ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

«Компьютерлік технологиялар» кафедрасы

«Қорғауға жіберілді»
Кафедра меңгерушісі
ф.-м.ғ.д., проф. Құралбаев З.Қ.
_____
(қолы)
«_____»_____2014 ж.

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: «Ипотека жүйесін автоматтандыру»
5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»
мамандығы бойынша

Орындаған Гульманова Жулдыз Манаскызы тобы: ВТк-10-1

Жетекші доцент, ф-м.ғ.к. Аманбаев А.А

Кеңесшілер :
Экономикалық бөлім бойынша :
доцент Боканова Г.Ш.
_____«20»05 2014ж.
(қолы)
Өмір тіршілігі қауіпсіздігі бойынша:
т.ғ.к.,аға оқытушы Муташева Г.С.
_____«24»04 2014 ж.
(қолы)
Есептеу техникасын қолдану бойынша :
доцент, ф-м.ғ.к. Аманбаев А.А.
_____«23»05 2014 ж.
(қолы)
Мөлшер бақылаушы:
аға оқытушы Рахимжанова З.М.
_____«28»05 2014 ж.
(қолы)
Пікір жазушы :
ҚазЭУ, қолданбалы информатика кафедрасының профессоры : т.ғ.д., Абдылдаев Э.К.
_____«____»_____ 2014 ж.
(қолы)

Алматы 2014

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

«Ақпараттық технологиялар» факультеті
«Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы
«Компьютерлік технологиялар» кафедрасы

жобаны орындауға берілген

ТАПСЫРМА

Студент Гульманова Жулдыз Манасқызы

Жоба тақырыбы «Ипотека жүйесін автоматтандыру»
ректордың «24» қаңтар №115 бұйрығы бойынша бекітілген.

Аяқталған жұмысты тапсыру мерзімі: «____»_____2014 ж.

Жобаға бастапқы деректер (талап етілетін жоба нәтижелерінің параметрлері және нысанның бастапқы деректері):
- Ақпараттық жүйелер, деректқорлар (ДҚ), деректқорларда басқарудың жүйелері (ДҚБЖ) түсініктері, жүйенің деректқорын жобалау.
- ER моделін жобалау, дерек қорда құру, пайдаланушы интерфейсін жобалау, деректқормен байланыс орнату.
- Техникалық-экономикалық негіздеуді.
- Өндірістегі еңбек қорғау мен қауіпсіздік шараларын ұйымдастыру.

Диплом жобасындағы әзірленуі тиіс сұрақтар тізімі немесе диплом жобасының қысқаша мазмұны:
Бұл дипломдық жобада ипотека жүйесін автоматтандыру мәселелері қарастырылады. Ипотека процессінің концептуалдық моделі, ипотека алу кезіндегі құжаттарды рәсімдеудің автоматтық жүзеге асырылады.
Бітіру жұмысында ипотека жүйесін автоматтандырудың тиімділігі мен ипотека жүйесін жетілдіру туралы нақты есептеулер келтірілген.

Сызба материалдарының (міндетті түрде дайындалатын сызуларды көрсету) тізімі:

Деректердің ER диаграммасы.
Жасалған бағдарламаның қосымшаның жұмыс істеу қабілеттілігі мен функционалдығын көрсететін графикалық материал.

Негізгі ұсынылатын әдебиеттер:

1. Алексей Вишневский. SQL Server . Эффективная работа. – Санкт-Петербург, 2009. – 541с.
2. Полякова Л.Н. Основы SQL. Курс лекций: учеб. пособие. – Москва, 2004. – 368с.
3. Глушаков С.В., Ломотько Д.В. Базы данных: Учебный курс. – М.: ООО «Издательство АСТ», 2000. – 504с.
4. Агапов В.П. Основы программирования на языке C# . учеб.пособие.- Москва,2012.-128с.
5. Голицына О.Л., Попов И.И.Основы алгоритмизации и программирования. учеб.пособие.-Форум 2008.-432с.

Жоба тараулары бойынша кеңес берушілер және оның мерзімі:

Бөлім	Кеңесші	Мерзімі	Қолы
Негізгі бөлім	Аманбаев А.А.	20.04.2014	
Тіршілік қауіпсіздігі	Муташева Г.С.	24.04.2014	
Экономикалық бөлім	Боканова Г.Ш.	20.05.2014	
Норма бақылаушы	Рахимжанова З.М.	28.05.2014	
Есептеу техникасын қолдану	Аманбаев А.А.	23.05.2014	

диплом жобасын дайындау

КЕСТЕСІ

№ р/с	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
1.	Автоматтандырылған жұмыс орнының анықтамасы (арналуы т.б)	06.09.2013	
2.	Ақпараттық жүйелер, дерекқорларды басқарудың жүйелері, түсініктері.	25.10.2013	
3.	Жүйенің дерекқорын жобалау және жүзеге асыру, жобалау құралдарын таңдау.	11.12.2013	
4.	ER моделін жобалау.	27.01.2014	
5	Пайдаланушы интерфейсін жобалау.	13.03.2014	
6.	Дерекқормен байланысын орнату.	27.03 2014	

Тапсырманың берілген уақыты «12» 11 2014 ж.

Кафедра меңгерушісі ________ (қолы) ф.-м.ғ.д., проф. Құралбаев З.Қ.

Жоба жетекшісі ________ (қолы) доцент, ф-м.ғ.к. Аманбаев А.А

Орындалатын тапсырманы қабылдаған студент ________ (қолы) Гульманова Жулдыз Манаскызы

Андатпа

Бұл дипломдық жобада ипотека жүйесін автоматтандыру мәселелері қарастырылды. Ипотека процессінің жылдамдығының артуы, ипотека алу кезіндегі құжаттарды рәсімдеудің оңтайлығы жүзеге асырылды.

Бітіру жұмысында ипотека жүйесін автоматтандырудың тиімділігі мен ипотека жүйесін жетілдіру туралы нақты есептеулер келтірілген. Деректер қорына қойылатын талаптар, пайдалану шарттары мен оны жүзеге асыру қарастырылған.

Ақпараттық жүйенің интерфейсі Microsoft Visual Studio 2012 құрылған, ал деректер қоры Microsoft SQL Server 2008 R2 негізінде жүргізілген

Өміртіршілік қауіпсіздігі бөлімінде жұмыс орнындағы бөлменің жасанды жарықтандыру сипаттамалары мен есептеулері жүргізілген. Экономикалық бөлімде жұмсалған күрделі шығындар мен пайдалану шығындары есептелген.

Аннотация

В данной дипломной работе рассматривается вопросы автоматизации ипотечной системы. Осуществляется увеличение скорости ипотечного процесса и удобство оформления документов при получении ипотеки.

Выпускной работе приведены точные расчеты эффективности автоматизации и совершенствования ипотечной системы. Рассмотрены требования к базе данных, условия использования и осуществление.

Интерфейс информационной системы разработана в среде Microsoft Visual Studio 2012, а база данных разрабатывается на Microsoft SQL Server 2008 R2.

В разделе безопасность жизнедеятельности произведены расчеты искусственного освещения комнаты. В экономической части произведены расчеты капитальных затрат и эксплуатационных расходов.

Annotation

In this thesis work is considered the problems of automation of the mortgage system. carried out increase of the mortgage process speed and convenience of clearance paperwork when obtaining a mortgage.

Exhaust work presents accurate calculations the efficiency of automate and improve of the mortgage system. Considered the requirements to the database, terms of use and implementation.

Interface of information system is developed in Microsoft Visual Studio 2012, and the database is developed on Microsoft SQL Server 2008 R2.

In the safety life section produced calculations are of artificial lighting of the room. In the economic part produced calculations of capital costs and operating expenditure

Кіріспе

Бұл дипломдық жобада ипотека жүйесін автоматтандыру атты жұмысы жобаланды.

Бұл бағдарлама ипотека жүйесін дамытып, ипотека алу мүмкіндігін жеңілдетуге арналған. Ипотека алу негізінен тұрғын үй мен ипотека алушылар, құрлысшылар, бөлінетін жүктемелер туралы ақпаратты жүйелеумен байланысты болады. Сондықтан, ипотека жүйесін автоматтандыру жүктемесін бөлу жұмысы ыңғайлы, әрі тез болу үшін, барлық деректерді жүйелі түрде сақтау, қажетті ақпаратқа тез қатынас алу, сонымен қатар жаңа деректерді ыңғайлы түрде енгізу мүмкіндіктері қарастырылған болу керек.

Дерекқор құру үшін алдымен зерттелетін облысының талдауы жасалады. Келесі кезеңде тұрғын үй мен ипотека алу туралы қажетті деректерді сақтайтын логикалық байланысқан кестелер құрылып, олардың негізінде жүйенің инфологиялық моделі алынады. Инфологиялық модель MS SQL Server-жүйесінде жасалады. Ипотека жүйесін автоматтандыру бағдарламасының дерекқоры MS SQL Server дерекқорларды басқару жүйесінде құрылғаннан кейін, пайдаланушылардың барлық сұраныстарын қанағаттандыратын клиенттік қосымша жобаланады. Клиенттік бағдарлама Microsoft Visual Studio 2012 Ultimate программалық ортасында іске асырылады.

Бұл бағдарлама ыңғайлы әрі уақытты үнемді пайдалануға мүмкіндік береді, сонымен қатар, аз уақыт ішінде бағдарламаны игеріп, оңай пайдалануға мүмкіндік береді.

Мазмұны

Кіріспе 9
1 Техникалық тапсырма 10
1.1 Тағайындалуы және қолдану аймағы 10
1.1.1 Сенімділікке қойылатын талаптар 10
1.1.2 Техникалық құрылғыларға қойылатын талаптар 10
1.1.3 Бағдарламалық құжатқа қойылатын талаптар 10
1.2 Құру кезеңдері мен сатылары 11
1.2.1 Кезең бойынша жұмыстардың мазмұны 11
1.2.2 Тестілеу 11
1.3 Автоматтандырылған жұмыс орнының (АЖО) арналуы мен құрылымы 13
1.3.1 АЖО құрылымы 14
1.3.2 АЖО-ға қойылатын жалпы талаптар 15
1.3.3 Дерекордағы шешілетін мәселелер 16
2 Ақпараттық жүйелер. Деректер қорын жобалау 17
2.1 Ақпараттық жүйелер туралы қысқаша мағлұматтар 17
2.1.1 Ақпараттық жүйе жобалау сатылары 17
2.1.2 Ақпараттық жүйенің тіршілік циклі 19
2.1.3 АЖ–ның басты сенімділік түсінігі 21
2.2 Деректер қорын жобалау 24
2.2.1Деректер моделі 24
2.2.2 Реляционды деректер қорына қойылатын талаптар 26
2.2.3 SQL тілі туралы негізгі деректер 27
2.2.4 Кесте құру және өшіру 28
3 Дерекқорды жобалау және оны іске асыру 31
3.1 ДҚ құрудың бастапқы кезеңі 31
3.1.1 Дерекқордың функционалдығын сипаттау 31
3.2 Концептуалды жобалау 31
3.2.1 Қажетті ақпарат: пайдаланушылардың талабы 31
3.2.2 Дерекор және қосымшаны жобалау құралдарын таңдаудың негіздемесі 31
3.3 ER моделін құру 33
3.3.1 Бизнес ережелер 33
3.3.2 Реляционды қатынастарды құрастыру 36
3.3.3 Логикалық жобалау 38
3.4 Физикалық жобалау 40
3.4.1 Дерек қорына қосымша құру 40
3.4.2 Бағдарламаны сипаттау 40
4 Тіршілік қауіпсіздігі 46
4.1 Жұмыс орнындағы еңбек жағдайын талдау 46
4.2 Жасанды жарықтандыруды есептеу 52
4.3 Меншікті қуат әдісі 56
5 Техника-экономикалық негізделуі 57

5.1 Жобаның мақсаты мен міндеттері 57
5.2 Маркетингтік стратегия 57
5.3 Бағдарламамен қамтамасыз етудегі еңбек сыйымдылығын есептеу 57
5.4 Бағдарламалық қамсыздандыру шығынының есебі 59
5.5 Бағдарлама өнімін сатып алуға кеткен бір жолғы шығындар есебі 64
5.6 Игеру саласындағы жылдық бір жолғы шығындар есебі 64
5.7 Ақпараттық жүйе енгізуден үнем мен табыс мөлшерінің есебі 66
5.8 Салыстырмалы экономикалық тиімділіктің көрсеткіштерін есептеу 67
5.9 Динамикалық көрсеткіштер негізінде жобаны өткізуде экономикалық тиімділігін бағалау 68
Қорытынды 71
Пайдаланылған әдебиеттер тізімі 72
А қосымшасы 73
Б қосымшасы 88

1 Техникалық тапсырма

1.1 Тағайындалуы және қолдану аймағы

Бұл жүйе автоматтандырылған қызмет көрсету орны болып табылады. Яғни ипотека алу мүмкіндіктері, тұрғын үйлерді таңдау, ипотека мөлшері туралы ақпарат алу, оңтайлысын таңдау. Программа керекті есептеулерді автоматты түрде жүргізеді, есептеу барысында кететін қателерді азайтады. Бұл жүйе келесідей мәліметтерді жүйелеп, қолданушының сұрақтарына жауап беруі қажет:

- тұрғын үйлер;
- ипотека мөлшері;
- құжаттарды толтыру;
- анықтама.

1.1.1 Сенімділікке қойылатын талаптар

Бағдарлама толық сенімді функционалды жұмыс жасауы үшін тапсырыс беруші ұйымдық – техникалық шаралардың жиынтығын қамтамасыз ету керек. Олар төмендегідей болады:

- техникалық құралдарға үзіліссіз қоректену көзін ұйымдастыру;
- лицензиялық бағдарламалық қамтуды пайдалану;
- ГОСТ 51188–98 талаптарының уақыттылы орындалуы. Ақпаратты қорғау;
- компьютерлік вирусқа бағдарламалық құралдарды тексеру.

1.1.2 Техникалық құрылғыларға қойылатын талаптар

Сервер қызметін атқаратын дербес компьютер болуы керек. Оның сипаттамалары төмендегідей:

- процессор Pentium–4 кем емес;
- жедел жады 2 Гигабайт көлемінен кем емес;
- HDD, 250 Гбайт аз емес;
- операционды жүйе Windows 2000 Server , Windows 2003, XP, 7 және Windows 8;
- Microsoft SQL Server 2008.

1.1.3 Бағдарламалық құжатқа қойылатын талаптар

Бағдарламалық құжаттың құрамы келесідей болу керек:

- техникалық тапсырма;
- бағдарлама және байқаулардың әдістері;
- пайдаланушыға көмек.

1.2 Құру кезеңдері мен сатылары

Құру сатылары.

Құру келесі үш сатыны қамту керек:

- техникалық тапсырманы құру;
- жобалау жұмысы;
- енгізу.

Құру кезеңдері.

Техникалық тапсырмаға сай құру жұмыстары кезеңмен орындалу керек. Құру кезеңдері келесідей болады:

- ақпарат жүйесін құру;
- бағдарламалық құжатты құру;
- мысалмен бағдарламаның тексерілуі.

Бағдарламаны енгізу кезінде құру кезеңдері орындалып және бағдарламаны беру болып табылады.

1.2.1 Кезең бойынша жұмыстардың мазмұны

Құру кезеңінде келесі жұмыстар орындалу керек:

- тапсырманың мақсатын анықтау;
- техникалық құрылғыларға қойылатын талаптарды анықтау;
- бағдарламаға қойылатын талаптарды анықтау;

1.2.2 Тестілеу

Тестілеу – бұл ақпараттық жүйеде қателерді іздеу болып табылады.

Қате – бұл бар нәрсенің бір–біріне сәйкес келмеуі.

Сенімділік – бұл әрбір қабылдалмаудың құнын есепке ала отырып есептелінген, белгілі бір уақыт ішіндегі қарсылық жасалмаған жұмыс ықтималдығы.

Тестілеу принциптері.

Тестілеу жақсы жобаланған жүйеде қалған қателерді табу үшін, сонымен қатар жүйенің сенімділігін арттыру үшін және құндылығын арттыру үшін жүргізіледі. Тестілеу арқылы нашар жобаланған жүйенің жоғарғы дәрежедегі сенімділігіне жетуге болмайды.

Егер біз бағдарламаны тестілейтін болсақ, онда қандай да бір жағдайлармен бағдарламаның бағасын көтеру арқылы тестілеуге кеткен шығынды қайтарып алуымыз қажет. Мұны тек бағдарламаның сенімділігін жоғарлату арқылы ғана жасауға болады. Сенімділікті тек үрдіске енгізілген қателерді түзету арқылы жоғарлатуға болады.

Сәтті тест деп, қатені тапқан тестті айтамыз. Егер бірде бір қате табылмаса, онда ол сәтсіз тест деген сөз.

– қателер топтасу қасиетіне ие;

– тестілеуді жеңілдету үшін еш уақытта бағдарламаны өзгертпеңіз;

– бағдарламаны тестілеудің авторымен кездесуден қашу керек. Егер программист бағдарламаны жазуда қате жіберсе, онда оны тестілеуде де дәл сол қатені жіберуі ықтимал;

– тестіні құру – бұл шығармашылық үрдіс, яғни кей жағдайда мидың жұмыс істеуін қажет ететін үрдіс;

– қатені жақсы табатын тест жақсы тест болып табылады;

– бағдарлама неге арналған, яғни тек өзіне ғана тиесілі істі істейді ме, әлде басқа жұмыстарды да істейді ме соны тексеру керек;

– комментариилердің жеткіліксіздігі қателерді іздеуді қиындатады;

– комментариилердің аздығы қатені іздеуді қиындатумен қатар, түсініктеме сөздігінсіз күрделі бағдарламаны талқылау да қиын.

Тестілеуден кейін қатенің жоқ екендігіне кепілдік беруге болмайды, тек белгілі бір деңгейде жүйе жұмысының дұрыстығының сенімділігі жайлы айтуға болады.

Тест – бұл кіріс деректерінің жиынтығы және (немесе нәтижені күтуші қолданушының қызметі) жауап бағдарламалардың іс–әрекеті. Қарапайым бағдарламамен толық, жан–жақты тестілеу жүргізу мүмкін емес, себебі оған уақытта, қажетті ресурстарда жетпейді. Сондықтан тестілеудің қатені табу ықтималдылығына байланысты, тестті құру әдісіне байланысты бірнеше түрі бар [1].

Тестілеу түрлері:

– құрылымдық тестілеу (ақ жәшік);

– функционалдық (қызметтік) тестілеу (қара жәшік).

Құрылымдық тестілеу.

Берілген тестілеуде бағдарламаның тексті көрініп тұрады. Шешімдердің блоктары, циклдері тестіленеді.

Құрылымдық тестілеудің бірнеше типі бар:

- операторларды жабу;
- шешімдерді жабу;
- шарттарды жабу;
- шарттардың аралас жабулары;
- циклдердің тестіленуі.

Толық түсініктер үшін құрылымдық тестілеудің мысалдарын қарастыруға болады.

Функционалдық (қызметтік) тестілеу**.**

Берілген тестілеуде бағдарламаның тексті көрінбейді және бағдарлама қара жәшік түрінде, яғни кіріс және шығыс шарттары белгілі, сонымен қоса жұмыстың жалпы схемасы қарастырылады. Бағдарлама оның спецификациясы бойынша тексеріледі.

Функционалдық тестілеудің бірнеше түрі бар:

- эквивалентті кластар;
- шекті мәндерінің анализі;
- белгілі бір мөлшердегі тестілеу;
- белгілі бір көлемдегі тестілеу;
- қорғанысты тестілеу;
- жүйе эксплуатациясы;
- тәжірибелі эксплуатация.

Үлкен түсініктер үшін функционалдық тестілеудің мысалдарын қарастыруға болады.

1.3 Автоматтандырылған жұмыс орнының (АЖО) арналуы мен құрылымы

Автоматтандырылған жұмыс орны (АЖО) немесе шетел терминологияда «жұмыс станса» (work–station) бұл анықталған функцияларды автоматтандыру үшін қажетті құралдармен жабдықталған белгілі бір мамандықтағы қолданушы маманның орны. Мұндай құралдар, керек жағдайда баска қосалқы электронды құрылғылармен, атап айтқанда, дискілік жинақтауыштармен, басу құрылғылармен, оптикалық оқитын құрылғылармен немесе штрих кодасын оқитын, графика құрылғыларымен, басқа АЖО–мен және басқа жергілікті есептеу желілерімен т.б. түйіндесетін құрылғылармен толықтырылатын әдетте дербес компьютер болып табылады.

Таралған ИС концепциясына сәйкес олардың ішінде ақпарат көлемінің төменнен жоғары қарай берілуі ИС дың ұйымдық иерархияда жоғарыдағы деңгейлердегі оның қаншалықты кажеттілігіне байланысты болады. Бұл жерде ақпаратты өңдеу нәтижелерінің басым бөлігі және бастапқы деректер деңгейлердің жергілікті дерекқорларда (ДБ) сақталған жөн. Осы таралған

басқару идеясын жүзеге асыруда әрбір ИС деңгейі үшін мамандардың автоматтандырылған жұмыс орындарын құруды талап етті.

АЖО–ның негізгі арналуы деп жұмыс орындарындағы ақпараттың орталықтандырылмаған өңдеуді, «өзінің» сәйкес келетін дерекқорларын қолданумен қатар жергілікті АЖО–мен ДК желілеріне, ал кейде қуатты ЭЕМ–лар бар ауқымды есептеу тораптарына ену мүмкіндігімен қолдануды атауға болады.

АЖО–да шешілетін мәселелерді шартты түрде ақпараттық және есептеуіш деп бөлуге болады.

Ақпараттық мәселелерге ақпаратты кодалау, классификациялау (жіктеу), жинау, құрылымдық ұйымдастыру, түзету, сақтау жатады. Ақпараттық мәселелер жиі түрде арифметикалық пен мәтіндік сипаттағы және байланыстағы қарапайым есептеу мен логикалық процедуралардан тұрады. Ақпараттық мәселелер әдетте күрделі болып табылады және мамандардың жұмыс уақытының көп бөлігін алады.

Есептеу мәселелер формальдандырылатын және жартылай формальдандырылатын болады. Формальдандырылатын мәселелер формальды алгоритм негізінде шешіледі және екі топқа бөлінеді: тура есептеу мәселелер мен математикалық модельдер негізіндегі мәселелер. Тура есептеу мәселелер қарапайым алгоритмдер көмегімен шешіледі. Одан күрделі мәселелерді шешу үшін әртүрлі математикалық модельдерді қолдану талап етіледі. Соңғы кезде жартылай формальданатын мәселелерді шешуге арналған сематикалық деп аталатын амалдарды жетілдіруге аса назар аударылады. Мұндай мәселелер жиі түрде экономикалық объектілерді оперативті басқару барысында, әсіресе толық емес ақпарат жағдайында шешімді қабылдау кезінде туады [2].

1.3.1 АЖО құрылымы

АЖО бұл ұйымдастыру, функционалды және қамтамасыз ету бөліктерінен тұратын кешенді жүйе.

Ұйымдастыру бөлігі АЖО қолданушылардың функцияларын бөліп таратуға арналған ұйымдастыру құрлымынан, сонымен қатар мамандарды даярлау, дамыту мен әкімшілдеу тәсілдерінің жиынтығынан тұрады. Әкімшілдеуге жұмысты жоспарлау, тіркеу, бақылау, сараптау, реттеу, АЖО қолданушыларының құқықтарын мен міндеттерін құжатты түрде рәсімдеуді жатқызады.

АЖО–дың функционалды бөлігі бұл оперативті және перспективалық жоспарлау мәселелердің шешімін, техника-экономикалық көрсеткіштердің тіркеуін мен сараптауын камтамасыз ететін экономика-математикалық тәсілдер кешені. Кешен құрылымы АЖО ның анықталған функцияларын жүзеге асыратын жүйенің біршама дербес ішжүйеден тұратыны болжамдалады.

Ішжүйелердің ерекшеленуі жұмыс орынның функционалды специализациясы мен ұйымдық құрылымдық есеппен функционалды ұйымдық белгісі бойынша жүргізіледі.

АЖО–ның қамтамасыз ету бөлігі:

– функционалды ішжүйелердің ақпараттық қамтамасыз етуі (АҚЕ);
– техникалық құралдар кешені (ТҚК);
– ішжүйелер функцияларын жүзеге асырудың программалық құралдары.

АҚЕ АЖО бұл құжаттардың, ақпарат массивтерінің (ДҚ) және кодалаудың стандартталған және бәрыңғай формалар жиынтығы.

АЖО–да қолданылатын ақпарат тиянақтылық, оның арналуы және қозғалу бағыты бойынша жіктеледі. Тұрақтылық бойынша ақпарат шартты тұрақты және айнымалы тұрақты, арналуы бойынша нормативті-анықтамалық, жоспарлау, есептемелік, сараптамалық және оперативті-өндірістік (директивті-оперативті және оперативті) деп бөлінеді. Қозғалу бағыты бойынша ақпарат кіріс, шығыс және аралық деп бөлінеді [2].

АЖО–ның техникалық құралдар кешеніне (ТҚК) ақпратты жинау, жеткізу, өңдеу және сақтау процестердің жүзеге асырылуын қамтамасыз ететін құрылғылар жиынтығы кіреді.

Біртекті операцияларды орындайтын құрылғылар техникалық құралдардың функционалды топтарын құрайды. Ішжүйелердің функцияларды жүзеге асырудың программалық құралдарына біріншіден адам-машина АЖО жүйесінде қатынасудың программалық тір құралдарына арнап, «жұмыс ортасын» қалыптастыратын операциялық жүйелерді жатқызуға болады. АЖО прогораммалық тіл құралдары келесімен қамтамасыз етеді:

– АЖО–ның функционалды мәселелерін шешу;
– АЖО қолданушылар жағынан мәселелердің шешімін ыңғайлы және түсінікті басқару – мәселелер интерфейстері.

1.3.2 АЖО-ға қойылатын жалпы талаптар

АЖО–ға қойылатын жалпы талаптар келесідей:

– жүйелілік;
– икемділік;
– тұрақтылық;
– эффективтілік (тиімділік).

Жүйелілік. АЖО жұмыс орынның функционалды арналуымен анықталатын құрылымы бар жүйе деп қарастырғаны жөн.

Икемділік. Элементтердің стандартталуы мен құрылудың модульдігінің арқасында жүйе мүмкін болатын қайта құруларға икемді болу керек.

Тұрақтылық. Ішкі мен сыртқы қоздыратын факторлардың әсеріне қарамастан жүйе негізгі функцияларды орындауы керек.

Эффективтілік. АЖО–ды жүйенің құрылуы мен қолдануына кеткен шығындарына апаратын жоғарыда келтірілген қағидалардың жүзеге асырылу деңшейінің интегралды көрсеткіші де қарастырғаны жөн.

АЖО бұл әрдайым арнаулы сипатты жүйе. Нақты бір маманға әкімші, экономист, инженер, конструктор, жобалаушы,сәулетші, дизайнер, дәрігер,

ұйымдастырушы, зерттеуші,кітапханашы,мұражай қызметкері және т.б. арналған техникалық құралдар мен программалық қамтамасыз етудің жиынтығы.

АЖО–ды ендіру компьютер ұйытқы болатын ақпаратты өңдеудің машиналық құралдар мен адам арасындағы функциялар мен жүктеменің дұрыс бөліп тарату жағдайында қалаған эффектті беруі мүмкін.

АЖО бұл тек еңбектің өнімділігін мен басқарудың эффективтілігін жоғарлатуының құралы ғана емес, сонымен қатар мамандардың қоғамдық қолайшылығының құралы.

АЖО жеке (дербес) , топтық, коллективті болуы мүмкін. Топтық пен коллективті АЖО–ға қатысты олардың ДК–дің жүйе ретінде эфефективті функционалдау мақсатында мамандарға (коллективке) әкімшілдеудің нақты функцияларын анықтау қажет. АЖО адам машиналық жүйе ретінде ашық, икемді, үнемі даму мен жетілдірілуіне бейімдіболу қажет [3].

Мұндай жүйеде келесі қамтамасыз етілу керек:

– мамандардың ақпаратты өңдеу машина құралдарына максималды жақындығы;
– диалогты тәртіпте жұмыс жасау;
– эргономика талаптарына сәйкестілік;
– компьютердің жоғары икемділігі;
– ескілік процестердің максималды автоматтандырылуы;
– жекелік жағдайда мамандардың творчестволық белсенділігін арттыратын, ал кейін жүйенің дамуына әкелетін мамандардың еңбек жағдайларына моральды түрде қанағат болуы;
– мамандардың өзін-өзі оқыту мүмкіндігі.

1.3.3 АЖО–ды жобалау сатылары

АЖО–ды дұрыс ұйымдастыру, құрастыру және енгізу мақсатында оның жобалау процесін келесі сатыларға жіктейді:

– бар басқару жүйесін зерттеу мен сараптама жасау;
– АЖО құруының техникалық тапсырманы құрстыру;
– АЖО–ды техникалық жобалау;
– АЖО–ды жұмыстық жобалау.

2 Ақпараттық жүйелер. Деректер қорын жобалау

2.1 Ақпараттық жүйелер туралы қысқаша мағлұматтар

Ақпараттық жүйе – қандай да бір адам қызметіне арналған ақпараттық моделі болып табылады. Ол жүйе төмендегідей ақпараттық үдерістерді қамтуы керек:

- сақтау;
- ақпарат ауысу;
- ақпаратты түрлендіру.

Қазіргі заманда күннен күнге ақпараттық жүйелер, қызметтердің әр саласына маңыздылығы артуда.

Ақпарат жүйесінің көмегімен шешілетін тапсырмалардың әр түрлі болып келуіне арқасында құрылу жағынан бір–бірінен айырмашылығы бар ақпараттық жүйелердің әр түрлі болып бөлуіне әкеп соқты.

Ақпараттық жүйелерді көптеген белгілері бойынша бөлуге болады. Негізгісі белгісі, функционалдық мүмкіндігі мен құрылу ерекшеліктерімен бөлу болып табылады.

Деректерді сақтау бойынша ақпараттық жүйелер фактографикалық және құжаттық болып бөлінеді. Фактографикалық жүйелер сан және мәтін түріндегі деректерді сақтауға және өңдеуге арналған. Мұндай деректермен көптеген операциялар орындауға болады. Құжаттық жүйелерде дерек құжат түрінде болып келеді. Алынған құжаттар пайдаланушыға беріледі, ал оны өңдеу құжаттық жүйелерде қарастырылмайды.

Автоматтандырылған ақпараттық жүйелерге негізделе отырып, ақпараттық жүйелер қолмен басқарылатын, автоматандырылған және автоматтандырылмаған болып бөлінеді [5].

2.1.1 Ақпараттық жүйе жобалау сатылары

Тап осы лекцияда қарастырылған типтік тіршілік цикл бірінші кезеңдері ақпараттық жүйе жасау, жүйеге техникалық тапсырма берумен және өңдеумен аяқталады.

Стандарттқа сәйкес ISO/IEC 12207 ақпараттық жүйені жобалау үрдістері үш топқа бөлінген:

Негізгі үрдістер:

- тауып алу;
- жеткізу;
- өңдеу;
- қанау;
- ере жүру.

Қосалқы үрдістер:

– құжаттау;
– кескін үйлесімімен басқару;
– сапасын қамтамасыз ету;
– верификацйя;
– іскерлік сараптау;
– бірге бағалау;
– тексеру;
– проблемалардың рұқсаты.

Ұйымдық процестер:

– басқару;
– инфражүйе жасауы;
– жетілдіру;
– оқу.

Күрделі жүйені жобалау процесі өз алдында шығармашылық жұмыс, ал мұнда қолданылатын технология – ұйымдасқан шығармашылық технология – пәндік облыстағы фундаментальдік зерттеу сатысы мен дайындау және қолдану жүйесі арасындағы процестердің жиынтығы.

Бұндай технологияның бірінші сатысы ақпараттық жүйе құру концепциясымен аяқталатын пәндік облысты зерттеу болып табылады.

Көптеген шынайы жүйе иерерхиялық құрылымға ие. Бұл жүйенің ерекшелігі:

– арнайыланған жеке жұмыстарды белгілеу және оны арнйы дайындалған элементтермен бөлек беру;

– координация – жүйенің негізгі мақсатына жетуге бағытталған арнайы элементтің жиынтығын қамтамасыз ету.

Иерархиялық жүйені қолдана отырып жеке шешім қабылдай алатын, иерархиялық элементтің өзара әсерінің жүйесіндегі ұйымды анықтауға болады. Ұйымның өз бағытында қолданып отырған қиындықтарды шешу ережелерінің жиынтығы стратегия деп аталады.

Барлық ақпараттық жүйе қабылдау сатысында жүйелік жақындау қолданылады. Бұнда жүйелік модельдің көмегімен іске асатын әр түрлі саладағы, типтегі, кластардағы ұйымдардың және пәндік облыстардың зерттелуі мен өңдеуі қарастырылады. Мамандар жүйелік жақындау жоқ деп қабылдау бұл осы жүйеге енудің жақсы жолы деп айтады. Жүйелік жақындау уақытпен тексерілген және қарапайым зерттеу іздеу әдісіне қарама – қарсы келеді.

Бұндай жүйе кәдімгі коммерциялық әдіспен алынбайды. Бәсекелестік бұл жағдайда тек ұсынысты зерттеу және алдын ала өңдеумен ғана шектеледі. Шешуші техникалық және жұмысты жобалау тек бір ғана ұйыммен келісім жасалған соң орындалады. Тапсырыс беруші тапсырманы бекіткеннен кейін нарыққа ұсынысын көрсете алмайды. Сондықтан жүйелік талдау сатысы АЖ жоғарғы мәнге ие. Мысалы, АҚШ – тың қорғаныс министрлігі 1965 жылы

№32009 директивасын шығарды, онда құрылым концепция сатысынсыз күрделі жүйені өңдеу туралы келісім қабылданбайды делінген.

Концепция түсінігі – ол анық түсінім әдісі, белгілі бір заттың, құбылыстың, процестің жеткізілуі. Кейбір сатыда бір – бірімен байланысқан білім, факт, тапсырма мен қызығушылқтар жиналып қалады.

Бұндай жағдай адам санасы тез дамып келе жатқан облыстарда көп кездеседі; тез дамып келе жатқан облыстарда бұл үрдіс дағдарысқа алып келеді.

Жүйенің ерекшелігі барлық компоненттердің байланысын анықтайтын модельдің болуы. әсіресе жеке компоненттердің қасиеттерін жүйенің қасиетімен байланыстыратын модель.

Бұл жүйенің шешімі теңдік жүйесін құру арқылы жүзеге асады. Онда түрлі компоненттердің қасиеттерінің бір – біріне және жүйе қасиетіне әсерін көрсетеді. Әр компонент жеке өз қасиетін жақсарта алмайды, себебі ол өзгеріс басқа компоненттреге ісерін тигізеді. Ол ішкі жүйеге кері әсерін тигізеді, тіпті жалпы жүйенің бұзылуына алып соғады.

Осылайша, тапсырма тек теңдік жүйесін құру арқылы ғана шешуге болады. Кейбір компоненттер жобаланады және осы жүйеге арналып жасалады. Барлық осы арнайы компоненттерді қолдану және осылайша түскен табысты компонентті құрғандарға бөліп беруді қамтамасыз ететін нарық болу үшін жүйе жеткілікті үлкен болуы керек [5].

Тапсырылымның бітетін ақпараттық жүйе құру сатысы рационалды жоспарлау және ұйымды жұмыстардың үйлесімімен ерекшеленеді. Ақпараттық жүйенің тіршілік циклына мына модельдер:

- каскадты;
- аралық бақылаумен жүретін тарауланған модель;
- спиральды модель.

2.1.2 Ақпараттық жүйенің тіршілік циклі

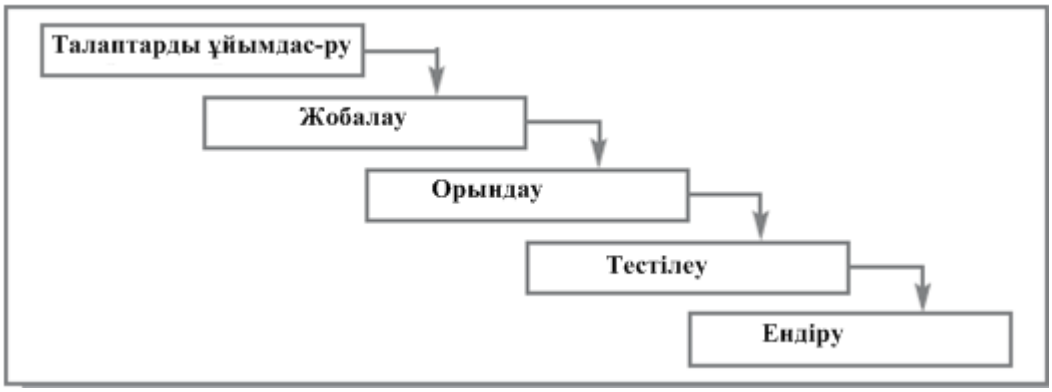
Ақпараттық жүйенің тіршілік циклін жүені құру үрдісі және оны пайдалану кезіндегі болатын оқиғалар түрінде көрсетуге болады.

Тіршілік циклі жүйенің әр түрлі күйін көрсетеді. Оның бастапқы қажеттілігенен бастап толық пайдалануынан тыс қалғанша қарастырады.

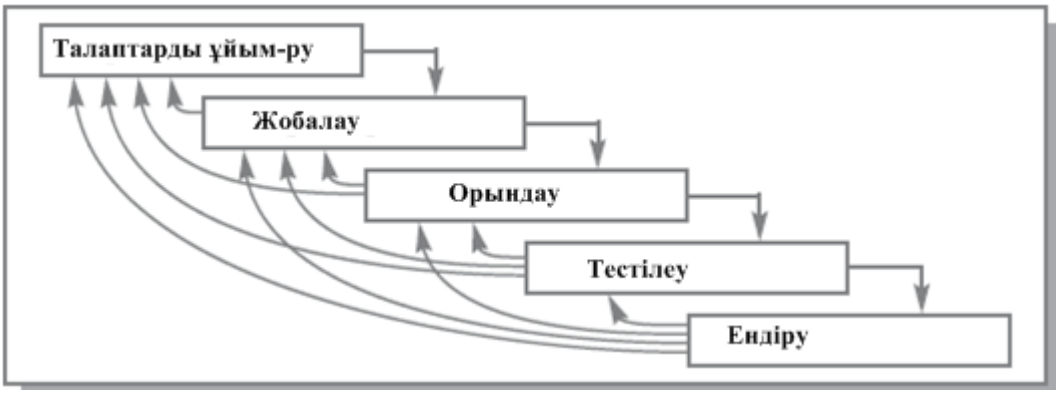
Қаіргі кезде жоғарыда айтылғандай тіршілік моделінің келесі түрлері қолданылады:

– Каскадты модель (сурет 2.1) жобаның бекітілген ретімен барлық кезеңдерінің орындалуын қамтамасыз етеді. Келесі кезеңге өту алдындағы кезеңдегі жұмыстын толықтай аяқталуын білдіреді.

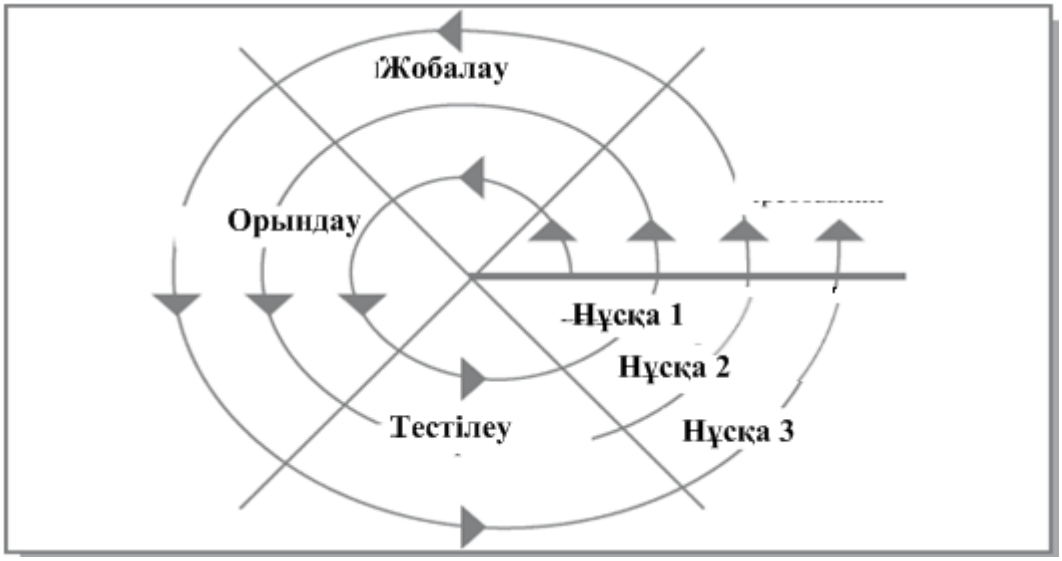
– Аралық бақылаумен жүретін тарауланған модель (сурет 2.2) ақпараттық жүйенің құрылуы, кезеңдер аралығымен кері байланыс ц-клының итерациясымен басқарылады. Аралық кезең түзетулері. Спиральды модель (сурет 2.3) әр спиральдың орамында өнімнің келесі нұсқасы құрылады, жобаға талаптар қарастырылады және оның сапасы анықталады [2].



Сурет 2.1 – Каскадты модель



Сурет 2.2 – Аралық бақылаумен жүретін тарауланған модель



Сурет 2.3 – Спиральды модель

Тәжірибе жүзінде тіршілік циклының ең таратылған екі түрі қолданылады:

- каскадты модель (мерзім 1970–1985 ж.);
- спиральды модель (мерзім 1986 ж.).

2.1.3 АЖ–ның басты сенімділік түсінігі

АЖ–ның кез келген шығу эталоны тапсырыс берушінің немесе потенциалды қолданушының техникалық тапсырмасы болып табылады (немесе спецификациялық талап ету).

Мұндай құжаттарды қолданушының белгілі шартын және шығу берілгенін алу кезінде, мазмұнын және мәнінің нәтижесінің құрамын құру кезінде алуға болады. Кез келген АЖ–ның шарттарын көрсеткен функционалды нәтижені қайтаруға және соған керек формалды эталонды қателік сияқты квалификация жасауға болады.

Практика жүзінде тапсырыс беруші мен өндірушінің өндіріс процесінде ПЖ–нің шығу талаптары бойынша нақтыланады және детальданады. Базаның мұндай нақтылануы формальданусыз көрсеткіші және мамандар білімі болып табылады, сонымен қатар уақытаралық жоба этабының тексеру нәтижесі болады. Нормативтік–техникалық құжаттарды орнату, бір немесе бірнеше АЖ–ның ережелерімен берілген мінездемелерін орнату қорытындысын АЖ–ның техникалық операциясы тексереді.

Оның қасиетінің бейнеленуі және мүмкіндігінің анықталуы және оны тура белгілеуде эффектті қолдану, кез келген өнім көрсеткіш жиынының сапасын көрсетеді. Өнім күрделі болған сайын, оның қасиетінің мінездемелік саны және оның сапасының адекватты бейнелену көрсеткішінің қажеттігі көп болады. АЖ сапасы және күрделі бағдарламалық құрал жиынтығының көрсеткіші – критериилермен жазылады, олардың әрқайсысын метрикалық және әдіс өлшемін анықтау қажет. Бұл критериилер және метрикалық бағдарламаларының соңғы өнімнің жетістік тәсілінің қатысуынсыз қасиетін бейнелейді.

Анықтау белгілері, АЖ–ның басты функциялары, тапсырыс беруші немесе потенциалды қолданушының техникалық талап етілуі, атрибуттар жиыны түрінде анықталатын, оның функционалды қажеттілігі АЖ–ның сапасына байланысты болады.

Көптеген деңгейде функционалды қажеттілік АЖ сенімділігінен байқалады. Оған қоса, функционалды қажеттілікті көптеген әр түрлі критериилер бейнелейді. Оларды басты сенімділік көрсеткішіне байланысты бөлім критериі немесе фактор ретінде қарастыруға болады.

АЖ–ның жұмысқа қабілеттілігі немесе олардың ірі компоненттерінен айырылу кедергілерін көрсеткіш сенімділігінде тіркеуге болады. Техникалық құжаттарды талап етуге орнатылған, берілген функция параметрін орындауға қабілетті объект жағдайын сенімділік теориясының жұмысқа қабілеттілігі деп аталады [5].

Тек уақыт аралығында байқалатын жүйені ішкі қасиеті сенімділік болып табылады.

Сапа критериі динамикалық және мүлікті статистикалық, АЖ–ның толық немесе ірі компоненттерінің функционалды мінездемелері бола алады. АЖ–

ның интегралды сапа көрсеткіші тәжірибе жүзінде тура бағаланып және анықталып өлшенеді.

Жинап қою және тексеруде қолданылатын шығу берілгендерге АЖ–ның жұмысқа қабілеттілігіне толық сенім береді. Нағыз шығу берілгендерінің мәні берілген техникалық тапсырма және тексеруде қолданылатын бағдарламада басқаша болады. Осындайда шығу берілгендері функционалды бағдарламаларын алдын ала болжау қиын, өйткені олар әр түрлі аномалия, қарсы аяқталулар шақыруы мүмкін.

Нормативтік–техникалық және (немесе) конструкторлық құжаттарды жүйеде орнату, бір немесе бірнеше функцияларды бұзу ең болмаса сапа талабының біреуінде орындалу жағдайын – жауапсыз дейді. Нормативтік–техникалық және (немесе) конструкторлық құжаттама және берілген функцияларды орындау жауапсыздығын анықтауды рұқсат етуге орнатылған, белгі немесе жиын белгілері жауапсыз критериилердің бағалануын жағдайға әкеледі.

Бұзудың «жауапсыздықтан» айрықшалығы уақытша көрсеткіштің ұзақтығын қайта құрудан кейін АЖ–ның функционалды бұзуына қатысты болады, бұзудан кейін жауапсыздан кейінгіге қарағанда қайта құру уақыты аз болады.

Бұзулар кезінде және қайта құрылу қасиетіне тура қатысты АЖ–ның тұрақтылық мінездемелерінің функционалдыққа қарсы жауапсыздыққа қабілетті болады. Қайта құрылу АЖ–ның функционалды толық қайта құрудан кейін қайта жіберумен (рестарт) мінездемеледі. Ресурстар және уақыт қажеттілігі АЖ–ның функционалды нормаларын жаңартуды қамтамасыз етуді қайта жіберуге болады, сондықтан функционалды қайта құрудың толықтығы және ұзақтығы АЖ–ның бейнелеу сапасынан кейінгі бұзылуды және белгілеуді оның тура мүмкіндікте қолдануға болады [5].

Басқа өндіріс өніміне қабылданған АЖ–ның сенімді көрсеткіші адекватты аналогикалық мінездеменің мәнділік деңгейі болады. Критерииді кең қолдануға қарсы өндірудің ұзақтығы – жұмысқа қабілеттілік жүйе жағдайының уақытаралық жауапсызы немесе оның жақсы функционалды жүйеден кейінгі бастамасы болады. Критерии сенімінің әр түрлілігі мүмкіндік мінездемесінің бірнеше форма көлемінде қолданылады.

Мүмкіндікті қайта құрылу жүйесінен алу жұмысқа қабілеттілік жағдайының соңғы уақытында бейнелейтін, мінездеме жауапсызын қайта құру критерии коэффициентін дайындауға өндіріледі. Жауапсыз және қайта құру құрамы коэффициент мәнінің дайындығы үлкен интервалға жететін жұмыс жүйесінің пайдалысы уақыт уысына байланысты болады.

Интегралды бағдарламаның сапа көрсеткішінің жақсы өлшемін, қатар дәлдігін алуға күрделі АЖ–ның сапасын бағалауды рұқсат етуге сенімділік теориясының басты түсінігі қолданылады. Келтірілген критериилер басты ретте АЖ–ны тексеру және комплексті жинап қоюдың аяқталу фазасында қолданылады.

Сұраныста АЖ қолданушыларының шығу ақпаратында сенімді көрсетілімі барлық АЖ типінің интегралдық көрсеткішінің маңыздыларының бірі болып табылады, сонымен қатар бағарламалық–техникалық комплекстердің әр түрлі белгілері болады ($Р_{нап}$ – мүмкіндік сенім көрсетілімі).

Қазіргі заманғы, корпоративті АЖ өзінің комплекстілігін көрсетеді:

– бағарламалық–техникалық құрылғылар;

– ақпараттық қамтамасыз ету (берілгендер базасы);

– телекоммуникация құрылғылары;

– оның функционалды қолданылуы үшін өңдеу сұраныстарын және шығу ақпаратының көрсетілімін қамтамасыз етеді.

Автоматтандырылған сұраныс өңдеуімен және эксплуатациялық шартты қолдануды бақылау кезінде алгоритмдік шындалған шығу ақпаратының келісімін көрсетумен және АЖ–ны техникалық қызмет етуімен, АЖ құрылымын қабылдауды қамтамасыз етумен АЖ қолданушыларының шығу ақпаратының сұраныс көрсетілімінің сенімділігімен түсіндіріледі ($Р_{нап}$ – мүмкіндік сенім көрсетілімі) [3].

АЖ қолданушысының шығу ақпаратының курстық жұмыста сұраныс көрсетілімінің сенімділігін бағалау келесі модельде қолданылады.

АЖ–ның кез келген уақытында (қолданушының көзқарасымен қарағанда) қатар екі жағдайдың біреуінде болады:

– жұмысқа қабілеттілік;

– жұмысқа қабілетсіздік.

АЖ–ның математикалық уақытты күтудің жұмысқа қабілеттілік жағдайы жауапсыздың орташа өңделіміне тең. АЖ–ның математикалық уақытты күтудің жұмысқа қабілетсіздік жағдайы жұмысқа қабілеттіліктің жауапсыздан кейінгі орташа қайта құру уақытына тең.

Кездейсоқ уақытта шығу ақпараты көрсетілім сұранысын орындайды (экран дисплейінде, файлды магнитті тасымалдаушыда немесе шығарғанда). 3 варианты мүмкін:

– АЖ–ның функционалды t уақытында түскен сұраныс, оның жұмысқа қабілеттілік жағдайы үстінен түседі және сұранысты өңдеу қажеттілігі және оған шығу ақпаратына байланысты берілуі АЖ–ның жағдайында барлық уақытында болады;

– АЖ–ның функционалды t уақытында түскен сұраныс, оның жұмысқа қабілеттілік жағдайы үстінен түседі, бірақ сұранысты өңдеуге қажетті және оған шығу ақпаратына қатысты берілуінде АЖ бұл жағдайда аз уақыт болады;

– АЖ–ның функционалды Т уақытында түскен сұраныс, оның жұмысқа қабілетсіздік жағдайы үстінен түседі.

Бірінші жағдайда сенімді ақпараттық көрсетілім болады, ал екінші және үшінші жағдайларда сұратылған ақпарат көрсетілмейді.

Зерттеулер кезінде алынған бағалар:

– жауапсыздыққа орташа өңделу;

– орташа қайта құрылу уақыты;

– шығу ақпаратының орташа уақыт көрсетілімі (АЖ–ға сұраныс уақыты);

– АЖ қолданушыларының шығу ақпаратына сұрастырылатын сенімді көрсетілім есептеулерін өткізу үшін шығу берілгендері болады. Шығу ақпараты көрсетілуі мүмкін барлық сұрастырылған ақпараттар нақты анықталған формада (немесе мазмұнда) типтерге бөлінеді.

2.2 Деректер қорын жобалау

2.2.1 Деректер моделі

Деректер қорын басқару жүйесі (ДҚБЖ) қолдайтын декректердің моделі өзінің логикалық құрылымы болады. Оған келесідей модельдер жатады:

– иерархиялық;
– желілік;
– реляционды.

Сонымен қатар кейінгі уақытта тәжірие жүзінде бастапқы үш модельдерден гөрі белсенді деректер моделі қолданылып жүр. Олар:

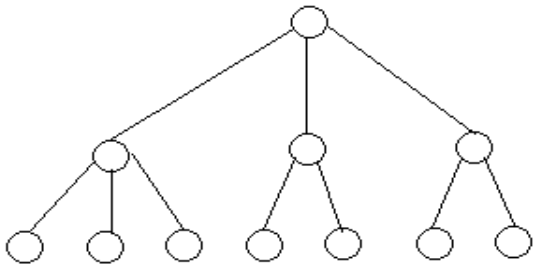
– постреляционды;
– көп өлшемді;
– нысанды–бағытталған.

Осы модельдерге негізделе отырып, қазіргі кезде басқа да деректер моделі құрылуда. Олардың қатарына нысанды–реляционды, деуктивті–нысанды–бағытталған, семантикалық, концептуалдық және бағытталған моделдерді жатады.

Кейбір ДҚБЖ бірнеше деректер моделін қамти алады. Мысалы, ИНТЕРБАЗА жүйесінде желілік деректер қорын басқарылады және пайдаланушы интерфейсінде SQL және QBE тілдері құрылған [7].

Иерархиялық модель.

Иерархиялық модельде деректер арасындағы байланыстар графа (ағаш тәріздес) түрінде болады. Иерархиялық модельдің сызбалық көрінісі келесідей болады. (сурет 2.4)



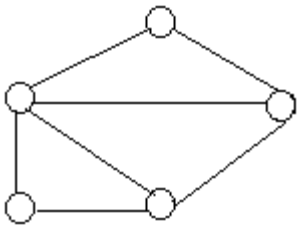
Сурет 2.4 – Иерархиялық модельдегі байланыстар көрінісі

Иерархиялық деректер қорын құру кезде арнайы бағдарламалау тіліндегі ағаш типі қолданылады. Ағаш типі бірнеше подтиптерден құралған болып келеді. Әр ағаштың өзіне сәйкес «тамырлары» (түбірі) болады. Сол тамырдан ббағынатын көптеген құрылымдық ағаш тәрісдес типтер де болады.

Иерархиялық модельдің бір артықшылығы, ол ЭЕМ жадысын тимімді қолдануы және деректерге қолданатын операциялардың жылдам орындалуымен сипатталады.

Желілік модель.

Желілік модель декректер элементтерінің өзара әр түрлі байланысын көрсетеді. Желілік модельдің сызбалық кескіні келесідей болады. (сурет 2.5)



Сурет 2.5–Желілік модельдегі байланыстар

Желілік модельдің артықшылығы – иерархиялық модельге қарағанда байланыстардың әр түрлі болып келуі болып табылады.

Реляционды модель.

Реляционды модель IBM фирмасының Эдгаром Коддом қызметкерімен ұсынылған болатын. Модельдің негізгі ұғымы қатынас (relation) болып табылады.

Қатынас, ол элементтердің жиынынан (кортеж) тұрады.

Кесте жолдар (жазбалар) мен бағандардан тұрады. Кесте жолдарына кортеж сәйкес келсе, бағанға қатынас атрибуттары сәйкес келеді.

Артықшылығы оның қарапайымдылығында. Осы қарапайымдылығының арқасында пайдаланушы оны кең қолдана алады. Сонымен қатар деректерді өңдеу жеңіл әрі тез орындалады [7].

Реляционды моделдің мысалы ретінде келесілерді қарастыруға болады: dBaselll Plus және dBase IY (Ashton–Tate фирмасы), DB2 (IBM), R:BASE (Microrim), FoxPro және FoxBase (Fox Software), Windows (Borland) үшін Paradox және dBASE for, Visual FoxPro және Access (Microsoft), Clarion (Clarion Software), Ingres (ASK Computer Systems), Oracle (Oracle) және Microsoft SQL Server, сонымен қатар ресейлік реляционды ДҚБЖ: ПАЛЬМА (ИК АН УССР) және НуТech (МИФИ).

Реляционды ДҚБЖ нысанды–бағытталған жүйелердің кейбір қасиеттерін қамтиды. Ондай ДҚБЖ көбіне нысанды–реляционды деп атайды. Оның мысалы ретінде Oracle 8.х. алуға болды. Ал, оның алдындағы нұсқасы, яғни Oracle 7.х таза реляционды болып табылады. Microsoft SQL Server 2005, PostgreSQL, сол

сияқты Sav Zigzag, IBM Cloudscape, FirstSQL бұлардың бұлардың бәрі нысанды–реляционды болып табылады.

Постреляционды модель.

Постреляционды модель реляционды модельдің кеңейтілген түрі болып есептелінеді. Ол кестеде сақталған деректерді бөлуге рұқсат етеді және өріс көп мәндік бола алады, яғни бір мәннің ішінде бірнеше мәндер болу мүмкін. Осылардан құралған кестелер негізгі кестеге бағынады.

Артықшылығы, реляционды кестелерді жиынын бір постреляционды кестемен беруге болады. Сонымен қатар, тағы артықшылықтары оның ақпараттық жоғары дәрежеде көрсетілуі мен оны өңдеудің тиімділігінде.

Постреляционды модельдің мысалы ретінде ДҚБЖ uniVers айтсақ болады.

Көп өлшемді модель.

Көп өлшемді модель, реляционды модельмен бірге пайда болды. Бұрын ол реляционды модель сияқты кеңінен қолданылмаған, тек 90 жылдардың ортасынан бастап оған қызығушылық артуда. Көп өлшемді модель ақпаратты интерактивті өңдеуге арналған.

Оның артықшылығы, реляционды модельге қарағанда нысандардың өзара күрделі байланыстарын қарастырады.

Көп өлшемді модельдің мысалы ретінде мына жүйелерді қарастыруға болады: Essbase (Arbor Software), Media Multi–matrix (Speedware), Oracle Express Server (Oracle) және Cache (InterSystems).

Нысанды–бағытталған модель.

Нысанды–бағытталған модельде деректер өзерінен бөлек жазбаларын идентифициялана алады және де оларды өңдейтін функцияларды анықтай алады. Ал кемшілігі, сұраныстардың жай орындалуында.

90 жылдарды нысанда–бағытталған модельдер тәжірибе жүзінде іске асырылды. Қазіргі кезде олар кеңінен қолданыста. Атап айтатын болсақ: ДҚБЖ: POET (POET Software), Jasmine (Computer Associates), Versant (Versant Technologies), O2 (Ardent Software), ODB–Jupiter («Интелтек Плюс» ғылыми–өндіріс орталығы), Iris, Orion және Postgres [8].

2.2.2 Реляционды деректер қорына қойылатын талаптар

Біз жұмыста реляциялық деректер қорын негізге алатын болғандықтан оған қойылатын талап – деректердің қалыптастырылуы. Қалыптастыру кестелердегі деректердің қайшылықтары және үйлесімсіз болмауын қадағалайды. Қалыптастырудың алты формасы бар, бірақта тәжірибеде деректер алдыңғы үш қалыптың талаптарын қанағаттандыратын болса жеткілікті.

Деректер базасының теориясы қалыптастыру принципіне негізделінеді. Ал, қалыптастыру ұғымы функционалдық тәуелділік ұғымына негізделінеді.

– R – қатынасының A атрибуттарының әрбір мәніне B атрибутының тек бір ғана мәні сәйкес болса, онда B→A. B атрибуты A атрибутына

функционалды тәуелді;

– егер A→B–ға, B→A тәуелді болса, онда A және B атрибуттары өзара функционалды өзара тәуелді;

– егер кілт емес атрибуттар құрама кілттің бір бөлігіне ғана тәуелді болса, онда ол кілтке жартылай тәуелді деп аталады;

– егер R қатысының кілт емес B атрибуты. A атрибуттар тобына, яғни құрама кілттің барлық атрибутына тәуелді болса және ешбір жартылай функционалдық тәуелділік болмаса, онда B атрибуты A атрибуттар тобына толық тәуелді;

– егер R қатынасында A→B–ға, B→C–ға тәуелді болса, онда C атрибуты A атрибутына транзитивті тәуелді.

ДҚБЖ, SQL бағдарламналау тілімен жүзеге асады. Енді SQL бағдарламалау тіліне қысқаша тоқталайық [8].

2.2.3 SQL тілі туралы негізгі деректер

Деректерге ену мүмкіндігінің реляциялық тәсілі жазбалар тобының амалдарына негізделген. Амалдарды орындауға SQL (Structured Query Language) құрылымдық сұраныс тілінің құралдары қолданылады. Delphi қосымшаларында BDE механизмін қодану барысында деректер жиыны ретінде SQL–сұранысты орындауды беретін Query немесе StoredProc компоненттері қолданылады. SQL құралдарын жергілікті және қашықтағы деректер қорымен амалдар орындауға қолдануға болады.

Бағдарламалаудың процедуралық тілінен айырмашылығы: есептеу процестерін басқару инструкциясы (циклдер, тармақталу, басқа операторға өту) және енгізу/шығару құралдары жоқ.

Microsift Access, Visual FoxPro немесе Paradox сияқты ДҚБЖ, SQL–де сұраныстарды бағдарламалаумен байланысты әрекеттерді өздері орындайды, мысалы Query By Example (QBE) – үлгі бойынша сұраныс, пайдаланушыға сұранысты визуальды құру құралдарын білдіреді.

SQL бағдарламалау тілінің толық функционалдық мүмкіндіктерін қамтымағанмен, ол деректерге ену мүмкіндігіне арналған және оны бағдарламаны жасау құралына қосады. және Delphi жүйесіне ендірілген. Сонымен қатар, SQL командаларымен жұмыс істеу үшін сәйкес құралдар және компоненттер бар. Delphi–де мұндай компоненттерге Query, SQLQuery және ADOQuery деректер жиыны жатады.

SQL тілінің функциялары.

Статистикалық функциялар:

– AVG() – орташа мән;
– MAX() – ең үлкен мән;
– MIN() – ең кіші мән;
– SUM() – қосынды;
– COUNT() – мәндердің саны;

– COUNT(*) – нөлдік емес мәндердің саны.

Жолдармен жұмыс істеуге арналған функциялар:

– UPPER(Str) – Str символдық жолды жоғарғы регистрге айыстыру;

– LOWER(Str) – Str символдық жолды төменгі регистрге айыстыру;

– TRIM(Str) – Str жолының бастапқы және соңғы бос орындарын өшіру;

– SUBSTR (Str FROM TO) – Str жолынан, өзі символдардан тұратын, n1 жолынан бастап, n2 жолымен аяқталатын ішкі жолды бөліп алу;

– CAST(<Expression> AS <Type>) – Expression өрнегін Type типіне келтіру.

Ай–күн (Дата) және уақытпен жұмыс істейтін функция:

– EXTRACT(<Элемент> FROM <Өрнек>) – дата және уақыт мәндерінен тұратын өрнектен, сәйкес көрсетілген элементтен мәндер алу; дата және уақыт элементі ретінде YEAR, MONTH. MINUTE және SECOND элементтерін көрсетуге болады.

Кестелермен келесі амалдарды орындауға болады:

– жаңа кесте құру;

– кестені өшіру;

– кесте өрістерінің құрамын өзгерту;

– индексті құру және өшіру.

Бұл әрекеттер SQL тілінің инструкцияларының көмегімен орындалады [4].

2.2.4 Кесте құру және өшіру

Кестені құру үшін CREATE TABLE инструкциясы қолданылады:

```
CREATE TABLE <Кесте аты>
        (<Өріс аты> <дерек типі>
         . . .
         <Өріс аты> <дерек типі>);
```

Файлдың кеңейтілуі бойынша кестенің форматы автоматты түрде, яғни .db – Paradox кестесі үшін, .dbf dBase кестесі үшін анықталады. Кесте файлы дерек қор псевдонимі көрсетілген, МҚ каталогында орналастыру керек.

SQL тілінің деректер типі және сәйкес Paradox–тың деректер типтері келесі кестеде көрсетілген (2.1-кесте).

К е с т е 2.1 – SQL тілінің деректер типі

SQL	Paradox
SMALLINT	Short
INTEGER	Long Integer
DECIMALL	BCD
NUMERIC(x,y)	Number
FLOAT(x,y)	Float(x,y)
CHARACTER(n)	Alpha

Кесте 2.1 – дің соңы

SQL	Paradox
VARCHAR(n)	Alpha
DATE	Date
BOOLEAN	Logical
BLOB(n,1)	Memo
BLOB(n,2)	Binary
BLOB(n,3)	Formatted memo
BLOB(n,4)	OLE
BLOB(n,5)	Graphic
TIME	Time
TIMESTAMP	Timestamp
MONEY	Money
AUTOINC	Autoincrement
BYTES(n)	Bytes

SQL тілінің көмегімен кесте құру мысалы:

```
CREATE TABLE Student.dbf
        (Numer Integer
         Name CHAR(15)
         Datarogd DATE);
```

Paradox кестесі үшін PRIMARY KEY сипаттамасын және жақшаның ішіне оның өрістерін көрсетіп, негізгі немесе алғашқы кілтті анықтауға болады. Кілттік өріс өрістер тізімінде бірінші болып орналасуы керек. Негізгі кілтті құрумен жаңа кестеге құруға мысал:

```
CREATE TABLE Personal.db
        (Code AUTOINC
         Name CHAR(15)
         Post CHAR(15)
         Oklad Numeric(10,2)
         PRIMARY KEY(Code));
```

Кестені өшіру үшін келесі инструкция қолданылады:

DROP TABLE <Кесте аты>;

Мысалы DROP TABLE <student>; Дискіден student атты кестеге қатысы бар барлық файлдар өшіріледі.

Кесте өрістерінің құрамын өзгерту.

Кесте өрістерінің құрамын өзгерту өрістерді қосу немесе өшіруді білдіреді және оның құрылымын өзгертуге әкеледі, сонымен қатар кестені басқа қосымшалар қолданбауы керек. Кесте өрістерінің құрамын өзгерту ALTER TABLE инструкциясының көмегімен орындалады:

```
ALTE TABLE <кесте аты>
        ADD <1–өріс аты> <1–өріс типі>,
        DROP <1–өріс аты>,
```

. . .
ADD <n–өріс аты> <n–өріс типі >,
DROP <n–өріс аты>;

ADD операндысы кестеге жаңа өріс қосады, өріс аты және типі CREATE TABLE инструкциясындағыдай беріледі, ал DROP операндысы көрсетілген атымен кестеден өрісті өшіреді. ADD және DROP операндылары бір–бірімен байланысты емес және еркін түрде орналасуы мүмкін [5].

3 Дерекқорды жобалау және оны іске асыру

3.1 ДҚ құрудың бастапқы кезеңі

Жүктемені бөлу жұмысының жалпы сипаттамасы.

Ипотека жүйесінің жүктемесін бөлу жұмысының негізін құрайды. Жүктемелерді бөліп, бөлінбеген жүктемелерді көріп, оларға өзгеріс енгізе алады.

Мәселелер және шектеулер:

– құпия сөзсіз кестені өңдеуге және өзгеріс енгізуге мүмкіндіктің болмауы;
– жүктеменің дұрыс бөлінуі;
– есеп беру жүйелі түрде болуы.

3.1.1 Дерекқордың функционалдығын сипаттау

Жобаланатын дерекқор және бағдарламалық интерфейс ипотека жүйеснің жүктемесін бөлуге арналған. Бағдарламалық қосымшада келесі функциялар іске асырылған:

– ақпараттық функция, жүктеме, бөлінген және бөлінбеген жүктемелер туралы толық ақпаратты береді;
– енгізу функциясы, дерекқорға жаңа ақпарат енгізуге мүмкіндік береді.

3.2 Концептуалды жобалау

3.2.1 Қажетті ақпарат: пайдаланушылардың талабы

Жоба пайдаланушының барлық негізгі талаптарын қанағаттандыру керек. Бұл жүйенің негізгі пайдаланушылары ипотека беретін мекемелердегі қызметкерлер болады. Негізгі талаптар алынған ақпараттың ұсынылатын тиімділік деңгейінде негізделеді. Жүйеге қойылатын негізгі талаптардың тізімі:

– жүйе қолданыста қарапайым болу керек. интерфейс ыңғайлы болу керек;
– жүйе қауіпсіздікті қамтамасыз ету керек;
– жүйе дерекқордың тұтастығын қамтамасыз ету керек;
– жүйе толығымен интеграциялану керек, яғни мәліметтердің артылуы және артық жаңартулар болмау керек.

3.2.2 Дерекор және қосымшаны жобалау құралдарын таңдаудың негіздемесі

Дерекқорларды басқару жүйесі.

Барлық талаптарды қамтамасыз ететін информациондық жүйені жүзеге асыру үшін бірінші кезекте реляциондық дерекқорын және серверлі архитектура клиентін қолданудың анық шешіміне әкелетін орталық деректер сақтаудың әдісін таңдау қажет.

Қазіргі кезде қойылған тапсырмалар талаптарын ДҚБЖ мүмкіншілігіне қарай және тапсырыс берушінің таңдауына байланысты таңдауға болатын бірнеше ДҚБЖ көрсетілген [2,4,7]:

- орнату және қолданудың жеңілдігі;
- кең таралған;
- қымбат серверлік қуатты орнатуды қажет етпейді;
- өндіргіштігінің жақсы көрсеткіші;
- деректерді сақтау сенімдігі мен қауіпсіздігін қамтамасыз етудің жеке құралдары;
- коммерциялық емес қолдану үшін өнімді сызғышта арзан түрлерінің болуы.

Microsoft SQL Server төменөндіргіштік серверлік қуаттарға есептелген, сонымен қатар қолданушылық жұмыс станциясында қолданушыға өндіргіштіктің төмендеуінің білінуінсіз жұмыс істеуге мүмкіндігі бар. Сонымен қатар кең таралғандық критерийларына және деректерді сақтау сенімдігі мен қауіпсіздігін қамтамасыз етудің жеке құралдарының бар екендігіне жауап береді.

SQL компьютерлік дерекқорда сақталатын деректерді өңдеу және оқу үшін арналған құрал болып табылады, және қолданушының дерекқормен өзара қатынас ұйымдастыру үшін қолданылатын программалау тілі болып табылады. Шындығында SQL тек бір белгілі типті – реляционды дерекқормен жұмыс істейді. SQL ДҚБЖ қолданушыға ұсынатын барлық функционалдық мүмкіншіліктерді жүзеге асыру үшін қолданылады, соның ішінде:

- деректерді ұйымдастыру. SQL қолданушыға деректердің көрсетілімі құрылымын өзгертуге, сонымен қатар дерекқордың элементтері арасында қатынас құруға мүмкіндік береді;
- деректерді оқу. SQL қолданушыға немесе қосымшаға дерекқордың ішіндегі деректерді оқуға және олармен қолдануға мүмкіндік береді;
- деректерді өңдеу. SQL қолданушыға немесе қосымшаға дерекқорды өзгертуге, яғни оған жаңа деректер енгізуге, оған қоса бұрыннан бар деректерді өшіруге немесе жаңартуға мүмкіндік береді;
- қатынас құруды басқару. SQL-дің көмегімен қолданушының деректерді оқу және өзгерту және рұқсат етілмеген қатынас құрудан қорғауға мүмкіндігін шектеуге болады;
- деректерді бірге қолдану. SQL параллельді түрде жұмыс істейтін қолданушыларға бір-біріне кедергі келтірмес үшін деректерді бірге қолдануын қадағалайды;

– деректердің тұтастығы. SQL дерекқордың тұтастығын қамтамасыз етіп, оны келісімсіз өзгерістерден немесе жүйенің бас тартуынан қираудан қорғауға рұқсат береді.

SQL – дерекқорды программалау тілі. Дерекқорға қатынас құру үшін бағдарлаушылар өздерінің бағдарламаларына SQL командаларын қояды. Бұл әдістеме қолданушылар жазған бағдарламаларда да, дерекқорлардың қызметтік бағдарламаларында да қолданылады (есеп беру генераторы және деректерді енгізу құралдарында). Осылайша, SQL адамдарға, бағдарламаларға және есептеуіш жүйелерге реляционды дерекқорлардағы ақпараттарға қатынас құруды қамтамасыз ететін қажетті және мықты құралға айналды.

3.3 ER моделін құру

ДҚ құрудың бастапқы кезеңінде алғашқы нысандар жиынтығын анықтау қажет. Бұл ақырғы пайдаланушылар мен жобалаушы көзқарасы бойынша жүйе объектілері туралы ең маңызды ақпарат болып табылады. Ипотека жүйесін автоматтандыру 3.1–кестеде көрсетілген нысандары қолданады.

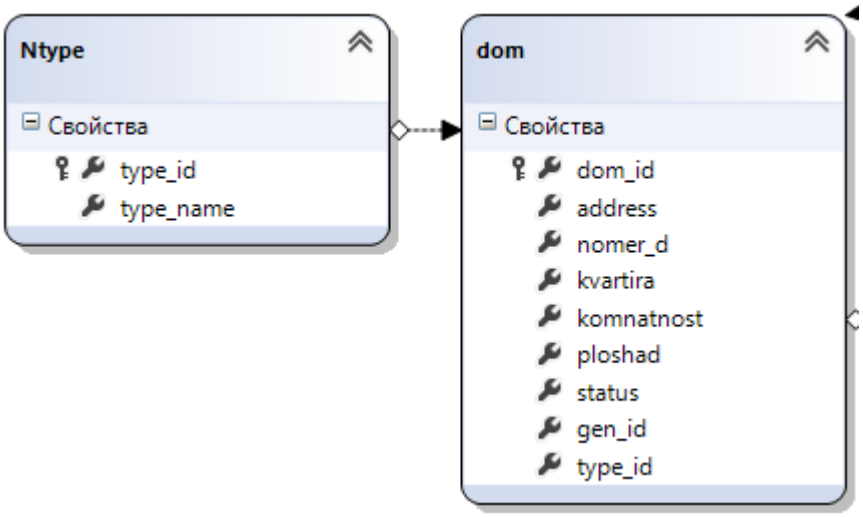
Кесте 3.1 – «Ипотека жүйесін автоматтандыру» дерекқорының нысандары

Нысан аты	Сипаттамасы
Dom	Тұрғын үй туралы мәліметтер
Klient	Ипотека алған тұлғалар туралы мәліметтер
Konsultant	Қызметкерлер туралы ақпараттар
Ntype	Тұрғын үйдің типі
Genpodryadchik	Құрлысты салушылыр туралы мәліметтер
Semestr	Семестр

Жобалаушы және ақырғы пайдаланушы нысандарды анықтау кезінде келісімге келу керек. Жобалаушы нысандар арасындағы байланыстарды операциялардың сипаттамасына негізделіп анықтайды. Нысандар арасындағы байланыстар операциялардың толық сипаттамасы негізінде құрылған бизнес-ережелерге негізделеді.

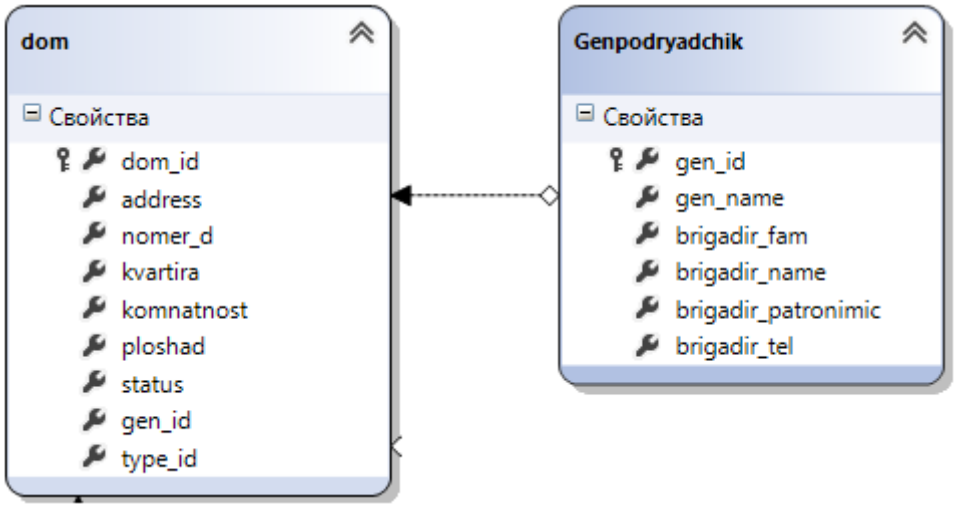
3.3.1 Бизнес ережелер

1) Әрбір тұрғын үйдің бір ғана типі болады. (сурет 3.1)



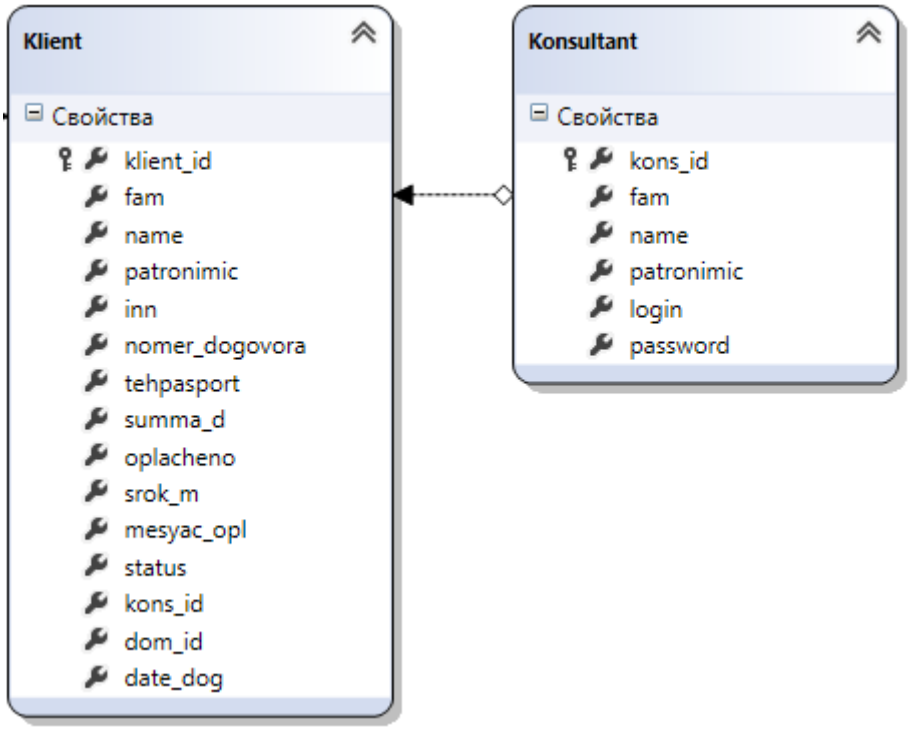
Сурет 3.1 – 1– бизнес ереже үшін ER сегменті

2) Әрбір тұрғын үйді бір компания салады, ал компания өзі бірнеше тұрғын үйлерді салады. (сурет 3.2)



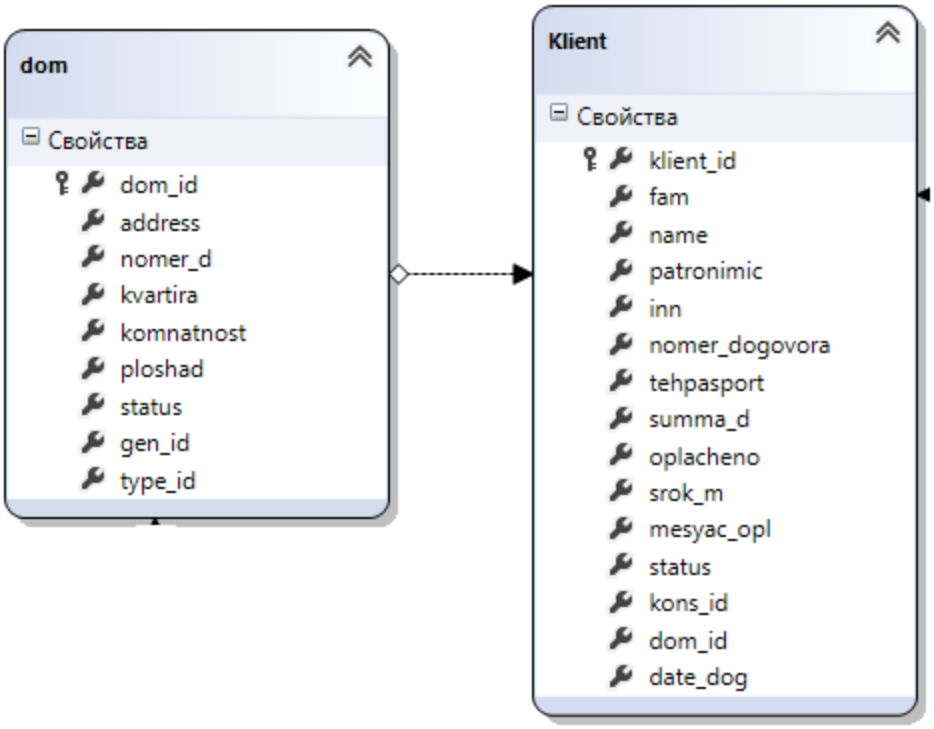
Сурет 3.2 – 2 – бизнес ереже үшін ER сегменті

3) Қызметкер бірнеше клиенттерді қабылдайды, сәйкесінше, бір клиентті бір ғана қызметкер қабылдай алады. (сурет 3.3)

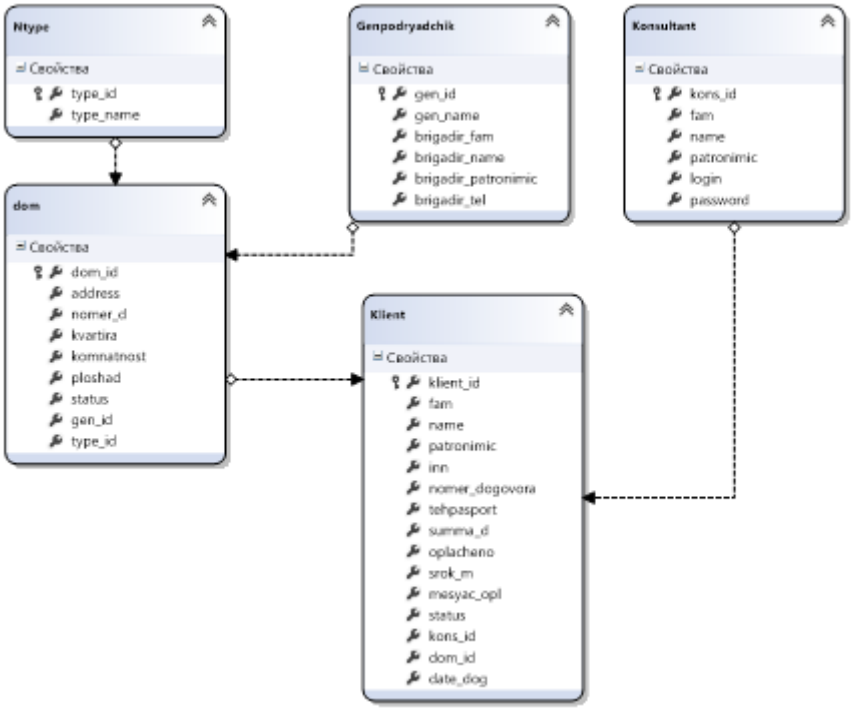


Сурет 3.3 – 3 – бизнес ереже үшін ER сегменті

4) Әрбір клиент бір пәтерді ипотекаға ала алады, сәйкесінше, бір тұрғын үйдің тек бір ғана иесі болады. (сурет 3.4)



Сурет 3.4 – 4 – бизнес ереже үшін ER сегменті



Сурет 3.5 – ER диаграмма

3.3.2 Реляционды қатынастарды құрастыру

Әрбір реляционды қатынас бір нысанға сәйкес келеді және оған нысанның барлық атрибуттары енгізіледі. Әрбір қатынас үшін алғашқы кілт және сыртқы кілттерді (егер бар болса) анықтау қажет. Қатынастар 3.2–3.6–кестелерінде келтірілген. Әрбір қатынас үшін ішкі аты, типі мен ұзындығы бар атрибуттар көрсетіледі Мәліметтер типі келесідей белгіленеді: int – сандық, char – символдық, Datetime – мезгіл (оның ұзындығы ДҚБЖ-не байланысты әртүрлі болады, сондықтан ол көрсетілмейді).

К е с т е 3.2 – Dom қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Тұрғын үй коды	dom_id	int	Алғашқы кілт
Адресі	Address	varchar(30)	Қажетті өріс
Үй нөмері	nomer_d	varchar(10)	Қажетті өріс
Пәтер нөмері	Kvartira	varchar(10)	Қажетті өріс
Пәтер ауданы	Ploshad	varchar(10)	Қажетті өріс
Құрылыс салушылар	gen_id	int	Сыртқы кілт
Тұрғын үй типі	type_id	int	Сыртқы кілт

К е с т е 3.3 – Ntype қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Тип коды	type_id	int	Алғашқы кілт
Тұрғын үй типі	type_name	varchar(30)	Қажетті өріс

К е с т е 3.4 – Konsultant қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Қызметкер коды	kons_id	int	Алғашқы кілт
Қызметкер фамилиясы	Fam	varchar(30)	Қажетті өріс
Қызметкер аты	Name	varchar(20)	Қажетті өріс
Қызметкер тегі	Otech	varchar(30)	Қажетті өріс
Логин	Login	varchar(30)	Қажетті өріс
Парол	Password	varchar(30)	Қажетті өріс

К е с т е 3.5 – Klient қатынасы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Клиент коды	klient_id	int	Алғашқы кілт
Клиент фамилиясы	Fam	varchar(30)	Қажетті өріс
Клиент аты	Name	varchar(20)	Қажетті өріс
Клиент тегі	Otech	varchar(30)	Қажетті өріс
ИНН	Inn	bigint	Қажетті өріс
Договор нөмері	nomer_dogovora	int	Қажетті өріс
Техпаспорт	Tehpasport	varchar(max)	Қажетті өріс
Ипотека соммасы	summa_d	bigint	Қажетті өріс
Төленген соммасы	Oplacheno	bigint	Қажетті өріс
Төлену мерзімі	srok_m	int	Қажетті өріс
Төленген мерзімі	mesyac_opl	int	Қажетті өріс
Ипотека статусы	Status	int	Қажетті өріс
Договор күні	date_dog	date	Қажетті өріс

Кесте 3.5 – дің соңы

Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Қызметкер коды	kons_id	int	Сыртқы кілт
Тұрғын үй	dom_id	int	Сыртқы кілт

К е с т е 3.6 – Genpodryadchik қатынасы

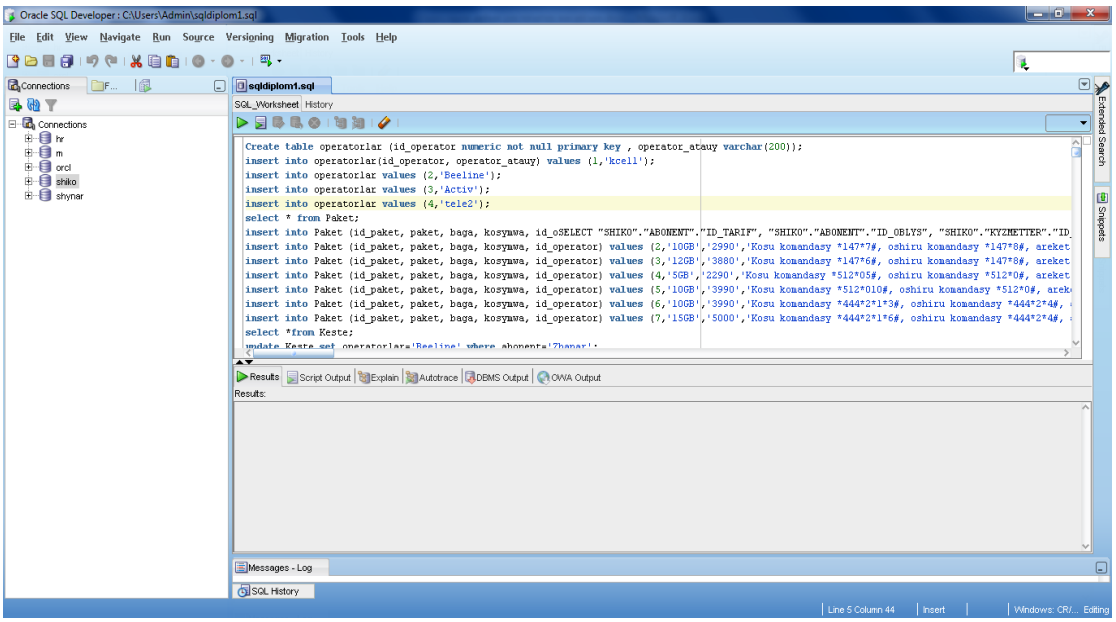
Өріс аты	Атрибут аты	Мәліметтер типі	Ескерту
Құрлыс коды	gen_id	int	Алғашқы кілт
Құрлыс компаниясының аты	gen_name	varchar(30)	Қажетті өріс
Бригадир фамилиясы	brigadir_fam	varchar(30)	Қажетті өріс
Бригадир аты	brigadir_name	varchar(20)	Қажетті өріс
Бригадир тегі	brigadir_otch	varchar(30)	Қажетті өріс
Телефон нөмері	brigadir_tel	varchar(20)	Қажетті өріс

3.3.3 Логикалық жобалау

Логикалық жобалау кезеңінде дерекқор моделі жүйені іске асыру кезінде қолданылатын белгілі бір ДҚБЖ (MS SQL Server) үшін жарамды форматқа трансформаланады. Логикалық жобалаудың мақсаты реляционды кестелік құрылымдарды құру болып табылады. Жобаланатын дерекқорда кестелердің және байланыстардың құрылымдарын MS SQL Server көмегімен жасаймыз.

Кестелер құру.

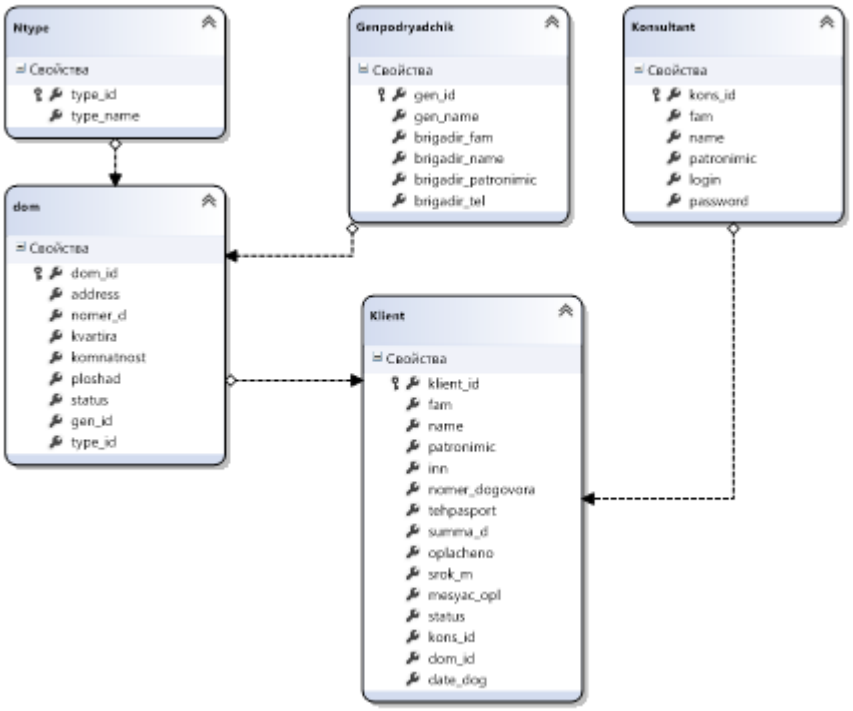
Кестелер CREATE TABLE командасымен құрылады. CREATE TABLE кесте атын және бағандар аттарын анықтайды. Сонымен қатар ол деректердің типін және бағандардың өлшемін анықтайды. Ең ыңғайлы бұл генерацияланған скриптті Query Analyzer–де (MS SQL Server) қосу. Кестелерді құруға арналған скрипт А қосымшасында келтірілген. (сурет 3.6)



Сурет 3.6 – Query Analyzer–де кестелерді құру

ДҚ диаграммасы.

Концептуалды моделді дерекқордың сәйкес құрылымына енгізілу тәсілі қандай болса да, ДҚ реляционды сұлбасы жобаға сәйкес болу керек. (сурет 3.7)



Сурет 3.7 – ДҚ диаграммасы

3.4 Физикалық жобалау

3.4.1 Дерек қорына қосымша құру

Тапсырманың қойылуы.

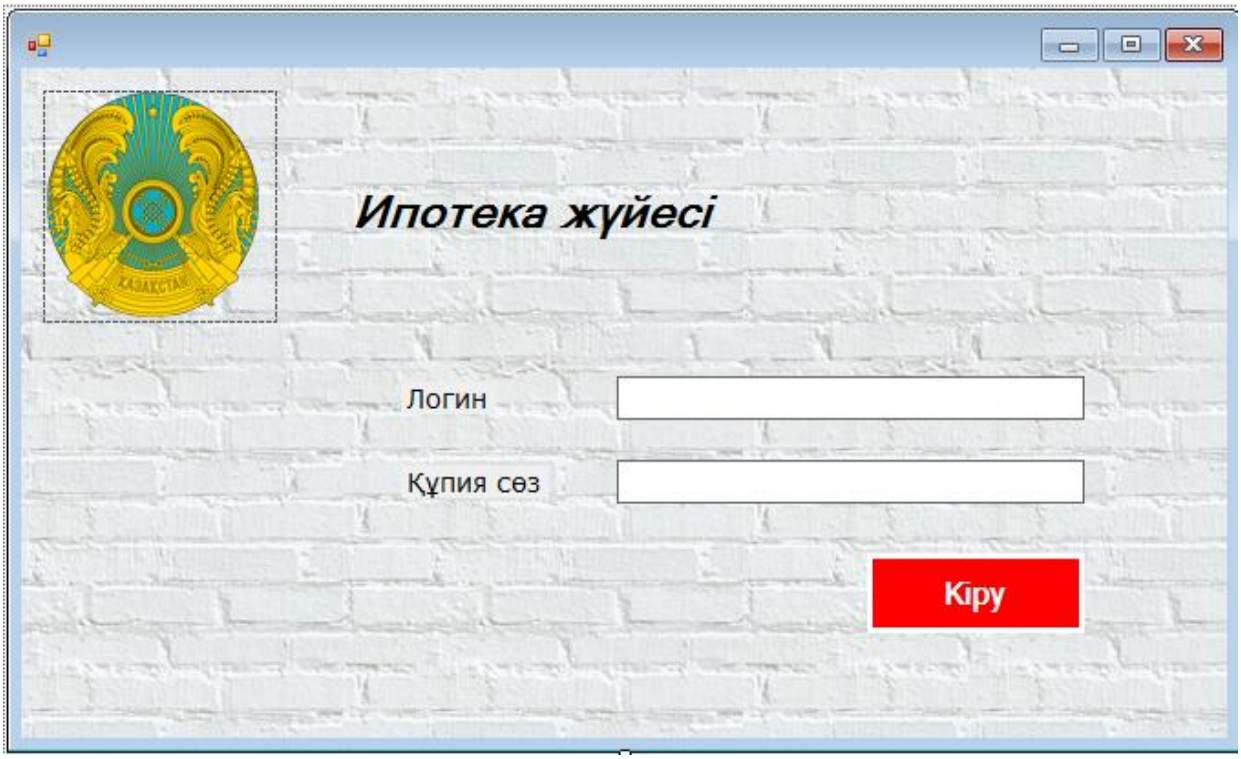
Мәліметтер қорымен жұмыс істеу қосымшасын жасау әдістемесі кәдімгі программаны құру әдістемесінен ешқандай айырмашылығы жоқ. Формаға керекті компоненттер енгізіледі, компоненттердің қасиеттерінің мәндері қойылады, керекті оқиғаны өңдеу процедурасы жасалады.

Мәліметтер қорымен жұмыс істеу қосымшасы мәліметтерге енуді қамтамасыз ететін компоненттерден, өрістердің мәліметтерін қарап шығу және редакциялау мүмкіндіктерінен тұрады. Мәліметтерге кіруді қамтамасыз ететін компоненттер Data Access компоненттер палитрасы қосымшасында, ал мәліметтерді бейнелеу компоненттері – Data Controls қосымшасында болады [1,5,9].

Database компоненті мәліметтер қорын кестелер жиынтығы ретінде, ал Table компоненті – мәліметтер қорының бір кестесі. DataSource (мәліметтер көзі) компоненті мәліметтерді бейнелеу-редакциялау компоненті (мысалы DBGrid компонентін) мен мәліметтер көзі (Table компоненті немесе SQL–сұраныс нәтижесі) арасындағы байланысты қамтамасыз етеді. DataSource компоненті мәліметтер көзін тез таңдауды, сол компоненттердің біреуін пайдаланып, мысалы DBGrid кестеден мәліметтерді көру үшін немесе осы кестеге SQL–сұранысты орындау нәтижесін көруге мүмкіндік береді [2,6].

3.4.2 Бағдарламаны сипаттау

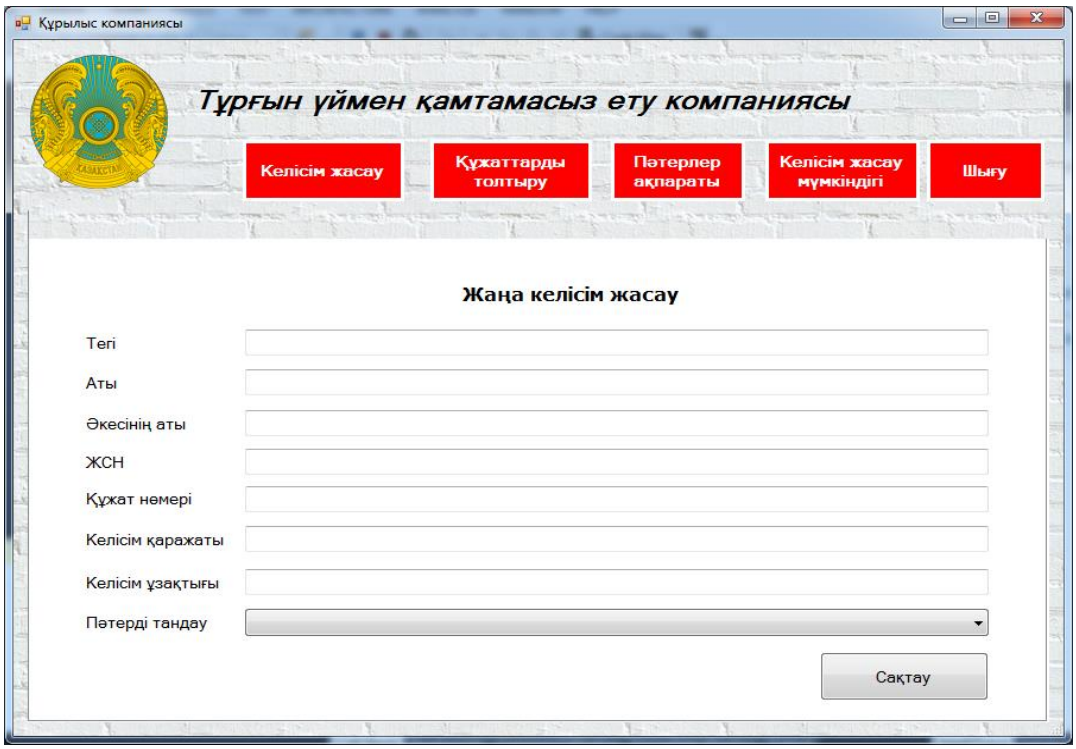
Бағдарламаның алғашқы беті авторизация бөлімінен тұрады. Жүйенің екі қолданішылары болады, администратор және қызметкерлер. Администратордың өзінің паролы, ал қызметкерлердің бөлек паролі болады. (сурет 3.8)



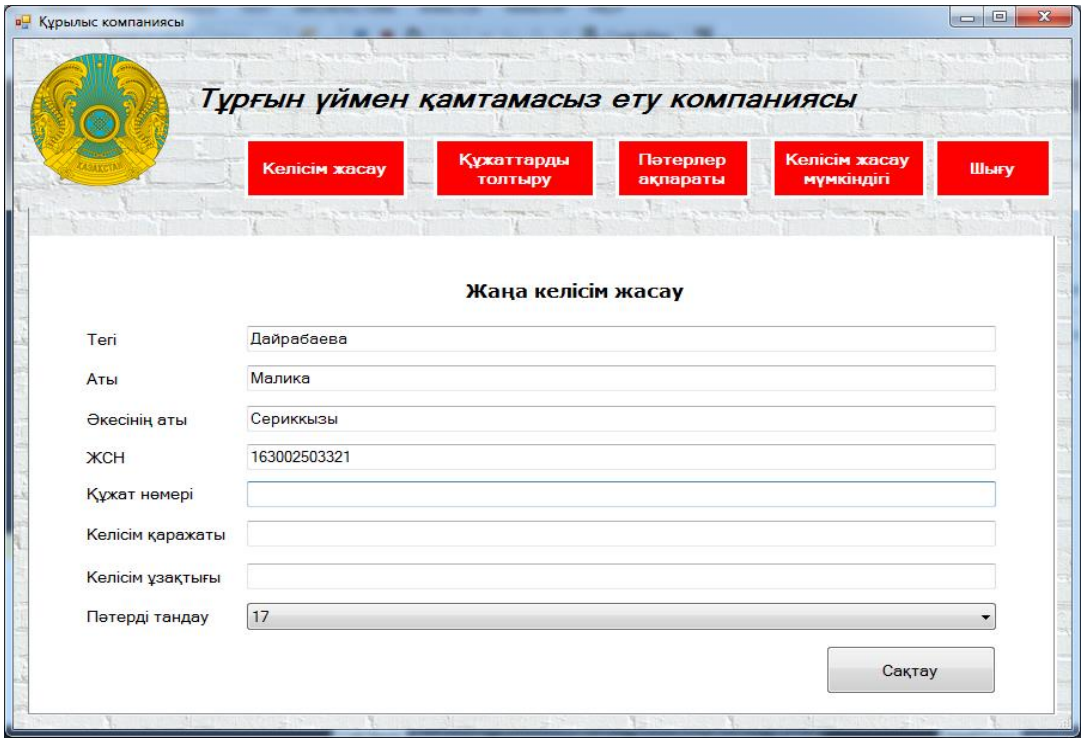
Сурет 3.8 – Программаның алғашқы терезесі

Қызметкерлер бөлімі.

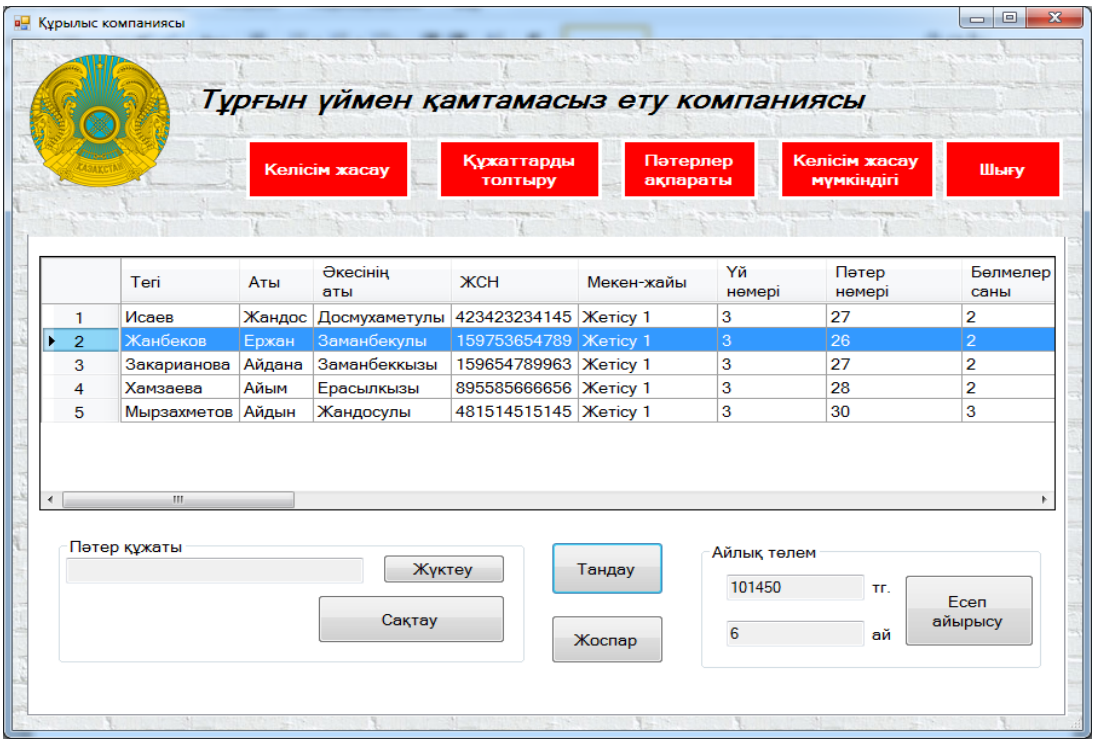
Бұл бөлімде 5 батырма бар: келісім жасау, құжаттарды толтыру және пәтерлер ақпараты,келісім жасау мүмкіндігі, шығу. Келісім жасау батырмасында ипотека алу кезіндегі келісім құжатын толтыру орындалады. Ал құжаттарды толтыру бөлімінде клиенттердің ипотеканы төлеу жоспарын көруге және Excel–ге шығартуға болады. Пәтерлер ақпараты батырмасын басып, пәтерлерді қарап, таңдауға мүмкіндік бар. Келісім жасау мүмкіндігі батырмасында келісім уақыты мен ай сайынғы төлемін есептеп береді. Шығу батырмасы жүйеден шығу үшін керек. (сурет 3.9 – 3.13)



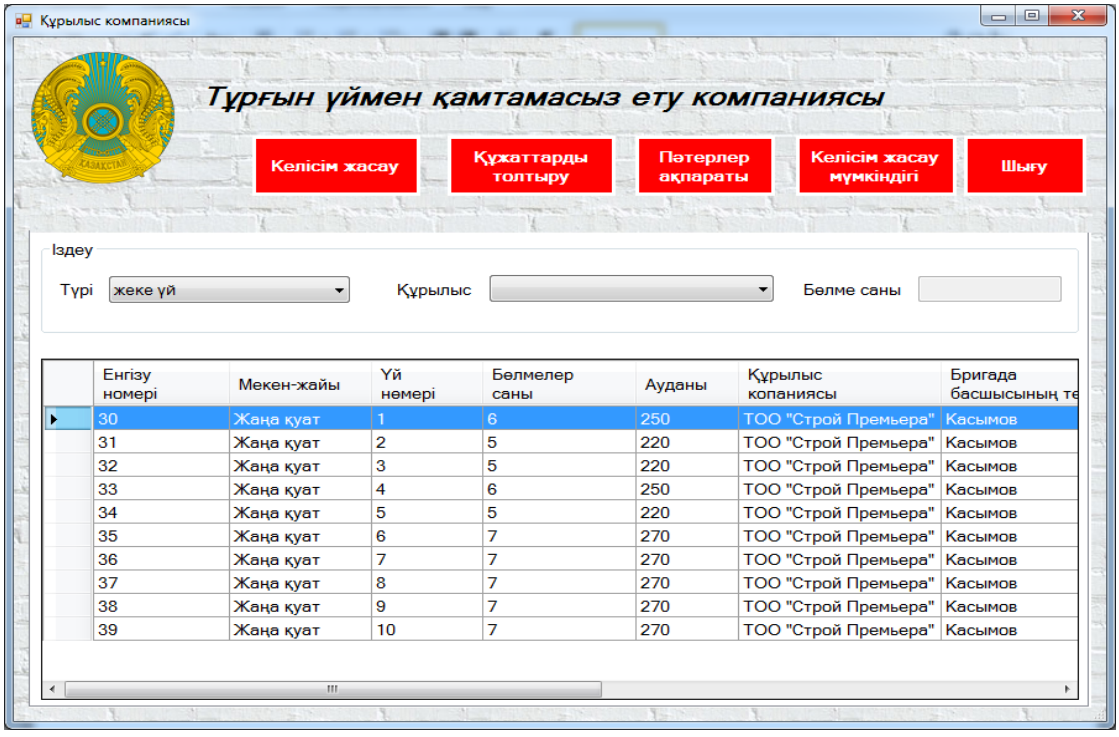
Сурет 3.9 – Қызметкерлер терезесі



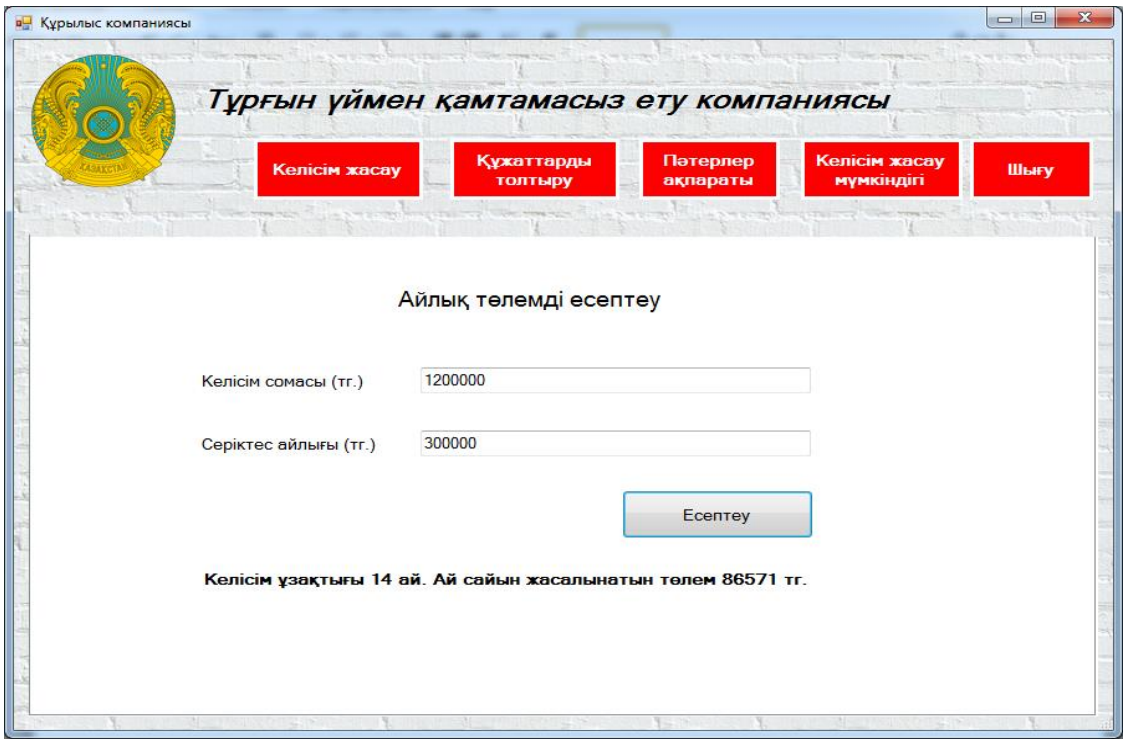
Сурет 3.10 – Жаңа келісім жасау терезесі



Сурет 3.11 – Құжаттарды толтыру терезесі



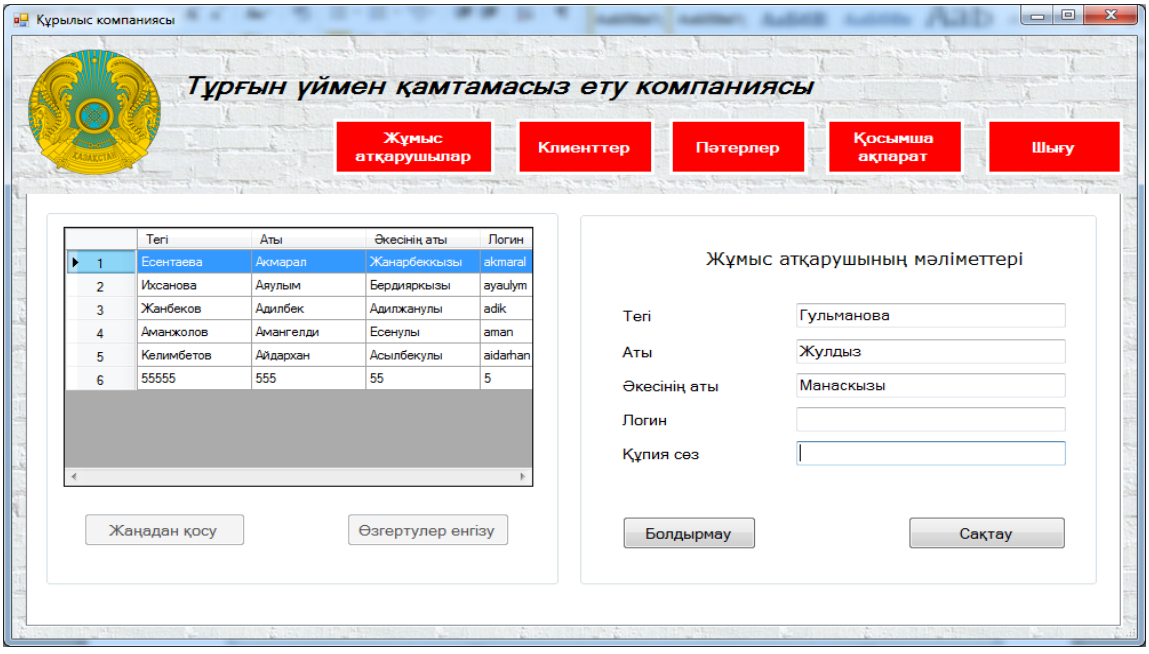
Сурет 3.12 – Пәтерлер ақпараты терезесі



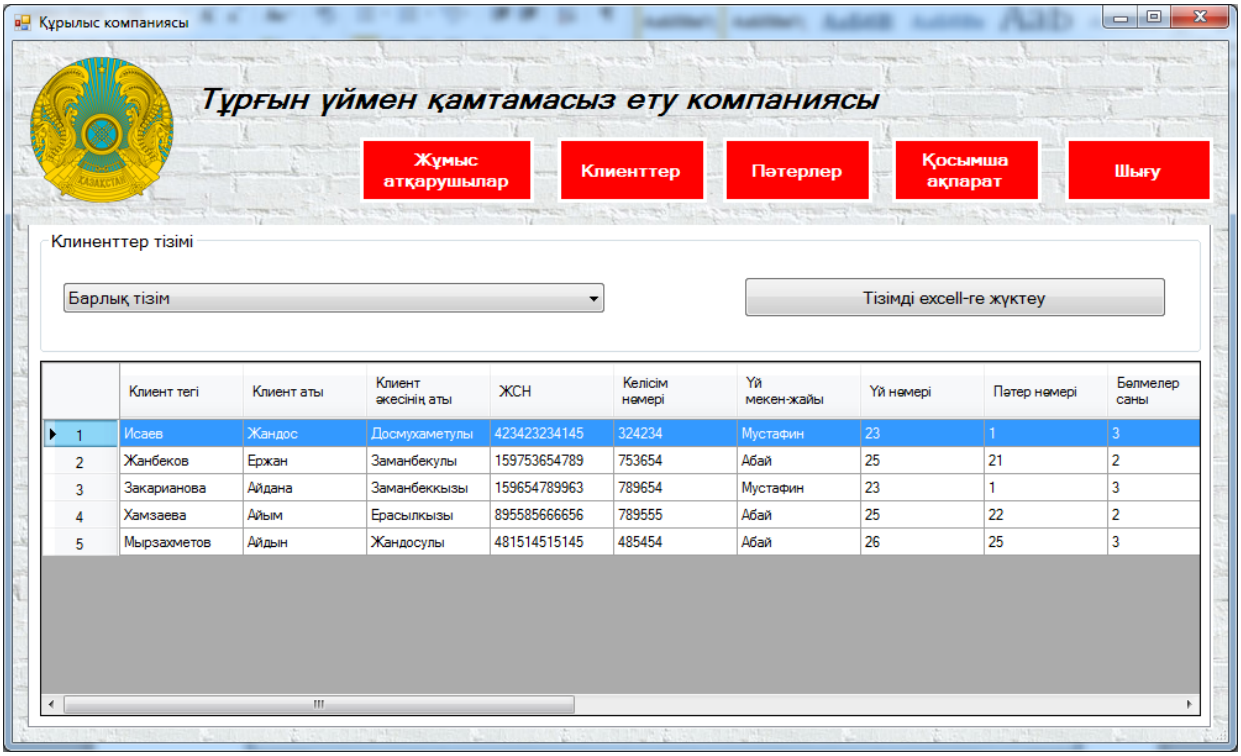
Сурет 3.13 – Келісім жасай мүмкіндігі терезесі

Администратор беті.

Бұл бет бес бөліктен тұрады: жұмыс атқарушылар бөлімі, клиенттер, пәтерлер, қосымша ақпарат, шығу бөлімі. Жұмыс атқарушылар бөлігінде қызметкерлерді қосу және жою орындалады. Клиенттер бөлігінде клиенттер тізімін қарап, Excelге шығаруға болады. Пәтерлер бөлігінде пәтерлер тізімі жаңартылады. (сурет 3.14–3.15)



Сурет 3.14 –Администратор терезесі



Сурет 3.15 –Клиенттер терезесі

4 Тіршілік қауіпсіздігі

4.1 Жұмыс орнындағы еңбек жағдайын талдау

Бұл дипломдық жобаның мақсаты ипотекалық жүйені автоматтандыру болып табылады. Бұл бағдарлама ыңғайлы әрі уақытты үнемді пайдалануға мүмкіндік береді, сонымен қатар, жұмыс барысында мүмкін болатын қателерді азайтады.

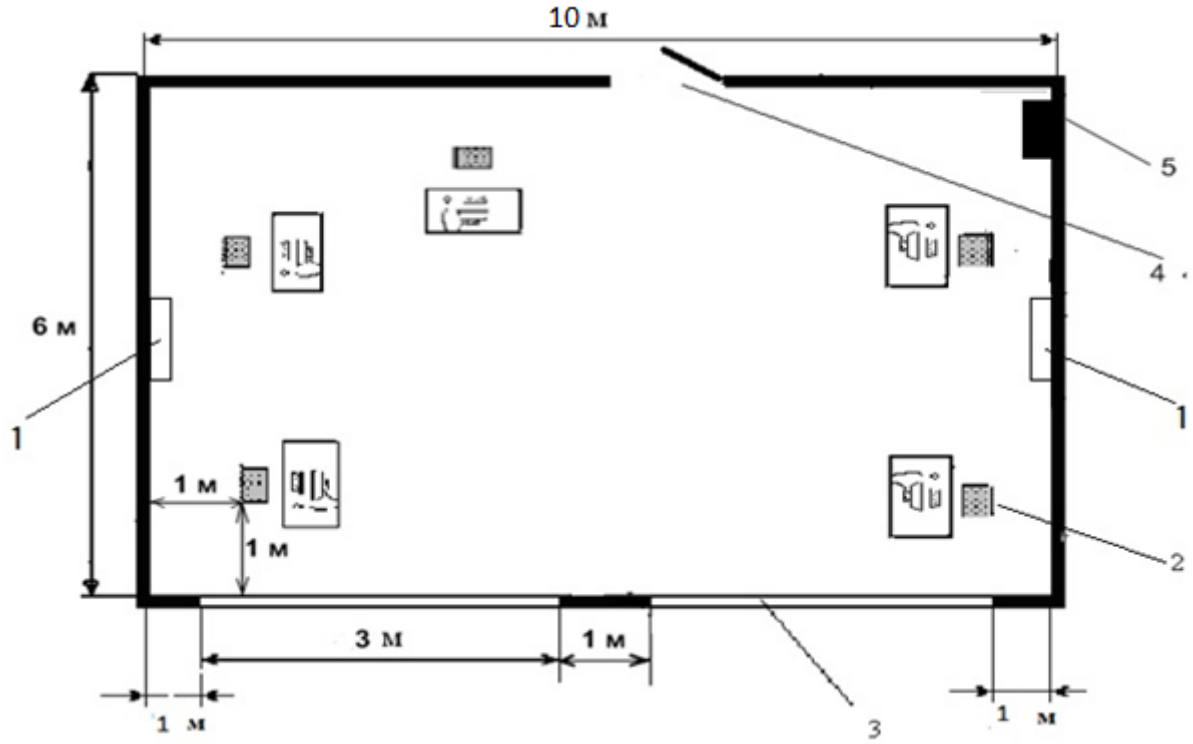
Бағдарламалық өнімді жасау барысында компьютермен жұмыс істеудің қауіпсіздігі мен еңбек жағдайының нұсқаулары өте маңызды. Әрбір ұлттық компанияның осындай нұсқаулары болады.

Дербес компьютерлер жерлеу (нөлдеу) элементтерімен қамтылып, ортақ жерленген (нөлденген) желіге қосылуы керек.

Жұмыс орындарын Қазақстан Республикасының Денсаулық сақтау министірлігінің 2010 жылы 28 қыркүйектегі «Дербес компьютерлерге, бейнетерминалдарға және олармен жұмыс істеу жағдайларына қойылатын санитариялық-эпидемиологиялық талаптар» атты №767 бұйрығымен орындалуы керек.

Бағдарламалық өнімді ипотека алушы адамдарды қабылдау бөліміндегі жұмысшылар қолданады. Осыған байланысты жұмысшылардың жұмыс қабілеттілігі мен денсаулығын сақтау, жұмыс уақытында жақсы сезінуі үшін оптималды жұмыс жағдайын қарастыруы керек. Сондықтан ең алдымен жұмыс орнының ауданын және жұмысшылардың санын ескеру керек. Есептеулер жұмыс орнындағы максималды адам санымен жүргізіледі.

Қабылдау бөлмесі сурет 4.1 – де көрсетілген. Бөлменің ұзындығы 10 м, ені 6 м және биіктігі 3 м, ұзындығы 2 м екі терезе бар. Бөлмеде 5 адам жұмыс істейді, жұмыс графигі – аптасына бес күн, күніне сегіз сағат. Дербес компьютер операторының жұмысы ұзақ көру жұмысымен байланысты болғандықтан, қабылдау бөлмесіндегі жарықтануды есепке алу қажет. Жарықтану деңгейі психикалық функциялардың күйіне және ағзадағы физиологиялық үрдістерге әсер етеді. Бөлме компьютерлік құрылғылар мен оргтехникамен жабдықталған, сол себептен кафедраның персоналы артық жылулық сәулеленуге шалдығады. Сондықтан персоналдың қолайлы еңбек ету шарттарын қамтамасыз ету үшін микроклимат параметрлерін нормалау қажет.



Сурет 4.1 – Жұмыс бөлмесінің жоспары

мұндағы 1 – Кондиционерлер орны

2 – операторлардың жұмыс орындары;

3– терезе жақтаулары;

4 – есік жақтауы;

5 – өрт сөндіргіш балоны.

Жұмысшының негізгі жұмысы ақпаратты қабылдау немесе енгізу, бағдарламалар бойынша ЭЕМ–да тапсырманың есептелуін түзету және бақылау, машина тоқтап немесе істен шыққан жағдайда дер кезінде шара қолдану, сонымен қатар тапсырыс берушіге қортындыны жібергені туралы ақпарат беру. Міне осыған орай, жұмысшы организмінің энергия шығынына байланысты жұмыс катергориясын таңдаймыз. МЕСТ 12.1.005-88 ССБТ «Жұмыс орнының ауасы, санитарлық-гигиенаның жалпы талаптары» 4.1 – кестеде келтірілген. Жұмысшылардың жасайтын еңбегі бірінші категорияға жатады (Іа).

К е с т е 4.1 – Организмнің энерго шығыны бойынша жұмыс категориясы

Жұмыс категориясы	Категория	Адам организмінің энергия шығыны, Ккал/сағ.	Жұмыс сипаттамасы
Жеңіл физикалық жұмыс I а	I а	138-172	Жұмыс отыру күйінде өтеді

Микроклиматтың күйін қадағалау қабылдау залында қолайлы жағдайды орнатуға көмектеседі. Ал жұмыс орнының қолайлы жағдайларында адамдардың жұмыс істеу қабілеттері жақсарады, жұмысшылардың ауруға шалдығуы азаяды сонымен қатар, еңбек өндірісі артады. Жұмыс орнындағы нормаланған климаттық параметрлер 4.2 – кестеде келтірілген.

К е с т е 4.2 – Температураның, ылғалдылық пен ауа қозғалысының нормалары МЕСТ 12.1.055.88ССБТ

Жыл мезгілі	Ауа температурасы, ^{0}С			Салыстырмалы ауа ылғалдылығы, %		Ауа қозғалысының жылдамдығы, м/с	
	Қолайлы	Рұқсат етілген		Қолай-лы	Рұқсат етілген	Қо-лайлы	Рұқсат етілген
		Жоғар-ғы	Төменгі				
Суық	21–23	24	20	40–60	75	0,1	<0,1
Жылы	22–24	28	22	40–60	60–70	0,2	0,1–0,3

Жазғы микроклиматтың қысқыға қарағанда, ауа баптауы сәл жоғарырақ. Кесте бойынша, ауа температурасы 23–25°, ылғалдылығы 30–50%, ауа қозғалысының жылдамдығы 0,2м/с. 55% дейінгі ауа ылғалдылығы құрғақ деп есептеледі, 56–70% - деңгейінде орташа құрғақ, 71–85% –орташа ылғалды және 85% жоғары – тым ылғалды. Ең үлкен ылғалдылық пен салыстырмалы ылғалдылық арасындағы айырым қанығу тапшылығы деп аталады. Қанығу тапшылығы қаншалықты үлкен болса, соншалықты ылғалды ауа жұтады. Ауаның температурасы 23–24^{0}С – де ауа қозғалысының жылдамдығы 0,15 м/с дейін болған жөн. Адамның өмір тіршілігіне байланысты ауа құрамының өзгеруі, оған берілген көмір қышқыл газымен есептеледі – CO_2. Бөлмедегі CO_2 шоғырлануы 4.3 – кестеде келтірілген.

К е с т е 4.3 – Бөлме ауасындағы көмір қышқыл газының (CO_2) шоғырлану шектері

Бөлме	CO_2 шоғырлану шегі	
	Салмағы бойынша, г/м^3	Көлеміне % қатысы
Балалар мен аурулар болу үшін	1,3	0,07
Адамдардың ұзағырақ болуы үшін	1,86	0,1
Адамдардың кезеңдік болуы үшін	2,32	0,125
Адамдардың қысқа уақыт болуы үшін	3,72	0,2

CO_2 жұтатын ауада артық не кем болуы ағзаға біркелкі зиянды. CO_2, жетімді шоғырлануы 0,03% кем болса, онда аталған органдардың жұмысы бұзылады, CO_2 > 1,5%-тен артық болса, онда наркотикалық әсері болады, бас ауырады және т.б. Егер дем алатын ауада CO_2= 0,5–1,5% мәнінде болса, онда ағзаға оның елеулі әсері болмайды. Ал ең қолайлы шоғырлану шамасы CO_2= 0,04–0,5% сәйкес келеді. Қолданыстағы санитарлық нормалар бойынша бір адамға 20–60 $м^3$/сағ таза ауа қажет.

Адамның электр тогынан зақымдану ықтималдығына әсер ететін біздің бөлмеміздің классын анықтайық:

– едендер бір қабатты поливинилхлоирдті антистатикалық линолеуммен қапталған, сондықтан ол ток өткізбейтін болып табылады;

– ауаның салыстырмалы ылғалдылығы 60%–дан аспайды, сондықтан бөлме құрғақ;

– ауа температурасы Цельсий бойынша плюс 30 градустан аспайды;

– адамның бір уақытта бір жақтан жермен байланысы бар технологиялық жабдықтардың корпустарымен және басқа жерлендірілген бөліктермен, екінші жақтан электр жабдықтарының металл корпустарымен немесе ток өткізуші бөліктермен жанасу мүмкіндіктерінің болмауы (кернеу 1000В мәнінен аспағандықтан сымдардың өте жақсы изоляциясында);

– химиялық белсенді заттар жоқ.

МЕСТ 12.1.013–78.ССБТ сәйкес осы бөлмені маңызды қаупі жоқ бөлме ретінде классификациялауға болады.

Біздің жағдайымызда электр қауіпсіздігін қамтамасыз ету үшін МЕСТ 12.1.030-81 бойынша жерлендіру мүмкіндігін қарастыру қажет. Біздің жағдайымыздағы кернеу – 220В, сондықтан жерлендіру мен нөлдеу міндеттелмейді, бірақ ұсынылады.

Құрылыс конструкцияларын дайындау үшін кірпіш, темір бетон, әйнек, металл және басқа жанбайтын материалдар қолданылады. Сонымен қатар жанбайтын материалдардан жасалған қоршаулар түріндегі өртке қарсы өткелдерді ескеру қажет, олар біздің офистің бөлмелері арасында орнатылады. Ғимараттарда өрт крандары дәлістерде, баспалдақ торларында және кіре беріс аумақтарында орнатылады. Дербес электрондық есептеуіш машинаны қолданушылар бөлмелерінде, архивте және қосымша, қызметтік бөлмелердегі өртті өшіру үшін су қолданылады. Дербес электрондық есептеуіш машина бар бөлмелерде, ақпаратты тасушыларды сақтау бөлмелерінде, қымбат құрылғыларды бұзу немесе толықтай істен шығару қаупінен бақылау–өлшеуіш жабдықтары бар бөлмелерде суды қолдану тек кейбір жағдайларда ғана рұқсат етіледі, мысалы өрт қауіпті ірі көлемде болғанда. Бірақ судың мөлшері минималды болуы және дербес электрондық есептеуіш машинаны, дыбыстық құрылғыларды брезентпен немесе матамен жауып судан қорғау керек.

Барлық бөлмелерді стационарлы автоматты өрт өшіргіш қондырғылармен жабдықтау қажет. Ауа құрамындағы оттегіні тез азайтатын от өшіргіш газбен

бөлмені бірден толтыруға негізделген өртті газбен өшіру қондырғыларын қолданған тиімдірек болып табылады.

Зиян химиялық заттардың деңгейін нормалау. Бөлмені ластау көздері сыртқы ортаның және ғимараттың құрылыс материалдарынан, жиһаздардан, киімнен, аяқ-киімнен бөлінетін жүздеген әрекеттесулердің зиян заттары және адамның биоактивті әрекеттесулері (антропотоксиндер) болып табылады.

Бөлменің сыртқы ортаның зиян заттарымен ластануын қарастыра отырып, ең алдымен ғимараттың орналасқан орнын ескеру қажет, біздің жағдайымызда ол автострадаға жақын орналасқан. Бөлмеге сыртқы ортадан келетін жиі ластағыштар көміртек оксиді, азот диоксиді, күкірт диоксиді, қорғасын, шаң және тағы басқалары болып табылады.

Жиһаз, киім және аяқ-киімдер минералды талшықты, көмір сутегісі, полиэфир қара майы және тағыбасқа зиянды заттары бар шаңды бөледі. Биоктивті әрекеттесулердің ең маңыздысы көміртек диоксиді, күкірт сутегісі және тағы басқалары болып табылады.

Дербес электрондық есептеуіш машина қолданушысының, оператордың, жұмыс орнындағы шу көздері – сөйлесіп тұрған адамдар, сыртқы ортаның – компьютердің, принтердің, желдеткіш қондырғының шуы болып табылады. Олар болмашы мәнде шуды тудырады, сондықтан бөлмеде дыбысты жұтқыштарды қолдану жеткілікті.

Ең жақсы дыбысты жұту қасиеті талшықты – ауа көлемді материалдарда: фибролитті плиталарда, жарықталшықтарында, минералды мақтада, полиуретанды поропластта, ауа көлемді поливинилхлоридте және басқаларында болады. Дыбыс жұтқыш материалдарға дыбыс жұту коэффициенті 0,2 мәнінен төмен емес материалдар жатады.

Дербес электрондық есептеуіш машинамен жұмыс істеуге арналған бөлменің тиімді жарықтандырылуы табиғи және жасанды жарықтың болуы есебінен жасалады.

Жұмыс орнындағы жарықтанудың жоғары болуы жұмыс жағдайына дұрыс әсерін тигізеді.

Авариялық жарықтану, жалпы жарықтанудан 5%–ке аз жарықтануды қамтамасыз етуі керек, бірақ ғимарат ішінде оның деңгейі 2 лк–дан аз болмауы керек.

Эвакуациялық жарықтану, адамнның қауіпті деген өтетін жерлеріне орналасады, ол өтетін жолдардың еденінде, баспалдақтарда орналасады, оның деңгейі ғимарат ішінде 0,1 лк, ал ашық территорияда 0,2 лк болу керек.

Жұмыс орынның жасанды жарықталуы.

Жұмыс бөлмесіндегі жасанды жарықталуының шарттары адамның көру жұмыс қабілеттігіне, физикалық және моральдық күйіне үлкен әсер етеді, ал бұл еңбек өнімділігіне және өнімнің сапасына әсер етуіне әкеледі.

Еңбектің жағымды шарттарын құру үшін өндірістік жарықталуы келесі талаптарға сай болу керек:

– жұмыс орындағы жарықталуы тазалық нормаға сәйкес болу керек;
– жұмыс бетіндегі және қоршаған кеңістіктің шектегі жарық

мүмкіндігінше біркелкі таратылу керек;

– жұмыс бетіндегі кенет көлеңкенің болуы жарықтың біркелкі емес таратылуына әкеледі, сондықтан оларды мүмкіндігінше жою керек;

– көру алаңында жарқылдық (тура немесе шағылысқан) болмау керек.

Жарықтың көзін таңдау

Жасанды жарық үшін қызу және газоразрядтық шамдар қолданылады. Жалпы жұмыстық қыздыру шамдардың типтерін белгілеу шартындағы келесі әріптер мынаны білдіреді: В–ваакумдық, Г–газбен толтырылған, Б–биспиральдық, БК–биспиральдық криптондық. Шамдардың жарықтық ағымдары, жалпы жұмыстық қыздырушылық және олардың қуаты 4.4 – кестеде келтірілген.

Кесте 4.4 – Жалпы жұмыстық қыздырушылық шамның қуаты мен жарықтық ағым

Шамның типі	Қуат Вт	220	200–235	Шамның типі	Қуат Вт	200	200–235
		кернеуіндегі жарықтық ағым				кернеуіндегі жарықтық ағым	
В	15	105	85	Б	150	2100	1840
В	25	220	190	Г	200	2800	-
Б	40	400	300	Б	200	2920	2540
БК	40	460	–	Г	300	4600	4000
Б	60	716	550	Г	500	8300	7200
БК	60	790	–	Г	750	13100	–
Б	100	1350	1090	Г	1000	18600	–
БК	100	1450	–	Г	1500	2900	–
Г	150	2000	–				

Кіші қысымды газоразрядтық шамдар ең үнемді болып саналады (ЛД, ЛДЦ, ЛХБ және т.б.). Газоразрядтық шамдардың кейбір техникалық сипаттамалары 4.5 – кестеде келтірілген.

Номиналды жарықтық ағым лм шамдар типі

Кесте 4.5 – Кіші қысымды газ разрядтық шамдардың кейбір сипаттамалары

Номиналдық қуат, Вт	Номиналды жарықтық ағым,лм. Шамдар типі					Шамның өлшемі, мм.	
	ЛДЦ	ЛД	ЛХБ	ЛТБ	ДБ	Диаметр	Шатырдың ұзындығы
15	500	590	675	700	760	27	451,6
20	820	920	935	975	1060	40	604,0
30	1450	1640	1720	1720	2100	27	908,8
40	2100	2340	3000	3000	3120	40	1213,6

Кесте 4.5 –соңы

Номиналдық қуат, Вт	Номиналды жарықтық ағым,лм. Шамдар типі					Шамның өлшемі, мм.	
	ЛДЦ	ЛД	ЛХБ	ЛДЦ	ДБ	Диаметр	Шатырдың ұзындығы
65	3050	3570	3820	3980	4650	40	1514,2
80	3740	4070	4440	4440	5220	40	1514,2

4.2 Жасанды жарықтандыруды есептеу

Жарықтандыру табиғи және жасанды болуы мүмкін. Жасанды жарықтандыру екі жүйелі болуы мүмкін: жалпы және аралас. Аралас жарықтандыру кезінде жалпыға жергілікті жарықтандыру қосылады.

Жасанды жарықтандыруды есептеу үшін пайдалану коэффициенті және нүктелік әдістер қолданылады. Пайдалану коэффициенті әдісі үлкен көлеңке түсіретін заттар болмаған жағдайдағы біркелкі горизонталь беттердегі жарықтандыруды есептеуге арналған. Бұл әдіс арқылы η коэффициентінің мәні анықталады. Ол есептелінетін бетке түсетін жарық ағынының жарықтандыру құралының толық ағынына қатынасы болып табылады.

Жарықтануды есептеу ауданы 60 м^2, ұзындығы – 10 м, ені – 6 м, биіктігі – 3 м бөлме үшін жасалады, көру жұмысының разряды жоғары дәлдікті I а. Пайдалану коэффициенті әдісі арқылы есептеуді бастаймыз.

Жасанды жарықтану үшін қыздыру шамдарын қолданамыз. $Р_{төбе}$, $Р_{қабырға}$ және $Р_{еден}$ коэффициенттерінің мәндерін шағылысу коэффициенттерінің жазықтық сипаттамасына тәуелділік кестесінен анықтаймыз:

$$\rho_{\text{төбе}} = 70\%;$$
$$\rho_{\text{қабырға}} = 50\%;$$
$$\rho_{\text{еден}} = 30\%;$$

Қызметкерлердің жұмыс орны үшін жұмыс бетінің деңгейі еденнен 0,8 м–ді құрайды. Онда шырақтың жұмыс бетінен іліну биіктігі мынаған тең

$$h = H - h_c - h_0 \text{, м} \qquad (4.1)$$

мұндағы Н – бөлме биіктігі;

h_c – шырақтың іліну биіктігі;

h_0 – жұмыс орнының биіктігі.

$$h = 3 - 0{,}2 - 0{,}8 = 2 \text{ м}$$

Шырақтардың арасындағы ең тиімді қашықтық мына формуламен анықталады

$$Z_1 = \lambda \cdot h, \text{ м} \quad (4.2)$$

мұндағы λ – шырақтардың ең тиімді орналасу коэффициенті (1,1–1,2).

$$z = 1.1 \cdot 2 = 2.2 \text{ м}.$$

Шырақтардың қатарлар санын санайық

$$n = \frac{B}{L}, \quad (4.3)$$

мұндағы В – бөлме ені, В = 6 м;

L – шырақтар арасындағы қашықтық, L = 2,2 м.

Осыдан

$$n = \frac{6}{2.2} = 2.3 \approx 2$$

Сәйкесінше шырақтар екі қатарда орналасады.

Бөлме индексі мына формуламен есептеледі

$$i = \frac{A \cdot B}{h \cdot (A + B)}, \quad (4.4)$$

мұндағы А – жарықталатын беттің ұзындығы, А = 10 м;

В – жарықталатын беттің ені, В = 6 м;

h – шырақтардың іліну биіктігі, h = 3 м.

$$i = \frac{10 \cdot 6}{3(10+6)} = 1.25$$

Қажетті шырақтар санын келесі өрнекпен есептейміз

$$N = \frac{E \cdot K_З \cdot S \cdot Z}{F \cdot \eta}, \quad (4.5)$$

мұндағы $K_З$ – қор коэффициенті, эксплуатация процесінде жарық көздерінің шығуы және шаңдануды есептейді, $К_З$ =1,5;

S – бөлме ауданы, S = 60 м^2;

F – есептелетін жарық ағыны, F=4600 Лм;

Z – жарықтанудың біркелкілік коэффициенті, Z = 1,1–1,2;

η – пайдалану коэффициенті, η = 55% (кесте 4.7);

Е – берілген минимальды жарықтандыру, 300 лм.

К е с т е 4.7 – Жарықтық ағынның пайдалану коэффициентінің мәні

Шырақ түрі	Шағылысу коэффициенті, %	Пайдалану коэффициенті бөлме индексіне байланысты, %								
		0,5	1	1,25	1,5	2	2,5	3	4	5
I топ	50	30	10	21	40	45	49	55	58	61

$$N=\frac{300\cdot1,5\cdot60\cdot1,2}{2\cdot4600\cdot0,55}\approx6 \text{ шам}$$

Шыққан есептеулерге байланысты 300 Вт қуатты, жалпы қолдануға арналған, Г қыздыру шамын таңдаймыз. Шамның негізгі параметрлері 4.8–кестеге енгізілген.

К е с т е 4 . 8 – Қыздыру шамының негізгі параметрлері.

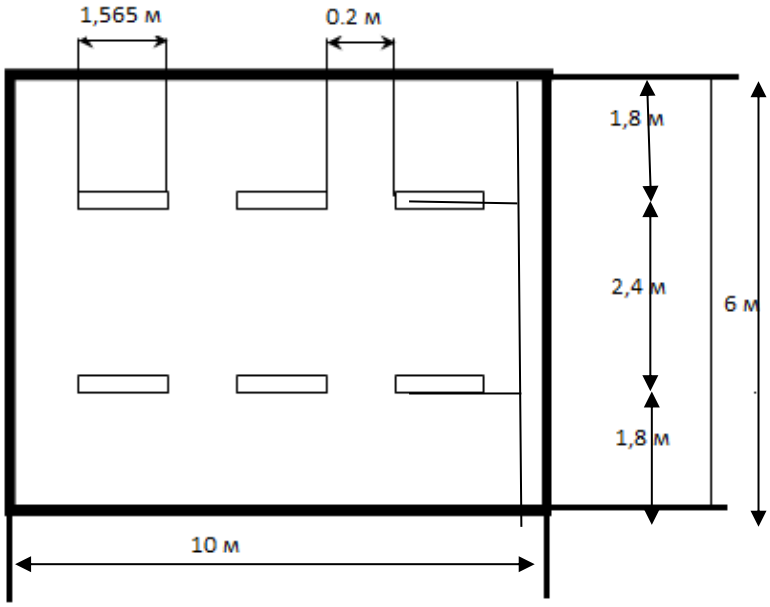
Г қыздыру шамының негізгі сипаттамалары.							
		Өлшемдері, мм		Жану ұзақтығы, сағ.		Жарық ағыны, лм	
Қуаты, Вт	Кернеуі, В	Ұзындығы	Диаметрі	Минимальды	Орташа	100 сағат жанудан кейінгі орташа мәні	Жану ұзақтығының минимальды мәні
300	220	1565	40	5200	13000	4400	3165

Барлығы 300 лк нормативтік жарықты құру үшін 300 Вт қуатты Г типті 6 қыздыру шамы қажет. Бір қыздыру шамы бар УСП–35 типті шырақтың ұзындығы $L_{СВ}$ = 1565 мм , ал олардың толық ұзындығы мынаған тең

$$L = N_0 \cdot L_{св}, \text{ м} \qquad (4.6)$$

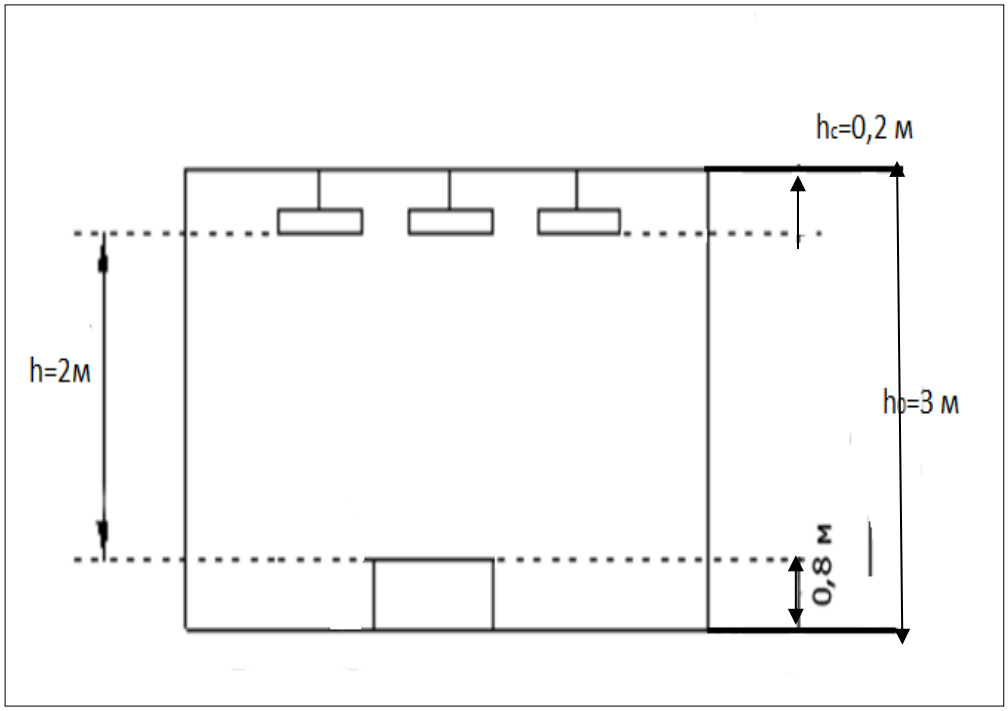
$$L= 2\cdot1,565 =3,13 \text{ м}$$

Яғни, 10 м ұзындықты қабылдау бөлмесіне екі қатарға үш шырақтан орналастырамыз, сурет 4.2 – де келтірілген.



Сурет 4.2 – Жасанды жарықтандыру сұлбасы

Шамдардың іліну сұлбасы сурет 4.3 көрсетілген.



Сурет 4.3 – Шамдардың іліну сұлбасы

4.3 Меншікті қуат әдісі

Меншікті қуат әдісімен жарықтандыруды шешудің маңызы, шырақтардың түрі мен оны орнатудың орнына байланысты, жұмыс бетінен ілгіштің биіктігі, горизонтальды беттегі жарықтандыру мен бөлменің ауданымен меншікті қуат мәні анықталаумен қорытындыланады.

Қыздыру шамдарын орнатуды қабылдаймыз және параллель екі қатарға үшеуден орналыстырамыз.

$$N=2\cdot3=6 \text{ шам.}$$

Шырақтың меншікті қуаты $\omega = 60$ Вт/м^2.

Бір шырақтың қуаты

$$\rho = \frac{\omega \cdot S}{N}. \tag{4.8}$$

$$\rho = \frac{60 \cdot 60}{6} = 600 \text{ Вт}$$

Әр шырақта екі шамнан. Әр шамның қуаты:

$$\rho_g = \frac{600}{2} = 300 \text{ Вт}$$

Сәйкесінше, екі әдіспен келтірілген есептер негізінде үш шырақты таңдап аламыз, олардың әр қайсысы 300 Вт екі шамнан тұрады.

5 Техника-экономикалық негізделуі

5.1 Жобаның мақсаты мен міндеттері

Бұл дипломдық жобаның мақсаты ипотека жүйесін автоматтандыру болып табылады. Бұл бағдарлама ыңғайлы, әрі уақытты үнемді пайдалануға мүмкіндік береді, сонымен қатар, жұмыс барысында мүмкін болатын қателерді азайтады.

Бағдарламаның тиімділігіне мыналарды жатқызуға болады:

– ипотека алу кезіндегі құжаттарды рәсімдеуде оңтайлығы;

– аз уақыт ішінде бағдарламаны игеріп, оңай пайдалануға мүмкіндік береді;

– бағдарламаның тез іздеу мүмкіндігі, қолданысқа қарапайымдылығы.

5.2 Маркетингтік стратегия

Маркетингтік стратегияның мақсаты – бағдарламалық қамтамасыз ету арқылы пайда табу. Жоспарланып отырған табыс 5.1 – кестеде көрсетілген.

К е с т е 5.1 – Жоспарланып отырған табыс мың.теңге

Жылдар	2014	2015	2016	Барлығы
Жоспарланған табыс, барлығы,мың.тг	486242,16	1458726,48	972484,32	2917452,96

Бағдарламаның сатылым жоспарын

Бұл бағдарлама ипотека жүйесін автоматтандыруға негізделген. Бағдарламаны алғашқы жылы сатылымға шығарғанда 1 жұмыс орнында ғана сынама жүргіземіз. Кейінрек, бағдарламаны тағы 3 жұмыс орнына сатамыз, 3–ші жылы Қазақстан рыногындағы басекелестікті ескеріп сатылымды 2 жұмыс орнына сатамыз деп жоспарлаймыз. Жалпы бағдарламадан түсетін пайда 2917452,96 теңге.

5.3 Бағдарламамен қамтамасыз етудегі еңбек сыйымдылығын есептеу

Еңбек шығыны құрамдасын есептеудегі базалық көрсеткіш мына формуламен есептелінеді:

$$Q = q \times c , \qquad (5.1)$$

мұндағы Q – шартты командалар саны;

q – есеп түріне қарай шартты командалар санын ескеретін коэффициент;

c – бағдарламаның қиындығы мен жаңалығын ескеретін коэффициент.

$$Q=q*c=1480*1.30=1924$$

Атап өткен *q* коэффициентінің мәнін 1– кестеден (Б– қосымшасынан) таңдап алуға болады.

Атап өткен *«c»* коэффициенті 2–кестеден (Б–қосымшасынан) анықталады, ол күрделік тобы бағанасы мен жаңалықтық дәрежесі бағанасының қиылысуы.

Бағдарламаның күрделілік коэффициенті c=1,30. Осы әдіспен базалық көрсеткішті табамыз: Q = 1924

Ары қарай бағдарламалық өнімді әзірлеуге кететін уақытты есептеу керек.

Жалпы бағдарлама жасауға кеткен уақыт әртүрлі компоненттен тұрады. Бағдарламалық өнімді жасауға кеткен жалпы уақыт құрамы 3– кестеде (Б– қосымшасынан) көрсетілген.

Уақыт адам-сағатпен есептеледі, ал $T_Д$ нақты істелген уақытпен алынады, ал қалған кезеңнің уақыты Q командасының шартты санына байланысты есептік жолмен анықталады,

Бағдарламалық өнімін дайындауға кеткен әр кезеңнің уақытын анықтаймыз:

1) $T_{ПО}$ (мақсат сипатын дайындау уақыты), нақтылы деректер бойынша алынады және келесі мәнге тең деп алынады (3– тен 5 күнге дейін, 8 сағаттан)

$$T_{ПО} = 24 \text{ адам / сағ.}$$

2) T_O (мақсат сипаттамасы уақыты) келесі формуламен анықталады

$$T_O = Q \times B / (50 \times K), \quad (5.2)$$

$$T_O = Q \times B / (50 \times K)=1924*1,3/(50*0,8)= 62,53 \text{ адам / сағ.}$$

мұндағы B – мақсат есебі өзгерісінің коэффициенті, B коэффициенті мақсат күрделігіне және өзгеріс санына тәуелді – 1,2-ден 1,5– ке дейін (2– кестені қара (Б– қосымшасы)).

К – бағдарлама жасаушы білектілігін ескеретін коэффициент. *К* коэффициенті мәнін 4–кестеден (Б– қосымшасынан) алуға болады.

3) T_A (алгоритм құруға кеткен уақыт) мына формуламен есептейміз

$$T_A = Q / (50 \times K), \quad (5.3)$$

$$T_A = Q / (50 \times K)=1924/(50*0.8)= 48,1 \text{ адам / сағ.}$$

4) $T_{БС}$ (блок – сұлба құруға кеткен уақыт) $T_А$ сияқты 5.3 – формуламен есептеленеді.

5) $T_Н$ (бағдарламаның тілінде жазуға кеткен уақыт) келесі формуламен анықталады

$$T_Н = Q \times 1,5 / (50 \times К), \quad (5.4)$$

$$T_Н = Q \times 1,5 / (50 \times К)=1924*1.5/(50*0.8)= 72,15 \text{ адам / сағ.}$$

6) $T_П$ (бағдарлама теру уақыты) келесі формуламен анықталады

$$T_П = Q / 50, \quad (5.5)$$

$$T_П = Q / 50=1924/50=38,48 \text{ адам / сағ.}$$

7) $T_{ОТ}$ (бағдарламаны реттеу және тестілеу уақыты) келесі формуламен анықталады

$$T_{ОТ} = Q \times 4,2/50 \times К, \quad (5.6)$$

$$T_{ОТ} = Q \times 4,2/50 \times К=1924*4.2/50*0.8=202,02 \text{ адам / сағ.}$$

8) $T_Д$ (құжаттарды рәсімдеу уақыты), нақтылы деректер бойынша алынады және құрылады (3–тен 5 күнге дейін, күніне 8 сағат)

$$T_Д = 24 \text{ адам / сағ.}$$

Еңбек шығындарының сомасы еңбек шығынының құрама сомасы ретінде 5.7 формуламен есептеленеді

$$T = T_{ПО}+ T_{ТО} + T_А + T_{БС} + T_Н + T_П + T_{ОТ} + T_Д, \quad (5.7)$$

$$T = T_{ПО}+ T_{ТО} + T_А + T_{БС} + T_Н + T_П + T_{ОТ} + T_{Д=}24+62,53+48,1 +48,1 +72,15 +38,48+202,02 +24=519,38 \text{ адам / сағ.}$$

5.4 Бағдарламалық қамсыздандыру шығынының есебі

Бағдарламалық қамсыздандыру шығыны ішіне еңбек ақы шығыны да, еңбек ақыдан аударылымдар, амортизациялық және тағы да басқа шығындар кіреді, олар мынандай формуламен анықталады:

$$C = ФОТ + О_{СН} + А + С_{ЭЭ} + С_{МжК} + С_{ТО} + С_{ПР} + С_Н, \quad (5.8)$$

Еңбек ақы екі жасаушыдан құрылады: негізгі еңбек ақы және қосымша

еңбек ақы сомасы (немесе еңбек ақы қоры, *ЕАҚ*) негізгі еңбек ақы және қосымша еңбек ақы сомасы мына формуламен есептеленеді

$$ФОТ = З_{осн} + З_{доп}, \quad (5.9)$$

Негізгі еңбек ақы төмендегідей анықталады

$$З_{осн} = Т \times ТС/t_{орт}, \quad (5.10)$$

$$З_{осн} = Т \times ТС/t_{орт} = 519{,}38 * 3000 * 2{,}25/21 = 166943.57 \text{ тг.}$$

Қосымша еңбек ақы негізгі еңбек ақының 20 % құрайды және келесі формуламен есептелінеді

$$З_{доп} = 0{,}2 \times З_{осн}, \quad (5.11)$$

$$З_{доп} = 0{,}2 \times З_{осн} = 0{,}2 * 1669943.57 = 33388{,}7 \text{ тг.}$$

Жалпы еңбек ақысы (еңбек төлеу фонды) негізгі және қосымша еңбек ақының қосындысы ретінде анықталады (9)

$$ФОТ = З_{осн} + З_{доп},$$

$$ФОТ = З_{осн} + З_{доп} = 1669943.57 + 33388{,}7 = 200332{,}3 \text{ тг.}$$

Әлеуметтік салық ЕАҚ 11 % құрайды (ҚР СК 358 б. 1– арау) жұмыскердің табысынан, мынандай формуламен есептеледі [16] (Б– қосымшасы)

$$О_{СН} = (ФОТ - ЗА) \times 11\%, \quad (5.12)$$

$$О_{СН} = (ФОТ - ЗА) \times 11\% = (200332{,}3 - 20033{,}23) * 0{,}11 = 19832{,}9 \text{ тг.}$$

мұндағы ЗА – зейнетақы аударылымдар, ЕАҚ– нан 10% құрайды және әлеуметтік салықпен міндеттелмейді [17]

$$ЗА = ЕАҚ * 10\%, \quad (5.13)$$

$$ЗА = ЕАҚ * 10\% = 200332{,}3 * 0{,}1 = 20033{,}23 \text{ тг.}$$

Амортизациялық аударылымдар амортизацияның тағайынды шамаларымен орындалады, пайыздармен жабдықтың баланстық құнына және мына формуламен есептеледі

$$А=\frac{Б_{бас}\times А_{ш}\times N}{100\times 12\times t}=\frac{149190*0,2375*21,64}{100*12*519,38}=1,2, \quad (5.14)$$

Амортизация шамалары ($А_Ш$), мына формуламен есептеледі

$$Н_А=\frac{Б_{бас}-К_{тар}}{Т_{норм}\cdot Б_{бас}}\times 100\%, \quad (5.15)$$

$$Н_А=\frac{Б_{бас}-К_{тар}}{Т_{норм}\cdot Б_{бас}}\times 100\%=\frac{149190-7459,5}{4*149190}*100\%=23,75\%$$

мұндағы $К_{тар}$ – таратылым құны, жабдықтың құнынан 5% құрайды (нұсқа бойынша); $К_{тар}$ =149190*0,05=7459,5 тг.

Құны $Б_{бас}$ = 149190 тг.

Компьютердің қазіргі уақыттағы құны 133200 тг [18].

Принтердің құны 15990 тг [19].

$Т_{норм}$ – жабдықтың нормативтік қызмет ету мерзімі (дербес компьютер үшін – 4жыл*)*.

Дербес компьютерде жалпы жұмыс істеу уақыты мына формуламен есептеленеді

$$Т = Т_А + Т_{БС} + Т_Н + Т_П + Т_{ОТ}, \quad (5.16)$$

$$Т = Т_А + Т_{БС} + Т_Н + Т_П + Т_{ОТ}=24+62,53+48,1+48,1+72,15+38,48+202,02+24$$
$$=519,38 \text{ адам / сағ.}$$

Амортизациялық аударылымдар(14)

$$А=\frac{Б_{бас}\times А_{ш}\times N}{100\times 12\times t}=\frac{88080*0.2375*21,87}{100*12*411,575}=0,9263$$

Электрэнергия шығындары мына формуламен есептеледі

$$С_{ЭЭ}= Қ\times k_з \times Т \times С_{кВт\text{-}сағ}, \quad (5.17)$$

$$С_{ЭЭ}= Қ\times k_з \times Т \times С_{кВт\text{-}сағ} =450*0,8*0,01937*8=55,78 \text{ тг.}$$

мұндағы Қ – ЭЕМ қуаты (450 Вт);

$k_з$ – жүтеме коэффициенті (0.8);

$С_{кВт.с}$ – 1 кВт–сағ электрэнергиясының құны19,37 тг [20];

Т – жұмыс уақыты, сағ

$$А_{жыл} = Б_{бас} \cdot \frac{Н_А}{100\%} = 149190 * \frac{23.75\%}{100\%} = 35432,625тг$$

Материалдар мен көмекші бөлшектер шығыны, бағдарламалық өнімді жазу барысында қолданылды ($С_{МжК}$), сонымен қатар техникалық қызмет көрсету шығыны ($С_{ТО}$), жабдықтың құнынан 1.5% және 2.5% құрайды және мына формулалар мен есептеледі (18 – 19)

$$С_{МжК} = 0,015 \times С_{обор,} \quad (5.18)$$

$$С_{МжК} = 0,015 \times С_{обор} = 0,015*149190 = 2237,85 \text{ тг},$$

$$С_{ТО} = 0,025 \times С_{обор}, \quad (5.19)$$

$$С_{ТО} = 0,025 \times С_{обор} = 0,025*149190 = 3729,75 \text{ тг}.$$

Басқару мен қызмет көрсетуге байланысты үстеме шығындар, сондай-ақ жабдықты пайдалану кезіндегі және де кәсіпорын үдерістері мен айналымдарынан қосымша шығындар еңбек ақы қорынан 50% құрайды және де мына формуламен есептеледі

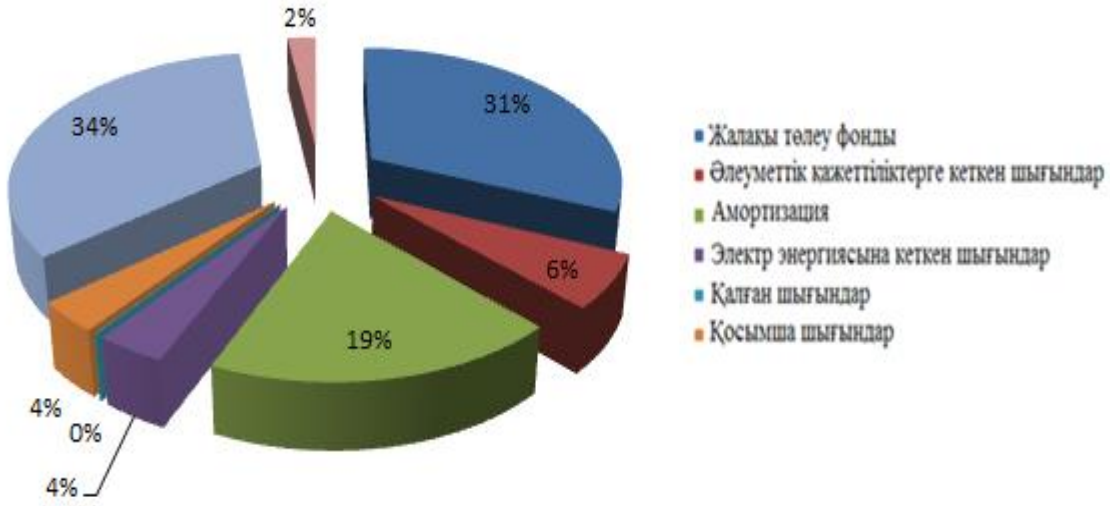
$$С_Н = 0,5 \times ЕАҚ, \quad (5.20)$$

$$С_Н = 0,5 \times ЕАҚ = 0,5* 200332,3 = 100166,15 \text{ тг}.$$

Бағдарламалық өнімнің өзіндік құнының есебінің жиынтық нәтижелерін кесте түрінде ұсыну керек (5.2 – кесте), шығын статьясын атап, және оның ортақ құндағы сыбағаларын пайызбен есептеп, бағдарламалық қамсыздандырудың өзіндік құнының құрылымының диаграммасын сызу керек. (сурет 5.1)

К е с т е 5.2 – Өзіндік құнның қорытынды кестесі

Шығын бабы атауы		Сомасы, теңге	Әр баптың үлесі, %
ЕАҚ	$ЕА_{нег}$	166943.57	46,14
	$ЕА_{кос}$	33388,7	9,23
Үстеме шығындар, $С_{Накл}$		100166,15	27,7
Әлеуметтік салық шығыны, ӘС		19832,9	5,5
Пайдалану шығындары	$С_{ээ}$	55,78	0,015
	$С_{ТО}$	3729,75	1,03
	$А_{жыл}$	35432,625	9,8
Материалдар және көмекші, $С_{МжК}$		2237,85	0,62
Барлығы:		361787,325	100



Сурет 5.1 – Өзіндік құн нәтижесінің диаграммасы

Бағдарламалық өнімді жүзеге асыру бағасы оның құны мен таза кірістің қосындысынан тұрады

$$Ц=С+П \qquad (5.21)$$

мұндағы, С – өнім бағасы;

П – таза кіріс.

Бастапқы бағаны анықтауда бағдарламалық өнімді жүзеге асыру үшін керекті рентабелдік деңгейін анықтау қажет (20%):

$$Ц_п=С\cdot(1+\frac{Р}{100}) \qquad (5.22)$$

мұндағы, Р – рентабельдік (20%).

$$Ц_п = 361787{,}325* (1 + 20/100)= 434144{,}79 \text{ теңге.}$$

Бағдарламаның орындалу бағасы келесі формула арқылы табылады.

$$Ц_р=Ц_п+ҚҚС. \qquad (5.23)$$

Қазіргі таңда ҚР-да НДС 12% құрайды

$$ҚҚС=Ц_п\cdot 12\% ; \qquad (5.24)$$

$$ҚҚС = 434144{,}79*12\%= 52097{,}37 \text{ теңге,}$$

$$Ц_р= 434144{,}79+52097{,}37 =486242{,}16 \text{ теңге.}$$

5.5 Бағдарлама өнімін сатып алуға кеткен бір жолғы шығындар есебі

Бағдарлама өнімін сатып алуға және оны өндіріске енгізу шығындары келесі шығындардан тұрады

$$\Sigma З = С_С + С_{ТР} + С_{О,} \quad (5.25)$$

$$\Sigma З = С_С + С_{ТР} + С_О = 486242,16+121560,54+89600 = 697402,7 \text{ тг.}$$

Бағдарламаның бағасы: $С_С$ =486242,16 тг

$С_{ТР}$ – көлік шығыны, жүйе құнынан - 25 %, мың теңге;

$$С_{ТР} = 0,25*С_С = 486242,16*0,25 = 121560,54 \text{ тг.}$$

Жүйе құны үстінде есептелінді, ал қалғандары келесі түрде есептеленеді. Өнімді игеруге деген маманды оқыту шығыны, оқытуға кеткен уақыт пен оған деген консалтингті фирмадағы мөлшерлемеден тұрады

$$С_О = Т \times С_{ОП}, \quad (5.26)$$

$$С_О = Т \times С_{ОП} = 4*8*2800 = 89600 \text{ тг.}$$

мұндағы $Т$ – оқытуға кеткен уақыт, сағ.;

$С_{ОП}$ =2600 тг. Себебі қазіргі танда сағатына 2500–3000 тг аралығында.

Бағдарлама өнімін сатып алуға кеткен бір жолғы шығындар есебін 5.3-кестеге келтіру керек.

Кесте 5.3 – Ақпарат жүйелерін енгізуге керекті бір жолымғы шығындар есебінің жиынтығы

Шығын бабы атауы	Сомасы, мың теңге
Жүйенің құны	*486242,16*
Көлік шығыны	*121560,54*
Жүйені оқуға кеткен шығыны	*89600*
Барлығы:	*697402,7*

5.6 Игеру саласындағы жылдық бір жолғы шығындар есебі

Ақпараттық технологияларын қолданған кездегі жылдық шығындары келесі баптардан тұрады:

- жылдық ЕАҚ;
- әлеуметтік салық аударым;

– басқа да шығындар.

Осының барлығысын енді формула арқылы жазайық

$$Сэ = ЗП + О_{СН} + П_Р, \quad (5.27)$$

$$Сэ = ЗП + О_{СН} + П_Р = 35280000+3492720+10584000=49356720 \text{ тг.}$$

АТ енгізгеннен кейінгі мамандардың жылдық еңбек ақы шығындары келесі формуламен есептеледі

$$ЗП = (О_С \times t \times К_Р) \times Ч \times (1 + К_Д), \quad (5.28)$$

$$ЗП = (О_С \times t \times К_Р) \times Ч \times (1 + К_Д) = (3000 * 8 * 245) * 5 * (1 + 0{,}2) = 35280000 \ тг,$$

$$О_{СН} = (ФОТ - ЗА) \times 11\% = (35280000\text{-}3528000) * 0{,}11 = 3492720 \text{ тг},$$

$$ЗА = ЕАҚ \times 10\% .= 35280000 * 0{,}1 = 3528000 \text{ тг.}$$

мұндағы $О_С$ – маманның сағаттық ақысы, 3000 мың теңге/сағ.;

t – жұмыс күнінің ұзақтығы, 8 сағ.;

$К_Р$ – жылдағы жұмыс күндер саны, 245 күн;

Ч – үдеріске қатысы бар мамандар саны, 5 адам;

$К_Д$ – қосымша еңбек ақы коэффициенті, 20 %.

Басқа да шығындар – материалға деген шығындар, үстеме шығындар жылдық еңбек ақы шығындарынан 30 % құрайды және келесі формуламен есептеледі

$$Пр = ЗП \times 0{,}3, \quad (29)$$

$$Пр = ЗП \times 0{,}3 = 35280000 * 0.3 = 10584000 \text{ тг.}$$

Ақпараттық технологияларын қолданған кездегі жылдық бір жолғы шығындарды келесі кестеге сомасын және әр баптарын көрсетіп толтырыңыз (5.4 – кесте).

К е с т е 5.4 – Ақпараттық технологияларын қолданған кездегі жылдық бір жолғы шығындар

Шығын баптары атауы	Сомасы, мың теңге
Жылдық ЕАҚ	35280000
Әлеуметтік салық аударым	3492720
Басқа да шығындар	10584000
Барлығы:	49356720

5.7 Ақпараттық жүйе енгізуден үнем мен табыс мөлшерінің есебі

Еңбек өнімділігі өсуінен түскен үнемділікті келтірінді шығындардың базалық ($С_о$) және ұсынылған ($С_1$) нұсқалар айырмасы ретінде шығаруға болады.

$$Э_{уг} = С_о – С_1, \quad (5.30)$$

$$Э_{уг} = С_о – С_1 = 78970752 – 49356720 = 29614032 \text{ тг.}$$

Ұйымда жүйе енгізілмей тұрғанда (қол еңбегімен айналысқанда) жылдық келтірінді шығындар келесі баптардан тұрады:

- еңбек ақы қоры;
- әлеуметтік салық;
- басқада шығындар.

Осының барлығын енді формула арқылы жазайық

$$Сэ = ЗП + О_{СН} + П_Р, \quad (5.31)$$

$$Сэ = ЗП + О_{СН} + П_Р = 56448000 + 5588352 + 16934400 = 78970752 \text{ тг.}$$

АТ енгізгеннен кейінгі мамандардың жылдық еңбек ақы шығындары келесі формуламен есептеледі

$$ЗП = (О_С \times t \times К_Р) \times Ч \times (1 + К_Д), \quad (5.32)$$

$$ЗП = (О_С \times t \times К_Р) \times Ч \times (1 + К_Д) = (3000 * 8 * 245) * 8 * (1 + 0{,}2) = 56448000 \text{ тг,}$$

$$О_{СН} = (ФОТ - ЗА) \times 11\% = (56448000 - 5644800) * 0{,}11 = 5588352 \text{ тг,}$$

$$ЗА = ЕАҚ * 10\% . = 56448000 * 0{,}1 = 5644800 \text{ тг.}$$

Басқа да шығындар – материалға деген шығындар, үстеме шығындар жылдық еңбек ақы шығындарынан 30 % құрайды және келесі формуламен есептеледі

$$Пр = ЗП \times 0{,}3, \quad (5.33)$$

$$Пр = ЗП \times 0{,}3 = 56448000 * 0.3 = 16934400 \text{ тг.}$$

К е с т е 5.5–Ұйымда жүйе енгізілмей тұрғанда жылдық бір жолғы шығындар

Шығын баптары атауы	Сомасы, мың теңге
Жылдық ЕАҚ	56448000
Әлеуметтік салық аударым	5588352
Басқа да шығындар	16934400
Барлығы:	78970752

5.8 Салыстырмалы экономикалық тиімділіктің көрсеткіштерін есептеу

Нормативтік күрделі қаржы салымының экономикалық тиімділігінің коэффициенті келесі формуламен есептеледі

$$E_н = \frac{1}{T_н}, \tag{5.34}$$

$$E_н = \frac{1}{T_н} = 0{,}33.$$

мұндағы $T_н$ – нормативтік күрделі қаржы салымын өтелу мерзімі, жыл. ($T_н$ = 1, 2, 3…n) бағдарлама өнімдерінің өтеу мерзімі 4 жыл.

Есептік күрделі қаржы салымының экономикалық тиімділігінің коэффициенті

$$E_р = \frac{Э_{уг}}{К}, \tag{5.35}$$

$$E_р = \frac{Э_{уг}}{К} = \frac{29614032}{49356720} = 0{,}6$$

Есептік күрделі қаржы салымын өтелу мерзімі

$$T_р = \frac{1}{E_р}, \tag{5.36}$$

$$T_р = \frac{1}{E_р} = \frac{1}{0{,}6} = 1{,}7$$

Дисконтты есептемегенде өтелу мерзімі 1,7 жыл.

К е с т е 5.6 – Бағдарлама өнімін енгізудің салыстырмалы экономикалық тиімділігінің көрсеткіштері

Көрсеткіштер атауы	Мәні
Шартты жылдық шығынды үнемдеу, мың теңге	29614032
Күрделі қаржы салымының экономикалық тиімділігінің коэффициенті ($E_р$)	0.6
Күрделі қаржы салымын өтелу мерзімі ($T_р$), жыл	1,7

5.9 Динамикалық көрсеткіштер негізінде жобаны өткізуде экономикалық тиімділігін бағалау

Таза дисконттық табыс (ЧДД) келесі формуламен есептеледі

$$\text{ЧДД} = \sum_{t=0}^{T} (P_t - З_t) \times \frac{1}{(1+E)^t} - K, \qquad (5.37)$$

$$Э_t = (P_t - З_t) = Э_{уг} = 29614032 \text{ мың теңге.}$$

Есеп аралығы – есептеу кезеңіндегі уақыттың бір бөлігі (жыл, квартал, ай және т. б.)

$$T_{ОК} = t + \frac{K - (dП_1 + dП_2)}{dП_З} = 1{,}67 + \frac{49356720 - 45243660}{17137750} = 1{,}91 = 1\text{жыл}11\text{ай}$$

Егер инвестициялық жобадан таза дисконттық табыс (*ЧДД*) оң болса, жоба тиімді (берілген дисконттау мөлшерінде).

Е – тұрақты дисконт мөлшері, 20 % [6].

Онда барлық есеп шегін ескергендегі таза дисконттық табыс сомасы

$$\text{ЧДД} = Э_1 \frac{1}{1+E} + Э_2 \frac{1}{(1+E)^2} + Э_3 \frac{1}{(1+E)^3} - K,$$

$$\text{ЧДД} = Э_1 \frac{1}{1+E} + Э_2 \frac{1}{(1+E)^2} + Э_3 \frac{1}{(1+E)^3} - K = \frac{29614032}{1{,}15} + \frac{29614032}{1{,}15^2} + \frac{29614032}{1{,}15^3} - 49356720 =$$
$$= 18258781{,}68\text{тг}$$

Таза дисконттық табыстың оң мәні ЧДД > 0, инвестицияның тиімділігін көрсетеді, ақпараттық жүйелер керекті көлемде пайда алып келеді.

Табыстық индексі (ИД) келтірілген әсерлердің сомасының күрделі қаржы салымына қатынасы. Ол келесі формуламен есептеледі

$$\text{ИД} = \frac{1}{K} \sum_{t=1}^{T} (P_t - З_t) \cdot \frac{1}{(1+E)^t},$$

мұндағы *К* – күрделі қаржы салымы немесе инвестицияның құны.

$$\text{ИД} = \frac{67615501,8}{49356720} = 1,4 .$$

Егер ИД > 1, жоба тиімді, егер ИД < 1 – тиімді емес.

Ішкі табыстық мөлшері (ВНД немесе IRR) - келтірілген әсер, келтірілген күрделі қаржы салымдарға тең болғандағы дисконттау мөлшерін *($r_{вн}$)* көрсетеді.

Бұл көрсеткіш $r_{вн}$ (ВНД) келесі теңдеуден шығады

$$\sum_{t=0}^{T} \frac{P_t - З_t}{(1+r_{вн})^t} = \sum_{t=0}^{T} \frac{K_t}{(1+r_{вн})^t} . \qquad (5.38)$$

Егер $r_{вн}$ мәні инвесторға керек табыс мөлшері салынған капиталдан көп не Ішкі табыстық мөлшері (ВНД)

$E_1 = 0,15.$

$$\text{ЧДД}_1 = \frac{29614032}{1,15} + \frac{29614032}{1,15^2} + \frac{29614032}{1,15^3} - 49356720 = 18258781,68 \text{ тг};$$

$E_2 = 0,20.$

$$\text{ЧДД}_2 = \frac{29614032}{1,2} + \frac{29614032}{1,2^2} + \frac{29614032}{1,2^3} - 49356720 = 13024690 \text{ тг};$$

$$E_1 \rightarrow \text{ЧДД}_1 > 0; \qquad E_2 \rightarrow \text{ЧДД}_2 > 0;$$

$$\text{IRR} = 0,20 + \frac{18258781,68}{18258781,68 + 13024690} \cdot (0,2 - 0,15) = 0,18 .$$

Сонымен, дисконттау мөлшері келесі аралықта 15 %….18 % (IRR=18%). Егер мына шарт орындалса, формула дұрыс: $r_a < \text{IRR} < r_b$ и NPVa > 0 > NPVb

К е с т е 5.7– Бағдарлама өнімін әзірлеуінің және енгізуінің экономикалық пайдалылығының көрсеткіштері

Көрсеткіштер атауы	Мәні
Бағдарлама өнімін әзірлеуге және енгізуге шығын, мың теңге	49356720
Бағдарлама өнімін енгізгеннен кейінгі болжалды үнем, мың теңге	29614032
Таза дисконттық табыс, мың теңге	18258781,7
Табыстық индекс	1,4
Ішкі табыстық мөлшері	0,18
Дисконталған өтелу мерзімі, жыл	1жыл 11ай
Моральдық ескіру мерзімі, жыл	3

Ақшаның жылжуы (қозғалысы).

К е с т е 5.8 – Ақшаның жылжуы

Атауы	Жылдар			Барлығы, тг.
	2014	2015	2016	
Бірреттік шығын, $К_в$	697402,7			697402,7
Үнемдеуден шыққан пайда		29614032	29614032	59228064
Ағымдағы шығын, К		49356720	49356720	98713440
Пайдаға салық		5922806,4	5922806,4	11845612,8
Таза пайда	-697402,7	23691225,6	23691225,6	
Дисконттау еселігі, $К_д$ (15% мөлшерлememен)	1	0,869	0,756	
Таза дисконтталған пайда, PV	-697402,7	20587675,05	17910566,55	
Жалпы өсім бойынша ТДП, NPV	-697402,7	19890272,35	37800838,8	56993708,5

Қорытынды

Бұл дипломдық жобада барлық қойылған мақсаттар іске асырылды. Ипотека жүйесін автоматтандыруға арналып дерекқор құрылды. Дерекқорды құру барысында барлық ерекшеліктер ескерілді.

Дерекқор MS SQL Server дерекқорларды басқару жүйесінде жасалды, жаңа түскен ақпаратқа байланысты деректерді тек құпия сөз арқылы енгізе алады. Мұндай қатынау принципі пайдаланушылар үшін онша ыңғайлы емес, сондықтан олар үшін Microsoft Visual Studia 2012 Ultimate программалық ортасында клиенттік қосымша жасалды. Сонымен қатар, құпия сөз қолдану арқылы деректердің қауіпсіздік пен құпиялылық мәселелері қарастырылды.

Бұл бағдарламалық өнімді банктерде ипотека құжаттарын рәсімдеу кезінде пайдалануға болады.

Тіршілік қауіпсіздігі бөлімінде қабылдау бөлмесінің жасанды жарықтандыру жүйесін есептедім. Мұнда сыртқы және ішкі микроклимат параметрлерін, температура, ылғалдылық және ауаның қозғалыс жылдамдығы, табиғи жарықтандыру мен жарық көздерін анықтап, есептеулер жүргізу арқылы бөлмеге 6 шам орналастыру керек және Г қыздыру шамы таңдалып алынды.

Техника-экономикалық негізделуі нәтижесі бойынша бұл программаны құру ақшалай қаржыландыруды қажет етеді. Бағдарламаның өзіндік құны қосымша құн салығынсыз 486242,16 теңгені қамтиды.

Берілген бағадарламаны қолдану мынадай мүмкіндіктер береді:

– ипотека құжаттарын рәсімдеуді жеңілдетеді;

– керекті есептеулерді автоматты түрде жүргізе отырып, уақытты үнемдейді;

– есептеу барысында кететін қателерді азайтады.

Ипотека жүйесін автоматтандыруға арналған бағдарламаны құру қиын әрі ұзақ уақытты қажет етуіне қарамастан ипотека алу кезіндегі таптырмас бағдарлама болып табылады.

Пайдаланылған әдебиеттер тізімі

1 Агапов В.П. Основы программирования на языке C# . учеб.пособие.-Москва,2012.-128с.

2 Голицына О.Л., Попов И.И.Основы алгоритмизации и программирования. учеб.пособие.-Форум 2008.-432с

3 Никита Культин. Основы программирования в Delphi 2010. 2-е издание. –Санкт-Петербург, 2010. – 640с.

4 Полякова Л.Н. Основы SQL. Курс лекций: учеб. пособие. – Москва, 2004. – 368с.

5 Глушаков С.В., Ломотько Д.В. Базы данных:Учебный курс. – М.: ООО «Издательство АСТ», 2000. – 504с.

6 Ибраева Л.К., Сатимова Е.Г. Проектирование баз данных. МУ к выполнению лабораторных работ. – Алматы: АИЭС, 2004.

7 Айтхожаева Е.Ж. Стандартный язык баз данных SQL: Учебное пособие. – Алматы: АИЭС, 2005. - 48 с.

8 Дунаев В.В. Базы данных. Язык SQL. – СПб.: БХВ – Петербург,2006.

9 Кандзюба С.П., Громов В.Н. Delphi 6/7. Базы данных и приложения. – СПб: ООО «ДиаСофт», 2002.

10 Хакимжанов Т.Е. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. – Алматы: АИЭС, 2002. – 30 с.

11 Ломакин В.К. Мировая экономика: Учебник для вузов. – М.: Финансы, ЮНИТИ, 2008. – 727 с.

12 Борисов, Е. Ф. Основы экономики: Учебное пособие / Е. Ф. Борисов. – М.: Юрайт – Издат, 2009. – 316 с.

13 Современная экономика: Учебное пособие /Под ред. О. Ю. Мамедова. – Ростов-на-Дону: Феникс, 2011. – 456 с.

14 Экономика: Учебник / Под ред. Р. П. Колосовой. – М.: Норма, 2011. – 345 с.

15 Экономика: Учебное пособие /Под ред. А.С. Булатова. – М.: Юристъ, 2009. – 896 с.

16 http://www.tfa.kz/ru/v-pomoshch-bukhgalteru/86-v-pomoshch-bukhgalteru/115-stavki-nalogov-na-2014-god

17 http://uchet.kz/news/detail.php?EID=138493

18 http://alfa.kz/pc/notebooks/lenovo/ideapad_z500/1447902

19 http://www.sulpak.kz/Goods/12/kopyuternaya_i_ofisnaya_tehnika/printeriy/0/printer_samsung_ml_2160__xev

20 http://www.inform.kz/kaz/article/2619716

21 http://uchet.kz/news/detail.php?EID=138493

А қосымшасы

Бағдарлама листингі

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace Administrations{
public partial class Form1 : Form{
public Form1(){
InitializeComponent();}
private void button1_Click(object sender, EventArgs e){
if (textBox1.Text == "admin" && textBox2.Text == "admin"){
new AdminControl().ShowDialog();}
else{
MessageBox.Show("Сіз логинді немесе құпия сөзді қате енгіздіңіз. Қайталап
көріңіз", "Назар аударыңыз!",
MessageBoxButtons.OK, MessageBoxIcon.Error);}}
private void label11_Click(object sender, EventArgs e)
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Windows.Forms.VisualStyles;
namespace Administrations{
 public partial class AdminControl : Form{
 private static int rec, rec2, rec3;
 public AdminControl(){
InitializeComponent();
comboBox1.Items.Add("Барлық тізім");
comboBox1.Items.Add("Пәтер төл құжатын алғандар тізімі");
```

```
comboBox1.Items.Add("Пәтер алуға тіркелгендер тізімі");
groupBox1.Enabled = false;
a1 = "select k.kons_id, k.fam as 'Тегі', k.name as 'Аты', k.patronimic as 'Әкесінің
аты', k.login as 'Логин', k.password as 'Құпия сөз' from Konsultant k";
 connect();
 dataGridView1.Columns["kons_id"].Visible = false;
c1 = "select d.dom_id, n.type_name as 'Үй типі', d.address as 'Мекен-жайы',
d.nomer_d as 'Үй нөмері', d.kvartira as 'Пәтер нөмері', d.komnatnost  as 'Бөлмелер
саны', d.ploshad as 'Ауданы', g.gen_name  as 'Құрылыс копаниясы', g.brigadir_fam
as 'Бригада басшысының тегі', g.brigadir_name as 'Бригада басшысының аты',
g.brigadir_patronimic as 'Бригада басшысының әкесінің аты', g.brigadir_tel as
'Бригада басшысының байланыс нөмері', d.status from dom d inner join
Genpodryadchik g on g.gen_id = d.gen_id inner join Ntype n on n.type_id =
d.type_id";
connect3();
dataGridView3.Columns["dom_id"].Visible = false;
groupBox5.Enabled = false;
tabControl1.SelectTab(tabPage1);
var kom = (from i in Program.db.Genpodryadchiks select i.gen_name);
foreach (var t in kom){
comboBox2.Items.Add(t);}
d1 = "select g.gen_id, g.gen_name as 'Құрылысқа жауапты компания',
g.brigadir_fam as 'Бригада басшысының тегі', g.brigadir_name as 'Бригада
басшысының аты', g.brigadir_patronimic as 'Бригада басшысының әкесінің аты',
g.brigadir_tel as 'Бригада басшысының байланыс ннөмері' from Genpodryadchik
g";
connect4();
dataGridView4.Columns["gen_id"].Visible = false;
groupBox7.Enabled = false;
var dt = from i in Program.db.Ntypes select i.type_name;
foreach (var i in dt){
comboBox3.Items.Add(i);} }
private void dataGridView1_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e){
int index = e.RowIndex;
string indexStr = (index + 1).ToString();
object header = this.dataGridView1.Rows[index].HeaderCell.Value;
if (header == null || !header.Equals(indexStr))
this.dataGridView1.Rows[index].HeaderCell.Value = indexStr;}
private void dataGridView2_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e){
```

```
string indexStr = (index + 1).ToString();
object header = this.dataGridView2.Rows[index].HeaderCell.Value;
if (header == null || !header.Equals(indexStr))
this.dataGridView2.Rows[index].HeaderCell.Value = indexStr;}
private void dataGridView3_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e){
int index = e.RowIndex;
string indexStr = (index + 1).ToString();
object header = this.dataGridView3.Rows[index].HeaderCell.Value;
if (header == null || !header.Equals(indexStr))
this.dataGridView3.Rows[index].HeaderCell.Value = indexStr;}
private void dataGridView4_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e){
int index = e.RowIndex;
string indexStr = (index + 1).ToString();
object header = this.dataGridView4.Rows[index].HeaderCell.Value;
if (header == null || !header.Equals(indexStr))
this.dataGridView4.Rows[index].HeaderCell.Value = indexStr;}
private string a, a1, b, b1, c, c1, d, d1, cn;
public void connect(){
SqlConnection cn = Program.conn;
a = a1;
DataSet ds = new DataSet();
SqlDataAdapter da = new SqlDataAdapter(a, cn);
da.Fill(ds);
dataGridView1.DataSource = ds.Tables[0];
public void connect2(){
SqlConnection cn = Program.conn;
b = b1;
DataSet ds1 = new DataSet();
SqlDataAdapter da1 = new SqlDataAdapter(b, cn);
da1.Fill(ds1);
dataGridView2.DataSource = ds1.Tables[0];}
public void connect3(){
SqlConnection cn = Program.conn;
c = c1;
DataSet ds2 = new DataSet();
SqlDataAdapter da2 = new SqlDataAdapter(c, cn);
da2.Fill(ds2);
dataGridView3.DataSource = ds2.Tables[0];}
public void connect4(){
```

```
SqlConnection cn = Program.conn;
d = d1;
DataSet ds3 = new DataSet();
SqlDataAdapter da3 = new SqlDataAdapter(d, cn);
da3.Fill(ds3);
dataGridView4.DataSource = ds3.Tables[0];}
private void button4_Click(object sender, EventArgs e){
groupBox1.Enabled = true;
panel1.Enabled = false;
dataGridView1.Enabled = false;
rec = 0;}
private void button5_Click(object sender, EventArgs e){
groupBox1.Enabled = true;
panel1.Enabled = false;
dataGridView1.Enabled = false;
int kons_id = Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value);
string kons_fam = (from i in Program.db.Konsultants where i.kons_id == kons_id
select i.fam).SingleOrDefault();
string kons_name = (from i in Program.db.Konsultants where i.kons_id == kons_id
select i.name).SingleOrDefault();
string kons_patronimic = (from i in Program.db.Konsultants where i.kons_id ==
kons_id select i.patronimic).SingleOrDefault();
string kons_login = (from i in Program.db.Konsultants where i.kons_id == kons_id
select i.login).SingleOrDefault();
string kons_pass = (from i in Program.db.Konsultants where i.kons_id == kons_id
select i.password).SingleOrDefault();
textBox1.Text = kons_fam;
textBox2.Text = kons_name;
textBox3.Text = kons_patronimic;
textBox4.Text = kons_login;
textBox5.Text = kons_pass;
rec = 1;}
private void button7_Click(object sender, EventArgs e){
groupBox1.Enabled = false;
panel1.Enabled = true;
dataGridView1.Enabled = true;
textBox1.Clear();
textBox2.Clear();
textBox3.Clear();
textBox4.Clear();
textBox5.Clear();}
```

```
private void button6_Click(object sender, EventArgs e){
if (rec == 0){
if (string.IsNullOrEmpty(textBox1.Text) || string.IsNullOrEmpty(textBox2.Text) ||
string.IsNullOrEmpty(textBox3.Text) || string.IsNullOrEmpty(textBox4.Text) ||
string.IsNullOrEmpty(textBox5.Text)){
MessageBox.Show("Барлық жолдарды толтырыңыз!", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);}
else{
try{
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "insert into Konsultant(fam, name, patronimic, login,
password) values(N'" + textBox1.Text + "', N'" + textBox2.Text + "', N'" +
textBox3.Text + "', N'" + textBox4.Text + "', N'" + textBox5.Text + "')";
cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
 sqlConnection1.Close();
MessageBox.Show("Ақпарат сәтті енгізілді", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);
groupBox1.Enabled = false;
panel1.Enabled = true;
dataGridView1.Enabled = true;
textBox1.Clear();
textBox2.Clear();
textBox3.Clear();
textBox4.Clear();
textBox5.Clear();
a1 = "select k.kons_id, k.fam as 'Тегі', k.name as 'Аты', k.patronimic as 'Әкесінің
аты', k.login as 'Логин', k.password as 'Құпия сөз' from Konsultant k";
connect();
dataGridView1.Columns["kons_id"].Visible = false;}
catch (Exception){
MessageBox.Show("Деректер қоры жабық", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);  } }}
else if (rec == 1){
if (string.IsNullOrEmpty(textBox1.Text) || string.IsNullOrEmpty(textBox2.Text) ||
string.IsNullOrEmpty(textBox3.Text) || string.IsNullOrEmpty(textBox4.Text) ||
string.IsNullOrEmpty(textBox5.Text)){
```

```
MessageBox.Show("Барлық жолдарды толтырыңыз!", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);}
else
{
try
{
int kons_id = Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value);
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "update Konsultant set fam = N'" + textBox1.Text + "', name =
N'" + textBox2.Text + "', patronimic = N'" + textBox3.Text + "', login = N'" +
textBox4.Text + "', password = N'" + textBox5.Text + "' where kons_id = " +
kons_id;
cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
sqlConnection1.Close();
MessageBox.Show("Ақпарат сәтті жаңартылды", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);
groupBox1.Enabled = false;
panel1.Enabled = true;
dataGridView1.Enabled = true;
textBox1.Clear();
textBox2.Clear();
textBox3.Clear();
textBox4.Clear();
textBox5.Clear();
a1 = "select k.kons_id, k.fam as 'Тегі', k.name as 'Аты', k.patronimic as 'Әкесінің
аты', k.login as 'Логин', k.password as 'Құпия сөз' from Konsultant k";
connect();
dataGridView1.Columns["kons_id"].Visible = false;}
catch (Exception){
MessageBox.Show("Деректер қоры жабық", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);}}}}
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e){
if (comboBox1.Text == "Барлық тізім"){
b1 = "select k.fam as 'Клиент тегі', k.name as 'Клиент аты', k.patronimic as 'Клиент
әкесінің аты', k.inn as 'ЖСН', k.nomer_dogovora as 'Келісім нөмері', d.address as
'Үй мекен-жайы', d.nomer_d as 'Үй нөмері', d.kvartira as 'Пәтер нөмері',
d.komnatnost as 'Бөлмелер саны', d.ploshad as 'Ауданы', k.tehpasport as 'Пәтер төл
```

```
құжаты', k.summa_d as 'Келісім сомасы', k.oplacheno as 'Төленгені', k.srok_m as
'Уақыты', k.mesyac_opl as 'Төленген ай саны', ks.fam as 'Жауаптының тегі',
ks.name as 'Жауаптының аты', ks.patronimic as 'Жауаптының әкесінің аты',
k.date_dog as 'Келісім жасалған уақыт' from Klient k inner join dom d on d.dom_id
= k.dom_id inner join Konsultant ks on ks.kons_id = k.kons_id";
connect2();
else if (comboBox1.Text == "Пәтер төл құжатын алғандар тізімі"){
b1 = "select k.fam as 'Клиент тегі', k.name as 'Клиент аты', k.patronimic as 'Клиент
әкесінің аты', k.inn as 'ЖСН', k.nomer_dogovora as 'Келісім нөмері', d.address as
'Үй мекен-жайы', d.nomer_d as 'Үй нөмері', d.kvartira as 'Пәтер нөмері',
d.komnatnost as 'Бөлмелер саны', d.ploshad as 'Ауданы', k.tehpasport as 'Пәтер төл
құжаты', k.summa_d as 'Келісім сомасы', k.oplacheno as 'Төленгені', k.srok_m as
'Уақыты', k.mesyac_opl as 'Төленген ай саны', ks.fam as 'Жауаптының тегі',
ks.name as 'Жауаптының аты', ks.patronimic as 'Жауаптының әкесінің аты',
k.date_dog as 'Келісім жасалған уақыт' from Klient k inner join dom d on d.dom_id
= k.dom_id inner join Konsultant ks on ks.kons_id = k.kons_id where k.status = 1";
connect2();}
else if (comboBox1.Text == "Пәтер алуға тіркелгендер тізімі"){
b1 = "select k.fam as 'Клиент тегі', k.name as 'Клиент аты', k.patronimic as 'Клиент
әкесінің аты', k.inn as 'ЖСН', k.nomer_dogovora as 'Келісім нөмері', d.address as
'Үй мекен-жайы', d.nomer_d as 'Үй нөмері', d.kvartira as 'Пәтер нөмері',
d.komnatnost as 'Бөлмелер саны', d.ploshad as 'Ауданы', k.tehpasport as 'Пәтер төл
құжаты', k.summa_d as 'Келісім сомасы', k.oplacheno as 'Төленгені', k.srok_m as
'Уақыты', k.mesyac_opl as 'Төленген ай саны', ks.fam as 'Жауаптының тегі',
ks.name as 'Жауаптының аты', ks.patronimic as 'Жауаптының әкесінің аты',
k.date_dog as 'Келісім жасалған уақыт' from Klient k inner join dom d on d.dom_id
= k.dom_id inner join Konsultant ks on ks.kons_id = k.kons_id where k.status = 0";
connect2();}
else{
b1 = "select k.fam as 'Клиент тегі', k.name as 'Клиент аты', k.patronimic as 'Клиент
әкесінің аты', k.inn as 'ЖСН', k.nomer_dogovora as 'Келісім нөмері', d.address as
'Үй мекен-жайы', d.nomer_d as 'Үй нөмері', d.kvartira as 'Пәтер нөмері',
d.komnatnost as 'Бөлмелер саны', d.ploshad as 'Ауданы', k.tehpasport as 'Пәтер төл
құжаты', k.summa_d as 'Келісім сомасы', k.oplacheno as 'Төленгені', k.srok_m as
'Уақыты', k.mesyac_opl as 'Төленген ай саны', ks.fam as 'Жауаптының тегі',
ks.name as 'Жауаптының аты', ks.patronimic as 'Жауаптының әкесінің аты',
k.date_dog as 'Келісім жасалған уақыт' from Klient k inner join dom d on d.dom_id
= k.dom_id inner join Konsultant ks on ks.kons_id = k.kons_id where k.status = 2";
connect2();}}
private void button8_Click(object sender, EventArgs e){
if (dataGridView1.Rows.Count == 0){}
```

```
else{
try{
string myDocsPath =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
SaveFileDialog saveResults = new SaveFileDialog();
saveResults.Filter = "Книга Excel (*.xls)|*.xlsx";
saveResults.FilterIndex = 2;
saveResults.RestoreDirectory = true;
saveResults.InitialDirectory = myDocsPath;
saveResults.FileName = comboBox1.Text;
saveResults.Title = "ЖСК: Экспорт результатов";
Microsoft.Office.Interop.Excel.Application ExcelApp;
ExcelApp = new Microsoft.Office.Interop.Excel.Application();
ExcelApp.Application.Workbooks.Add(Type.Missing);
ExcelApp.Columns.ColumnWidth = 19;
ExcelApp.Cells[1, 1] = "Клиент тегі";
ExcelApp.Cells[1, 2] = "Клиент аты";
ExcelApp.Cells[1, 3] = "Клиент әкесінің аты";
ExcelApp.Cells[1, 4] = "ЖСН";
ExcelApp.Cells[1, 5] = "Келісім нөмері";
ExcelApp.Cells[1, 6] = "Үй мекен-жайы";
ExcelApp.Cells[1, 7] = "Үй нөмері";
ExcelApp.Cells[1, 8] = "Пәтер нөмері";
ExcelApp.Cells[1, 9] = "Бөлмелер саны";
ExcelApp.Cells[1, 10] = "Ауданы";
ExcelApp.Cells[1, 11] = "Пәтер төл құжаты";
ExcelApp.Cells[1, 12] = "Келісім сомасы";
ExcelApp.Cells[1, 13] = "Төленгені";
ExcelApp.Cells[1, 14] = "Уақыты";
ExcelApp.Cells[1, 15] = "Төленген ай саны";
ExcelApp.Cells[1, 16] = "Жауаптының тегі";
ExcelApp.Cells[1, 17] = "Жауаптының аты";
ExcelApp.Cells[1, 18] = "Жауаптының әкесінің аты";
ExcelApp.Cells[1, 19] = "Келісім жасалған уақыт ";
for (int i = 0; i < dataGridView2.ColumnCount; i++){
for (int j = 0; j < dataGridView2.RowCount; j++){
ExcelApp.Cells[j + 2, i + 1] = (dataGridView2[i, j].Value).ToString();}}
ExcelApp.ActiveWorkbook.SaveAs(saveResults.FileName,
saveResults.ShowDialog());
ExcelApp.Workbooks.Close();
ExcelApp.Application.Quit();
```

```
MessageBox.Show("Файл сақталды.", "Назар аударыңыз!",
MessageBoxButtons.OK, MessageBoxIcon.Information);}
catch (Exception){
MessageBox.Show("Файлды сақтау кезінде қателіктер пайда болды.", "Назар
аударыңыз!", MessageBoxButtons.OK, MessageBoxIcon.Warning);}}}
private void button1_Click(object sender, EventArgs e){
tabControl1.SelectTab(tabPage1);}
private void button2_Click(object sender, EventArgs e){
tabControl1.SelectTab(tabPage2);}
private void button3_Click(object sender, EventArgs e){
this.Close();}
private void button9_Click(object sender, EventArgs e){
tabControl1.SelectTab(tabPage3);}
private void button10_Click(object sender, EventArgs e){
rec2 = 0;
groupBox5.Enabled = true;
groupBox4.Enabled = false;}
private void button11_Click(object sender, EventArgs e){
rec2 = 1;
groupBox5.Enabled = true;
groupBox4.Enabled = false;
string dom_id = dataGridView3.CurrentRow.Cells[0].Value.ToString();
comboBox3.Text = dataGridView3.CurrentRow.Cells[1].Value.ToString();
string address = dataGridView3.CurrentRow.Cells[2].Value.ToString();
string nomer_d = dataGridView3.CurrentRow.Cells[3].Value.ToString();
string kvart = dataGridView3.CurrentRow.Cells[4].Value.ToString();
string komn = dataGridView3.CurrentRow.Cells[5].Value.ToString();
string plosh = dataGridView3.CurrentRow.Cells[6].Value.ToString();
string kom = dataGridView3.CurrentRow.Cells[7].Value.ToString();
textBox6.Text = address;
textBox7.Text = nomer_d;
textBox8.Text = kvart;
textBox9.Text = komn;
textBox10.Text = plosh;
comboBox2.Text = kom;}
private void button12_Click(object sender, EventArgs e){
if (rec2 == 1){
if (string.IsNullOrEmpty(textBox6.Text) || string.IsNullOrEmpty(textBox7.Text) ||
string.IsNullOrEmpty(textBox9.Text) ||
string.IsNullOrEmpty(textBox10.Text)){
```

```
MessageBox.Show("Барлық жолдарды толтырыңыз!", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);}
else{
try{
var dt = (from i in Program.db.Ntypes where i.type_name == comboBox3.Text select
i.type_id).SingleOrDefault();
int komid = (from i in Program.db.Genpodryadchiks where i.gen_name ==
comboBox2.Text select i.gen_id).SingleOrDefault();
int dom_id = Convert.ToInt32(dataGridView3.CurrentRow.Cells[0].Value);
int kons_id = Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value);
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "update dom set address = N'" + textBox6.Text + "', nomer_d
= N'" + textBox7.Text + "', kvartira = N'" + textBox8.Text + "', komnatnost = N'" +
textBox9.Text + "', ploshad = N'" + textBox10.Text + "', gen_id = '" + komid + "',
type_id = '" + dt + "' where dom_id = " + dom_id;
                    cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
sqlConnection1.Close();
MessageBox.Show("Ақпарат сәтті жаңартылды", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);
textBox6.Clear();
textBox7.Clear();
textBox8.Clear();
textBox9.Clear();
textBox10.Clear();
comboBox2.Items.Clear();
var kom = (from i in Program.db.Genpodryadchiks select i.gen_name);
foreach (var t in kom){
comboBox2.Items.Add(t);}
comboBox3.Items.Clear();
var dt1 = from i in Program.db.Ntypes select i.type_name;
foreach (var i in dt1){
comboBox3.Items.Add(i);}
groupBox5.Enabled = false;
groupBox4.Enabled = true;
c1 = "select d.dom_id, n.type_name as 'Үй типі', d.address as 'Мекен-жайы',
d.nomer_d as 'Үй нөмері', d.kvartira as 'Пәтер нөмері', d.komnatnost  as 'Бөлмелер
саны', d.ploshad as 'Ауданы', g.gen_name  as 'Құрылыс копаниясы', g.brigadir_fam
```

```
as 'Бригада басшысының тегі', g.brigadir_name as 'Бригада басшысының аты',
g.brigadir_patronimic as 'Бригада басшысының әкесінің аты', g.brigadir_tel as
'Бригада басшысының байланыс нөмері', d.status from dom d inner join
Genpodryadchik g on g.gen_id = d.gen_id inner join Ntype n on n.type_id =
d.type_id";
connect3();
dataGridView3.Columns["dom_id"].Visible = false;}
catch (Exception){
MessageBox.Show("Деректер қоры жабық", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);}}}
else if (rec2 == 0){
if (string.IsNullOrEmpty(textBox6.Text) || string.IsNullOrEmpty(textBox7.Text) ||
string.IsNullOrEmpty(textBox8.Text) || string.IsNullOrEmpty(textBox9.Text) ||
string.IsNullOrEmpty(textBox10.Text)){
MessageBox.Show("Барлық жолдарды толтырыңыз!", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);}
else{
try{
var dt = (from i in Program.db.Ntypes where i.type_name == comboBox3.Text select
i.type_id).SingleOrDefault();
int komid = (from i in Program.db.Genpodryadchiks where i.gen_name ==
comboBox2.Text select i.gen_id).SingleOrDefault();
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "insert into dom(address, nomer_d, kvartira, komnatnost,
ploshad, gen_id, type_id, status) values(N'" + textBox6.Text + "', N'" +
textBox7.Text + "', N'" + textBox8.Text + "', N'" + textBox9.Text + "', N'" +
textBox10.Text + "', '" + komid + "', '" + dt + "', '0')";
cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
sqlConnection1.Close();
MessageBox.Show("Ақпарат сәтті енгізілді", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);
textBox6.Clear();
textBox7.Clear();
textBox8.Clear();
textBox9.Clear();
textBox10.Clear();
comboBox2.Items.Clear();
```

```
var kom = (from i in Program.db.Genpodryadchiks select i.gen_name);
foreach (var t in kom){
comboBox2.Items.Add(t);}
comboBox3.Items.Clear();
var dt1 = from i in Program.db.Ntypes select i.type_name;
foreach (var i in dt1){
comboBox3.Items.Add(i);}
groupBox5.Enabled = false;
groupBox4.Enabled = true;
c1 = "select d.dom_id, n.type_name as 'Үй типі', d.address as 'Мекен-жайы',
d.nomer_d as 'Үй нөмері', d.kvartira as 'Пәтер нөмері', d.komnatnost  as 'Бөлмелер
саны', d.ploshad as 'Ауданы', g.gen_name  as 'Құрылыс копаниясы', g.brigadir_fam
as 'Бригада басшысының тегі', g.brigadir_name as 'Бригада басшысының аты',
g.brigadir_patronimic as 'Бригада басшысының әкесінің аты', g.brigadir_tel as
'Бригада басшысының байланыс нөмері', d.status from dom d inner join
Genpodryadchik g on g.gen_id = d.gen_id inner join Ntype n on n.type_id =
d.type_id";
connect3();
dataGridView3.Columns["dom_id"].Visible = false;}
catch (Exception){
MessageBox.Show("Деректер қоры жабық", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);}}}}
private void button13_Click(object sender, EventArgs e){
textBox6.Clear();
textBox7.Clear();
textBox8.Clear();
textBox9.Clear();
textBox10.Clear();
groupBox5.Enabled = false;
groupBox4.Enabled = true;}
private void textBox7_KeyPress(object sender, KeyPressEventArgs e){
if (!(Char.IsDigit(e.KeyChar))){
if (e.KeyChar != (char)Keys.Back){
e.Handled = true;}}}
private void textBox8_KeyPress(object sender, KeyPressEventArgs e){
if (!(Char.IsDigit(e.KeyChar))){
if (e.KeyChar != (char)Keys.Back){
e.Handled = true;}}}
private void textBox9_KeyPress(object sender, KeyPressEventArgs e){
if (!(Char.IsDigit(e.KeyChar))){
if (e.KeyChar != (char)Keys.Back){
```

```
e.Handled = true;}}}
private void textBox10_KeyPress(object sender, KeyPressEventArgs e){
if (!(Char.IsDigit(e.KeyChar))){
if (e.KeyChar != (char)Keys.Back){
e.Handled = true;}}}
private void textBox15_KeyPress(object sender, KeyPressEventArgs e){
if (!(Char.IsDigit(e.KeyChar))){
if (e.KeyChar != (char)Keys.Back){
e.Handled = true;}}}
private void button15_Click(object sender, EventArgs e){
groupBox6.Enabled = false;
groupBox7.Enabled = true;
rec3 = 0;}
private void button14_Click(object sender, EventArgs e){
groupBox6.Enabled = false;
groupBox7.Enabled = true;
rec3 = 1;
textBox11.Text = dataGridView4.CurrentRow.Cells[1].Value.ToString();
textBox12.Text = dataGridView4.CurrentRow.Cells[2].Value.ToString();
textBox13.Text = dataGridView4.CurrentRow.Cells[3].Value.ToString();
textBox14.Text = dataGridView4.CurrentRow.Cells[4].Value.ToString();
textBox15.Text = dataGridView4.CurrentRow.Cells[5].Value.ToString();}
private void button16_Click(object sender, EventArgs e){
textBox11.Clear();
textBox12.Clear();
textBox13.Clear();
textBox14.Clear();
textBox15.Clear();
groupBox6.Enabled = true;
groupBox7.Enabled = false;}
private void button17_Click(object sender, EventArgs e){
if (string.IsNullOrEmpty(textBox11.Text) || string.IsNullOrEmpty(textBox12.Text) ||
string.IsNullOrEmpty(textBox13.Text) || string.IsNullOrEmpty(textBox14.Text) ||
string.IsNullOrEmpty(textBox15.Text)){
MessageBox.Show("Барлық жолдарды толтырыңыз!", "Назар аударыңыз",
MessageBoxButtons.OK,
MessageBoxIcon.Information);}
else{
if (rec3 == 0){
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
```

```
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "insert into Genpodryadchik(gen_name, brigadir_fam,
brigadir_name, brigadir_patronimic, brigadir_tel) values(N'" + textBox11.Text + "',
N'" + textBox12.Text + "', N'" + textBox13.Text + "', N'" + textBox14.Text + "', N'"
+ textBox15.Text + "')";
cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
sqlConnection1.Close();
MessageBox.Show("Ақпарат сәтті енгізілді", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);
comboBox2.Items.Clear();
var kom = (from i in Program.db.Genpodryadchiks select i.gen_name);
foreach (var t in kom){
comboBox2.Items.Add(t);}
d1 = "select g.gen_id, g.gen_name as 'Құрылысқа жауапты компания',
g.brigadir_fam as 'Бригада басшысының тегі', g.brigadir_name as 'Бригада
басшысының аты', g.brigadir_patronimic as 'Бригада басшысының әкесінің аты',
g.brigadir_tel as 'Бригада басшысының байланыс ннөмері' from Genpodryadchik
g";
connect4();
dataGridView4.Columns["gen_id"].Visible = false;
textBox11.Clear();
textBox12.Clear();
textBox13.Clear();
textBox14.Clear();
textBox15.Clear();
groupBox6.Enabled = true;
groupBox7.Enabled = false;}
else if (rec3 == 1){
int genid = Convert.ToInt32(dataGridView4.CurrentRow.Cells[0].Value);
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "update Genpodryadchik set gen_name = N'" +
textBox11.Text + "', brigadir_fam = N' " + textBox12.Text + "', brigadir_name = N'"
+ textBox13.Text + "', brigadir_patronimic = N'" + textBox14.Text + "', brigadir_tel
= N'" + textBox15.Text + "' where gen_id = " + genid;
cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
```

```
sqlConnection1.Close();
MessageBox.Show("Ақпарат сәтті жаңартылды", "Назар аударыңыз",
MessageBoxButtons.OK, MessageBoxIcon.Information);
comboBox2.Items.Clear();
var kom = (from i in Program.db.Genpodryadchiks select i.gen_name);
foreach (var t in kom){
comboBox2.Items.Add(t);}
dataGridView4.Columns["gen_id"].Visible = false;
d1 = "select g.gen_id, g.gen_name as 'Құрылысқа жауапты компания',
g.brigadir_fam as 'Бригада басшысының тегі', g.brigadir_name as 'Бригада
басшысының аты', g.brigadir_patronimic as 'Бригада басшысының әкесінің аты',
g.brigadir_tel as 'Бригада басшысының байланыс ннөмері' from Genpodryadchik
g";
connect4();
textBox11.Clear();
textBox12.Clear();
textBox13.Clear();
textBox14.Clear();
textBox15.Clear();
groupBox6.Enabled = true;
groupBox7.Enabled = false;}}}
private void button18_Click(object sender, EventArgs e){
tabControl1.SelectTab(tabPage4);}
private void comboBox3_SelectedIndexChanged(object sender, EventArgs e){
if (comboBox3.Text == "жеке үй"){
textBox8.Clear();
textBox8.Enabled = false;}
else if (comboBox3.Text == "пәтер"){
textBox8.Enabled = true;
textBox8.Text = dataGridView3.CurrentRow.Cells[4].Value.ToString();}}
private void label11_Click(object sender, EventArgs e){}}}
```

Б қосымшасы

Техника-экономикалық негізделуі бөлімі бойынша қосымша

К е с т е 1 – q коэффициентінің мәні

Тапсырма түрлері	Коэффициенттің өзгеру аралығы
Есептеу тапсырмалары	1400 ден 1500
Оперативті басқару тапсырмалары	1500 ден 1700
Жоспарлау тапсырмалары	3000 ден 3500
Көп вариантты	4500 ден 5000
Комплекстік тапсырма	5000 ден 5500

К е с т е 2 – Еңбек сыйымдылығын есептейтін коэффициент

Бағдарлама тілі	Күрделік тобы	Жаңалықтық дәрежесі				В коэффициенті
		А	Б	В	Г	
Жоғарғы деңгей	1	1,38	1,26	1,15	0,69	1,2
	2	1,30	1,19	1,08	0,65	1,35
	3	1,20	1,10	1,00	0,60	1,5
Төменгі деңгей	1	1,58	1,45	1,32	0,79	1,2
	2	1,49	1,37	1,24	0,74	1,35
	3	1,38	1,26	1,15	0,69	1,5

К е с т е 3 – Бағдарламалық өнімді жасауға жалпы уақыт құрамы

Кезең №	Дәл кезеңдегі уақыт белгісі	Кезеңнің мазмұны
1	$T_{ПО}$	Мақсат сипатын дайындау
2	$T_{О}$	Мақсат сипаттамасы
3	$T_{А}$	Алгоритм құру
4	$T_{БС}$	Алгоритмнің блок-схемасын құру
5	$T_{Н}$	Бағдарламаны ... тілде жазу
6	$T_{П}$	Бағдарламаны теру
7	$T_{ОП}$	Бағдарламаны реттеу және тестілеу
8	$T_{Д}$	Құжаттарды рәсімдеу, пайдаланушыға нұсқаулар және түсіндірмелер жазу

Б қосымшасының соңы

К е с т е 4 – Бағдарлама жасаушы білектілігін ескеретін коэффициент

Жұмыс тәжірибиесі	Білектілік коэффициенті
Екі жылға дейін	0.8
2-3 жыл	1
3-5 жыл	1.1 – 1.2
5-7 жыл	1.3 – 1.4
7 жылдан көп	1.5– 1.6

Бірыңғай тарифтік сеткадан көшірме (БТС)

К е с т е 5 – Бірыңғай тарифтік сеткадан көшірме (БТС)

Еңбек ақы разрядры	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Тарифтік коэффициент	1,0	1,07	1,15	1,24	1,33	1,43	1,54	1,66	1,78	1,91	2,05	2,25	2,37	2,55

К е с т е 5 жалғасы

Еңбек ақы разряды	15	16	17	18	19	20	21
Тарифтік коэффициент	2,74	2,95	3,17	3,41	3,67	3,94	4,24