны енгізумен айналысатын интернет-дүкен үшін бағдарламалық жасақтама әзірледім.

Қазіргі уақытта компьютерлік техника адам қызметінің барлық салаларында кеңінен қолданылады. Компьютерде жұмыс істегенде адам қауіпті және зиянды өндірістік факторларға ұшырайды: электромагниттік өрістер, инфрақызыл және иондаушы сәуле, шу және діріл, статикалық электр т.с. Дүкенде 6 адам (4 ер адам және 2 әйел) жұмыс істейді. Дүкеннің бастапқы деректері 5.1-кестеде келтірілген.

5.1-кесте – Бастапқы деректер

Үй-жайдың типі	Параметры помещения				Разряд бекер. жұмыстарды	$\rho_{пот.}$	$\rho_{стен}$	$\rho_{пол}$	$h_{нок}$,м	Жарық белдеуі	Нзд	Қашықтық. жақын тұрған ғимаратқа дейін, Р
	L, м	В, м	Н, м	$h_{ок}$, м								
Интернет дүкеннің үй-жайы	6	8	6	3	I, а	70	50	30	0,5	Алматы облысы	15	12

Бөлмеде LG P09SP2 Dual INV кондиционері, HP ENVY AiO 27-b113ur 2CW08EA (6 дана) және принтер Canon i-SENSYS LBP212dw ақ-қара (3 дана) компьютерлерімен жабдықталған.

Компьютермен жұмыс істеу операторлардың нейро-эмоционалдық жүктемесі, визуалды жұмысдың жоғары кернеулері және ДК пернетақтасымен жұмыс істеген кезде қолдың бұлшықеттеріне айтарлықтай жүктемеден

тұрады. Жұмыс орнының тиімді құрылымы мен элементтері өте маңызды, бұл адамның оңтайлы жұмыс жағдайын сақтау үшін маңызды.

Бұл бөлмеде жұмыс істеу шарттары. Микроклиматтың параметрлері кеңінен өзгеруі мүмкін және адам өмірінің қажетті шарты терморегуляция есебінен дене температурасының тұрақтылығын сақтау, яғни дененің қоршаған ортаға жылуды жеткізуді реттеу қабілеттілігі болып табылады.

Бөлмеде желдету жүйесі болмаса, адамдарға көміртегі диоксиді мен басқа зиянды заттар жинауға болады, бұл жаман көңіл-күйге және адамдардың тиімділігін төмендетуге әкеледі. Бөлмені ашық Windows шаң буларымен және пайдаланылған газдармен байланыста желдетуге көмектеспеңіз. Бұл проблемалардың көпшілігі ауа тазартуға, салқындатуға, жылытуға, ауа мен құрғатуға қабілетті кондиционерлермен шешіледі.

ГОСТ 30494 сәйкес ГОСТ 12.1.005 сәйкес үй-жайларды жылыту және желдету кезінде микроклимат параметрлерін қамтамасыз ету.

Осыған байланысты желдету және ауа баптау жүйелерін қолдану қажет. Біздің бөлмемізде жаңа LG P09SP2 Dual INV кондиционері бар. Салқындату және жылытудың негізгі режимдері.

Өндіріс ортасының кемшіліктерінің бірі – компьютерлермен жұмыс істеу кезінде туындайтын шудың жоғары деңгейі. Компьютерлер жаңа, шу жоқ болғандықтан, шу деңгейі рұқсат етілген деңгейден аспайды.

Бірнеше рет айтылғандай, дербес компьютермен жұмыс жасауда маңызды рөл атқарады, онда жұмыс жүргізіледі. Қандай сіңіру болмаса, жұмысшы аппараттың жұмысына, бас ауырсынуына, тітіркенуіне, ұйқының бұзылуына, шаршағандыққа және көздің ауырсынуына қанағаттанбау туралы шағымдары пайда болады.

Бұл бөлмеде Windows және 8 шам бар. Терезенің өлшемі кішкентай, ал шамдардың күші қажетті жарықтандыруға мүмкіндік бермейді. Сондықтан менің бөлімімде ерік деп санаймын.

Жұмыс орнының жарықтандыруы жарықтандыру жүйесін таңдауға, лампалардың қажетті санын, олардың типін және орналасқан жерін анықтауға негізделген. Осыған байланысты жасанды жарықтандыру параметрлерін есептейміз.

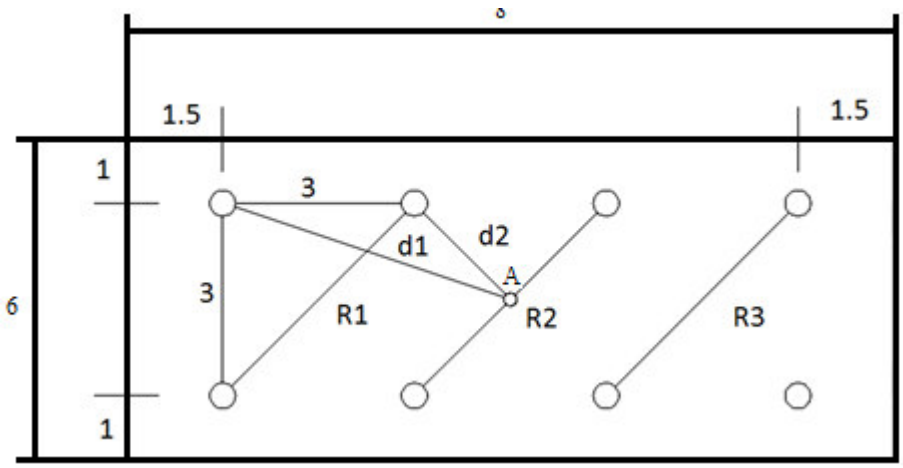
Жасанды жарықтандыруды есептеу. Жарықтандыру нүктесінің әдісін есептеу. Есептің биіктігін анықтаңыз.

$$h_р = H_п - h_{свеса} - h_{р.пов.}$$

Қабылдаймыз $h_{свеса} = 0,5$ м и $h_{р.пов.} = 0,8$ м

$$h_р = 4 - 0,5 - 0,8 = 2,7 \text{ м}$$

Шамдардың жалпы схемасы [3 х 4] 5.1-суретте көрсетілген.



5.1-сурет – Шамдардың орналасу схемасы

Шам арасындағы қашықтықты табамыз $\lambda=0.6\div2.0$.

$$L_A=\lambda\cdot h_p=1\cdot3=3$$
$$L_B=\lambda\cdot h_p=1{,}34\cdot3=4$$

Есептеу үшін арқасында бақылау нүктесі А табу қажет d8; d7,10; d3,9; d2,4,6,11; d1,5 – проекциялау қашықтығы төбе арасындағы нүктесі А және тиісті үлкен шаммен:

$$d_8=8 \text{ м};$$
$$d_{1,5}=3 \text{ м};$$
$$d_{3,9}=4 \text{ м}$$
$$d_{7,10}=\sqrt{8^2+3^2}=\sqrt{73} \text{ м}$$
$$d_{2,4,6,11}=\sqrt{4^2+3^2}=5 \text{ м}$$

Бұдан әрі төбенің биіктігі мен d сәйкес кесіндісінің арасындағы бұрышты анықтаймыз:

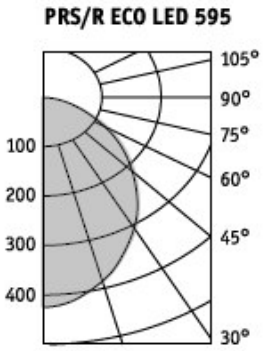
$$\text{tg}\alpha_1=\frac{d_{1,5}}{h\text{расч}}=\frac{3}{2{,}7}=1{,}11\rightarrow\alpha_1=48°;$$
$$\cos^3\alpha_1=0{,}299$$

$$\text{tg}\alpha_2=\frac{d_{2,4,6,11}}{h_{\text{расч}}}=\frac{5}{2{,}7}=1{,}85\rightarrow\alpha_2=62°;$$
$$\cos^3\alpha_2=0{,}103$$
$$\text{tg}\alpha_3=\frac{d_{3,9}}{h\text{расч}}=\frac{4}{2{,}7}=1{,}48\rightarrow\alpha_1=56°;$$
$$\cos^3\alpha_3=0{,}175$$
$$\text{tg}\alpha_4=\frac{d_{7,10}}{h\text{расч}}=\frac{\sqrt{73}}{2{,}7}=3{,}16\rightarrow\alpha_1=73°;$$
$$\cos^3\alpha_4=0{,}0249$$
$$\text{tg}\alpha_5=\frac{d_8}{h\text{расч}}=\frac{8}{2{,}7}=2{,}96\rightarrow\alpha_1=71°;$$

$$\cos^3 \alpha_5 = 0{,}0345$$

Шам түрін таңдаймыз PRS/RECOLED 300 4000К$Р_н$=2*18

5.2-сурет – 2-шамның түрі PRS/RECOLED 300 4000К



5.3-сурет – КСС

Осы бұрышта әрбір көзден 5.3-сурет бойынша жарық Күшін табамыз

$I_{\alpha 1}$=310 кд
$I_{\alpha 2}$=180 кд
$I_{\alpha 3}$=370 кд
$I_{\alpha 4}$=80 кд
$I_{\alpha 5}$=100 кд

Әрбір көзден бақылау нүктесіне қатысты үй-жайдың жарықтандырылуы:

$$e_{АГ} = \frac{n \cdot I_\alpha \cos^3 \alpha}{h_p}$$

$$e_{АГ1} = \frac{2 \cdot 310 \cdot 0{,}299}{2{,}7^2} = 25{,}43\text{лк}; \qquad e_{АГ2} = \frac{4 \cdot 180 \cdot 0{,}103}{2{,}7^2} = 10{,}17\text{лк};$$

$$e_{АГ3} = \frac{2 \cdot 370 \cdot 0{,}175}{2{,}7^2} = 17{,}76\text{лк}; \qquad e_{АГ4} = \frac{2 \cdot 80 \cdot 0{,}0249}{2{,}7^2} = 0{,}54\text{л, ;}$$

$$e_{АГ5} = \frac{2 \cdot 100 \cdot 0.0345}{2,7^2} = 0,94\text{лк};$$

$$\sum_{i=1}^{6} e_{АГi} = e_{АГ1} + e_{АГ2} + e_{АГ3} + e_{АГ4} + e_{АГ5} + e_{АГ6} = 25,43+10,17+17,76+0,54+0,9=54,84\text{лк}$$

Қосынды жарықтануы:

$$E = \frac{\mu \cdot Ф_л \cdot n}{1000 \cdot K_з} \cdot \sum_{i=1}^{6} e_{АГi}$$

мұндағы, μ - "қашықтағы" шамдардың әсерін ескеретін коэффициент (1,1 ÷ 1,25).

$$E_{АГ} = \frac{1.2 \cdot 1060 \cdot 2}{1000 \cdot 1.2} \cdot 54,84 = 116,26\text{лк}$$

Emin = 200 лк 3.12 кестеден аламыз (бірінші МУ әдебиеттер тізімін қараңыз))

ҚАДАМ = E min (т.к. жарықтануы аздап көп нормаланған жарықтандыру санын көбейту керек шам).

Қолдану коэффициентінің әдісі. Анықтаймыз индексі үй-жайлар.

$$i = \frac{L \cdot B}{h_p \cdot (L + B)}$$

$$i = \frac{14 \cdot 8}{2,7 \cdot (14 + 8)} = 1.88$$

Пайдалану коэффициенті жарық ағынының алайық 5.4-суреттен, ескере отырып, табылған индексі үй-жайлар мен берілген коэффициенттері көрсету:

PRS/R ECO LED								
потолок	80	80	80	70	50	50	30	0
стены	80	50	50	50	50	30	30	0
пол	30	30	10	20	10	10	10	0
0.6	65	43	34	41	40	34	33	28
0.8	74	53	43	50	48	42	41	36
1	81	60	49	57	54	48	48	42
1.25	87	69	57	64	61	56	55	49
1.5	91	74	62	69	65	60	59	54
2	96	82	68	76	70	66	65	60
2.5	100	87	73	80	74	71	70	65
3	102	92	77	84	78	75	73	69
4	105	96	80	87	80	78	76	72
5	106	99	83	90	82	80	79	75

5.4-сурет – Пайдалану коэффициенті шамның түрі PRS/RECOLED 300 4000K

η=70%

Қажетті жарықтандыру кезіндегі шамдардың саны E=200 лк:

$$N = \frac{E_{min} \cdot S \cdot Z \cdot K_з}{\Phi_л \cdot \eta \cdot n}$$

мұндағы, Z – 1.1÷1.2 тең жарықтандырудың әркелкілік коэффициенті;

Кз – қор коэффициенті, тең қабылданатын 1.3 үшін белгіленген үлгідегі үй-кесте 3.11 (қараңыз әдебиеттер тізімі алғашқы ШІ)

$$N = \frac{200 \cdot (14 \cdot 8) \cdot 1.1 \cdot 1.3}{1060 \cdot 0.7 \cdot 2} \approx 22(\text{шт});$$

Шамдардың санын 22-ге дейін ұлғайтамыз.

Қорытынды. Жұмыстың бірінші бөлігінде табиғи жарық есептеу. Үй-жайларды табиғи жарықтандыруды жобалау кезінде Кео-ның нормаланған мәнін қамтамасыз ететін жарық саңылауларының ауданын анықтау қажет. Кео-ның нормаланған мәндері eN = 0.84 бөлмесінде компьютерлік класс түрін қамтамасыз ету үшін III разрядтағы көрнекі жұмыс көлемін қажет етеді. III - 35 м2 ашық саңылаулар.

Осы жұмыстың екінші бөлігінде жасанды жарықтандыру есептеу. Дүңгіршектегі жарықтандыру есептеуі бөлменің жеткілікті жарықтылығын қамтамасыз ету үшін белгіленген шамдардың шамалы екенін көрсетті. Компьютерлік класс бөлмесінің қажетті жарықтандыруын қамтамасыз ету үшін шамдардың санын 36 данаға дейін арттыру қажет.

Қорытынды

Дипломдық жобаның тақырыбы «Компьютерлік техниканы сатуға онлайн-дүкен жасау», мақсаты – компьютерлік техниканы сатуды автоматтандыру.

Қазіргі уақытта компьютерлерді пайдалану арқылы кәсіпорындарды автоматтандыру бойынша көптеген жұмыстар жалғасуда. Бұл, әсіресе, көптеген қағаз ақпараттары бар мекемелерде орынды. Мен бұл диссертацияны осы бағдарламаны құру жолымен жаздым. Осы бағдарламаны әзірлеу кезінде арнайы байланыс дүкендерінің құжат айналымы қаралды, онда жұмыс күнінің мақаласы қарастырылып, тұжырымдамалар әзірленді. Осы ескертулер мен тұжырымдамалар негізінде бағдарлама Visual Studio C#-де әзірленді.

Дипломдық жобада компьютерлік техниканы сатушының жұмысын жеңілдететін автоматтандырылған ақпараттық технологияларды құру қарастырылады. Мамандандырылған дерекқор әзірленді, белгілі бір ақпараттық құрылымды басқару жүйесін әзірледі және кітапханаға қатысты нақты және шектеулі жұмыстарды жүргізді. Бұдан басқа, дерекқор деректері мен оларды басқару құралдарын зерттеу болашақ пайдаланушылармен толығымен қамтылды.

Әдебиеттер тізімі

1 Автоматизированые информационые технологи в банковской деяттельности. //под реда. Титореннко Г.А. – М.: Финстаттинформ. 2000.

2 Культинн Н.Б. Основы програмирования в С#. – Санкт-Петрбург, “БХВ–Петебург”, 2003.

3 Введениие в Microsoft Windowss 95. – издателство Microsofft 1995.

4 Дантеман Д. Програмирование в среде С#. – Киев, DiaSoft Ltd., 1995.

5 Гейл Д. Теория линейнных экономических моделей. – М., 1963.

6 Архангельски А.Я., Програмирование в С#. – Москва, Издательство “Бином”, 2004.

7 Яковицки Э. Ф. Автоматизированые системы обрабтки информаци в учрежденниях Сбербанка учебное псобие. – Минск, 1991.

8 Гончарев А. Microsoft Acces 2002 в примерх. – Питер, 1999.

9 Microsoft Acces 2002. Шаг за шагом: Практ.Посбие. – Москва, ЭКОМ.-1999.

10 Дженингс Р. Microsoft Access 97. – Санкт-Петербург, 1999.

11 Нортн П., Андерсн В. Разработка приложени в Microsoft Access 97.– Санкт-Петербуг, 1999.-656 б.

12 Норис Послед. Microsoft Access 2000. Базы даных и приложеня. – ДиаСофт, 2000.-510 б.

13 Бекаревич Ю.А., Пушкин Н. В. Microsoft Access 2003. – СПб.: БХВ-Петербург, 2006. – 752 б.

14 Р. Ахан. Эфективная работа с СУБД. – Санкт-Петербург.:БХВ-Петербург, 2005.

15 Автоматизированое рабочее места в системе управленя предпрятием, Сборник научнх трудов. – Ленинград, 1989.

16 В.В. Шураков. Автоматизированное рабочее место для статической обработки данных. – Ленинград, 1990.

17 Қазақстан Республикасынң Еңбек қорғау кодексі, 2007 жылы15 мамырда бекітілген (06.12.2008ж қосымшаларымен).

18 Венгров А.М., Федотов К. Проектирование програмного обеспечения информационых систем. – М.: Финансы и статистика, 2000. – 260 с.

19 Буч Г. Объектно-ориентированый анализ и проектированние с приме-рами приложений на Rational Rose 2-е изд.: Пер. с англ. – М.: Издателство Бином, – СПб.: Невски диалект, 1999. – 238 с.

20 Шарыпов А., Ыскакова А., Рақметова А. Сөздік-словарь. Ағылшынша – Орысша – Қазақша информатикадан түсіндірме сөздік. – Алматы, 2000. – 673 б.

21 Абдимуратв Ж. С., Мананбаева С. Е. тіршілік Қауіпсздігі. Барлық мамандықтарға арналған бітіру жұмыстарнда "өндірістік жарықтандыруды

есептеу" бөлімін орындауға арналған әдістемелік нұсқаулар. Бакалавриат-Алматы: АЭжБИ, 2009. – 20 б.

22 ҚР ҚНжЕ 2.04-05-2002 ескерту. Табииғи және жасанды жарықтанндыру. Жалпы талаптар. – М.: Құрылысиздат, 2002.

23 Абдімуратов Ж. С., Мананбаева С. Е. тіршілік Қауіпсіздігі. Өндірістік жарықтандру. Дипломдық жобаны орындауға арналған әдістеммелік нұсқаулар, Алматы энергетика және байланыс инситуты.-Алматы, 2010.

24 Кнорринг Г. Н. Справычная книга для проектирвания электрического освещения. - Л.: Энергия, 1986.