

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

ҚОРҒАУҒА ЖІБЕРІЛДІ

Кафедра меңгерушісі

PhD, доцент

_____ Т.С. Картбаев
« ____ » _____ 2019 ж.

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: “Банктің несие бөлімі” ақпараттық жүйесін әзірлеу

Мамандығы: 5B060200 – «Информатика»

Орындаған: Жолжаксынов Е.Т. Тобы: ИНФк-15-1

Ғылыми жетекші: аға оқытушы Тоғжанова К.О.

Кеңесшілер:

Экономикалық бөлім: аға оқытушы _____ *Тул* С.К. Тулегенова
« 19 » 04 2019 ж.

Өміртіршілік қауіпсіздігі: т.ғ.д., аға оқытушы _____ *Тр* Ш.Ш. Бекбасаров
« 10 » 03 2019 ж.

Есептеу техникасын қолдану: аға оқытушы _____ *Айт* Ж.С. Айткулов
« 13 » 05 2019 ж.

Норма бақылаушы: аға оқытушы _____ *Мукапил* К. Мукапил
« 15 » 05 2019 ж.

Сын-пікір беруші: т.ғ.к., доцент _____ Д.М. Ескендирова
« ____ » _____ 2019 ж.

Алматы 2019

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

Басқару жүйелері және ақпараттық технологиялар институты

IT-инжиниринг кафедрасы

Мамандығы 5B060200 – «Информатика»

Дипломдық жобаны орындауға берілген

ТАПСЫРМА

Білім алушы Жолжаксынов Ердаулет Талгатович

Жобаның тақырыбы: «Банктің несие бөлімі» ақпараттық жүйесін әзірлеу

2019 жылғы «01» наурыз № 33 университет бұйрығымен бекітілген.

Аяқталған жобаны тапсыру мерзімі: «24» мамыр 2019 ж.

Дипломдық жобаның бастапқы мәліметтері (зерттеу (жоба) нәтижелерінің талап етілген параметрлері мен объектінің бастапқы мәліметтері): Ұсынылып отырған дипломдық жобада банктерге арналған “Банктің несие бөлімі” ақпараттық жүйесін әзірлеу. Жобаны орындау барысында C# бағдарламалау тілін, MS SQL мәліметтер қорын қолдандым.

Дипломдық жобада қарастырылған мәселелер тізімі немесе дипломдық жобаның қысқаша мазмұны:

- талдау бөлімі;
- жобалау бөлімі;
- жүзеге асыру және тестілеу бөлімі;
- экономикалық бөлім;
- өміртіршілік қауіпсіздігі;
- А қосымшасы. Техникалық тапсырма;
- Ә қосымшасы. Программа листингі;
- Б қосымшасы. Ендіру актісі.

Графикалық материалдар тізімі (міндетті сызбалар дәл көрсетілуі тиіс):
16 кесте, 31 сурет ұсынылған.

Ұсынылатын негізгі әдебиеттер:

- 1 Линн Б. Изучаем SQL. – Издательство: Питер, 2012.100-110 с.

2 Молиново Э. SQL. Сборник рецептов. – Пер. с англ. – СПб.: Символ-Плюс, 2009. 20-35 с.

3 Дж. Грофф, П. Вайнберг. SQL: Полное руководство: Пер. с англ. – 2-е изд., перераб. и доп. – К.: Издательская группа BHV, 2001. 74-80 с.

4 АндонФ, Резниченко В. Язык запросов SQL. Учебный курс. – СПб.: Питер; Киев: Издательская группа BHV, 2006. 81-101 с.

Дипломдық жобаның бөлімдеріне қатысты белгіленген кеңес берушілер

Бөлімдер	Кеңесшілер	Мерзімі	Қолы
Экономикалық бөлім	Тулегенова С.К.	19.04.2019	<i>Тул</i>
Өміртіршілік қауіпсіздігі	Бекбасаров Ш.Ш.	19.02.2019 – 10.03.2019	<i>Бек</i>
Программалық қамтама	Айтқулов Ж.С.	01.03.2019 – 30.04.2019	<i>ЖС</i>
Норма бақылау	Мукапил К.	04.04.2019 – 10.05.2019	<i>Мукапил</i>

Дипломдық жобаны дайындау
КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекшіге ұсыну мерзімдері	Ескерту
Талдау бөлімі	15.01.19 – 31.01.19	<i>Организац</i>
Жобалау бөлімі	01.02.19 – 28.02.19	<i>Организац</i>
Жүзеге асыру және тестілеу бөлімі	01.03.19 – 20.03.19	<i>Организац</i>

Тапсырманың берілген күні «29» қазан 2018 ж.

Кафедра меңгерушісі _____ Т.С. Картбаев

Жобаның ғылыми жетекшісі *Жеку* _____ К.О. Тоғжанова

Тапсырманы орындауға алған білім алушы *Молд* _____ Е.Т. Жолжаксынов

Аңдатпа

Осы дипломдық жобада «Банктің несие бөлімі» ақпараттық жүйесі әзірленді.

Ақпараттық қосымша төмендегідей қызмет атқарады: несие алушылар жөнінде толық ақпарат беру, несие алушы клиенттерді тіркеу, клиенттің аты-жөні, мекен-жайы, келісім шарт нөмірінің өсуі бойынша сұрыптау, клиенттердің id-номері бойынша іздеу жүйесін ұйымдастыру, несие алушылардың толық статистикалық мәліметін шығару және .т.б.

Ақпараттық іздеу жүйесін құрастыру барысында объектілі-бағытталған бағдарламалау тілінің бірі – C# және MS SQL Server мәліметтер қоры қолданылған.

Аннотация

В дипломном проекте разработано информационная система «Кредитный отдел банка».

Целью дипломной работы является разработка информационного приложения «Банковская система отдел кредитование». Данное информационное приложение выполняет следующие задачи: предоставлении информации о заемщиках, регистрация нового клиента, Ф.И.О, адрес клиента, сортировка по возрастанию номера договора, организация системы поиска по id-номеров клиентов, вывод полных статистических данных заемщиков.

При разработке программного обеспечения дипломной работы использовалась объектно-ориентированная среда программирования С# и СУБД MS SQL Server.

Annotation

In the research paper developed informative annex "credit department of the banking system".

The aim of the thesis is to develop an information application "The banking system lending department". This data application performs the following tasks: providing information on borrowers, the registration of a new client, full name, address of the client Sorting in increasing numbers treaty organization look-id-numbers of clients, conclude comprehensive statistics borrowers. In software development, the thesis used object-oriented programming environment C# and database MS SQL Server.

Мазмұны

	Кіріспе	8
1	Талдау бөлімі	9
1.1	Банктік қызыметтің автоматтандырылған жүйелері	9
1.2	Несие түсінігі және оның түрлері	14
1.3	Банктің ақпараттық технологияларына шолу	17
2	Жобалау бөлімі	19
2.1	Мәліметтер қорын басқару жүйесін таңдауды негіздеу	19
2.2	Программа құрудың интегралдық ортасын сипаттау	25
2.3	SQL тілі – мәліметтер қорының негізгі тілі	28
2.4	Мәліметтер қорының құрылымын сипаттау	32
2.5	Пәндік саланы модельдеу	37
2.6	Прецеденттер диаграммасы	38
2.7	Тізбек диаграммасы	40
2.8	Кооперация диаграммасы	42
2.9	Қызмет диаграммасы	42
2.10	Кластар диаграммасы	43
3	Жүзеге асыру және тестілеу бөлімі	46
3.1	Программалау тілін таңдауды негіздеу	46
3.2	Банктің несие бөлімінің ақпараттық қосымшасын пайдаланушыға нұсқау	47
3.3	Программаның баяндалуы	52
3.4	Функционалдық тағайындалуы	52
3.5	Қажетті техникалық жабдықтар	52
4	Экономикалық бөлім	53
4.1	Экономикалық бөлімдегі орындалатын мақсаттар мен тапсырмалар	53
4.2	Программалық өнімді құрудың еңбек сыйымдылығын есептеу	53
4.3	Программалық өнім құрудағы шығындарды есептеу	53
4.4	Еңбекке ақы төлеу шығындарын есептеу	55
4.5	Программалық өнімнің мүмкін болатын (келісімді) бағасын анықтау	59
4.6	Программалық өнімнің экономикалық бөліміне қорытынды	59
5	Өміртіршілік қауіпсіздігі	60
5.1	Еңбек жағдайларын талдау	60
5.2	Табиғи жарықтандыруды есептеу	64
5.3	Жасанды жарықтандыруды есептеу	66
	Қорытынды	71
	Әдебиеттер тізімі	72
	А қосымшасы. Техникалық тапсырма	74
	Ә қосымшасы. Программа листингі	78
	Б қосымшасы. Ендіру актісі.	89

Кіріспе

Ақпараттық қосымша – қолданушымен тікелей байланыс жасауға негізделген, қолданушылар тапсырмасын орындауға арналған программа.

Қолдану аймағына және функцияларына байланысты ақпараттық қосымшаларға келесі талаптар қойылады:

– ақпараттық қосымша ақпараттарды жинауға, өңдеуге және сақтауға арналған. Сондықтан ақпараттық қосымшаның сақтау ортасы мен мәліметтерге қатынасы оның негізі болып табылады, сақтау ортасы сәйкес аймаққа қатысты және сақтау деңгейі сенімді болуы керек;

– ақпараттық қосымшаның интерфейстері қолданушы үшін оңай игерілетін және ыңғайлы болуы тиіс, қазіргі кездегі ақпараттық қосымшаларда дамыған графикалық интерфейстің болуы міндетті болып саналады.

Мәліметтер қоры дегеніміз – ақпараттарды сақтау үшін арналып ұйымдастырылған құрылым. Алғашқы кезде, яғни, мәліметтер базасы түсінігі қалыптаса бастаған жылдары, бұл қорларда тек қана мәліметтер сақталып жүрді. Бірақ бүгінгі күнгі мәліметтер қорын басқару жүйелері (МҚБЖ) өз құрылымында тек қана мәліметтерді ғана емес, сонымен қатар тұтынушылармен немесе басқа да программалық-ақпараттық кешендермен қарым-қатынас жүзеге асырылатын әдістерді (яғни, программалық код) де орналастыруға мүмкіндік береді. Осылайша біз қазіргі заманғы мәліметтер қорында тек қана мәліметтер ғана емес, сонымен бірге ақпарат та сақталатындығын айта аламыз.

Бұл дипломдық жобаның мақсаты:

– несиелік операциялар туралы ақпаратты өңдеуші банк қызметкерлерінің және несие бөлімінің қызметкерлерінің еңбек өнімділігін арттыру;

– клиентті несие түрлерімен таныстыру;

– клиенттерге қызмет көрсету сапасын арттыру;

– сонымен қатар клиенттің арызын қабылдап, кредит алуына келісім жасайтын, клиенттің төлемдерін қабылдайтын банктің несие бөлімінің автоматтандырылған ақпараттық қосымшасын құру болып табылады;

Ақпараттық қосымшаның негізгі орындайтын функциялары:

– клиентке кредит түрлері туралы ақпарат беру;

– жылдық несиеден түскен пайданы есептеу, мониторинг жасау.

Бұл қосымша несие алушыны несие туралы ақпаратпен қамтамасыз етіп, несие беру операцияларын жүзеге асырып, қағазбасты технологиядан құтылуға жағдай тудырады.

Дипломдық жұмыс нәтижесінде құрастырылған жүйе тұтынушы қауымға белгілі деңгейде пайдасын тигізеді.

1 Талдау бөлімі

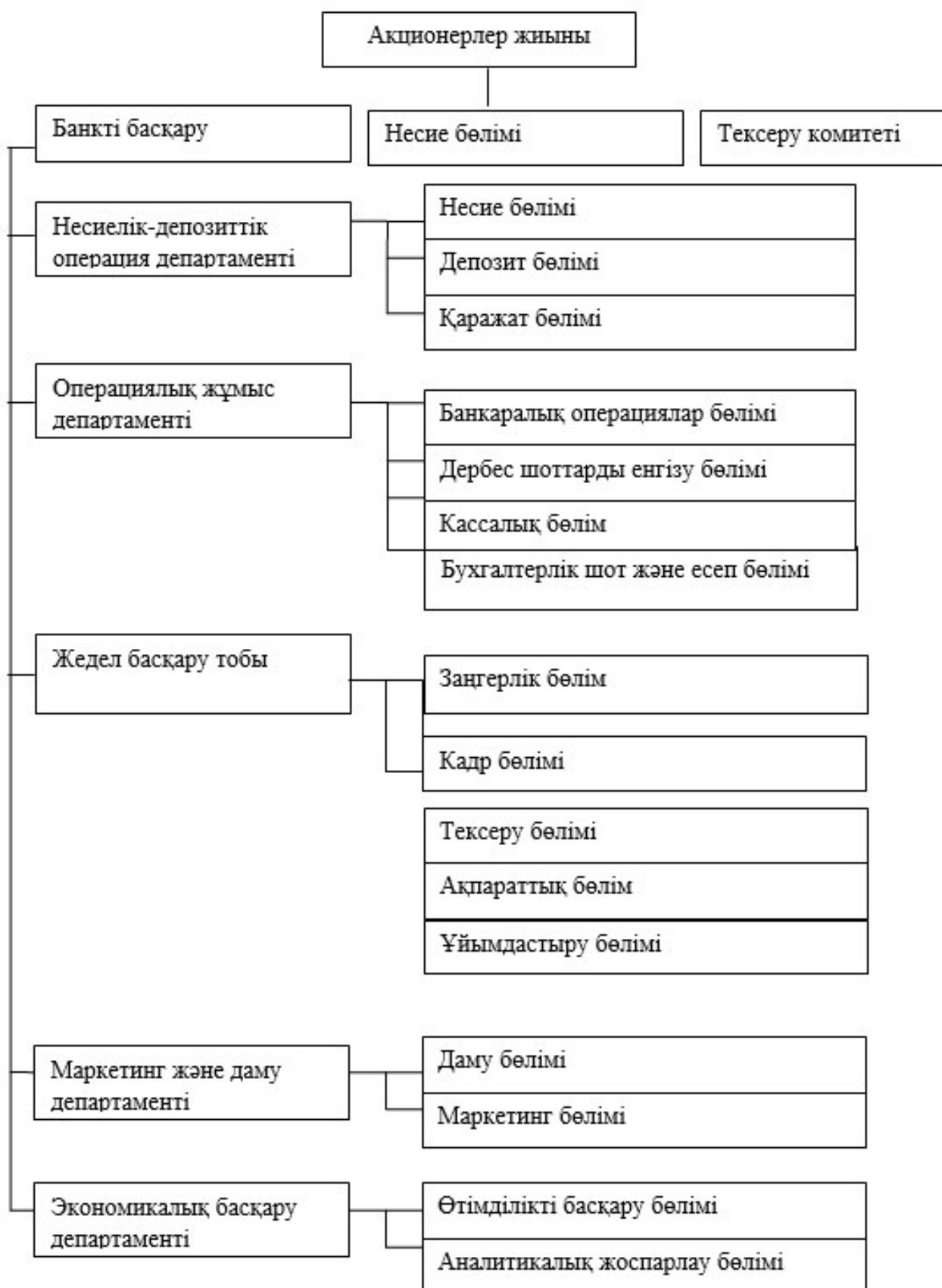
1.1 Банктік қызметтің автоматтандырылған жүйелері

Банк – ақша қаржылары мен қорларын жинақтау, кредит беру, мемлекеттер, кәсіпорындар (фирмалар), мекемелер және жеке адамдар арасындағы өзара төлемдер мен ақшалай есеп айырысуда делдалдықты жүзеге асыру, ақшаның белгілі бір түрін айналымға қосу, ақша мен құнды қағаздар шығару, алтынмен, шетелдік валютамен түрлі операциялар жасау және басқа да қызметтер атқаратын қаржы мекемесі. Банктің пайда болуы мен дамуы тауар- ақша қатынастарының дамуымен тығыз байланысты. Банктер кәсіпорындардың, фирмалардың, компаниялардың, сонымен қатар тұрғындардың уақытша пайдаланылмай бос тұрған қаражаттарын жинақтап, қажет етушілерге өсіммен уақытша кредит беру арқылы қаржы мұқтаждығын өтейді. Қызмет көрсету аясындағы банктің өнеркәсіптік кәсіпорындардан ерекшелігі оның несие беруінен байқалады.[3]

Қазіргі кездегі банк ісі – ұзақ мерзімдік дамудың нәтижесі. Банктер уақытша пайдаланылмай тұрған бос ақша қаражаттарын өзіне жинақтау, клиенттерге есепшот ашып, кассалық қызмет көрсету, қысқа және ұзақ мерзімге кредит беру, инвестициялық қызмет атқару (ұзақ мерзімді күрделі шығындарды қаржыландыру, құнды қағаздар шығару) сияқты өзіндік арнаулы қызметтерге қоса, нарықтық экономика жағдайында лизингтік, факторингтік, кеңесшілік, ақпараттық қызметтерді де атқарады, банк басқарудың құрылымдық сұлбасы төменде 1.1- суретте көрсетілген. Банк жүргізетін операциялар пассивті (ресурстарды банктегі есепшотқа енгізу) және активті болып бөлінеді. Негізгі пассивті операциялардың қатарына банктің өзіндік капиталы депозиттер қабылдау және банкаралық ақша пиасасынан қосымша ресурстар алу жатса, ал негізгі активті операцияларға банкаралық кредит беру, инвестициялық қаржы бөлу жатады.

Банк технологиясын автоматтандыру процесінің актуалдылығы 80-жылдардың соңы және 90-жылдардың басында ерекше көзге көрінді. Бұл 1989 жылдағы банк реформасына байланысты еді. Осы кезде жұмыс атқарушы банктер өз алдына ауқымды еркіншілікке ие болды да, бұрынғы мемлекеттік банктермен қатар жаңа коммерциялық банктер пайда болды. Осындай жағдайда банк ақпараттарын өңдеуге пайдаланылатын есептеу цифрлары банктерге қаржы ахуалын болжау, анализдеу және жұмысын жеңілдететін қызмет ауқымын ұсына алмайтын болды.[4]

Банк және өзге де қаржы құрылымдары қызметін автоматтандыру процесінің дамуы көптеген ақпараттық технологиялардың пайда болуына әкелді. Отандық банк инфрақұрылымының компьютерлік платформасы тұрғысынан қарағанда шет елдік жүйелерден айырмашылығы – олардың жоғарғы деңгейдегі біртектілігі болды.



1.1-сурет – Банк жүйесін басқарудың құрылымдық сұлбасы

«Қазақстан Республикасындағы банктер және банктік қызмет туралы» заңының 1-бабына сәйкес, «банк – осы заңға сай банктік қызметті жүзеге асыруға құқылы коммерциялық ұйым болып табылатын заңды тұлға».

Банктік қызмет – бұл банктік операцияларды жүзеге асырумен байланысты қызметті білдіреді. Аталған заңның 30-бабына сәйкес банктік операцияларға мыналар жатады:

- заңды тұлғалардың депозиттерін қабылдау, банктік шоттарын ашу және жүргізу;
- жеке тұлғалардың депозиттерін қабылдау, банктік шоттарын ашу және жүргізу;
- банктердің және банктік операциялардың жекелеген түрлерін жүзеге асыратын ұйымдардың корреспонденттік шоттарын ашу және жүргізу;
- заңды және жеке тұлғалардың металдық шоттарын ашу және жүргізу;
- кассалық операциялар: банкнота мен монетаны қабылдау, беру, қайта санау, айырбастау, сорттау, қаптау және сақтау;
- аударым операциялары: заңды және жеке тұлғалардың ақшаны аударумен байланысты тапсырмаларын орындау;
- есепке алу операциялары: заңды және жеке тұлғалардың вексельдерін және өзге борыштық міндеттемелерін есепке алу;
- заемдық операциялар: ақы төлеу, мерзімін белгілеу және қайтару шартымен ақшалай нысанда несиелер беру;
- заңды және жеке тұлғалардың, оның ішінде корреспондент-банктердің тапсырмаларына байланысты, олардың банктік шоттары бойынша есеп айырысу операцияларын жүргізу;
- сенім операциялары: сенім білдірушінің тапсырмасы бойынша және оның мүддесіне сай, ақшасын, құйма бағалы металын және бағалы қағаздарын басқару;
- сейфтік операциялар: клиенттердің құжатты формада шығарылған бағалы қағаздарын, құжаттарын және бағалы заттарын сақтау қызметін көрсету, сондай-ақ жәшіктерді, шкафтарды және бөлмелерді жалға беру;
- ломбардтық операциялар: тез іске асатын бағалы қағаздар мен жылжитын мүліктерді кепілге алып, қысқа мерзімді несиелер беру;
- төлем карточкаларын шығару;
- банкнот пен монеталарды және бағалы заттарды жөнелту;
- шетел валютасымен айырбас операцияларын ұйымдастыру;
- төлем құжаттарын инкассоға қабылдау;
- чек кітапшаларын шығару;
- бағалы қағаздар нарығындағы клирингтік қызмет;
- аккредитивті ашу, растау және ол бойынша міндеттемені орындау;
- ақшалай нысанда орындалуды көздейтін, банктік кепіл-хаттарды беру;
- үшінші тұлғаның атынан ақшалай банктік кепілдеме беру.[3]

Әрбір үлкен не кіші банкте бөлімдер болады. Олардың әрқайсысы бір бірінен ерекшеленген әлем болып саналады, онда өзіне тән принциптер мен

ережелер сақталады. Ұйымның жұмысы тиімді болу үшін бөлімдер өзара тиімді әрекеттесе алу қажет. Қайсы бөлім маңызды, қайсысы маңызды емес екенін анықтап отырудың қажеті жоқ. Олардың бәрі маңызды. Алайда олардың ішінен бірінші дәрежелі рөл ойнайтындарын ерекшелеп көрсетуге болады.

Несиелік ұйымдардағы ең маңызды бөлімдерді атап көрсетейік:

- есептік-кассалық қызмет;
- валюталық операциялар;
- бағалы қағаздар;
- пластикалық карталар;
- несие бөлімі.

Бұл бөлімдер банктің экономикалық жағдайын анықтайды. Сондықтан осы бөлімдердің жұмысын ұтымды ұйымдастыра алу қажет.

Енді несие бөлімінің жұмысы үш кезеңнен тұрады:

- несие беру жөніндегі арызды рәсімдеу;
- арызды қарастыру және несие беру жөнінде шешім шығару;
- несие беру және қарыз сомасы жөнінде келісім шартқа қол қою.

Әрбір кезең жекелеген қызметкерлердің жұмысымен байланысты:

а) бірінші кезеңнің жүзеге асуы, ең алдымен, несиелік кеңесшілердің жұмысына тәуелді. Кеңесшілер потенциалды клиенттермен операциялық залда немесе кредиттік бөлімде мәміле жүргізеді. Олардың міндетіне потенциалды қарыз алушыларға тиімді кредит беру нұсқасын таңдауда көмектесу және қажетті құжаттарды дұрыс толтыруға көмектесу сияқты әрекеттер кіреді, ол арыз беру үрдісін барынша жылдамдату және оңайлатуды көздейді. Әрі қарай іске несиелік инспекторлар кіріседі. Олардың міндеті – несиені рәсімдеу. Инспектор құжаттар пакетін тексереді; несие сомасын және мерзімін есептейді; кредитті қамсыздандыру қажеттігін анықтайды. Сонымен қоса инспектор қарыз алушының берген мәліметтеріне сүйене отырып оның төлемге қабілеттілігін зерттейді. Кредит алушының келісімімен инспектор несиелік бюроға, оның несиелік тарихын зерттеу мақсатында сұраныс жасайды.

ә) банк арызды ұзақ уақыт қарастыруы мүмкін – шамамен 10 күнге дейін. Бұл үрдіс кредит алушының несие ісін рәсімдеумен басталады, оны кейін қауіпсіздік қызметіне жібереді. Бұл бөлімнің қызметкерлері қарыз алушының берген мәліметтерін мұқият тексереді. Сонымен қоса, қарыз алушы немесе оның туыстары қылмыстық іске тартылған ба, соны да тексереді. Бұдан кейін іс қайтадан кредиттік инспекторға қайтып келеді. Кейін толық рәсімделгеннен кейін ол несие бөлімінің бастығына тексеруге жіберіледі. Одан кейін несиелік комитеттің хатшысы ол істі «қарастырылатын істер реестріне» қосады.[4]

Банктің несиелік комитетінде басқа бөлімдердің бастықтары да болуы мүмкін: әділет, экономикалық, қауіпсіздік, бухгалтерлік есеп бөлімдері. Егер филиалдың несиелік бөлімінде осы аталған бөлімдердің адамдары болмаса, онда белгілі бір шешім қабылдау тәртібі орнатылады, бірақ оған дейін іс

аталған барлық инстанциялардан өтеді. Нәтижесінде инспектор барлық клиенттердің істерін жинап комитетте жариялайды.

б) оң шешім қабылданған жағдайда инспектор бұл жөнінде қарыз алушыға хабарлайды, кредит сомасын беру датасын белгілейді және соған қажетті құжаттарды жинай бастайды (кредиттік келісімшарт, залог келісімі). Келісімшарт пішіндері банк заңгерлерімен келісілген және бастық қолы қойылған болу керек.

Несие жүйесінің рөлі мен маңызы келесідей бірқатар көрсеткіштермен сипатталады: несиелік жұмсалымдардың жалпы көлемі, кәсіпорындар мен ұйымдардың негізгі және айналым капиталын қалыптастыруға банктегі қарыздардың үлесі, жалпы төлем айналымы және т.б. Несие туралы келісімшартқа несиелік берілетін күні екіжақты қол қойылады. Қарыз алушының қатысуы міндетті. Қарыз алушы келісімшартпен танысып, қол қойып, қажетті данарларды таратып болған соң, банк кассасына барып шартта көрсетілген сомада ақшалай қаржыны алады. Кредит келісімшарты қол қойылған күннен бастап күшіне енеді. Қарыз алушы тек оны уақытылы жауып отырса болғаны. Бірақ осымен инспектор жұмысы біткен жоқ. Жинақталған мәліметтерді ол несиелік тарих бюросына жібереді, сонымен қатар ол қарыздың уақытылы төленуін қадағалап отыруы тиіс. Аса ірі банктерде бұл функцияны кредит келісімшартын сүйемелдеу бөлімінің қызметкерлері атқаруы да мүмкін.

Несиелік бөлімнің қызметкерлерінің жұмысы үлкен жауапкершілікті, талапкерлікті қажет етеді, алайда бұл қасиеттер клиенттерге байқатпай орындалуы тиіс. Клиенттің несиелік алу процесі олар үшін ыңғайлы және жағымды болуы тиіс, олардың банкке деген сенімі арта түсуі керек. Банктің ақпараттық және өзге де технологияларын автоматтандыру стратегиясы міндеттерді жедел түрде шешуге әсерін тигізуде. Клиенттерге қызмет көрсетуді автоматтандырудың басым бағыттары банктердің бәсекелестік позициясына әсер етеді. Олар: қосымша шығындарды кеміту және қызмет көрсету сапасын арттыру.[7]

Электронды техника жетістігі мүлдем жаңа қызмет түрін – ақша массасындағы нақты қаржыны автоматты басқару жүйесін ұсынды.

Қағаз қолданылмайтын технологияның оң аспектілері:

- іс жүзінде әп-сәттік мәлімет жіберу;
- сақтаудағы ерекше артықшылық;
- құжат өңдеудегі еңбекті қажет етудің күрт кемуі.

Банк технологиясын автоматтандыру процесінің актуалдылығы 80-жылдардың соңы және 90-жылдардың басында ерекше көзге көрінді. Бұл 1989- жылдағы банк реформасына байланысты еді. Осы кезде жұмыс атқарушы банктер өз алдына ауқымды еркіншілікке ие болды да, бұрынғы мемлекеттік банктермен қатар жаңа коммерциялық банктер пайда болды. Осындай жағдайда банк ақпараттарын өңдеуге пайдаланылатын есептеу цифрлары банктерге қаржы ахуалын болжау, анализдеу және жұмысын жеңілдететін қызмет ауқымын ұсына алмайтын болды.

Банк және өзге де қаржы құрылымдары қызметін автоматтандыру процесінің дамуы көптеген ақпараттық технологиялардың пайда болуына әкелді. Отандық банк инфрақұрылымының компьютерлік платформасы тұрғысынан қарағанда шет елдік жүйелерден айырмашылығы – олардың жоғарғы деңгейдегі біртектілігі болды.[8]

1.2 Несие түсінігі және оның түрлері

Кредит (лат. creditum – несие, сенемін, сенім білдіремін) – белгілі мерзім бойы пайдаланып, қайтарылу үшін әдетте, пайыз төлеу шартымен ақшалай немесе тауар түрінде берілетін қарыз. Оның қозғалысы кезінде кредит пен қарызгер арасында белгілі бір экономикалық қатынастар қалыптасады. Кредиттің дамуы ұдайы өндіріс үрдісіндегі капиталдың ауыспалы айналымының заңдылықтарына орайлас келеді: шаруашылықтың белгілі бір телімдерінде уақытша еркін айналымдағы ақша капиталы бос қалады да, басқаларында оған деген мұқтаждық, яғни кредит алу қажеттілігі туады. Капиталдың ауыспалы айналымы негізінде пайда болған уақытша еркін айналымдағы ақша капиталы, халық пен мемлекеттің ақшалай жинақтары кредит көзін құрайды.

Кредиттік қатынастар жүйесінде кредит операцияларын жасау үшін қаражат тартудың да маңызы зор. Мұндай қызмет әр түрлі кредиттік қатынастар үшін біркелкі бола бермейді. Кредит өнімнің өндірілуі, өткізілуі мен тұтынылуы үрдісіне және ақша айналысы саласына әсерін тигізеді, яғни өнімді өндіру мен өткізу үрдістерінің кемшіліксіз болуына ықпал етеді, өндірістің маусымдылығына орай туындаған қаражатқа деген қажеттілікті уақытша қанағаттандыруда, өндірісті кеңейтуде үлкен рөл атқарады. Тұтыну кредиті халықтың әр түрлі қажеттіліктерін қанағаттандыруға септігін тигізеді.[3]

Қазақстанда кредиттің мынадай түрлері пайдаланылады:

- ұлттық банк кредиттері – аукциондық, ломбардтық, бюджеттік кредиттер;
- екінші деңгейдегі банктер кредиттері – меншікті айналым қаражатының жетіспеушілігін уақытша толтыруға, күрделі салымдарды қаржыландыруға, тұтыну мақсатына арналған кредиттер; импорттық, ипотекалық, консорциалдық, банкаралық кредиттер. Қазақстан Ұлттық банкінде кредит сапасына қарай стандарттық, бейстандарттық, қанағаттанусыз, күмәнді және зиян келтіруші кредит деп бөлінеді.

Кредиттің формасы – бұл кредит қатынасының сырттай нақты көрініс табуы. Ол кредит қатынасының мәні мен ұйымдастырылуын синтездейді. Кредит қатынасының формасы мен мазмұны ажырағысыз әрі диалектикалық жағынан біртұтас болады. Кредит қатынасының формасы оның мазмұны мен дамуына сәйкес келуі керек.[4]

Таңдап алынған жіктеу өлшеміне қарай кредиттің мынадай ең маңызды формаларын бөліп көрсетуге болады.

– қызмет саласына қарай – ұлттық және халықаралық кредит;
– кредит мәмілесінің объектісіне қарай – ақшалай және тауарлық кредит;

– кредит қатынасының субъектісіне қарай – банктік, коммерциялық, халықаралық, тұтынушылық кредиттер.

Қазақстанда кредит түрлері былайша жіктеледі:

а) кредиттеу объектінің экономикалық белгілері бойынша:

- 1) айналым қаражатын қалыптастыруға берілетін несиелер;
- 2) негізгі құрал-жабдықтарды қалыптастыруға берілетін несиелер;
- 3) өндірістің маусымдық шығыны аясында берілетін несиелер;
- 4) жол үстіндегі есеп айырысу құжаттары аясында берілетін

кредит;

5) төлем кредиттері.

ә) қамтамасыз етілуі бойынша:

1) жылжымалы және жылжымайтын мүлікпен, кепілдікпен, сақтандыру келісім шартымен толық қамтамасыз етілген;

2) ішінара қамтамасыз етілген;

3) қамтамасыз етілуі болмайтын банкілік.

б) қайтарылу мерзімі бойынша:

1) қысқа мерзімді;

2) орта мерзімді;

3) ұзақ мерзімді.

в) өтелу тәртібі бойынша:

1) бөліп – бөліп төлеу;

2) бір жолғы өтеу;

3) кезең сайын бір қалыпты өтеу.

г) тәуекелді деңгейі бойынша:

1) субстандартты;

2) стандартты;

3) күмәнді;

4) сенімді;

5) сенімсіз;

6) ұзартпалы.

ғ) ақылығы бойынша:

1) қалыпты пайыздық мөлшерлемесі;

2) жоғары пайыздың мөлшерлемесі;

3) төмен пайыздың мөлшерлемесі;

4) пайызсыз.

д) салалық бағыты бойынша:

1) сауда - саттық кредиті;

2) өнеркәсіп кредиті;

3) ауыл шаруашылық кредиті;

4) құрылыс кредиті.

Несие – нарықтық экономиканың тірегі ретінде, экономикалық дамудың ажырамас элементін білдіреді. Оны барлық шарушылық субъектілерімен қатар, мемлекет те, үкімет те, сондай-ақ жеке азаматтар да пайдаланады. Несиенің пайда болуын өнімдерді өндіру сферасынан емес олардың айырбас сферасынан іздеу қажет. Тауар айырбастау – бұл тауардың бір қолдан екінші қолға өтуін білдіреді десе, шынымен де, осындай айырбас кезінде несиеге байланысты қатынас туындайды.[7]

Несие – бұл пайыз төлеу және қайтару шартында уақытша пайдалануға (қарызға) берілетін ссудалық капитал қозғалысы. Несие ақшалай капиталдың ссудалық капиталға өтуін қамтамасыз ете отырып, несие берушілер мен қарыз алушылар арасындағы несиелік қатынасты бейнелейді. Несиенің көмегімен заңды және жеке тұлғалардың уақытша бос қаражаттары мен табыстары экономикалық жүйе төңірегінде жинақтала отырып, уақытша және ақылы негізде пайдалануға берілетін ссудалық капиталға айналады.[3]

Несиенің экономикадағы орны мен рөлі, оның атқаратын қызметтерімен сипатталады. Жалпы несие экономикалық категория ретінде төмендегідей қызметтерді атқарады:

- қайта бөлу;
- айналыс шығындарын үнемдеу;
- айналыстағы нақты ақшалардың орнын уақытша алмастыру;
- капиталдың шоғырлануын жеделдету;
- ғылыми-техникалық прогрессті жеделдету.

Несиенің қайта бөлу қызметі, кез келген елдің ұлттық экономикасының толық қанды жұмыс жасауына өз үлесін қосады. Несиенің бұл қызметінің көмегімен экономикалық жүйенің бір саласынан екінші бір саласына капитал ағымы болады. Несиенің бұл қызметінің, қаржының қайта бөлу қызметінен айырмашылығы қаржының бөлінуі әкімшілік негізде жүргізілсе, ал салалар мен аймақтар арасындағы капитал ағымы несие арқылы, яғни ол нарықтық механизм негізінде жүзеге асырылады. Несиенің айналыс шығындарын үнемдеу қызметінің іс жүзіне асуы несиенің экономикалық мәнінен туындайды. Шаруашылық субъектілеріндегі ақшалай қаражаттардың түсуі мен жұмсалуды арасындағы уақытша болатын алшақтық кей жағдайларда қаржылай ресурстарға деген қажеттілікті туындатады. Міне, сондықтан да мұндай жағдайларда қарыз алушылардың барлық категориялары өздерінің меншікті қаражатқа деген жетіспеушіліктің орнын толтыру үшін несие пайдаланады. Бұл дегеніміз капитал айналымын қамтамасыз етіп қана қоймай, айналыс шығындарын үнемдеуге де мүмкіндік жасайды.[8]

Несие бөлімінің ақпараттық жүйесін құрудағы программалық өнім келесі бөлімдерден тұрады:

а) кредит 1.2 программасы – бұл программа несиелік міндеткерлікті төлеуді есептеуге арналған. Программаны организациялармен қатар жеке адамдар кредит төлемдерін тез есептеуге қолданады. Программаға енгізілетін мәліметтер: несие суммасы, жылдық пайыз, ай бойынша есептелген несие төлеу мерзімі, алғашқы төлем.

Программаның нәтижесінде әр ай сайын төленетін сумма, пайызбен қосылған сумма, төленетін уақыттар шығарылады. Есептеме ай бойынша не күн бойынша есептеледі. Соңғы бағанда пайызбен есептелгендегі нәтижелік сумма көрсетіледі. Сонымен қатар программаға төленген сумманы курсор арқылы енгізуге болады, сонда программа сол сумма бойынша қайта есептеме жүргізеді. Алынған нәтижені программаның архивына сақтауға болады. Нәтижелік мәліметті баспаға шығаруға болады.

ә) AS-Retail программасы – ол жеке беттің қызмет көрсетуінің қазіргі арнаулы банктік жүйесі. Жүйенің қайталанбас ерекшелігі барлық клиенттер, шоттар мен карталар бойынша орталықтанған мәліметтер базасын енгізу мүмкіндігі болып табылады, ол банк қызметі нарығындағы жұмысты ұйымдастырудың жоғары тиімділігін қамтамасыз етеді:

- банктің эквайрингтік желісін дамыту;
- жалақы жобаларын ұйымдастыру.

Негізгі артықшылықтар:

а) модульдік. AS-Retail жүйесі тәуелсіз модульдік кешенін ұсынады, ол оның мүмкіндігін банктің бизнес- процесінің өзгерісіне лайық өсіруге мүмкіндік береді.

ә) қолжетімділік. AS-Retail банкінің клиенттерін тәулікте 24 сағат, жылда 365 күн қызмет көрсетеді. Жүйенің баяулауы кезінде жабдықтың апатты істен шығуынан аулақ болу үшін, AS-Retail программалық компоненттің бөлек-бөлек компьютерлерде қосарландыруының мүмкіндігін көздейді. Бұл ретте олардан бірінің істен шығуы жүйенің қолжетімділігіне зиян келтірмейді, тек оның жалпы өнімділігін төмендетеді.

б) жүйенің жеке комплектін нақты банктің ағымдағы мақсаттарына лайықтау. Жүйені кірістірмес бұрын сол банктің қызметкерлері мен Asoft компаниясының мамандары қосылып оның ең маңызды есептерін анықтап алу қажет болады, нәтижеде банктің негізгі мәселелерін қанағаттандыратын және оның дамуын көздейтін тиімді шешім қабылданады.

в) жүйедегі амалдар мен әрекеттердің кешенді мониторингі. Жүйеде қолданушылардың әрекеті мен жеке шоттардың, штраф тарифтерінің және комиссияның, пайыздық жүктеменің өзгерістері толық бейнеленеді. Жүйе жұмысы кезіндегі барлық құжаттардың архиві жүргізіледі.

г) масштабталу. Жүйенің ақпараттық негізін орталықтанған мәліметтер қоры құрайды, ол аз уақыт ішінде заманауи телекоммуникациялық құралдар көмегімен банктің филиалдар желісін құрауға мүмкіндік береді. Масштабталу дегеніміз жүйені өнімділігі төмен аппараттарда қолдануға және қажет болғанда мәліметтер көлемі мен қолданушылар саны мың есе өскен жағдайда да қолдануға болады деген сөз.

ғ) көпвалюталық – несие берудің кез келген валюта түрлерінде алу мүмкіндігі;

д) пайызды жүктеудің дамыған сұлбасы;

е) басқа жүйелермен бірлесе жұмыс жасауға қабілеті – бір жүйеден екінші жүйеге автоматты түрде өте алу қабілеті;

ж) ақпаратты қорғаудың жоғары деңгейі – бұл ақпараттық жүйе қолданушысының тексеру ықтималдығын ойластыру, вирустардың жайылып кетпеуін қадағалау.[3]

1.3 Банктің ақпараттық технологияларына шолу

Қазіргі заманғы ақпараттық технологиялар банк құрылымдарының қызметін үйлестіріп, банкаралық байланыстарды кеңейтіп, бірнеше мемлекеттің қаржы нарығына бір мезгілде әрекет ету мүмкіншілігін ұсына алады.

Банктің пайда болуы мен дамуы тауар-ақша қатынастарының дамуымен тығыз байланысты. Банктер кәсіпорындардың, фирмалардың, компаниялардың, сонымен қатар тұрғындардың уақытша пайдаланылмай бос тұрған қаражаттарын жинақтап, қажет етушілерге өсіммен уақытша кредит беру арқылы қаржы мұқтажығын өтейді. Банктің ақпараттық және өзге де технологияларының ақпараттық қосымшасын құру стратегиясы міндеттерді жедел түрде шешуге әсерін тигізуде. Клиенттерге қызмет көрсетуді автоматтандырудың басым бағыттары банктердің бәсекелестік позициясына әсер етеді. Олар: қосымша шығындарды кеміту және қызмет көрсету сапасын арттыру.[4]

Несие өнімнің өндірілуі, өткізілуі мен тұтынылуы үрдісіне және ақша айналымы саласына әсерін тигізеді, яғни өнімді өндіру мен өткізу үрдістерінің кедергісіз болуына ықпал етеді, өндірістің маусымдылығына орай туындаған қаражатқа деген қажеттілікті уақытша қанағаттандыруда, өндірісті кеңейтуде пайдалану үлкен рөл атқарады.

2 Жобалау бөлімі

2.1 Мәліметтер қорын басқару жүйесін таңдауды негіздеу

Мәліметтер қоры – белгілі бір арнаулы ауқымға қатысты мәліметтерді баяндау, сақтау мен амал-тәсілдермен ұқсата білудің жалпы принциптерін қарастыратын белгілі бір ережелер бойынша ұйымдастырылған мәліметтердің жиынтығы. Адамның қатысуы ықтимал, автоматтық құралдармен өңдеуге жарамды түрде ұсынылған ақпарат мәліметтер ретінде түсініледі.

МҚБЖ – жаңа қор құрылымын құруға арналған программаның құралдар кешені, оның құрамын толықтыру, құрамында барды өңдеу және ақпаратпен көрсетушілік (визуализация). МҚБЖ-ның тағы бір функциясы – мәліметтерді басқару. Мәліметтерді басқару ретінде, әдетте, мәліметтерге рұқсат етілмей қол жетуден қорғау, мәліметтермен жұмыс режимін көп мәрте пайдалануды қолдау және мәліметтердің тұтастығы мен үйлесімділігін қамтамасыз ету түсініледі.[14]

Объект деп – мағлұматтары мәліметтер қорында сақталатын ақпараттық жүйенің элементтері аталады.

2.1-кесте – Ақпараттық жүйедегі мәліметтер қорының сипаттамасы

Элементтер	Сипаттама
Ақпараттық жүйе	Барлық ақпараттар МҚ сақталады
МҚ типі	Реляциондық
Жүйе классификациясы	Фасеттік, классификацияның иерархиялық сұлбасы
Кодтау жүйесі	Сәйкестендіру тәсілі, кодтаудың реттік жүйесі
Қауіпсіздік	Үзілген ақпараттардың қайта қалпына келуі, күнделікті құпия сөздің ауыстырылуы, зақымданған бөлімдерді толық тазалау администратордың нұсқауымен толық қадағаланады

2.1.1 Мәліметтер модельдері

Мәліметтер қорының моделі дегенде әдетте қордың құрылымы және онымен жұмыс істеу әдісі түсініледі. Модельдердің бірнеше түрлері бар және жаңа тұжырымдама ұсынатын, нақты талаптарға жауап беретін жаңа түрлері үнемі шығып, дамып тұрады.

Иерархиялық модель. Иерархиялық модель графпен байланысқан ағаш түрінде көрсетіледі. Ағаштың ұштары әр түрлі деңгейлерде орналасады. Модельдің ережелері:

- байланыс түрлері функциональды болуы керек (1:1, 1:К, К:1);
- ағаш айналым бола алмайтын бағытталмаған граф түрінде болады;
- ағаш бір «түбірлік» жазба түрінен және әрқайсысы өзіндік ағаш түрінде көрінетін бұтақшалардан тұрады;
- ағашта әр түйін тек бір ғана аталық түйінмен байланысады;

– ағаш бұтақтары «бастапқы-пайда болған» байланысқа сәйкес келеді;
– әр пайда болған түйінге қатынас жасау тікелей бастапқы түйін арқылы орындалады;

– кез- келген түйінге қатынас жасау үшін ағаштың түбірінен басталатын тек жалғыз ғана сызықтық-иерархиялық жол болады.

Иерархиялық модельдің анықталынған кемшіліктері бар:

– иерархиялық жүйеде ақпаратты іздеу тек жоғарыдан төменге қарай ғана орындалады, яғни нақты аталық сегмент данасы арқылы бүкіл ұрпақтық сегмент көшірмесін табуға болады. Ал кері жағдайда іздестіру қиындалған, тіпті көп жағдайда мүмкін емес;

– логикалық деңгейде мәліметтердің қайталануы болады;

– иерархиялық модельде ешбір ұрпақ аталық түйінсіз сақталмайды, сондықтан егер аталық түйін өшірілсе, онымен байланысқан ұрпақ түйіндер де өшіп кетеді.

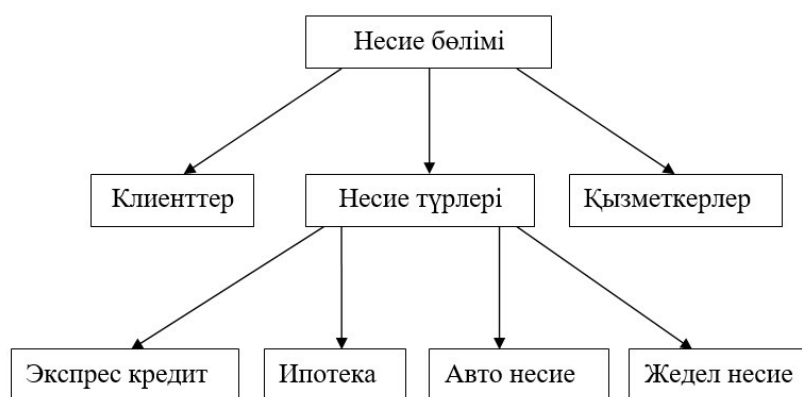
Реляциялық модель. Реляциялық модельде байланыс екі өлшемді массив түрінде көрсетіледі. Әр жол белгілі бір затқа немесе оның бөлігіне қатысты мәліметтен тұрады. Ал баған оның қандай да бір қасиетін сипаттайды. Кестеге келесі талаптар қойылады:

- ұяшықтағы мән бірімәнді болуы керек;
- бағандағы барлық жазбалар бір типті болуы керек;
- әр бағанның жеке қайталанбас атауы болуы керек;
- екі бірдей жол атауы болмауы керек;
- жол реттілігі сақталуы шарт емес.

Реляциялық мәліметтер қорының пайда болуы ақпаратты іздеудің өте жылдам алгоритмдерінің пайда болуының алғы шарты болды.

Желілік модель. Желілік модель иерархиялық модельдің кеңейтілген түрі болып табылады. Құрушылардың мақсаты – иерархиялық модельдің кемшіліктерін жөндеу және M:N байланыс түрін қолдануды қолдайтын модель түрін құру.[14]

Желілік модель де мәліметтерді граф түрінде ұсынуға негізделген. Бірақ иерархиялық модельде ұрпақ түйін тек бір ғана аталық түйінді иемдене алатын болса, желілік модельде көптеген аталық түйінді иемдене алады.



2.1-сурет – «Несие бөлімі» байланысы

2.1.2 Мәліметтер қорына қойылатын талаптар

Жақсы жобаланған мәліметтер қоры:

– мәліметтер қорының құрамына қатысты қолданушының барлық талаптарын қанағаттандырады. Қорды жобалау алдында қолданушылар талаптары туралы зерттеулер жүргізу керек.

– мәліметтердің толықтығына, дәлдігіне және қайшылықсыз екендігіне кепілдік береді. Кестені жобалағанда оның қасиеттерін анықтап алу қажет және қолданушының қате мәліметтер енгізбеуі үшін шектеулер қою қажет. Кестеге мәлімет енгізбес бұрын мәліметтер моделінің ережелерін жүзеге асырып, сонымен қатар ақпараттың бүтіндігіне кепілдік беру керек.

– ақпараттың құрылымын қабылдауды жеңіл қамтамасыз етеді. Сапалы қордың құрылуы базаға түсінуге жеңіл сұраныстар жасауға мүмкіндік береді және дұрыс емес мәліметтердің ену ықтималдылығы азаяды.

– қолданушылардың мәліметтер қорының өнімділігіне қойылатын талаптарын қанағаттандырады. Көлемді ақпаратпен жұмыс кезінде өнімділіктің сақталуы маңызды болып табылады.

Мәліметтер қорында модельдеу керек болатын дүниедегі нақты объектілерді талдау. Осы объектілерден мәнді және олардың қасиеттерін құрастыру.

Байланыс түрлері:

- «бірдің бірге» байланысы;
- «бірдің көпке» байланысы;
- «көптің көпке» байланысы.[21]

«Бірдің бірге» байланысы кезінде бастапқы кілт сыртқы кілттің де орнында болып келеді. Бұл байланыс кестедегі мәліметтерді белгілері бойынша екі кестеге бөліп қолданғанда пайдаланылады.

«Бірдің көпке» байланысы көп жағдайда мәндер арасындағы нақты байланысты сипаттайды. Мұнда «сырқы кілт – бастапқы кілт» жұбы бойынша байланыс орнатылады. Бастапқы кілтке мәндер, ал сыртқы кілтке олардың кодтары байланыстырылады.

«Көптің көпке» байланысын реляциялық мәліметтер қоры қолдамайды. Тек оны жаңа бір кесте қосу арқылы ғана орындауға болады. Ол кестеде тек сыртқы кілттер қолданылады.

Қажеттілік жағдайда ақпараттың құпиялылығын сақтау және сенімділігін жоспарлау. Бұл қадамда келесі сұрақтарға жауап беру керек:

- мәліметтер қорын қолдануға кімнің және қандай мүмкіндігі бар;
- мәліметтерді енгізуге және өшіруге кімнің құқығы бар;
- қолжетімділік құқықтарына айырмашылық енгізу керек пе;
- қалай ақпаратты қорғау тәртібін қамтамасыз етуге болады.

2.2- суретте «Банктің несие бөлімі» тұжырымдамалық кестесі көрсетілген.

2.1.3 Мәліметтер қорын басқару жүйесін құруға арналған программалық өнімдерге шолу

MySQL, проект қайнарын GNU General Public License (GPL) және әр түрлі жеке меншік лицензияларын да шығарды. MySQL-ды кезінде MySQL AP атты Шведттық фирма демеу еткен, ал қазір оны толығымен Oracle корпорациясы сатып алды.

Толық функционалды мәліметтер қорын қажет ететін, қайнары тегін проекттер жиі MySQL-ды қолданады. Серверді коммерциялық проекттерде қолдану үшін, қосымша функциялармен ұсынылатын бірнеше нұсқалары бар. MySQL-ді қолданатын бағдарламаларға TYPO3, Joomla, WordPress, phpBB, MyBB, Drupal және де басқа LAMP бағдарламаларын жатқызуға болады. MySQL сонымен қатар Google, Wikipedia, Facebook және Twitter сияқты дүниежүзілік және кең ауқымды web-бағдарламаларында қолданылады.[14]

Программалау тілі. MySQL серверімен жұмыс істеуге және оны басқаруға арналған басқа да клиенттік бағдарламалар бар. Кейбір кең қолданылатындары төменде алфавит бойынша тізімделген:

Adminer – CSS стилін қолдануға болатын, бір ғана PHP скрипттен және бірнеше мәліметтер қорымен жұмыс істей алатын, тегін клиенттік бағдарлама.

DaDaBIK – мәліметтер қорына CRUD операцияларын web арқылы жасауға мүмкіндік беретін, қайнары ашық және тегін PHP тілінде жасалған клиенттік бағдарлама.

DBEdit – MySQL және басқа да мәліметтер қорын басқаруға арналған тегін программа.

dbForge GUI Tools – схемаларды және мәліметтерді салыстыруға және синхрондауға, тапсырыстарды құруға арналған жеке модульдерден тұратын программа.

HeidiSQL – Windows платформасына арналған толық функционалды клиенттік программа. Басқару жүйесіне жергілікті және алыстағы компьютердегі мәліметтер қорын, кестелерді, кесте бағаналарын және жеке мәлімет жолын өңдеу операциялары жатады. Сонымен қатар уақыт/күн және мәндік типті жолдарды өзгертуге арналған клиенттік функциялары бар.[10]

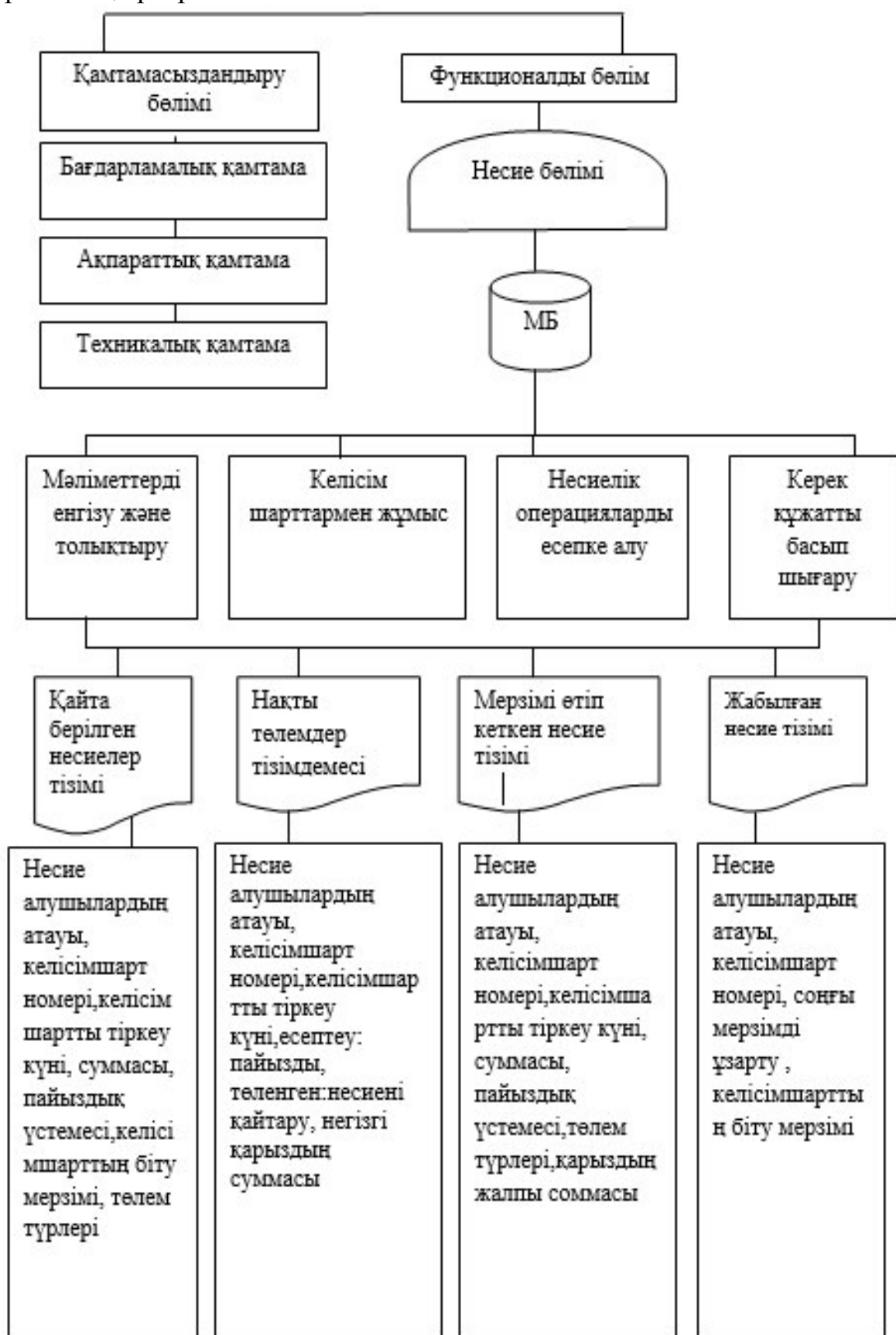
Navicat – Windows, Macintosh және Linux операциялық жүйелеріне арналған мәліметтер қорын басқаруға арналған графикалық программалардан тұратын программалар жиынтығы.

OpenOffice.org – OpenOffice.org Base, MySQL мәліметтер қорын басқара алады. (Ол үшін тегін және қайнары ашық OpenOffice.org suite программасын толығымен орнату керек).

phpMyAdmin – тегін web арқылы қолдануға арналған және көбінесе web-hosting- тарда кең қолданылатын web- бағдарлама. Ол PHP- да жасалғандықтан, LAMP, MAMP және WAMP программаларының ішінде кең қолданылады.

SQLBuddy – PHP-да жасалған тегін web-программа.

Sequel Pro – Mac OS X операциялық жүйесіне арналған тегін және қайнары ашық программа-клиент.



2.2-сурет – Кестелер арасындағы байланыс

SQLYog – MySQL-ді басқаруға арналған тегін немесе ақылы программа. Бұлардың бірнеше нұсқалары бар: Community, Ultimate, Enterprise және Professional. Қоғамдастық нұсқасы (Community Edition) тегін және қайнары ашық программа.[14]

Toad for MySQL – Quest Software компаниясының MySQL-ді басқаруға арналған тегін клиенттік программасы.

Бұлардан басқа ақылы программаларға кіретіндер dbForge Studio for MySQL, Epictetus, Oracle SQL Developer, SchemaBank, SQLPro SQL Client, Toad Data Modeler.

Қолданылуы. Веб-программаларында MySQL кең қолданылады және кең тараған LAMP программаларының ішінде басты компонент рөлін атқарады. LAMP сөзі – «Linux, Apache, MySQL, Perl/PHP/Python» сөйлемінің акронимі.

MySQL – қайнары тегін программалар мен Интернеттегі танымал Flickr, Nokia.com, YouTube, Wikipedia, Google, Facebook және Twitter сайттарында қолданылады.

Мәліметтер қорымен жұмыс істеу үшін, көптеген программалау тілдеріне кітапханалар жасалынған.

Мысалы, C# және VB.NET үшін MySQLConnector/Net, ал Java үшін JDBC драйвері. Бұнымен қатар ODBC интерфейсін қолдайтын ASP және ColdFusion сияқты тілдер үшін MyODBC кітапханалары бар. Ал URL арқылы кез-келген web-клиенттің MySQL серверімен араласуға мүмкіндік беретін HTSQL технологиясы, MySQL серверімен бірге келеді.[21]

Ресми программалар. Графикалық интерфейсте MySQL мәліметтер қорын басқаруға және олардың құрылымын жасау үшін, MySQL AB фирмасы MySQL Workbench атты программасын шығарды. Ол осыған дейін болған MySQL GUI Tools программасының орнына келді. MySQL Workbench қолданушыларға келесідей мүмкіншіліктер ұсынады:

- мәліметтер қорын дизайндау және модельдеу
- sql программалау – MySQL Query Browser программасының орнына
- мәліметтер қорын басқару – MySQL Administrator-дың орнына

MySQL Workbench екі нұсқада келеді. Олар, қайнары тегін қоғамдастық нұсқасы (Community Edition) және қосымша мүмкіншіліктері бар стандартты нұсқасы (Standard Edition).

Мәліметтермен жұмыс істеу функцияларын келесі стандартты топтарға бөлуге болады:

- кестелермен жұмыс;
- бөлек жазбалармен жұмыс;
- есеп дайындау;
- сұраныстармен жұмыс;
- желімен жұмыс;
- макростармен жұмыс;
- жұмыстық графикамен жұмыс.

Кестелермен жұмыс функциялары кестелерді құруға, көшіруге, атын қоюға, өшіруге мүмкіндік береді. Біріктіру және алып тастау секілді реляциялық алгебра операцияларын да орындауға болады.

2.2 Программа құрудың интегралдық ортасын сипаттау

Кейінгі кездегі дамып келе жатқан визуалды программалау программаның текстін қолдан жазғанның орнына, программалау пакетінің ішіндегі дайындықтарды қолдануды қолдайды, ол дайындықтардың тек қана кейбір параметрлерін өзгертіп немесе модификациалау қажет.

Программамен жұмыс істегенде, екі кезеңге бөлуге болады. Бірінші кезең – жобалау кезеңі. Мұнда программа бөлек құрама бөліктерден құралады да, оған қажетті параметрлермен мінездемелер белгіленеді. Дәл осы кезеңде, программа жұмыс атқарғанға дейінгі нәтижелерін көруге болады және визуалды программалаудың әдістері қолданылады. Екінші кезең – программаның жұмыс атқару кезеңі, және аралық кезеңді бөліп қарастыруға болады, бұл тексеру кезеңі, программаның қателіктерін, дұрыс жұмыс атқарылуын қарастырады.[15]

Жобалау кезеңінде бірнеше файлдар құрастырылып қолданылады. Негізгі бөлігі – жоба. Әдетте бұл бөлік бүкіл программаның жинақтаушы бөлігі болып келеді, бірақ керек жағдайда өзіне қажетті өзгертулер енгізуге болады.

Программаның жұмыс істеу кезеңінде әртүрлі динамикалық кітапханалы файлдарды (DLL – DynamicLinkLibrary) қолдануы мүмкін, кеңейтілуі *.dll. Программаның жұмыс істеу барысында әртүрлі тексттік немесе тексттік емес файлдарды қолдануы мүмкін, мұнда қажетті мәліметтер орналасады.

C# – ол бірнеше маңызды технологиялар комбинациясы:

- машиналық кодтағы жоғарғы өнімділік компиляторы;
- компоненттің объектілі-бағытталған моделі;
- программалық көшірмеден визуалды (жылдамдатылған) қосымшаларды құру;
- мәліметтер қорын құруға арналған құрал.

C# программалау тілі кез-келген қосымшаны дайындауға болатын жылдамдығы тез, қуатты тіл.[13]

1994 жылы Borland фирмасы C# атты қосымшаларды жылдам жетілдіруге арналған аспапты құралдарды жетілдіруді бастады. Оның негізіне Object Pascal тілі болды.

Delphi тілінің пайда болуының алдында, Borland фирмасы Pascal тілін конструктивті өзгерістерге ұшыратты. Мысалы: Object Pascal тіліне класстар енгізілгеннен кейін объектілі-бағытталған тілдерінің тобына ауыстырылды. C# құрылған кезде жаңа кілтті сөздер (published, property және т.б.) енгізілді. Бірте-бірте кейінгі жылдары Delphi-дің 2,3,4,5,6-шы нұсқалары пайда болды. 5-ші мен 6-шы нұсқаларының бір-бірінен айырмашылығы жоқ деуге болады,

екеуі де Windows 32 операциялық жүйесінің негізін пайдаланған. С# 6-да интернет үшін қосымшаларды дайындауға мүмкіндігі кеңейтілген және берілгендер қорымен жұмыс істеуде өзгешеліктері енгізілген.[14].

С# көмегімен Windows-қа арналған программалық қамсыздандыру өндірісі тез және көп күш алғызбайды. Сонымен бірге консольдік және Win 32 GUI қосымшаларын құра аласыз. Соңғы үлгіде компиляцияланған бағдарламалау тілімен (Object Pascal) кездесесіз. Ол пайдаланушы интерфейсін (оған әртүрлі мәзірлерді, сұхбаттық терезе, негізгі терезе және т.б. жатады) жобалаған кезде drag-and-drop механикасымен қолдана аласыз. С#-тың интеграцияланған жетілдіру ортасы үш бөлімнен тұрады. Олар:

- негізгі саймандар тақтасы;
- объект инспекторы;
- форма мен код редакторы.

Жоғарғы жақта негізгі терезе орналастырылған. С#-тың саймандар тақтасындағы бір тетікті басқан кезде проектінің ашылу, сақталуы немесе компиляциялау тапсырмаларын орындауға мүмкіндік береді. Саймандар тақтасы формаға орналастыра алатын көп жиынтықты компоненттерден тұрады (компоненттерге мәтіндік белгілер, редакторлау алаңы, тізім, тетік және т.б. кіреді). Оларды тиімді пайдалану үшін компоненттер топтарға бөлінген. Компоненттерді формаға орналастыру үшін оны бір рет таңдап форма бетіне апару керек.[16]

Негізгі терезе. Негізгі Терезе С#-тың «басқару орталығы» болып табылады. Ол негізгі мәзірді, құралдар тақтасы және құрауыштар палитрасы қосымшаларын құру процесін бақылайтын негізгі үш бөліктен тұрады.

Негізгі мәзір. Негізгі мәзір проектімен жұмыс кезінде қажетті барлық құралдарды шығаруға мүмкіндік береді. Мәзір бөлімдері және олармен байланысқан функциялар:

File (Файл) – файлмен жұмыс жасайтын командалар жиынтығын құрайды, оларды проектіге қосуға, шаблондардың көмегімен жаңа файл құруға, өшіруге және басып шығаруға мүмкіндік береді. Сонымен осы бетте жаңа қалып жасауға арналған командалар және шығу командасы орналасқан. С# 5-те Frame (фрейм) жаңа элементін құруға арналған команда қосылған.

Edit (Түзету) – атында көрсетілгендей, мұнда мәтін түзететін, өшіретін және буферге салынатын немесе керісінше буферден қоятын және түзетілген амалдарды кері қайтаратын командалар орналасқан.

Басқару командалары және де өзінің өңделген ActiveX интерфейсіне процедуралар немесе функциялар жаңа қасиет қосатын команда енгізілген. Бөлімнің бір опциясының орналасуын өзгертуге рұқсаттамауға да болады.

Search (Іздеу) – мәтінді іздеу және алмастыру амалдарын орындайтын командалар тобы.[15]

View (Қарау) – проектілерді басқаруда көп қолданылатын құралдарды шақыратын командалар берілген. Олар: Project Manager (Проект менеджері), Translation Manager (DLL тілінің менеджері), Object Inspector (Объектілердің инспекторы), To-Do-List (Есептер тізімі), Alignment Palette (Компоненттерді

теңестіретін терезе), Browser (Зерттеу), Code Explorer (Программа бойынша жол сілтеуіш), Component List (Компоненттер тізімі), Window List (Терезелер тізімі), Type Library (Кітапхана типтері).

Project – проектиге модульдерді қосуға және өшіруге, кітапханалар типін импорттауға, проектіні объектілер архивіне (Repository) сақтауға, проект мәтіндік терезесін іске қосуға, проектiлердi топқа қосуға мүмкіндік беретін тетік.

Run (Қосу) – өңделген программаларды жіберуге, оған параметр қатарын беруге мүмкіндік беретін тетік.

Component (Компонент) – мұнда компоненттерді орнатуға, ActiveX компоненттерін импорттауға, жаңа компоненттер және компоненттер шаблонын құруға, пакеттерді түзету және компоненттер политрасын баптауға арналған командалар бар.[14]

Database (Деректер қоры) – деректер қорымен жұмыс істейтін командалар бар, олар: SQL Explorer (деректер қорын зерттеу), SQL Monitor (SQL Manitor) және Form Wizard (Форма жасау).

Tools (Құралдар) – C#-тың жұмыс ортасының қасиеттерін орнатуға, проектiлер орнатуға, проектiлер архивіне бастау жүргізу, проектімен жұмыс жасауға, қосымша утилиттерді қосуға және өшіруге мүмкіндік береді. C# 5-те осы бөлімде ресурстар қатары ұлттық алфавитпен сақталған тағы бір архив – Translaron Repository-ді ашуға болатын команда қосылған.

Help (Көмек) – C#-тың анықтамалық жүйесін шақыруға мүмкіндік береді.

C#-тың жұмыс ортасы. C#-тың интеграцияланған жетілдіру ортасының орталық жерінде жұмыс ортасы орналасқан. Онда форма конструкторы көрсетілген. Ол қосымшалардың формасын құруға арналған. C#-та форма программаның терезесі ретінде белгіленген. Жұмыс істеу барысында форма үстінде форма конструкторы арқылы компоненттерді орнатуға, олардың орнын алмастыруға және өлшемін өзгертуге болады. Форма конструкторымен код редакторы тығыз байланысты. Оның көмегімен қосымша құрған кезде, программалық кодты енгізуге болады. Барлық алда өткен құрылғылар – объект инспекторы, форма конструкторы, код редакторы мен саймандар тақтасы бір-бірімен қолданбалы программаларды құру барысында, өте тығыз байланысады.

C# ортасында жиі кездесетін оқиғалар:

- On Click тышқан тетігін бір рет басу;
- On Dbl Click тышқан тетігін екі рет басу;
- On Key Down клавишті басу;
- On Create форманы екі рет шерту.[16]

2.2.1 Мәліметтер қорымен жұмыс істеу

Басты модульдің орындалатын бөлігі begin, end арасында орналасады. Орындалатын бөлік нұсқауы қосымшаны инициализациялау және экранға

шығаруды қамтамасыз етеді. Әрбір программада басты модульден басқа да, кем дегенде бір модуль формалары болады, ол үрдістің орындалуын қамтамасыз етеді. Delphi-де әр форманың өз модулі болады.

Модуль unit сөзімен басталады, кейін модуль аты жазылады. Осы ат қосымшаның басты модульдегі uses нұсқауында жазылған, оның тексті екінші листингте жазылған. Модуль 3 бөліктен тұрады: интерфейс, іске асыру, инициализация.[10]

Интерфейс бөлігі (interface сөзімен басталады) компиляторға, басқа программа модульдеріне модульдің қай бөлігі сай келетінін көрсетеді. Бұл бөлікте (uses сөзінен кейін) модульдің пайдаланатын кітапханалық модульдері және type сөзінен кейін форманың сипаттамасы жазылады.

2.3 SQL тілі – мәліметтер қорының негізгі тілі

SQL – МҚБЖ- мен басқарылатын ерікті реляциялық мәліметтер қорында сақталатын ақпаратты іздеу, шығару, түрлендіруге қолданылатын программалаудың стандартты тілі. SQL күшті тіл болу себебінен оны өте танымал Microsoft Access, Oracle, MySQL МҚБЖ қолданады.

SQL – бұл:

- программалау тілі;
- үйренуге жеңіл;
- процедуралы емес;
- интерактивті немесе кірістірілген;
- стандартталған.[17]

Программалау тілі. SQL – оның көмегімен компьютерге программа деп аталатын нұсқау беруге болатын құрал, формальді тілдердің бірі. Мәліметтер қорының программалық қамтамасы SQL тілінде жазылған программаны орындайды. Яғни, МҚБЖ оған жіберілген сұранысты және сұраныс мәліметтерін оқуды, түзетуді орындайды және жұмыс нәтижесін көрсетеді.

Басқа да формальді тілдер секілді SQL-дің де өзіндік синтаксисі мен семантикасы бар. Синтаксис дегеніміз – программада қолданылатын сөздер мен символдар, сонымен қатар ережелер. Семантика кез-келген синтаксистік жағынан дұрыс команданың нақты мағынасын анықтауға көмектеседі.[10]

Үйренуге жеңіл. Программалаудың басқа тілдеріне қарағанда SQL үйренуге жеңіл әрі тез тіл. SQL формальді емес тілдердің командалары сияқты оқылады. Мысалы кез-келген программалауды жаңадан үйреніп жатқан адам `SELECT au_fname, au_lname FROM authors ORDER BY au_lname` SQL командасын оңай түсінеді. Осы адамға осы командамен эквивалентті болып келетін C немесе Perl программаларын көрсетсеңіз ол мүлдем түсінбеуі мүмкін.[10]

Процедуралы емес. Егер сіз C немесе Perl тілдерінде программа жазып жүрген болсаңыз, онда процедуралық тіл дегенді білесіз. Бұл процедуралық тілдерге программист өзіне керек нәтиже алу үшін компьютер орындауы керек болған қадамдарды жазу керек болады. Ал SQL процедуралық емес

тілге жатады, оны декларативті деп те атайды. Сондықтан да мұнда IF-THEN-ELSE, WHILE басқару операторлары және GOTO өту операторлары қолданылмайды. Процедуралық және процедуралық емес тілдерді салыстыру үшін бір-біріне эквивалентті, бірдей тапсырманы орындайтын екі программаны алайық. Бірінші программа авторлардың есімдерін шығару үшін Do Until қолданған. Бұл жағдайда программа 20 қатарлы кодпен жазылған. Ал SQL тек бір ғана команда қолданып нәтижені ала алды. Бұл тек қарапайым ғана сұраныс. Егер оған сұрыптау, сүзгілеу, біріктіру операцияларын қосатын болсақ Microsoft Access Visual Basic программасының коды 100 қатарға жетіп қалады.[17]

Интерактивті немесе кірістірілген. Егер процедуралық тілде жазылған қандай да бір программаға SQL командаларын кіргізетін болсақ, ол кіргізілген SQL, ал процедуралық тіл базалық деп аталады. Яғни, C, Java, COBOL, Perl, PHP, Python тілдеріне SQL командаларын кірістіруге болады.

Егер команданы жазғанда ол дереу нәтижені шығаратын болса, онда ол интерактивті не динамикалық SQL деп аталады.

2.2-кесте – Программаларды салыстыру

Microsoft Access Visual Basic	SQL
<pre> Sub GetAuthorNames() Dim db As Database Dim rs As Recordset Dim i As Integer Dim au_names() As String Set db = CurrentDb() Set rs = db.OpenRecordset("authors") rs.MoveLast ReDim au_names(rs.RecordCount - 1, 1) With rs.MoveFirst i = 0 Do Until .EOF au_names(i,0) = ![au_fname] au_names(i,1) = ![au_lname] i = i + 1 .MoveNext Loop End Withrs.Close db.CloseEnd SUB </pre>	<pre> SELECT au_fname, au_lname FROM authors; </pre>

SQL-дің атқаратын қызметтері көп:

– SQL интерактивті сұраныстар тілі болып табылады. Қолданушылар мәліметтерді таңдау және оларды экранға шығару мақсатымен интерактивті программаларға SQL командаларын енгізеді. Бұл арнайы сұраныстарды орындаудың тиімді тәсілі.

– SQL – мәліметтер қорын программалау тілі. Мәліметтер қорына қатынас жасау үшін программисттер өз программаларына SQL командаларын

қояды. Бұл әдіс қолданушылар тарапынан жазылған программалармен қатар қызметтік программаларда да қолданылады.

– SQL мәліметтер қорын әкімшілік ету тілі болып табылады. Сервердегі не жұмыс станциясындағы мәліметтер қорының администраторы мәліметтер қорының құрамын анықтау және мәліметтерге қатынас жасауды басқару үшін SQL-ді қолданады.

– SQL клиент/ сервер қосымшаларын құру кезінде қолданылады. Дербес компьютер программалары үшін SQL мәліметтер қоры сервері мен жергілікті желі арасындағы байланысты орнату құралы ретінде қолданылады.

– SQL Internet ортасындағы мәліметтерге қатынас жасау қызметін атқарады. Web-серверлерде SQL корпоративтік мәліметтер қорына қатынас жасаудың стандартты тілі ретінде қолданылады.

– SQL үлестірілген мәліметтер қоры тілі болып саналады. Үлестірілген мәліметтер қорын басқару жүйелерінде өзара әсерлесуші есептеуіш жүйелеріне мәліметтерді үлестіруге көмектеседі. Әр жүйенің программалық қамтамасы SQL көмегімен басқа жүйелермен оларға сұраныс жіберу арқылы байланысады.

– SQL – мәліметтер қорының шлюздерінің тілі. Әр түрлі МҚБЖ-нен тұратын есептеу желілерінде SQL бір МҚБЖ-ін екінші МҚБЖ-мен байланыстыратын шлюздік программа ретінде жиі қолданылады.[10]

SQL-дің артықшылықтары. SQL – мәліметтерді басқарудың әмбебап программалық құралы және түсінуге жеңіл тіл. Оның жетістікке жетуіне келесі ерекшеліктері әсер етті:

- нақты бір МҚБЖ-нен тәуелсіздігі;
- платформа аралық тасымалдағыштық;
- стандарттың бар болуы;
- IBM компаниясының мақұлдауы;
- Microsoft компаниясының қолдауы;
- реляциялық негіз;
- жоғары деңгейлі құрылым;
- арнайы интерактивті сұраныстарды орындау мүмкіндігі;
- мәліметтер қорына программалық қатынастың қамтамасыз етілуі;
- мәліметтерді әр түрлі көрсету мүмкіндігі;
- мәліметтер қорымен жұмыс жасауға арналған тіл ретіндегі

ТОЛЫҚТЫҒЫ;

- мәліметтерді динамикалық түрде анықтау мүмкіндігі;
- клиент/сервер архитектурасын қолдау;
- объектіге бағытталған технологияны қолдау;
- Internet ортасындағы мәліметтерге қатынас жасау;
- Java тілімен интеграция.

Жоғарыда келтірілген факторлар SQL-дің жұмыс станцияларындағы және ірі серверлердегі дербес компьютерлердегі мәліметтерді басқаруға арналған стандартты құрал болуының себептері болды.[17]

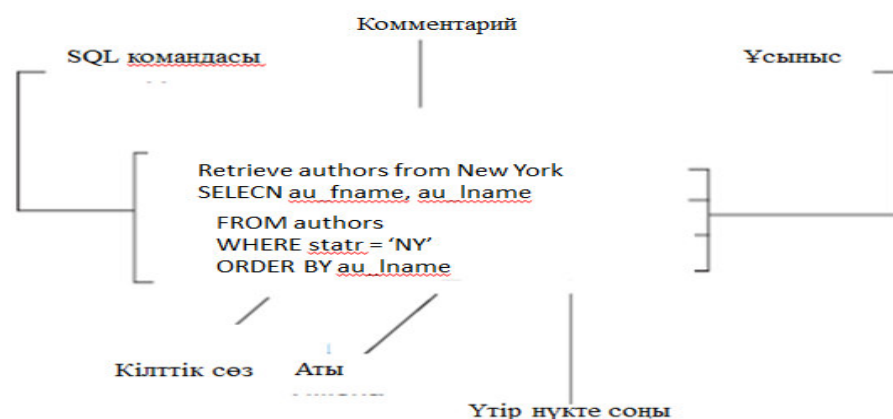
SQL синтаксисі. Түсіндірме – жеке жолға программаны түсіндіру үшін жазуға болатын жазылуы міндетті емес текст. Ол екі сызықшадан басталуы керек. МҚБЖ сызықшаны программа ретінде санамай өтіп кетеді.

SQL командасы. Кез-келген SQL командасы кілттік сөздерді еске салатын лексемалар жиынынан тұрады. Лексема дегеніміз – SQL тілінің бөлінбейтін, негізгі бөлігі. Лексемаға кілттік сөздер, операторлар, идентификаторлар, литералдар және басқа да символдар кіреді.

Сөйлем. SQL командасы кем дегенде бір сөйлемнен тұрады. Жалпы жағдайда ол анықталған рет бойынша жазылатын, қандай да бір кілттік сөзден басталатын SQL командасының фрагменті. Қарастырылып отырған мысалда олар төртеу – SELECT, FROM, WHERE, ORDER.

Кілттік сөздер. Кілттік сөз – SQL-де қолданылуы қатаң реттелген, анықталған мәні бар сөз.

Идентификаторлар. Идентификатор – мәліметтер қорындағы объекттердің, кестелердің, бағандардың аттары. Идентификатор кілттік сөз болмауы керек, өлшемі 128 таңбадан аспайды. Таңба дегеніміз кез-келген алфавит символы.[10]



2.3-сурет – SQL командасы

SQL – сөйлемнің еркін форматындағы тіл. Оның кез-келген командасы:
– төменгі және жоғарғы регистрде де жазыла береді;
– өзіңіз бөлмегенше бір қатарда жазыла береді;
– басқа командамен бір қатарда жазыла береді;
– экран не беттің көлбеу белгісінің кез-келген позициясынан бастала береді.

Дегенмен командаларды жазудың арнайы стилін қолдану тиімді. Себебі, оның орындалмауы келесі қателіктерге әкеліп соғуы мүмкін:

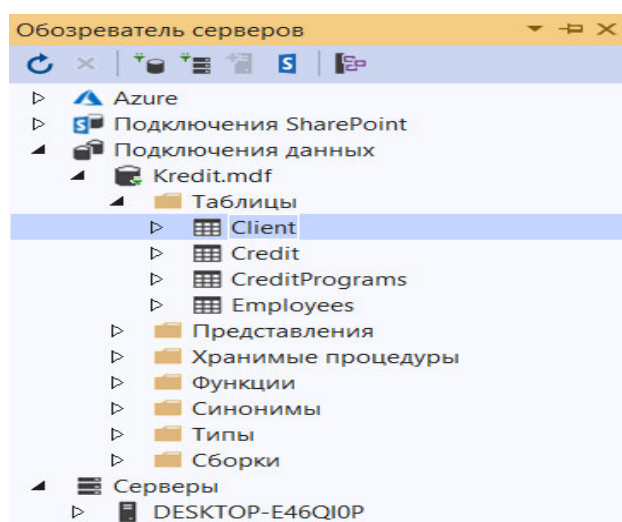
- қандай да бір команда немесе идентификаторды жазған кездегі қате орфография;
- команданың соңындағы үтірлі нүктенің қойылмай қалуы;
- команда денесіндегі сөйлемнің қате орналасуы;

- литералдар мен уақыт тұрақты шамаларының, тырнақшаларының қойылмауы;
- сандық тұрақты шамаларға тырнақшаның қойылып кетуі.

2.4 Мәліметтер қорының құрылымын сипаттау

Ақпараттық жүйені құру барысында MSSQL МҚБЖ, Visual Studio ортасын қолданып, C#-та орындадым. Мәліметтер қоры келесі кестелерден тұрады (2.4-сурет):

- «employees»;
- «client»;
- «credit»;
- «CreditProgramms»



2.4-сурет – Мәліметтер қорындағы кестелер

Бұл программаның негізгі тапсырмасы несие бөлімінің жалпы құрылым жұмысы болып отыр. Әр пункттің тиісті барлық кестелер «id» типі «сандық/ұзын бүтін» қатары коды бойынша байланысады. Әр кестелер байланысы «id» кілттік қатары болып табылады, осы арқылы барлық кестелермен байланысады.

«Employees» (2.5-сурет) :

- «id» қатары (типі «int»), AUTO_INCREMENT мәні автоматты түрде санын 1-ге арттырып отыратынды білдіреді;
- «Фамилия» қатары (типі «nchar»), мәні қызметкерлердің тегін білдіреді;
- «Имя» қатары (типі «nchar»), мәні қызметкерлердің атын білдіреді;
- «Отчество» қатары (типі «nchar»), мәні қызметкерлердің әкесінің атын білдіреді;
- «Дата_рождения» қатары (типі «nchar»), мәні қызметкерлердің туылған жылын білдіреді;

- «Адрес_прописки» (типi «nchar»), мәні қызметкерлердің тұратын мекен жайын білдіреді;
- «Адрес_проживания» (типi «nchar»), мәні қызметкерлердің тұрақты мекен жайын білдіреді;
- «Домашний» қатары (типi «nchar»), қызметкердің телефон номерін сақтайды;
- «Мобильный» қатары (типi «nchar»), қызметкердің ұялы телефон номерін сақтайды;
- «Номер_УД» қатары (типi «nchar»), қызметкердің жеке куәлік номерін сақтайды;
- «ИИН» қатары (типi «nchar»), қызметкердің ЖСН сақтайды;
- «Адресная справка» қатары (типi «varbinary(MAX)»), мекен жай анықтамасын сақтайды;
- «Должность» қатары (типi «nchar»), лауазымдардың реттік номерін сақтайды.

	Имя	Тип данных	Допустимы значения NULL	По умолчанию
Id		int	<input type="checkbox"/>	
Фамилия		nchar(51)	<input checked="" type="checkbox"/>	
Имя		nchar(50)	<input checked="" type="checkbox"/>	
Отчество		nchar(50)	<input checked="" type="checkbox"/>	
Дата_рождения		nchar(50)	<input checked="" type="checkbox"/>	
Адрес_прописки		nchar(50)	<input checked="" type="checkbox"/>	
Адрес_проживания		nchar(50)	<input checked="" type="checkbox"/>	
Домашний		nchar(50)	<input checked="" type="checkbox"/>	
Мобильный		nchar(50)	<input checked="" type="checkbox"/>	
Номер_УД		nchar(50)	<input checked="" type="checkbox"/>	
ИИН		nchar(50)	<input checked="" type="checkbox"/>	
Адресная_справка		varbinary(MAX)	<input checked="" type="checkbox"/>	
Должность		nchar(50)	<input checked="" type="checkbox"/>	

2.5-сурет – «Employees» кестесінің құрылымы

«Client» (2.6-сурет) :

- «id» қатары (типi «int»), AUTO_INCREMENT мәні автоматты түрде санын 1-ге арттырып отыратынды білдіреді;
- «Фамилия» қатары (типi «nchar»), мәні клиенттің тегін білдіреді;
- «Имя» қатары (типi «nchar»), мәні клиенттің атын білдіреді;
- «Отчество» қатары (типi «nchar»), мәні клиенттің әкесінің атын білдіреді;
- «Дата_рождения» қатары (типi «nchar»), мәні клиенттің туылған жылын білдіреді;
- «Адрес_прописки» (типi «nchar»), мәні клиенттің тұратын мекен жайын білдіреді;

- «Адрес_проживания» (типi «nchar»), мәні клиенттің тұрақты мекен жайын білдіреді;
- «Домашний» қатары (типi «nchar»), клиенттің телефон номерін сақтайды;
- «Мобильный» қатары (типi «nchar»), клиенттің ұялы телефон номерін сақтайды;
- «Номер_УД» қатары (типi «nchar»), клиенттің жеке куәлік номерін сақтайды;
- «ИИН» қатары (типi «nchar»), клиенттің ЖСН сақтайды;
- «Тип клиента» қатары (типi «nchar»), клиенттің типін сақтайды;
- «Место работы» қатары (типi «nchar»), клиенттің жұмыс орнын сақтайды;
- «Адрес работы» қатары (типi «nchar»), клиенттің жұмыс орны адресін сақтайды.

Имя	Тип данных	Допустимы значения NULL	По умолчанию
Id	int	<input type="checkbox"/>	
Фамилия	nchar(51)	<input checked="" type="checkbox"/>	
Имя	nchar(50)	<input checked="" type="checkbox"/>	
Отчество	nchar(50)	<input checked="" type="checkbox"/>	
Дата_рождения	nchar(50)	<input checked="" type="checkbox"/>	
Адрес_прописки	nchar(50)	<input checked="" type="checkbox"/>	
Адрес_проживания	nchar(50)	<input checked="" type="checkbox"/>	
Домашний	nchar(50)	<input checked="" type="checkbox"/>	
Мобильный	nchar(50)	<input checked="" type="checkbox"/>	
Номер_УД	nchar(50)	<input checked="" type="checkbox"/>	
ИИН	nchar(50)	<input checked="" type="checkbox"/>	
Тип_клиента	nchar(50)	<input checked="" type="checkbox"/>	
Место_работы	nchar(50)	<input checked="" type="checkbox"/>	
Адрес_работы	nchar(50)	<input checked="" type="checkbox"/>	

2.6-сурет – «Client» кестесінің құрылымы

- «Credit» (2.7-сурет):
- «id» қатары (типi «int»), мәні несие келісімшарт номерін білдіреді;
 - «Программа» қатары (типi «nchar»), несие түрі жайлы мәлімет сақтайды;
 - «Размер» қатары (типi «nchar»), несие өлшемі жайлы мәлімет сақтайды;
 - «Валюта» қатары (типi «nchar»), несие түрі жайлы мәлімет сақтайды;
 - «Срок» қатары (типi «nchar»), несие мерзімі жайлы мәлімет сақтайды;
 - «Дата открытия» қатары (типi «nchar»), несие ашқан мерзімі жайлы мәлімет сақтайды;
 - «Заемщик» қатары (типi «nchar»), несиеге кепілдеуші жайлы мәлімет сақтайды;
 - «Статус» қатары (типi «nchar»), несие алушының статусы жайлы мәлімет сақтайды;
 - «Скидка» қатары (типi «nchar»), несие алушыға жеңілдік жайлы мәлімет сақтайды;

Имя	Тип данных	Допустимы значения NULL	По умолчанию
Id	int	<input type="checkbox"/>	
Программа	nchar(51)	<input checked="" type="checkbox"/>	
Размер	nchar(50)	<input checked="" type="checkbox"/>	
Валюта	nchar(50)	<input checked="" type="checkbox"/>	
Срок	nchar(50)	<input checked="" type="checkbox"/>	
Дата_открытия	datetime	<input checked="" type="checkbox"/>	
Заемщик	nchar(50)	<input checked="" type="checkbox"/>	
Статус	nchar(50)	<input checked="" type="checkbox"/>	
Скидка	nchar(50)	<input checked="" type="checkbox"/>	

2.7-сурет – «Credit» кестесінің құрылымы

«CreditProgramms» (2.8-сурет):

- «id» қатары (типі «int»), мәні несие келісімшарт номерін білдіреді;
- «Название продукта» қатары (типі «nchar»), несие программасы түрі жайлы мәлімет сақтайды;
- «Процентная_ставка» қатары (типі «nchar»), несие пайызы жайлы мәлімет сақтайды;
- «Максимальная_сумма» қатары (типі «nchar»), берілетін несиең ең үлкен соммасы жайлы мәлімет сақтайды;
- «Валюта» қатары (типі «nchar»), несие соммасы жайлы мәлімет сақтайды;
- «Максимальный срок» қатары (типі «nchar»), несиең ең үлкен мерзімі жайлы мәлімет сақтайды;

Имя	Тип данных	Допустимы значения NULL	По умолчанию
Id	int	<input type="checkbox"/>	
Название_продукта	nchar(30)	<input checked="" type="checkbox"/>	
Процентная_ставка	nchar(10)	<input checked="" type="checkbox"/>	
Максимальная_сумма	nchar(15)	<input checked="" type="checkbox"/>	
Валюта	nchar(8)	<input checked="" type="checkbox"/>	
Максимальный_срок	nchar(10)	<input checked="" type="checkbox"/>	

2.8-сурет – «CreditProgramms» кестесінің құрылымы

«Client» кестесінің құрылымы мен банк клиенттері жайлы (2.9-суретте) мәліметтер көрсетілген.

Id	Фамилия	Имя	Отчество	Дата_рожд...	Адрес_про...	Адрес_про...	Домашний	Мобильный	Номер_УД	ИИН	Тип_ клиен...	Место_ раб...	Адрес_ раб...
2	Жанашев	Данияр	Дауленулы	9 Октябрь	Шевченко	г. Алматы	37125	8777856231	0354579	9710096545	-	КазахТелек	Аль-Фараб...
3	Бирликбаев	Таншолпан	Ермуханбе...	5 Февраль	Сатпаева 1	г. Алматы	-	8747569231	3453455	9702054565	-	Мега Парк	Сейфуллин...
4	Жиенбаев	Олжас	Ермуханов...	26 Июль 19...	Абая 12	г. Алматы	27171	8705213256	0265874	9407265645	-	Кафе "Адил	Назарбаев ...
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2.9-суретте – «Client» кестесінің құрылымы

«Credit» кестесінің құрылымы мен банк клиенттері алған несиелік шарттары жайлы (2.10-суретте) мәліметтер көрсетілген.

Id	Программа	Размер	Валюта	Срок	Дата_откр...	Заемщик	Статус	Скидка
1	Простой кр...	100000	KZT	92	02.05.2019	Жанашев Д...	Новый кли...	0%
2	Евроремонт...	200000	KZT	1095	13.03.2019	Бирликбаев...	Новый кли...	10%
3	Для бизнес...	50000	USD	365	17.04.2019	Жиенбаев ...	Новый кли...	0%
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2.10-сурет – «Credit» кестесінің құрылымы

«employees» кестесінің құрылымы мен банк - қызметкерлер жайлы (2.11-суретте) мәліметтер көрсетілген.

Id	Фамилия	Имя	Отчество	Дата_рожд...	Адрес_про...	Адрес_про...	Домашний	Мобильный	Номер_УД	ИИН	Адресная_...	Должность
1	Капар	Дулат	Мерекеулы	4 Ноябрь 1...	г. Алматы	Шашкина 1...	NULL	37112	0412563	9711045645	0x01	Финансовы...
2	Сархытов	Асхат	Серикулы	23 Декабрь	г. Алматы	Тимирязев...	NULL	37214	0365478	9712232145	0x02	Бухгалтер ...
3	Ануарбеко...	Аяулым	Ануарбеко...	14 Апрель	г. Алматы	Саина 17	NULL	37984	0389456	9504145678	0x03	Эксперт по ...
*>	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2.11-сурет – «Employees» кестесінің құрылымы

«CreditProgramms» кестесінің құрылымы мен несиелік түрлері жайлы (2.12-суретте) мәліметтер көрсетілген.

Id	Название_...	Процентн...	Максимальн...	Валюта	Максимальн...
1	ЕВРОРЕМО...	15	3000000	KZT	60
2	АВТОКЭШ ...	15	3000000	KZT	120
3	НЕОТЛОЖ...	15	1000000	KZT	60
4	Экспресс ...	12	750000	KZT	24
5	Простой кр...	13	1500000	KZT	36
6	Для бизнес...	17	5000000	USD	48
*	NULL	NULL	NULL	NULL	NULL

2.12-сурет – «CreditProgramms» кестесінің құрылымы

2.5 Пәндік саланы модельдеу

2.5.1 Rational Rose аспабымен жұмыс

UML (ағылш. Unified Modeling Language – унифицирленген модельдеу тілі) – бағдарламалық қамтаманы өңдеу аймағындағы объектіні моделдеу үшін қолданылатын графикалық сипаттау тілі. UML кең профильді тіл, UML-моделі деп аталатын жүйенің абстрактілі үлгісін жасау үшін графикалық түрде пайдаланатын ашық стандарт болып табылады. UML негізінде программалық жүйелерді анықтау, визуализациялау, жобалау және құжаттау үшін құрылған. UML программалау тілі болып табылмайды.[2]

UML программалық қамтаманы модельдеумен ғана шектелмейді. Сонымен қатар UML-ді бизнес-процесстерін моделдеу, жүйелік жобалау және ұйымдастыру құрылымын бейнелей үшін де қолданады.

UML тілінің құрылыс блоктарын бір-бірінен өз бетінше біріктіруге болмайды. Сондықтан UML тәртіптер тізімінен тұрады, ол дұрыс анықталған, яғни семантикалық түрде басқа байланысқан модельдермен сәйкестендірілген болуы керек.

Rational Rose өнімдерінің сериясы құрастырушыны нақты уақыт жүйелерінде және «клиент/сервер» орталарында қолдануға жарайтын және қазіргі кездегі бизнес талаптарын қанағаттандыратын тиімді де сенімді шешімдерді қабылдауға көмектесетін визуалдық моделдеудің толық құралдар жиынымен қамтамасыз етеді. Rational Rose құралдары біркелкі стандарттарға негізделген және моделдеуді, оларға жақын сфералардағы бизнес-процестерді оптимизациялауға талаптанатын, компьютерлік ғылымдармен онша таныс емес тұлғалармен қатар, программалық қолданбалардың логикасын моделдеу құралдарын қажет ететін мамандар үшін оңай етеді.[4]

Rational Rose өнімдерінің сериясы құрастырушыны нақты уақыт жүйелерінде және «клиент/сервер» орталарында қолдануға жарайтын және қазіргі кездегі бизнес талаптарын қанағаттандыратын тиімді де сенімді шешімдерді қабылдауға көмектесетін визуалдық модельдеудің толық құралдар жиынымен қамтамасыз етеді. Rational Rose құралдары біркелкі стандарттарға негізделген және модельдеуді, оларға жақын сфералардағы бизнес-процестерді оптимизациялауға талаптанатын, компьютерлік ғылымдармен онша таныс емес тұлғалармен қатар, программалық қолданбалардың логикасын модельдеу құралдарын қажет ететін мамандар үшін оңай етеді. Rational Rose тексермелік (пробная) пакетінің көшірмесін Rational Software Corporation сайтында <http://www.rational.Com> адресі бойынша табуға болады. Rational Rose – UML әдістемесін тарату аспабы. Ұқсас есептерді шешу үшін қолданылатын басқа да визуальды аспаптар. Өңдеу функцияналдығын сипаттау – қолдану варианттар диаграммасы (талаптар және шектеулер), функция орындалу реттері әрекеттердің тізбектер диаграммасы (талаптар және шектеулер), өңдеу элементерінің бір-бірімен

байланыс және бар болу сипаттамасы – кооперативті диаграммалар (талаптар және шектеулер). Unified Modeling Language (UML) – жүйеленген модельдеу тілі. Визуальды моделдеу нақты өмірдің объектілері мен түсініктерін көрсету. Біз көре алатын модельдер арқылы проблемаларды қабылдау тәсілі модельдер проблемаларды талдаудың ақпарат алмасуының программалық қамтама және деректер базасын жобалаудың құжаттарды дайындауда қолданылады. Модельдер арқылы талаптарды жақсырақ қабылдауға, жүйені дизайн сапасын арттыруға және оны басқара алу мүмкіндігін қолдана білуге септігін тигізеді.[5]

Интерфейсті жобалау. Объектілік жобаның өңделуі, ОМ 6 қосымшасына сәйкес, негізінен «прецеденттер диаграммасы» (ПД) немесе, екінші аталуы – "Қолдану варианттар диаграммасы" (ҚВД) басталады. Диаграмма прецедент барлық қалған жоба диаграммалары үшін арналып жасалады. Ал оның екінші аталуы, негізінен, оның атқаратын ролын толық көрсетеді деп айтуға болады – пайдаланушы жүйесінің қолдану варианттар диаграммасы (актерлар, әрекет жасаушы объект).

Дипломдық жобада «Кітап әлемі» интернет-дүкенінің веб-сайтының UML диаграммасы жасалды. Диаграммаларды сызуда Rational Rose арнайы аспабы пайдаланылады. Бұл программа UML диаграммалар бойынша визуалды диаграмма типтерін құруға мүмкіндік береді. Ол негізінен құрастырылатын жүйенің бейнесін көрсетеді. Rational Rose- та келесі диаграммалар тұрғызылмақ:

- прецеденттер диаграммасы;
- тізбекті және кооперативті диаграмма;
- қызмет диаграммасы;
- кластар диаграммасы.[6]

2.6 Прецеденттер диаграммасы

Қолдану вариантын идентификациялаудың көзі болып сыртқы оқиғалар қызмет атқарады. Алдымен жүйе реакция беретін сыртқы әлемде болып жатқан оқиғаларды тізбектеуден бастаған дұрыс. Кейбір нақты оқиғалар қолданушыны араластырмай жүйенің реакциясын тудыруы немесе қолданушының өз реакциясын тудыруы мүмкін. Реакция беруді қажет ететін оқиғаларды идентификациялау қолдану варианттарын айқындауға көмектеседі.[4]

Орындаушы және қолдану варианттары арасындағы байланысқа қосымша, «қолдану» (uses) және «кеңейту» (extends) деген байланыстың екі типі бар. Қолдану вариантының бірі екіншісіне ұқсас, бірақ одан көбірек жүктеме алған жағдайда «кеңейту» типті байланыс қолданылады.

Жүйе тәртібінің бір фрагменті бірнеше қолдану вариантында қайталанса, онда оның сипаттамасын әр вариантқа көшірмеу үшін «қолдану» байланысы қолданылады. «Кеңейту» және «қолдану» байланыстарының арасындағы ұқсастық пен айырмашылықтар. Бұл екеуі де бірнеше қолдану

варианттарының ортақ тәртіптер фрагменттерін жеке вариантқа бөлектеп, оны басқада варианттарға «қолданады» немесе оларды «кеңейтеді». Екінші жағынан, бұл жағдайлар оны әртүрлі мақсатта жасайды.

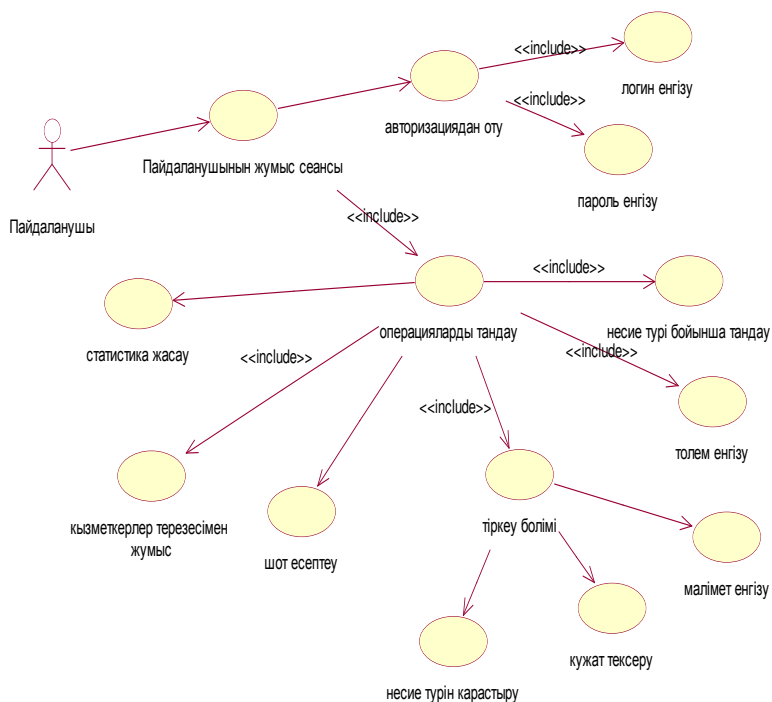
Жүйені құру үшін зат обласын білу қажет. Сонымен қатар жүйенің қалай жұмыс істейтіні туралы хабарлар болу керек. Диаграмма жүйесі орындайтын операция тізімін құру үшін арналған. Ол берілген функциядағы жүйе объектісі анықталған тізімдер жүйесін орындайды. Осындай түрде орындалатын жүйенің функция жүйесі құрылады және жүйемен қарым-қатынастағы объектілер әрекетінің сценариялары жазылады. Прецедент процесін моделдеу үшін қолданушы іс-әрекетін толық көрсетуіне және толық реакциясына бағытталған. Use Case диаграммасы 2.13- суреттерде көрсетілген.[6]

Прецедентер диаграммасы – бұл жүйе әрекетінің болжамының құжатталған моделі.

Әр прецедент құжат түрінде тіркелген оқиғалар ағынының көмегімен сипатталу керек. Оған сәйкес мәтіндік құжат актер прецеденті жасағанда жүйенің не істеу керектігін анықтайды.

Прецедентті сипаттайтын құжаттың құрылымы әртүрлі, бірақ жалпы сипаттамасы келесі бөлімдерден тұру керек:

- қысқаша сипаттама;
- алғы шарттар;
- оқиғалар ағынын бөлшектеп сипаттау негізгі ағын және альтернативті ағын;
- артқы шарт.



2.13-сурет – Use Case диаграммасы (пайдаланушы)

Қолдану вариантының диаграммасы (Use Case Diagram) – бұл активті субъектілердің көптеген графикалық көрсетімі оны қолданудың, сол немесе басқа варианттардың амалдарымен өзара әрекеттеседі. Жүйені жоспарлағанда жүйенің кілттік функциясын және көптеген қолданушыларды ұсынатын негізгі диаграмма (Main Use Case Diagram) конструкцияланады.

Use Case – бұл белгілі бір актер (Actor) үшін арналған, белгілі бір нәтиже бере алатын, жүйемен орындалатын әрекеттер тізбегінің сипаттамасы. Прецедент негізінен модельдегі мәндердің тәртіптік құрылымын жасайды және прецеденттер кооперацияның қатысуымен таратылады. Графикалық түрде ол тұтас сызықпен сызылған эллипс түрінде беріледі, оның ішіне аты жазылады.

Актер (Actor) – бұл жүйедегі негізгі элементтерімен байланыста болатын объектінің рөлі болып есептеледі. Актер мен пайдаланушының айырмашылығы келесідей: пайдаланушы жүйені пайдаланатын физикалық объект. Ол бірнеше рөлде ойнауы мүмкін сондықтан ол бірнеше актер болуы мүмкін.[2]

Альтернативті ағын. Оқытушы алған нәтижелерді MS EXCEL-ге немесе мәтіндік файлға енгізеді. Өйткені бастапқы мәліметтерді қарау – бұл пәндер туралы мәліметтер, оларға қойылатын жүктемелер және тапсыру формасы.

2.7 Тізбек диаграммасы

Тізбектелген диаграммасы уақыт бойынша объектілердің өзара бірлесу операцияларының орындалу ретін безендіреді және сценарийлермен қарастырылған функциялардың орындалу процесіндегі алмасатын объектілердің хабарламалар тізбектерінің реттелген сценарийге кіретін объектілер мен класстар бейнеленеді. Тізбектелген диаграммалар жалпыда Logical View пакетіндегі есептелінген қолдану варианттарының жүзеге асырумен бірге ассоциацияланады. Тізбек диаграммасы төрт негізгі элементтерден тұрады:

- прецеденттегі ізбасар мәтінінің іс-әрекеті. Ол сол жақтан жоғарыдан төменге жазылады. Сол терезеде іс-әрекет сипатталуы болып, жұмыс уақытындағы орындалатын ақпараттар қызмет етеді;

- объекттер "объект-класс" форматында аты немесе объект данасының номері және класс объектісінің аты жазылады;

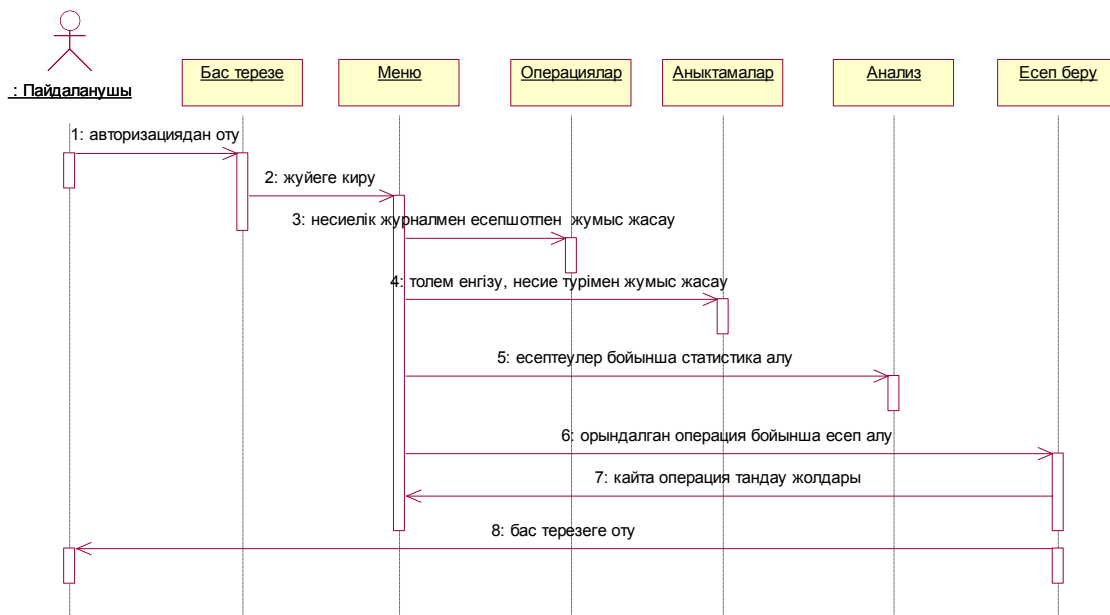
- хабарландыру, бағытпен көрсетілген бір объектіден келесіге бағытталған іс-әрекет туралы ақпарат жолдамасынан тұрады. Белгілі бір уақытта орындалуы және осы іс-әрекеттегі жүйенің жауапты реакциясы болуы мүмкін;

- әдістері (операциялар). Тікбұрыш түрінде көрсетілген. Олар үздік сызықта орналасқан. Яғни, әдістерге кіретін сол объектілер келеді. Тікбұрыш ұзындығы ізбасарда басқару фокусын көрсетуде қолдануға болады: Тікбұрыш

бітетін әдіс түгелдей нүктесіне дейін басқарумен иеленеді. Бұл үшбұрыштар өмір объекті түзуі деп аталады.[5]

UML талаптарымен сәйкес объект тізбектелген диаграммасында тіктөртбұрыш түрінде көрсетіледі. Жоғарыда көрсетілгендей объектіні 3 түрлі әдіспен атауға болады: тек оның атауын көрсету, объект пен класстың атауын беру, не класстың атауымен шектелу қажет.

Мен әрбір тыңдарман топтарымен кезекті жұмыс жасағанда, маған сұрақ қояды: тізбектелген диаграммалар қаншалықта қиын болуы мүмкін. Менің жауабым өзгертілмей қалады: неғұрлым оңай болса, соғұрлым жақсы болады. Осындай диаграммалардың күрделілігі олардың қарапайым болуында тұрады. Олар объектілерді, хабарламаларды және олардың алмасуы, сценарийлермен қамтылған функцияларды бірден көруге рұқсат етеді. Келесіде көбінесе мынадай сұрақ қойылады: оқу аймағының жүзеге асырылуын өңдесе, онда логикалық өрнек шартын қалай жасайды. Осыдан менің дәстүрлі жауабым бар, бірақ та ол субъективті әжептәуір. Егер де логика қарапайым болса, яғни бірнеше хабарламалармен сипатталатын болса, мен оларды бір диаграммаға қосамын және мүмкін болатын таңдау варианттарын сипаттау үшін қосымша ескертулерді қолданамын. Кері жағдайда жөнді бөлек диаграммаларды құру біреуін егер-онда оқиғасы үшін, ал басқасын басқа тармақтарды безендіру үшін. Бұл қабылдау модельдерін едәуір жеңілдетуге рұқсат береді. Тізбектер диаграммасы 2.14- суретте көрсетілген.[6]



2.14-сурет – Тізбектер диаграммасы (пайдаланушы)

Тізбектелген диаграммасында активті субъектілермен өзара бірлесу фактілер жүйесінің көрсетілуін рұқсат ететін шекаралар класы қосылады

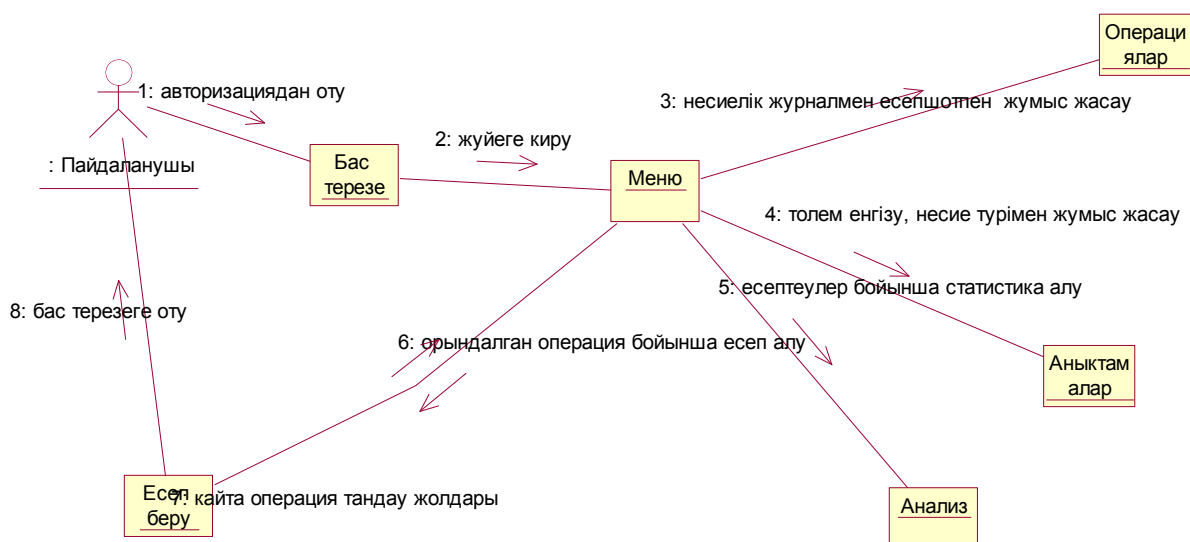
(қолданушылармен және басқа жүйелермен). Осындай қабылдау анализдің ерте сатыларында бекітілуін талап етеді және интерфейстер талаптарының құжатталуына рұқсат етеді.

2.8 Кооперация диаграммасы

Кооперация диаграммасы – бұл хабарламаларды жіберетін және қабылдайтын объектілерді құрылымдық жақтан жан-жақтылы ұйымдасуын айқындайтын өзара әрекеттесу диаграммасы. Кооперация диаграммалары жүйе жұмысы барысында объектілердің өзара әрекеттесуін бейнелейді. Мұндай диаграммалар жүйе тәртібінің сценарийлерін модельдейді. Объект атының асты сызылады және әрдайым беріледі, ал қасиеттері таңдаулы түрде көрсетіледі.[5]

Кооператив диаграммалар оқиғаларда пайдалы объектілерде істелген өзгертулердің зардаптары керек бағалағанда және қандай басқа объектілерге мынау әсер етеді әрекеттестік диаграммалары жасайды, есте сақтауға ереді, не жауаптылық айқын сайып келгенде объектілерге тағайындалады.

Кооперативтік диаграммаларды жатқызуға болады, олар негізінен тәртіп детализациясы үшін қолданылады, олар уақиғалар аймағын және олардың арасындағы байланысты анықтайды, қосымша қолданушыларды анықтайды, олардың жалпы және мінездемелік анықталарын береді, яғни соңғы «Класстар диаграммасын» салу үшін керекті деректердің барлығын алуға мүмкіндік береді. Кооперация диаграммасы 2.15-суретте көрсетілген.



2.15-сурет – Кооперация диаграммасы

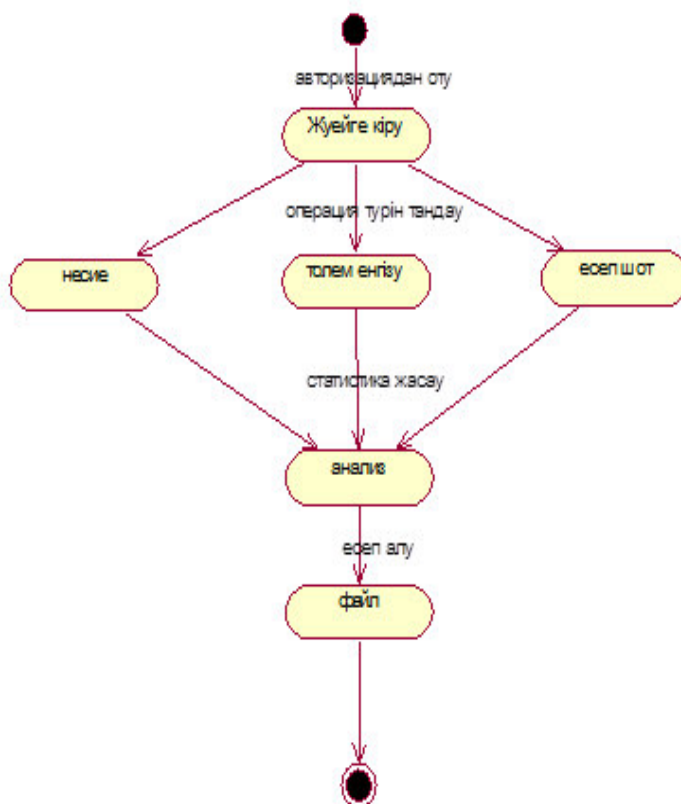
2.9 Қызмет диаграммасы

Қызмет (Activity) – бұл кейбір объектілерді орындау кезінде белгілі бір

әрекет немесе сол не басқа оқиғаның түсуін күтуде мекендеуді жүзеге асыратын жиын шарты. Объектінің қызметі класы бір немесе бірнеше атрибуттарымен сипатталады. Қызмет диаграммасы – объектінің бір жағдайдан басқасына ауысуын арандататын, және әрекеттері жағдайларын алмастырумен ескертілінген, оқиғалардың немесе хабарлаулардың объектік жағдайларының графикалық ұсынысы. Қызмет диаграммасы 2.16- суретте көрсетілген.

Объектінің өз қызметін уақыт бойында қалай өзгертетінін түсіну үшін қызмет өзгеруінің спецификациясы бар. Объектінің қызметі оның атрибуттарының және оған қоса қатынас атрибуттарының мәнімен анықталады.

Қызмет спецификациясы класс атрибутын анықтайды. Тәртіп спецификациясы класс операцияларын анықтайды, олардың кейбірі объект қызметін өзгерте алады. Объектілер қызметтерін моделдеу қызметтер диаграммасының көмегімен жүзеге асады. Қызметтер графы(автомат) бұл қызметтер мен ауысулар графы. Қызметтер моделі жүйеге маңызды кластар үшін құрылады.[2]



2.16-сурет – Қызмет диаграммасы

2.10 Класстар диаграммасы

Класс диаграммасы – қосымша кодын енгізу үшін негізгі диаграмма

классы диаграмма көмегімен жүйенің ішкі жүйесі шығарылады, яғни мұралауды сипаттауын және қосымша класстар бір-бірімен байланысты. Осында жүйенің логикалық көрінісі сипатталады. Класстар ол тақырыпша. Кейіннен физикалық объектілер және ақпарат кодтары анықталады.

Диаграмма кластары жүйенің ортақ көрінісін сипаттайды, яғни, жүйе объект көрінісі. Диаграмма класстар қорытындысында кез-келген уақытта өзгертілуі мүмкін. Осыдан диаграммалар және спецификациялық класс өзгертулері автоматты түрде жаңартады.[2]

Диаграмма класстары дайын жүйе талдау тәрізді қолдануы мүмкін және сонымен қатар жаңа әзірлеуде де. Ерекшеленген класстарда әртүрлі әдістер және жүрістер қолданылады. Осылай Барами (Bahrami) класстарға белгіленуіне (class discovery) 4 негізгі жүрістерді анық оқиды. Келесі жүрістерден тұрады:

- аты қолданылатын группа негізіндегі жүріс (сөйлемдегі бар аттар);
- класс үшін ортақ шаблондарды қолдану негізіндегі жүріс;
- прецеденттерді қолдану негізіндегі жүріс;
- CRC жүрісі (class-repository-collaborators класс, міндет, қызметкерлер спецификация, құрылымы).

Класстың атрибуттарын сипаттау үшін орындалатын операцияны оңай құру спецификация классы.

Класс (Class) ортақ қасиеттері (атрибуттары), тәртібі (функциялары), семантикасы және басқа объектермен байланысы бар объектер тобын анықтайды. Класстың объектіні құруға арналған шаблон ретінде қарауға болады. Әрбір объект қандайда бір ғана кластың нұсқасы мысал ретінде келесідей сипаттамалары бар курсты ұсыну класс анықтамасын қарастырайық:

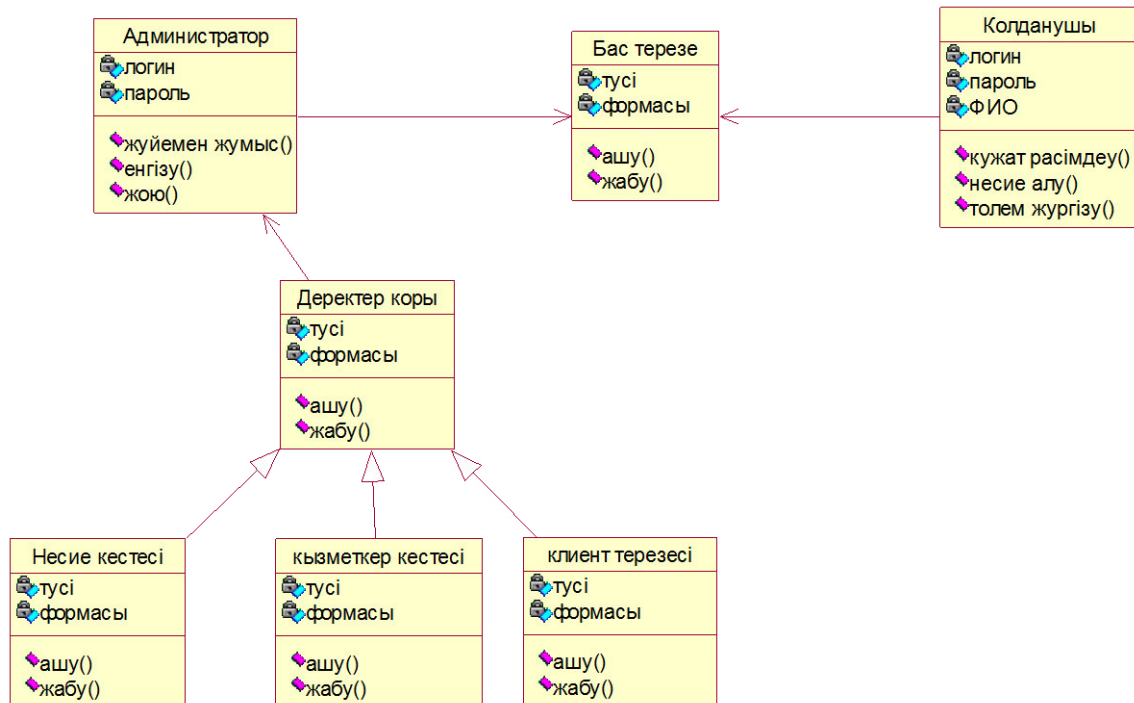
- атрибуттар;
- функциялар.

Диаграмманың бұл түрі сипатталған класстардың негізгі кодын негізге ала отырып, логикалық көрсетілім жүйесін құруға рұқсат етеді. Диаграмма белгілері класстар мен интерфейстердің өзара байланысын қиын иерархиялық жүйеде бейнелеуге рұқсат етеді. Диаграмманың берілген түрі жүйе объектілері бейнеленетін Collaboration диаграммасына қарама қарсы. Rational Rose диаграмманың берілген түрі көмегімен түрлі нотациялар құруға рұқсат етеді. Класстар арасындағы қарым-қатынасты орнату. Агрегаттық қатынастар және мұрагер қатынастар. Қолдану қатынастары. Ассоциативті байланыстар, бағыттық, уақыттық және қалыпты байланыстар.[5]

Класстарды құжаттандырады. Класстарды құру үшін міндетті түрде құжаттандыру керек. Сипаттама кластың құрылымын емес, мәнін көрсету керек. Класстар диаграммасын алу тәртібі. Функциональдық және актерлармен функциялардың бірлесе қолданылуы. Класстар диаграммасы 2.17-суретте көрсетілген.

Кластар диаграммасында кластар, интерфейстер, объектілер, мен кооперациялар және олардың арасындағы қатынастар көрсетіледі. Объектілі-

бағытталған жүйені моделдеу кезінде бұл диаграмма көп пайдаланылады. Кластар диаграммасы жобалау жағынан жүйенің статикалық түріне сәйкес келеді. Активті кластардан тұратын, кластар диаграммасы, процестер жағынан жүйенің статикалық түріне сәйкестендіріледі.[6]



2.17-сурет – Класстар диаграммасы

3 Жүзеге асыру және тестілеу бөлімі

3.1 Программалау тілін таңдауды негіздеу

Қазіргі таңда программа құру үшін кеңінен қолданылатын көптеген программалау тілі бар және бұдан басқа қазіргі кезде ұмытылып кеткен мыңнан астам түрі бар. Программалау тілі кілтті сөздердің және әртүрлі көмекші сөздердің көмегімен алгоритімді сипаттауға мүмкіндік береді. Кілтті сөздерді (олар бұйрықтар немесе операторлар деп аталады) қолдана отырып программист берілген тапсырманы шешу үшін компьютер орындайтын іс-әрекеттің дәлме-дәл ретін анықтай алады.

Қазіргі уақытта C# программалау тілі кең ауқымды түрде программистер үшін жобалау-өңдеуші бағдарлама ретінде танымал, оның негізгі тілі нысанды бағытталған болып табылады. Ол программа құру жұмысын жеңілдету мақсатында өзінен бұрын пайда болған Object Pascal үлгісі бойынша құрылған. C# программистерге кәдімгі жалпыхалықтық түсінік бойынша алгоритм құруына көмектесті.

Бұл программалау тілі қазіргі таңдағы Windows-ке арналған жылдам құрастырудың өте жетілген визуальдық ортасы болып табылады. Оның құрамына 200- жақын әртүрлі компоненттер кіреді, ал аяқталған программаны жасап шығару аз уақыт алады. Алайда C# программалау тілі арқылы құрылған программалардың артықшылығы MS Visual Basic құрылған программалардың жылдамдығынан он есе артық шапшаң жұмыс істейді.

Қазір компьютерлер біздің тілді түсінбейді. Оларға біздің тілді түсіндіру үшін арнайы тілді – программалау тілін қолдануға тура келеді. Компьютерлер адамдармен салыстырғанда жылдамырақ және сенімді, бірақ жаңа ештеңе ойлап таба алмайды! Сіздер үшін программалау тілдері кәдімгі тілдерден әлдеқайда жеңіл екені қызық болып көрінуі мүмкін, бірақ бұл солай.

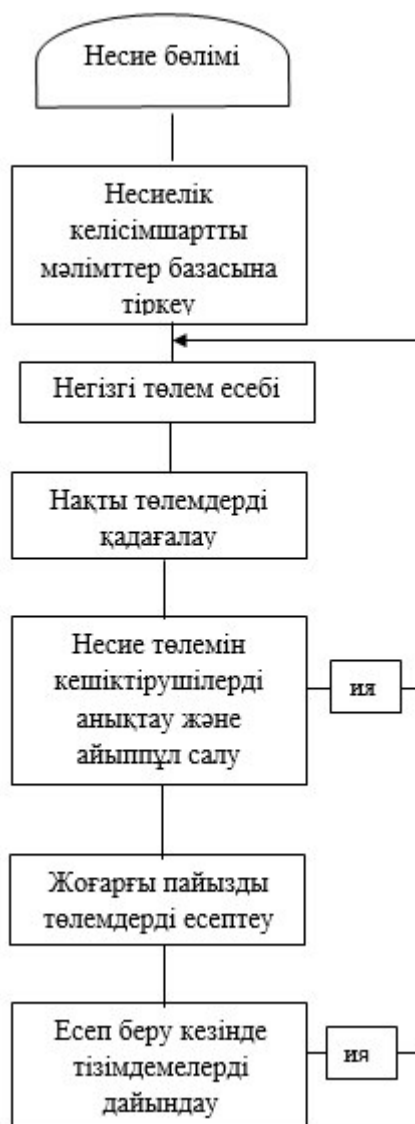
Қазіргі таңда қолданылатын программалау тілдерінің барлығы компьютерге біздің не қалайтынымызды түсіндіреді. Windows операциялық жүйесінде Паскаль, С және басқа да тілдерді қолданып программа құру – оңай жұмыс емес. Windows-тың ерекшеліктеріне аса мән бермей программа құруға болады. Бұл оның тағы бір жетістігі.

Кейінгі кездегі дамып келе жатқан визуалды программалау программаның текстін қолдан жазғанның орнына программалау пакетінің ішіндегі дайындықтарды қолдануды қолдайды, ол дайындықтардың тек қана кейбір параметрлерін керегіміз бойынша өзгертіп немесе модификациалау қажет.

C# қиын программаларды программалаудың қарапайым және әрлеу әдістерімен құруға арналған. Программаның жұмысын екі кезеңге бөлуге болады. Бірінші кезең – жобалау кезеңі. Бұнда программа бөлек құрама бөліктерден құралады да оған қажетті параметрлермен мінездемелер белгіленеді. Дәл осы кезеңде әлі программа жұмыс атқарғанға дейінгі

нәтижелерін көруге болады және визуалды программалаудың әдістері қолданылады. Екінші кезең – программаның жұмыс атқару кезеңі. Және аралық, үшінші кезеңді бөліп қарастыруға болады, бұл кезеңде программаның қателіктерін тексеру, дұрыс жұмыс атқарылуын қарастырады.

Жобалау кезеңінде бірнеше файлдар құрастырылып қолданылады. Программаның жұмыс істеу стадиясында әртүрлі динамикалық кітапханалы файлдарды қолдануы мүмкін, кеңейтілуі *.dll. Программаның жұмыс істеу барысында әртүрлі текстік немесе текстік емес файлдарды қолдануы мүмкін, мұнда қажетті мәліметтер орналасады.



3.1-сурет – Несие бөлімінің блок сұлбасы

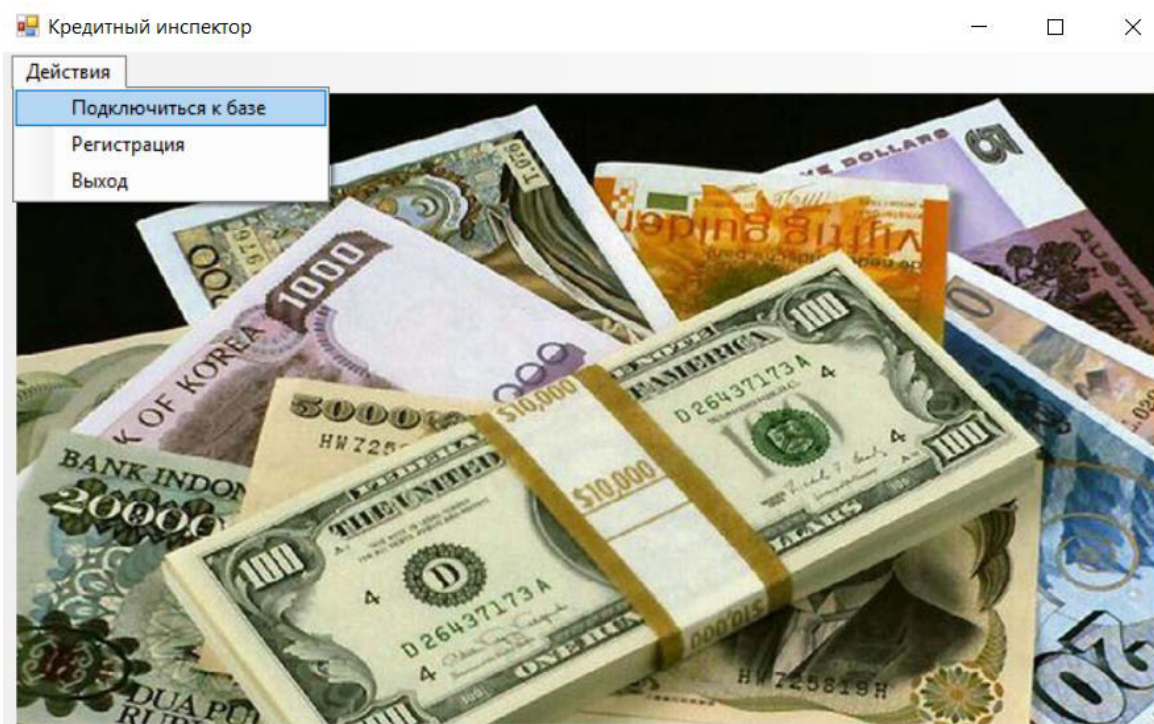
3.2 Банктің несие бөлімінің ақпараттық қосымшасын пайдаланушыға нұсқау

«Кредитный инспектор.ехе» файлын жүктегеннен кейін, негізгі бас терезе ашылады (3.2-сурет).



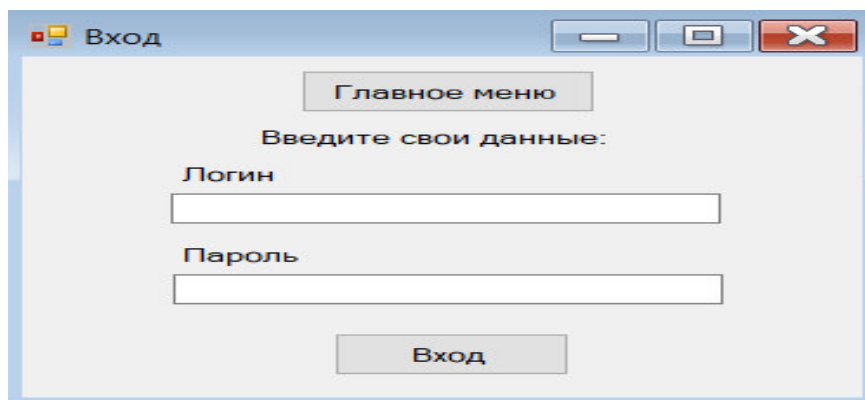
3.2-сурет – Негізгі терезесі

«Действия» мәзірінде деректер қорына қосылу және тіркеу жайлы (3.3-суретте) мәліметтер көрсетілген.



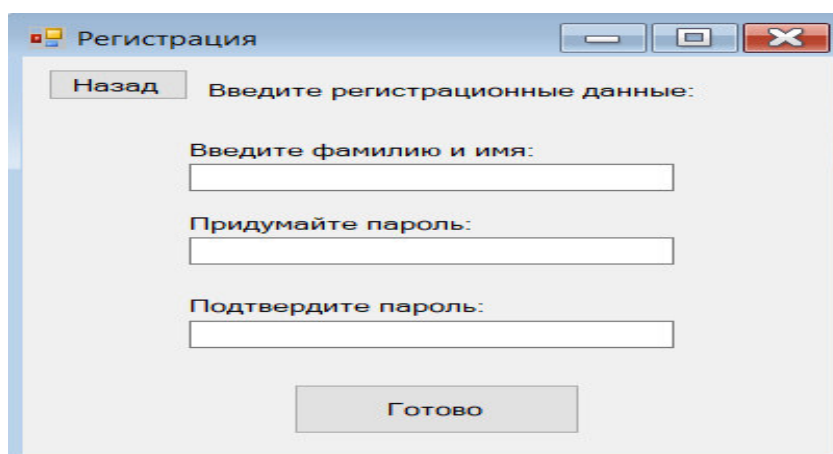
3.3-сурет – «Действия» мәзірі терезесі

«Действия» мәзірінен деректер қорына қосылу жолын таңдаған кезде 3.4-суретте көрсетілгендей экранға авторизация терезесі шығады.



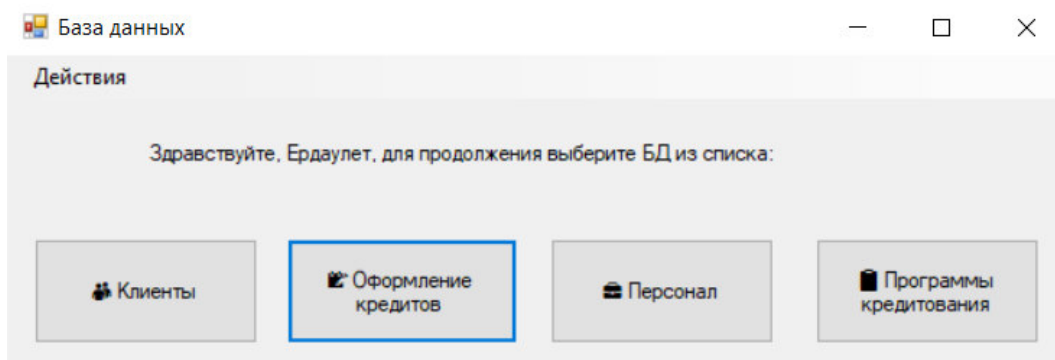
3.4-сурет – Авторизация терезесі

3.5- суретте көрсетілгендей « Действия» мәзірінен тіркеу жолын таңдаған кезде экранға пайдаланушыны тіркеу терезесі шығады.



3.5-сурет – Тіркеу терезесі

Қолданушы авторизациядан өтіп болған соң, « ОК» батырмасын басқаннан кейін 3.6-суретте көрсетілген мәліметтер қоры терезесі ашылады. Терезеден MS SQL мәліметтер қорында құрылған кестелер тізімін көруге болады.



3.6-сурет – Мәліметтер қоры терезесі

Пайдаланушы клиент жөнінде толық мәліметті енгізе алады, жаңа клиент қоса алады, деректер қорындағы клиенттерді іздей алады және жаңарту жұмыстарын атқара алады.

Клиенты

Личные данные

Фамилия:

Имя:

Отчество:

Дата рождения: День: Месяц: Год:

Документы

Номер удостоверения личности РК:

ИИН гражданина РК:

Адреса и телефоны

Адрес прописки:

Адрес фактического проживания:

Домашний телефон:

Мобильный телефон:

Дополнительные сведения

Тип клиента:

Место работы:

Адрес работы:

Добавить клиента

Сохранить изменения клиента

Очистить поля

Показать данные

Обновить таблицу

Удалить выбранного клиента

Ввод

Изменить

Назад

Поиск

Id	Фамилия	Имя	Отчество	Дата_рождения
2	Жанашев	Данияр	Дауленулы	9 Октябрь 1997
3	Бирликбаева	Таншолпан	Ермуханбетовна	5 Февраль 1997
4	Жиенбаев	Олжас	Ермуханович	26 Июль 1994

3.7-сурет – «Клиенты» терезесі

Төмендегі 3.8- сурет – «Оформление кредита» мәзірінің құрамы келтірілген. Терезеде несиелеу программасы, алынатын соммасы, сомма түрі, мерзімі, алған уақыты және кепілдеме тұлға жөнінде сияқты ақпараттарды енгізу, жою және жаңарту жұмыстарын жүргізуге болады.

Оформление кредита

Программа кредитования:

Размер займа:

Валюта займа:

Срок займа (дней):

Дата открытия займа:

Заемщик:

Добавить заемщика

Очистить поля

Список заемщиков

Id	Программа	Размер	Валюта	Срок	Дата_открытия
1	Простой кредит...	100000	KZT	92	02.05.2019 14:4
2	Евроремонт	200000	KZT	1095	13.03.2019 13:1
3	Для бизнеса	50000	USD	365	17.04.2019 16:2

Поиск

Ввод

Изменить

Удалить выбранного заемщика

Обновить таблицу

PDF

3.8-сурет – «Оформление кредита» мәзірінің құрамы

Төмендегі суретте «Программа кредитования» батырмасы көрсетілген. Бұл батырмада несие түрлері, төлем мерзімдері, төменгі және жоғарғы сумма, валюта түрлері (3.9-суретте) көрсетілген. Қордағы мәліметтерді өзгертуге, толықтыруға, өшіруге болады.

Программы кредитования

Id	Название_продукта	Процентная_ставка	Максимальная_сумма	Валюта	Максимальный_срок
1	ЕВРОРЕМОНТ	15	3000000	KZT	60
2	АВТОКЭШ	15	3000000	KZT	120
3	НЕОТЛОЖНЫЕ НУЖДЫ	15	1000000	KZT	60
4	Экспресс	12	750000	KZT	24
5	Простой кредит	13	1500000	KZT	36
6	Для бизнеса	17	5000000	USD	48

Добавить или изменить список

Название продукта:

Годовая процентная ставка:

Максимальная сумма кредита:

Максимальный срок (в месяцах):

Валюта:

Добавить программу

Показать данные

Сохранить изменения

Обновить таблицу

Удалить выбранную программу

Очистить поля

Ввод

Изменить

3.9-сурет – «Программа кредитования» терезесі

«Персонал» батырмасын басу арқылы ашылатын терезеде банк қызметкерлері жайлы толық мәлімет енгізуге болады (3.10-сурет).

Персонал

Личные данные

Фамилия:

Имя:

Отчество:

Дата рождения:

День: Месяц: Год:

Адреса и телефоны

Адрес прописки:

Адрес фактического проживания:

Домашний телефон:

Мобильный телефон:

Документы

Номер удостоверения личности РК:

ИИН гражданина РК:

Загрузить адресную справку

Скачать адресную справку

Дополнительные сведения

Должность:

Назад

Добавить сотрудника

Сохранить изменения сотрудника

Очистить поля

Показать данные

Обновить таблицу

Удалить выбранного сотрудника

Ввод

Изменить

Поиск

Id	Фамилия	Имя	Отчество	Дата_рождения
1	Капар	Дулат	Мерекеулы	4 Ноябрь 1997
2	Сарыитов	Асхат	Серикулы	23 Декабрь 1997
3	Ануарбекова	Аялым	Ануарбековна	14 Апрель 1995

3.3 Программаның баяндалуы

Бұл ақпараттық жүйе Visual Studio ортасында құрылды. Программалық ортада арқылы кредиттік бөлімнің жұмысымен танысуға, клиенттің алған кредитін есептеуге, кредиттің түрлері туралы мәліметтер алуға болады. Қолданушылық интерфейс программалаушымен байланысты құруды қамтамасыз етеді және әртүрлі қолданбалы басқару элементтері бар бірнеше терезелерден тұрады.

Орта көптеген есептерді шешуге мүмкіндік береді, оның ішінде:

– Әртүрлі бағыттағы аяқталған қосымшаларды Windows үшін құру;

Кез келген тілде кез келген қосымшалар үшін профессионалды көрінетін интерфейсті тез құру.

Ортаның артықшылықтары:

– қосымшаны өндеу жылдамдығы;

– жасалған қосымшаның өнімділігінің жоғарылығы;

– жасалған қосымшаның компьютер ресурстарына төмен қажеттіліктерін талап етуі;

– жаңа компоненттер мен құрылғыларды қосу арқылы қосымшаның жұмысын арттыру.

Құрылған ақпараттық қосымша банк қызметкерлерінің және қолданушы жұмысын жеңілдетеді:

– кредиттерді автоматты түрде есептейді;

– мәліметтерді енгізуді және есеп беруді жеңілдетеді;

– қолданушыға кредит түрлері туралы ақпаратты қолжетімді қылады.

Бұл ақпараттық жүйені банкте жеке қолданушылар қолдануы мүмкін.

3.4 Функционалдық тағайындалуы

Ақпараттық қосымша банктің кредиттік бөлімінің қызметкерлерінің жұмысын автоматтандыруға, қолданушыларға кредит түрлері, пайыздары туралы ақпарат беруге, кредиттік келісім жасауға, клиенттердің айлық төлемін шығаруға, клиенттің төлемдерін қабылдауға, борышкерлерге ескерту хатын жіберуге, берілген аралықтағы банктен алынған кредиттердің жалпы сомасын шығаратын және борышкерлердің тізімін және кредиттерінің сомасын шығаратын есеп берулер шығаруға арналған.

3.5 Қажетті техникалық жабдықтар

Программаның қалыпты жұмыс жасауы үшін төмендегідей оптималды конфигурациялы компьютер қолданылуы қажет:

– тактілік жиілігі 2.20 ГГц-тен жоғары Intel Core i5 процессоры;

– 2 Гб оперативті жад;

– VGA монитор;

– пернетақта және тышқан.

4 Экономикалық бөлім

4.1 Экономикалық бөлімдегі орындалатын мақсаттар мен тапсырмалар.

Бұл дипломдық жобада ақпараттық жүйе құрылды. Бұл бөлімнің мақсаты – шығындарды есептеу. Есептеу нәтижесінде қолданбалы программаның өзіндік құны болады.[8]

Өзіндік құнын табу үшін мыналарды ескеру қажет:

- программалық өнімді құрудың еңбек сыйымдылығы;
- материалдық шығындар;
- еңбекке ақы төлеу шығындары;
- әлеуметтік салық;
- негізгі қорларды амортизациялау;
- басқа да шығындар.

4.2 Программалық өнімді құрудың еңбек сыйымдылығын есептеу.

4.1-кесте – Жұмыстарды кезеңдер мен түрлер бойынша бөлу және олардың еңбек сыйымдылығын бағалау

Құрастыру кезеңі	Жұмыс түрі	Жұмыстың еңбек сыйымдылығы, ад.× сағ.
1	Тапсырманың сипаттамасы	34
2	Алгоритм құрастыру	60
3	Алгоритмнің блок-схемасын құрастыру	51
4	Проектің клиенттік бөлімін реализациялау	123
5	Жобаның администрациялық бөлімін құрастыру	198
6	Программаны баптау және тестілеу	105
7	Құжаттарды, қолданушыға нұсқаулықты, түсіндірме жазбаны дайындау.	51
Барлығы		622

4.3 Программалық өнім құрудағы шығындарды есептеу

Программалық өнім құруға кеткен шығындарды анықтау тиісті сметаны жасау жолымен жүргізіледі, ол мынадай баптарды қамтиды:

- материалдық шығындар;
- еңбекке ақы төлеу шығындары;
- әлеуметтік салық;
- негізгі қорларды амортизациялау;
- басқа да шығындар.

Программалық өнімді құрастыру барысында керекті программалық қамтамаға (IBM Rational Rose Modeler), орындыққа, үстелге және кеңсе тауарларына кеткен материалдық шығындар есептелген. Олар 4.2-кестеде көрсетілген.

4.2-кесте – Материалдық ресурстарға кеткен шығындар

Материалдық ресурстың атауы	Өлшеу бірлігі	Саны	Бірлік үшін бағасы, тг	Суммасы, тг
IBM Rational Rose Modeler	Дана	1	72000	72000
Үстел	Дана	1	12000	12000
Орындық	Дана	1	4000	4000
Қалам	Дана	4	100	400
Папка	Дана	3	90	270
Қысқыш	Дана	3	10	30
Қағаз	Бума	6	1500	9000
Барлығы				97700

Материалдық ресурстарға кеткен шығындардың (Z_M) жалпы суммасы (4.1) формула бойынша анықталады:

$$Z_M = \sum_{i=1}^n P_i \cdot C_i \quad (4.1)$$

мұндағы, P_i – материалдық ресурстың i түрінің шығыны, натуралды бірліктер;
 C_i – материалдық ресурстың i түрінің бірлігінің бағасы, тг;
 i – материалдық ресурстың түрі;
 n – материалдық ресурстардың түрлерінің саны.

Егер программалық өнімді құру үшін электр құрылғысы қолданылса, онда 4.2-кестеде келтірілген форма бойынша электр энергиясына арналған шығындарды есептеу қажет.[8]

Электр энергиясына кеткен шығындардың (Z_E) жалпы суммасы (4.2) формула бойынша есептеледі:

$$Z_E = \sum_{i=1}^n M_i \cdot K_i \cdot T_i \cdot C \quad (4.2)$$

мұндағы, M_i – i түрі электр құрылғысының паспорттық қуаты, кВт;
 K_i – i түрі электр құрылғысының қуатын қолдану коэффициенті ($K_i=0.9$ алынады);
 T_i – программалық өнімді құру кезіндегі i түрі құрылғысының жұмыс уақыты, сағ;
 C – электр энергиясының бағасы, тг/кВт×сағ;
 i – электр құрылғысының түрі;
 n – электр құрылғысының саны.

Электр энергиясына кететін шығындар ПҚ құру кезеңінің ұзақтығына, ПҚ жобалауға жұмсалған кВт/сағ санына және 1 кВт/сағ тарифіне сүйене отырып табылады. 2019 жылы заңды тұлғалар үшін Алматы қаласы бойынша Тариф ҚҚС есебімен 1 кВт/сағ үшін 17,81 теңгені құрайды. («АлматыЭнергоСбыт» ЖШС ресми сайтында ұсынылған мәліметтерге сәйкес).[9]

$$Z_3 = 0,8 \cdot 0,9 \cdot 622 \cdot 18,32 \approx 8204,43 \text{ тг}$$

$$Z_3 = 0,8 \cdot 0,9 \cdot 183 \cdot 18,32 \approx 2413,84 \text{ тг}$$

$$Z_3 = 0,3 \cdot 0,7 \cdot 622 \cdot 18,32 \approx 2392,96 \text{ тг}$$

4.3-кесте – Электр энергиясына кеткен шығындар

Құрылғының атауы	Паспорттық қуат, кВт	Қуатты қолдану коэффициенті	ПӨ құру үшін құрылғының жұмыс уақыты, сағ	Электр энергиясының бағасы, тг/кВт·сағ	Суммасы, тг
Ноутбук ASUS X541U	0,8	0,9	622	17,81	8204,43
Принтер	0,8	0,9	183	17,81	2413,84
Жарық	0,3	0,7	622	17,81	2392,96
Барлығы					13011,23

4.4 Еңбекке ақы төлеу шығындарын есептеу

2019 жылы программисттің орташа жалақысы 185000 тг (Алматы қаласы үшін) құрайды. Қызметкердің бір айдағы жұмыс сағаттары (4.3) формула бойынша анықталады:

$$Ч_м = N_м \cdot Ч_{рд} \quad (4.3)$$

мұндағы, $Ч_м$ – қызметкердің бір айдағы жұмыс сағаттары;
 $N_м$ – бір айдағы жұмыс күндерінің саны;
 $Ч_{рд}$ – бір күндегі жұмыс сағаттарының саны.

$$Ч_м = 21 \cdot 8 = 168 \text{ сағ}$$

Қызметкердің сағаттық қойылымы (4.4) формуласы бойынша есептеле алады:

$$ЧC_i = \frac{ЗП_i}{ФРВ_i} \quad (4.4)$$

Программист:

$$\text{ЧС}_i = \frac{185000}{168} = 1101,19 \text{ тг}$$

мұндағы, ЗП_i – i қызметкердің айлық жалақысы, тг;

ФРВ_i – i қызметкердің айлық жұмыс уақытының қоры, сағат.

Программалық өнімді құрудың еңбек сыйымдылығын анықтау үшін 4.1-кестедегі мәліметтер қолданылады.[9]

Программисттің программалық өнімді құрудағы еңбек сыйымдылығы 3 ад.× сағ. (алгоритмнің блок-схемасын құру, жобаның администраторлық бөлімін құрастыру, программаны баптау және тестілеу).

$$T_2 = 34 + 60 + 51 + 123 + 198 + 105 + 51 = 622 \text{ ад.} \times \text{сағ.}$$

Еңбекке ақы төлеуге кеткен шығындардың жалпы суммасы ($\text{З}_{\text{ТР}}$) (4.5) формуласы бойынша есептеледі:

$$\text{З}_{\text{ТР}} = \sum_{i=1}^n \text{ЧС}_i * T_i \quad (4.5)$$

мұндағы, ЧС_i – i қызметкердің сағаттық қойылымы, тг;

T_i – программалық өнімнің еңбек сыйымдылығын құру, ад.× сағ.;

i – қызметкер категориясы;

n – ПӨ құрумен айналысатын қызметкерлердің саны.

Программист:

$$\text{З}_{\text{ТР}} = 1101,19 \cdot 622 = 684940,18$$

4.4-кесте – Еңбекке ақы төлеу шығындары

Квалификация	ПӨ құрудың еңбек сыйымдылығы, ад.×сағ.	Сағаттық қойылым, тг/сағ	Суммасы, тг
Программист	622	1101,19	684940,18
Барлығы			684940,18

Қосымша жалақы:

$$\text{З}_{\text{доп}} = \text{З}_{\text{ТР}} \cdot 10\% \quad (4.6)$$

$$\text{З}_{\text{доп}} = 684940,18 \cdot 0,1 = 68494,02 \text{ тг}$$

Жалақы қоры:

$$\Phi_{зп} = З_{тр.о} + З_{доп} \quad (4.7)$$

$$\Phi_{зп} = 684940,18 \text{ тг} + 68494,02 = 753434,2 \text{ тг}$$

Әлеуметтік салықты есептеу:

$$H_c = (\Phi_{зп} - ОПВ) \cdot 11\% \quad (4.8)$$

мұндағы, ОПВ – міндетті зейнетақы жарналары -11% $\Phi_{зп}$ -тан.

$$H_c = (753434,2 - (753434,2 \cdot 0,1)) \cdot 0,11 = 74589,98 \text{ тг}$$

Амортизациялық негізгі қорларды есептеу

Амортизациялық аударымдардың жалпы сомасы мынадай формула бойынша анықталады:

$$З_{AM} = \sum_{i=1}^n \frac{\Phi_i \cdot H_{Ai} \cdot T_{НИРi}}{100 \cdot T_{Э\Phi i}} \quad (4.9)$$

мұндағы, Φ_i – i негізгі қорлардың құны, тг;

H_{Ai} – i негізгі қорларды амортизациялаудың бір жылдық нормасы, %;

$T_{НИРi}$ – программалық өнімді құрастыру кезеңіндегі i негізгі қорлардың жұмыс уақыты, сағ;

$T_{Э\Phi i}$ – i негізгі қорлардың бір жылдағы тиімді қор жұмыс уақыты, сағ/жыл;

i – негізгі қорлардың түрі;

n – негізгі қорлардың саны .

Негізгі қорларды амортизациялаудың бір жылдық нормасын есептеу:

$$H_{Ai} = \frac{100}{T_{Ni}} \quad (4.10)$$

мұндағы, T_{Ni} – i негізгі қорларды қолданудың мүмкін уақыты, жыл;

$$H_{Ai} = \frac{100}{4} = 25$$

Программалық өнімді құру үшін программалық қамтаманың жұмыс істеу уақытын анықтау үшін 4.1-кестедегі мәліметтер қолданылады.[8]

Программалық өнімді құру үшін IBM Rational Rose Modeler және MS Visual Studio бойынша жұмыс уақыты 430 сағатты құрайды (жобаның клиенттік бөлігін реализациялау, жобаның администраторлық бөлігін реализациялау, программаны баптау және тестілеу).

$$T_i = 123 + 198 + 105 = 426 \text{ сағ.}$$

Құрылғылар:

$$Z_{AM} = \frac{300000 \cdot 25 \cdot 622}{100 \cdot 1920} = 24296,87 \text{ тг}$$

$$Z_{AM} = \frac{52000 \cdot 25 \cdot 622}{100 \cdot 1920} = 4211,46 \text{ тг}$$

Программалық қамтама:

$$Z_{AM} = \frac{72000 \cdot 25 \cdot 426}{100 \cdot 1920} = 3993,75 \text{ тг}$$

4.5-кесте – Негізгі қорларды амортизациялау (НҚ)

Құрылғы мен ПҚ атауы	Құрылғы мен ПҚ құны, тг	Амортизацияның бір жылдық нормасы, %	Құрылғы мен ПҚ-ның жұмыс уақытының тиімді қоры, сағ/жыл	ПӨ-ді құру үшін құрылғы мен ПҚ-ның жұмыс уақыты, сағ	Суммасы, тг
Ноутбук ASUS X541U	300000	25	1920	622	24296,87
ОС Windows 10	Тегін	-	1920	622	-
IBM Rational Rose Modeler	72000	25	1920	426	3993,75
Toad for Oracle	Тегін	-	1920	426	-
Принтер	52000	25	1920	622	4211,46
Барлығы					32502,08

4.6-кесте – ПӨ құруға арналған шығындар сметасы

Шығындар баптары		Суммасы, тг
Материалдық шығындар, оның ішінде	материалдар	97700
	электр энергиясы	13011,23
Еңбекке ақы төлеу шығындары		680897,18
Әлеуметтік қажеттіліктерге аударымдар		74589,98
Негізгі қорларды амортизациялау		32502,08
Басқа да шығындар		9000

4.5 Программалық өнімнің мүмкін болатын (келісімді) бағасын анықтау

Қолданбалы ПӨ үшін келісімді баға ($Ц_d$) (4.11) формула бойынша есептеледі:

$$Ц_d = Z_{\text{НИР}} \cdot \left(1 + \frac{P}{100}\right) \quad (4.11)$$

мұндағы, $Z_{\text{НИР}}$ – ПӨ құруға кеткен шығындар (4.6-кестеден), тг;
 P – ПӨ рентабельділігінің орташа деңгейі – 25%.

$$Ц_d = 907700,47 \cdot (1 + 0,25) = 1134625,59 \text{ тг}$$

Бұдан әрі қосылған құн салығын (ҚҚС) есепке ала отырып, өткізу бағасы анықталады, ҚҚС ставкасы ҚР заңнамалық Салық Кодексімен белгіленеді. 2019 жылға ҚҚС ставкасы 12% мөлшерінде белгіленген.[9]

ҚҚС есебімен өткізу бағасы (4.12) формула бойынша есептеледі:

$$Ц_p = Ц_d + Ц_d \cdot \text{НДС} \quad (4.12)$$

$$Ц_p = 907700,47 + 907700,47 \cdot 0,12 = 1016624,53$$

4.6 Программалық өнімнің экономикалық бөліміне қорытынды

Банктің несие бөлімінің ақпараттық жүйесін әзірлеуді реализациялау бағасы, барлық мүмкін болатын шығындарды ескере отырып, 907700,47 теңгені құрайды. Шығынның негізгі бөлігін еңбекақы төлеу шығындары құрайды (75,01%).

Программалық өнім құрастыру – бұл үлкен зияткерлік және қаржылық шығындарды, сондай-ақ компьютерлік техниканы талап ететін қымбат жоба. Есептеулерден жобаны реализациялау экономикалық тұрғыдан орынды екенін көруге болады, яғни сапалы өнім мен белсенді жарнама болған кезде белгілі бір уақыт аралығында бұл жоба пайда әкеледі.

5 Өміртіршілік қауіпсіздігі

5.1 Еңбек жағдайларын талдау

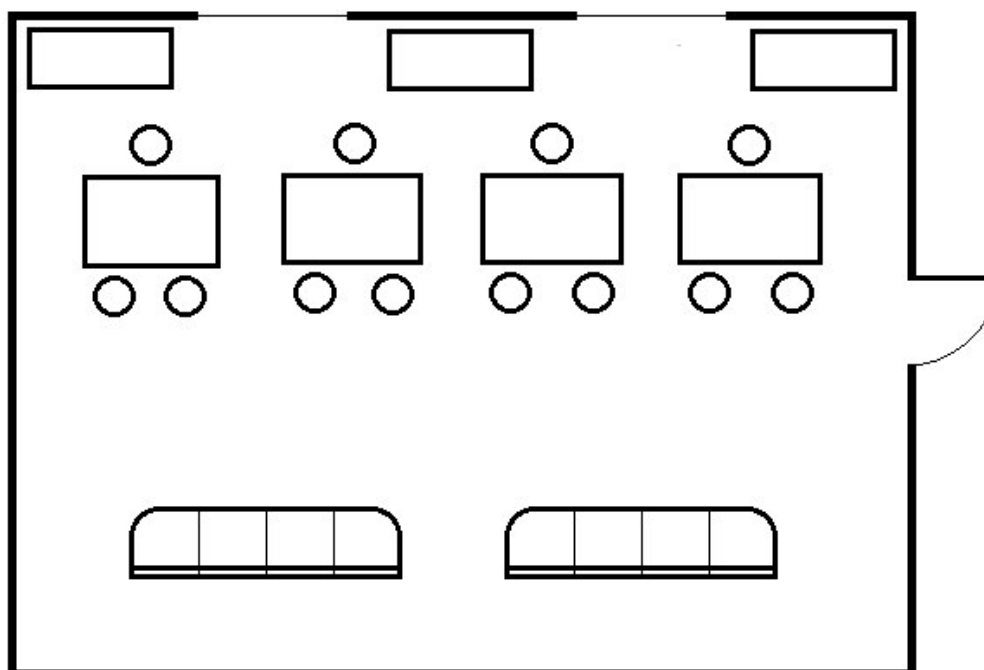
Дипломдық жобаның тақырыбы: “Банктің несиесі бөлімі” ақпараттық жүйесін әзірлеу. Бұл жобаның басты мақсаты – еңбек өнімділігін арттыру, өнім сапасын жақсарту және басқаруды оңтайландыру үшін банктің автоматтандырылған жүйесін әзірлеу.

Персонал кредиттік іс бойынша төрт маманнан тұрады.

Бұл бөлмеде шулы құралдар жоқ және қазіргі кондиционер қажетті температураны, ауа айналымын және ылғалдылықты қолдайды, сондықтан желдеткіш және шу оқшаулағыш жұмыс орнының талабына сәйкес келеді. Бірақ жарық жетіспейді. Бұл дипломдық жобаның осы бөлімінде бөлмені жарықтандыруды есептеу туралы шешім қабылданды.

Бұл бөлменің ұзындығы 10 метр, ені 8 метр және биіктігі 3 метр. Бөлме 4 жұмыс орнына есептелген.

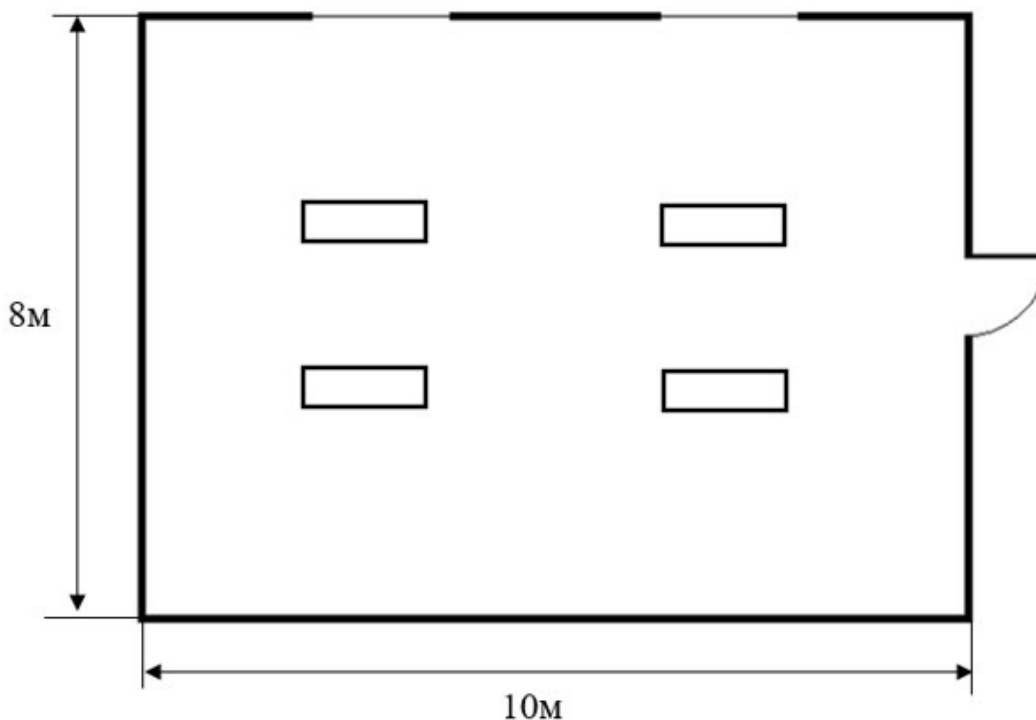
Техникалық қызметкерлер мен жабдықтарды орналастыру үшін таңдалған бөлменің жоспары 5.1-суретте көрсетілген.



5.1-сурет – Бөлменің жоспары

5.2 Табиғи жарықтандыруды есептеу

Бөлменің өлшемдері мынадай: ұзындығы $L = 10$ м; ені $B = 8$ м; биіктігі $H = 3$ м (5.2-сурет). Сондай-ақ екі терезе бар, өлшемдері мынадай: ені $L_o = 2,8$ м және биіктігі $H_o = 1,8$ м. Терезе ауданы – $S_o = 5,04$ м². Төбенің, қабырғалардың және еденнің шағылысу коэффициенттері: 60%, 35%, 25%. Қарсы ғимарат мынадай қашықтықта орналасқан: $L_{зд} = 20$ м, $H_{зд} = 15$ м.



5.2-сурет – Жұмыс кабинетінің схемасы

Жарық ойықтарының ауданы $S_o = 5,04$ м² үй- жайдың қалыпты жарықтандырылуы үшін жеткілікті екенін тексереміз. Есептеу (5.1) формуласы бойынша бүйірлік жарықтандыру кезінде жарық ойықтарының ауданын алдын ала анықтаудан тұрады. [20]

$$100 \frac{S_o}{S_n} = \frac{e_n K_3 \eta_0}{\tau_0 r_1} K_{зд} \quad (5.1)$$

мұндағы, S_o – бүйірлік жарықтандыру кезіндегі жарық ойықтарының ауданы, м²;

S_n – бөлменің еденінің ауданы, м²;

e_n – нормаланатын мән КЕО;

$K_3 = 1,5$ – жарық өткізетін материалдың тік орналасуы кезіндегі қор коэффициенті, ҚНМЕ II-4-79 бойынша таңдалады.

τ_0 – жарық өткізудің жалпы коэффициенті;

r_1 – бөлме бетінен шағылысқан және ғимаратқа жапсарлас төселетін қабаттың арқасында бүйірлік жарықтандыру кезінде КЕО көтерілуін ескеретін коэффициент;

$K_{зд} = 1$ – қарсы тұрған ғимараттармен терезелердің қараңғылануын ескеретін коэффициент, 5.1-кесте бойынша қабылдайды.

5.1- кесте – Қарсы тұрған ғимараттармен терезелердің қараңғылануын ескеретін коэффициенттің мәні

$L_{зд}/H_{зд}$	0,5	1	1,5	2	3 и более
$K_{зд}$	1,7	1,4	1,2	1,1	1

Еден ауданы (5.2) формула бойынша анықталады

$$S_n = L * B \quad (5.2)$$

Осылайша, (5.2) формуласы бойынша еден ауданы тең

$$\begin{aligned} L &= 10 \text{ м,} \\ B &= 8 \text{ м,} \\ S_n &= 10 \cdot 8 = 80 \text{ м}^2 \end{aligned}$$

Табиғи жарықтандыру коэффициентінің нормаланған мәні (5.3) формула бойынша есептеледі

$$e_N = e_n * m_N \quad (5.3)$$

мұндағы, N – табиғи жарықпен қамтамасыз ету жөніндегі әкімшілік-аумақтық аудан тобының нөмірі;

$e_n = 1,5$ – осы бөлмедегі көру жұмыстарының сипаттамасына және табиғи жарықтандыру жүйесіне байланысты ҚНЖЕ 23-05-95 бойынша таңдалатын табиғи жарықтандыру коэффициентінің мәні, яғни бүйірдегі табиғи жарықтандыру кезіндегі КЕО мәні;

$m_N = 0,85$ – ҚНЖЕ 23-05-95 кестесі бойынша жарық ойықтарының түріне, олардың көкжиектің жақтарына және әкімшілік ауданы тобының нөміріне бағытталуына байланысты жарық климатының коэффициенті, оңтүстіктен сыртқы қабырғалардағы жарық климатының коэффициенті.[19]

(5.3) формула бойынша әртүрлі аудандарда орналасқан ғимараттар үшін КЕО e_N нормаланған мәндері мынаған тең

$$e_N = 1,5 \cdot 0,85 = 1,275 \%$$

Бір жақты жарықтандыру кезіндегі бөлменің тереңдігі

$$l = B - 1 = 8 - 1 = 7 \text{ м}$$

Жарықтылық сипаттамасының η_0 мәнін алу үшін, төмендегі қатынастар есептелген

$$\frac{L}{l} = \frac{10}{7} = 1,43$$

$$\frac{l}{h_1} = \frac{7}{2} = 3,5$$

$$\eta_0 = 8.5$$

Жарықтылық сипаттамасының η_0 мәні 5.2-кесте бойынша қабылданады.

5.2-кесте – Жарықтылық сипаттамасының мәні

Бөлменің ұзындығының (L) оның тереңдігіне (l) қатынасы	Бөлменің тереңдігінің оның шартты жұмыс бетінің деңгейінен терезенің үстіне дейінгі биіктігіне қатынасы кезіндегі жарық сипаттамасының мәні (h_1)							
	1	1,5	2	3	4	5	7,5	10
4 және жоғары	6,5	7	7,5	8	9	10	11	12,5
3	7,5	6	8,5	9,6	10	11	12,5	14
2	8,5	9	9,5	10,5	11,35	15	17	17
1,5	9,5	10,5	13	15	17	19	21	23
1	11	15	16	18	21	23	26,5	29
0,5	18	23	31	31	45	54	66	-

Жарық өткізудің жалпы коэффициенті τ_0 (5.4) формула бойынша анықталады

$$\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4 \quad (5.4)$$

мұндағы, $\tau_1 = 0,8$ – 5.3-кесте бойынша анықталатын материалды жарық өткізу коэффициенті;

$\tau_2 = 0,7$ – 5.4-кесте бойынша анықталатын терезе түптеуіндегі жарықтың жоғалуын ескеретін коэффициент;

$\tau_3 = 0,8$ – 5.5- кесте бойынша анықталатын көтергіш конструкциялардағы жарықтың жоғалуын ескеретін коэффициент;

$\tau_4 = 0,8$ – күннен қорғайтын құрылғылардағы жарықтың жоғалуын ескеретін коэффициент. 5.6-кесте бойынша анықталады;[19]

5.3-кесте – τ_1 коэффициентінің мәні

Жарық өткізетін материалдың түрі		τ_1
Терезелік беттік әйнек	бірлік	0,9
	екілік	0,8

5.3-кестенің жалғасы

Жарық өткізетін материалдың түрі		τ_1
Терезелік беттік әйнек	үштік	0,75
Қалыңдығы 6-8 мм витринді әйнек		0,8
Арматураланған беттік әйнек		0,6
Өрнектелген беттік әйнек		0,65
Арнайы қасиеттері бар беттік әйнек	күннен қорғайтын	0,65
	контрастты	0,75
Органикалық әйнек	мөлдір	0,9
	сүттік	0,6
Бос әйнек блоктар		0,5
Жарық мөлдір		0,55
Әйнек пакеттер		0,8

5.4-кесте – τ_2 коэффициентінің мәні

Өнеркәсіптік ғимараттардың терезелеріне арналған түптеу түрі		τ_2
Ағаш қаптамалар	бірлік	0,75
	қосарланған	0,7
	екілік, бөлектенген	0,6
Болат қаптамалар	бірлік, ашылатын	0,75
	бірлік, естімейтін	0,9
	екілік, ашылатын	0,6
	екілік, естімейтін	0,8

5.5-кесте – τ_3 коэффициентінің мәні

Жабындардың салмақ түсетін конструкциялары		τ_3
Болат пішіндер		0,9
Темір бетонды және ағаш қалыптар мен аркалар		0,8
Қима биіктігі кезінде тұтас аркалықтар мен рамалар	50 см және жоғары	0,8
	50 см төмен	0,9

5.6-кесте – τ_4 коэффициентінің мәні

Күннен қорғайтын құрылғылар, бұйымдар мен материалдар		τ_4
Реттелетін жалюздер мен перделер (қабырға аралық ішкі, сыртқы)		1
Терезе жазықтығына 900 бұрыш астында жалюзи пластиналары немесе экрандар орналасқан кезде қорғаныс бұрышы 450-ден аспайтын стационарлық жалюзи мен экрандар	көлденең	0,65
	тік	0,75
Көлденең күнқағарлар	Қорғау бұрышы 300-ден аспайтын	0,8
	Қорғау бұрышы 150-ден 450-ге дейін	0,9-0,6

(5.4) формулаға белгілі мәндерді қойып, τ_0 табамыз

$$\tau_0 = 0,8 \cdot 0,7 \cdot 0,8 \cdot 0,8 = 0,36$$

5.7-кесте бойынша бүйірлік жарықтандыру үшін r_1 коэффициентін анықтаймыз. Ол үшін мыналарды табамыз:

а) бөлменің тереңдігінің шартты жұмыс бетінің деңгейінен терезенің үстіне дейінгі биіктікке қатынасы:

$$\frac{l}{h_1} = \frac{7}{2} = 3,5$$

ә) бөлменің тереңдігінің бөлменің еніне қатынасы

$$\frac{l}{B} = \frac{7}{8} = 0,87$$

б) бөлменің ұзындығының оның тереңдігіне қатынасы

$$\frac{L}{l} = \frac{10}{7} = 1,43$$

Төбенің, қабырғалардың және еденнің орташа өлшенген коэффициенті ρ_{cp} (5.5) формула бойынша анықталады

$$\rho = \frac{\rho_{пот} + \rho_{ст} + \rho_{пол}}{3} \quad (5.5)$$

Осылайша, (5.5) формуласы бойынша

$$\rho = \frac{\rho_{пот} + \rho_{ст} + \rho_{пол}}{3} = \frac{60 + 35 + 25}{3} = 40\% = 0,4$$

$$r_1 = 1,04$$

5.7-кесте – Бүйірлік жарықтандыру үшін r_1 коэффициентінің мәні

Бөлменің тереңдігінің (l) терезенің үстіне дейінгі шартты жұмыс беті деңгейінің биіктігіне қатынасы (h_1)	Бөлменің тереңдігінің (l) бөлменің еніне (B) қатынасы	Еденнің, қабырғалардың және төбенің орташа өлшенген коэффициенті, ρ_{cp}		
		0,4		
		Бөлменің ұзындығының (L) оның тереңдігіне (l) қатынасы		
		0,5	1	2
2,5-тен 3,5-ке дейін	0,1	1	1	1
	0,2	1,12	1,01	1,01
	0,4	1,28	1,25	1,20
	0,6	1,95	1,86	1,67

5.7-кестенің жалғасы

Бөлменің тереңдігінің (<i>l</i>) терезенің үстіне дейінгі шартты жұмыс беті деңгейінің биіктігіне қатынасы (<i>h</i> ₁)	Бөлменің тереңдігінің (<i>l</i>) бөлменің еніне (<i>B</i>) қатынасы	Еденнің, қабырғалардың және төбенің орташа өлшенген коэффициенті, $\rho_{\text{ср}}$		
		0,4		
		Бөлменің ұзындығының (<i>L</i>) оның тереңдігіне (<i>l</i>) қатынасы		
		0,5	1	2
2,5-тен 3,5-ке дейін	0,8	1,04	1,04	1,03
	1	1,14	1,12	1,1

Барлық табылған мәндерді орналастыра отырып, (5.1) формуласымен бүйірлік жарықтандыру кезінде жарық ойықтарының ауданын S_0 табамыз

$$S_0 = \frac{S_n e_N K_3 \eta_0 K_{зд}}{100 \tau_0 r_1} = \frac{80 \cdot 1,275 \cdot 1,5 \cdot 8,5 \cdot 1}{100 * 0,36 * 1,04} = 34,73 \text{ м}^2 \approx 35 \text{ м}^2$$

Қорытынды. Бұл бөлмеде жарық ойығының ауданы тексерілді және ол жеткілікті емес ($S_0 = 35 > S_{\phi} = 10,08$). Осыдан, тәуліктің кешкі уақытында және қыста қосымша жарық көздері қажет деген қорытынды жасадым.

5.3 Жасанды жарықтандыруды есептеу

Бұл бөлмеде жасанды жарықтандыру жарық ағыны 3570 лм болатын ЛЛ Т8 G13 (65 Вт) типті люминесцентті шамдар қолданылатын TLPL236 типті 4 шамшырақ пайдалану арқылы жүзеге асырылады. 300 лк жарықтандыру жасау үшін қажетті шамдардың санын қолдану коэффициенті әдісімен есептейміз. Ол көру жұмыстарының III–ші разрядын қамтамасыз ету үшін жеткілікті болып табылады.

Шамдардың санын анықтау үшін мына формуланы қолданамыз

$$N_{\text{л}} = \frac{S_{\text{помещ}} K_3 Z E_{\text{н}}}{\Phi_{\text{л}} n \eta} \quad (5.6)$$

мұндағы, Z – минималды жарықтану коэффициенті (орташа және минималды жарықтану қатынасы). Есептерде Z коэффициенті мына шектерде қабылданады: $1,1 \div 1,2$;

$n = 2$ – шамшырақтағы шамдардың саны;

η – шағылысу коэффициенттері мен бөлменің индексіне байланысты пайдалану коэффициенті.

Жарықтандырудың бірқалыпты еместігін сипаттайтын Z коэффициенті көптеген айнымалылардың функциясы болып табылады және ең үлкен

дәрежеде шамдардың арасындағы қашықтықтың шамдардың ілінуінің есептік биіктігіне (L/h) қатынасына байланысты болады, оны ұлғайтумен ұсынылған Z мәні күрт жоғары өседі. λ ұсынылған мәндерден аспайтын жағдайда мынадай мәндерді қабылдауға болады:

а) $Z = 1,15$ – қыздыру шамдары және ДРЛ үшін;

ә) $Z = 1,1$ – люминесцентті шамдар үшін.[19]

Жарықтандыру қондырғысын пайдалану кезінде жұмыс беттеріндегі жарықтандыру жарық көздерінің жарық ағынын азайту, шамдар мен жарықтандыру арматурасының ластануы, сондай-ақ жарықтандырылатын үй-жайдың қабырғалары мен төбесінің ластануы есебінен төмендейді. Сондықтан жарықтандыру қондырғысының қуатын анықтау кезінде қор коэффициенті енгізіледі. Қор коэффициенті ауаның шаңмен, түтінмен және т. б. ластану дәрежесіне байланысты. [20]

Жарықтану қондырғысын пайдалану коэффициенті η – бұл жұмыс бетіне түсетін жарық ағынының, жіберілетін жарық ағынына қатынасы. Пайдалану коэффициенті шамның түріне, бөлменің геометриялық өлшемдеріне және беттердің шағылысу коэффициенттеріне байланысты. Пайдалану коэффициентін анықтау үшін бөлменің индексін анықтау қажет:

$$i = \frac{S_{\Pi}}{(L+B)h_1} = \frac{A \cdot B}{(L+B)h_1} \quad (5.7)$$

мұндағы, S , L , B – сәйкесінше бөлменің ауданы, ұзындығы және ені. Барлық мәндерді (5.7) формулаға қойып, мынадай нәтиже аламыз

$$i = \frac{A \cdot B}{(L + B)h_1} = \frac{10 \cdot 8}{(10 + 8) \cdot 2} = 2,22$$

Шамшырақтың тобын анықтаймыз және 5.8-кесте бойынша $\eta = 0.65$ табамыз.

5.8-кесте – Люминесцентті шамдардың жарық ағынын пайдалану коэффициенті

Төбе	80	80	80	70	50	50	30	0
Қабырғалар	80	50	50	50	50	30	30	0
Еден	30	30	10	20	10	10	10	0
0,6	65	43	34	41	40	34	33	28
0,8	74	53	43	50	48	42	41	36
1	81	60	49	57	54	48	48	42
1,25	87	69	57	64	61	56	55	49
1,5	91	74	62	69	65	60	59	54
2	96	82	68	76	70	66	65	60
2,5	100	87	73	80	74	71	70	65
3	102	92	77	84	78	75	73	69

5.8-кестенің жалғасы

Төбе	80	80	80	70	50	50	30	0
Қабырғалар	80	50	50	50	50	30	30	0
Еден	30	30	10	20	10	10	10	0
4	105	96	80	87	80	78	76	72
5	106	99	83	90	82	80	79	75

Көрсетілген әдіс бойынша есептеу кезінде бір шамның қажетті жарық ағыны (5.8) формула бойынша анықталады:

$$\Phi_{л} = \frac{E_{\min} \cdot S \cdot Z \cdot K_3}{N \cdot \eta \cdot n} \quad (5.8)$$

мұндағы, E_{\min} – минималды нормаланған жарықтану, лк;
 K_3 – қор коэффициенті;
 S – жарықтанатын аудан, м²;
 Z – минималды жарықтану коэффициенті (жарықтанудың біркелкі емес коэффициенті);
 N – шамшырақтардың саны;
 n – шамшырақтағы шамның саны;
 η – бірлік үлесіндегі жарық ағынын пайдалану коэффициенті.

Жасанды жарықтандырудың жеткілікті немесе жеткіліксіз екенін тексеру үшін E_{\min} есептеу керек, ол 300 лк-ден жоғары болуы керек

$$E_{\min} = \frac{\Phi_{л} \cdot N \cdot n \cdot \eta}{S \cdot Z \cdot K} = \frac{3570 \cdot 4 \cdot 2 \cdot 0,65}{80 \cdot 1,1 \cdot 1,3} = 162$$

Есептеу бойынша, E_{\min} бізге керек мәннен аз. Сондықтан, бұл проблеманы шешу керек. Мен жарық ағыны 3570 лм, қуаты 65 Вт болатын ескі люминесцентті шамдарды жарық ағыны 4510 лм, қуаты 80 Вт болатын жаңа люминесцентті шамдарға ауыстыру керек деп шештім.

(5.6) формуласы бойынша, жарық ағынының мәнін (4510 лм) қолданып, 300 лк жарықтықты құру үшін қажетті шамшырақтар санын анықтау керек

$$N = \frac{S_{\text{помещ}} K_3 Z E_H}{\Phi_{л} n \eta} = \frac{80 \cdot 1,3 \cdot 1,1 \cdot 300}{4510 \cdot 2 \cdot 0,65} = 5,85 \approx 6 \text{ шамшырақ}$$

Яғни, көру жұмыстарының III–ші разряды үшін және 300 лк жарықтық құру үшін қуаты 80 Вт және жарық ағыны $\Phi = 4510$ лм болатын 6 шамшырақ (12 шам) қажет.

Жұмыс орнының үстінде шамдарды ілу биіктігін мына формула бойынша анықтаймыз

$$h_{\text{расч}} = H - (H_{\text{св}} + H_{\text{р.п.}}) \quad (5.9)$$

мұндағы, $H_{\text{св}} = 0,3$ – шамдардың салбырау биіктігі, м;
 $H_{\text{р.п.}} = 1$ – еден үстіндегі жұмыс бетінің қашықтығы, м;
 $H = 3$ – бөлменің биіктігі, м.

Сонда шамдарды ілу биіктігі

$$h_{\text{расч}} = H_{\text{пом}} - (H_{\text{св}} + H_{\text{р.п.}}) = 3 - (0,3 + 1) = 1,7 \text{ м}$$

Шамдардың арасындағы қажетті қашықтықты мына формула бойынша анықтаймыз

$$L = \lambda \cdot h \quad (5.10)$$

мұндағы, L – Көрші шамдардың немесе люминесцентті шамдардың қатарлары арасындағы қашықтық;

h – Шамның жұмыс бетінен ілу биіктігі.

Люминесцентті шамдары бар шамшырақтарды пайдаланған кезде
 $\lambda = 1,2 \div 2,4$.

Осылайша, шамдар арасындағы қажетті қашықтық

$$L = \lambda * h = 2,4 * 1,7 = 4,08 \text{ м}$$

Шамдардың қатарлары арасындағы қашықтық

$$L_b = \lambda \cdot h_p = 2,4 \cdot 1,3 = 3,12 \text{ м}$$

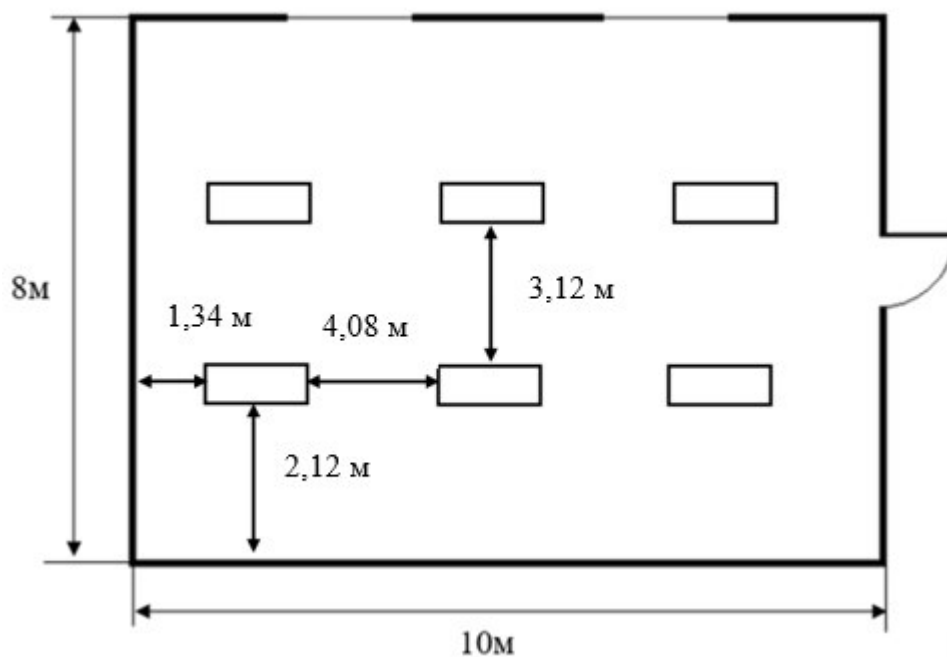
Шам мен қабырға арасындағы қашықтық

$$l_a = l_b = \frac{L_b}{3} + [0,3; 0,5] = \frac{3,12}{3} + 0,3 = 1,34 \text{ м}$$

Шамдар мен қабырға арасындағы қашықтық

$$L_a = L_b - 1 = 3,12 - 1 = 2,12 \text{ м}$$

Жарық шамдары мен жарық ойықтарын орналастыру сұлбасы 5.3-суретте көрсетілген.



5.3-сурет – Бөлмедегі жарық шамдары мен жарық ойықтарын орналастыру сұлбасы

Қорытынды. Жүргізілген есептеулер негізінде жасанды жарықтандыру қайта жанартылды. Жарық ағыны 3570 лм және қуаты 65 Вт болатын ескі люминесцентті шамдарды жарық ағыны 4510 лм және қуаты 80 Вт болатын жаңа люминесцентті шамдарға ауыстыру және шамшырақтардың (типі – TLPL236) санын 4-тен 6-ға дейін көбейту керек деген шешім қабылданды.

Қорытынды

Бұл дипломдық жобаның нәтижесінде банктің кредиттік бөлімінің жұмысын автоматтандыратын ақпараттық қосымша құрылды. Ақпараттық қосымшаны құру үшін дипломдық жұмыстың бірінші бөлімінде қазіргі кезде қолданылып жатқан кредитті есептейтін программалық өнімдері және кредиттік бөлімнің қызметкерлерінің жұмысы, кредит беру барысында орындалатын банктік операциялар туралы мәліметтер жиналды.

Дипломдық жұмыстың екінші бөлімінде мәліметтер қоры түсінігі, мәліметтер модельдері, мәліметтер қорына қойылатын талаптар және мәліметтер қорын басқару жүйесін құруға арналған программалық өнімдер, яғни, MySQL, SQL, МҚБЖ-нің артықшылықтары қарастырылып, нәтижесінде басқа МББЖ-лармен салыстырғанда мәліметтер типтерінің көптігімен және қолдануға жеңілдігімен ерекшеленетін MS SQL МҚБЖ қолданылды. С# МҚБЖ-мен байланыс жасалынды, сонымен қатар сұраныс жасау үшін SQL тілі қолданылды. Бөлімнің соңында мәліметтер қорының құрылымы және кестелер арасындағы байланыс сипатталды.

Үшінші бөлімде қарастырылған объектіге бағытталған программалау ортасының ерекшеліктеріне қарап, қосымша осы ортада құрылды. Қолданушы үшін банктің кредиттік бөлімінің ақпараттық қосымшасына нұсқау жазылды. Қосымша келесі қызметтерді орындайды: клиентті банктің кредиттері туралы мәліметтермен таныстыру, клиенттің арызын қабылдау, келісім жасау, клиенттің айлық төлемін баспаға шығару, олардың төлемдерін қабылдау, төлемді растайтын чек шығару, борышкерлердің электронды почтасына ескерту хатын жіберу, берілген аралықтағы банктен алынған кредиттердің жалпы сомасын шығаратын және борышкерлердің тізімін және кредиттерінің сомасын шығаратын есеп берулер шығару.

Құрылған ақпараттық қосымша банктің кредиттік бөлімінің қызметкерлерінің жұмысын анағұрлым жеңілдетеді. Бұл қосымша болашақта жетілдірілуі мүмкін.

Әдебиеттер тізімі

- 1 Г. Буч, Объектно ориентированный анализ и проектирование с примерами приложений на C++. – М.: «Издательство Бином», 1999г.
- 2 У. Боггс, М.Боггс Rational Rose & UML – М.: «Издательство Лори», 2000.
- 3 Белоглазова Г.Н., Кроливецкая Л.П. Банковское дело организация деятельности коммерческого банка. Учебник для вузов. – М.:Издательство Юрайт, 2011 г. – 422 с.
- 4 Киреев, В. Л. Банковское дело : учебник / В. Л. Киреев, О. Л. Козлова. – М: КНОРУС, 2012. – 239 с
- 5 Вендров А.М. Проектирование программного обеспечения экономических информационных систем – М.: Финансы и статистика, 2000.
- 6 Т. Квартони, Rational Rose и UML. Визуальное моделирование. – М.: ДМК, 2001.
- 7 Горелая, Н. В. Организация кредитования в коммерческом банке : учеб. пособие / Н. В. Горелая. – М. : Форум : ИНФРА-М, 2012. – 207 с.
- 8 Курсов, В. Н. Бухгалтерский учет в коммерческом банке: новые типовые бухгалтерские проводки операций банка : учеб. пособие / В. Н. Курсов, Г. А. Яковлев. – 15-е изд., испр. и доп. – М. : ИНФРА-М, 2012. – 364 с.
- 9 Фаулер М., Скот К., UML в кратком изложении. Применение стандартного языка объектного ориентирования. – М.: ДМК, 2002.
- 10 Г. Буч, Рембо Д., Джекобсен А., Язык UML. Руководство пользователя – М.: ДМК 2000.
- 11 Схемы алгоритмов, программ, данных и систем. ГОСТ 19.701-90. ЕСПД. – М.: Изд-во стандартов, 1994.
- 12 Бекишева А.И. Методические указания к выполнению экономической части дипломной работы для бакалавров специальности 5В0703 – Информационные системы – Алматы: АУЭС; 2013.-24 с
- 13 Боканова Г.Ш., Еркешева З.Д. Экономика и организация производства. Методические указания по выполнению расчетно-графических работ для студентов специальности 5В070200 – Автоматизация и управление. – Алматы: АУЭС, 2017. – 29 с.
- 14 Токмаков Г.П. Базы данных. Концепция баз данных, реляционная модель данных, языки SQL и XML: учебное пособие / Г.П. Токмаков. – Ульяновск: УлГТУ, 2010. – 192 с.
- 15 Автоматизированные информационные технологии в экономике: Учебник / Под ред. проф. Титоренко Г.А. – М.: Компьютер, ЮНИТИ, 1999.
- 16 Мацяшек Л.А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML. Пер. с англ. – М.: Издательский дом «Вильямс», 2002.
- 17 Адам Фримен. ASP.NET MVC 4 с примерами на C# 5.0 для профессионалов. Вильямс, 2013. - 688 с

18 Кристиан Нейгел, Билл Ивьен, Джей Глинн, Морган Скиннер, Карли Уотсон. С# 2005 и платформа .NET 3.0 для профессионалов. Вильямс, 2008. - 1790 с

19 Кристиан Нейгел, Билл Ивьен, Джей Глинн, Карли Уотсон, Морган Скиннер. С# 4.0 и платформа .NET 4 для профессионалов (+ CD-ROM). Вильямс, 2011. - 1440 с

20 Герберт Шилдт. С# 4.0. Полное руководство. Вильямс, 2017. - 1056 с

21 Яргер Р., Риз Дж., Кинг Т. MySQL и MSSQL, Арачу. Базы данных для небольших предприятий и Интернета – СПб.: Символ-Плюс, 2000. – 150 с.

22 Леоненков А. «Самоучитель UML». – СПб.: БХВ – Петербург, 2004. – 215 с.

23 Косолапова, Н.В. Безопасность жизнедеятельности: Учебник / Н.В. Косолапова, Н.А. Прокопенко. – М.: КноРус, 2013. – 192 с.

24 Маликов, А.Н. Безопасность жизнедеятельности: Учебное пособие / Ш.А. Халилов, А.Н. Маликов, В.П. Гневанов; Под ред. Ш.А. Халилов. – М.: ИД ФОРУМ, ИНФРА-М, 2012. – 576 с.

А Қосымшасы (міндетті)

Техникалық тапсырма

А.1 Кіріспе

Ақпараттық қосымша – қолданушымен тікелей байланыс жасауға негізделген, қолданушылар тапсырмасын орындауға арналған программа.

Қолдану аймағына және функцияларына байланысты ақпараттық қосымшаларға келесі талаптар қойылады:

– ақпараттық қосымша ақпараттарды жинауға, өңдеуге және сақтауға арналған. Сондықтан ақпараттық қосымшаның сақтау ортасы мен мәліметтерге қатынасы оның негізі болып табылады, сақтау ортасы сәйкес аймаққа қатысты және сақтау деңгейі сенімді болуы керек;

– ақпараттық қосымшаның интерфейстері қолданушы үшін оңай игерілетін және ыңғайлы болуы тиіс, қазіргі кездегі ақпараттық қосымшаларда дамыған графикалық интерфейснің болуы міндетті болып саналады.

А.1.1 Мақсаты

Бұл дипломдық жұмыстың мақсаты:

– несиелік операциялар туралы ақпаратты өңдеуші банк қызметкерлерінің және несие бөлімінің қызметкерлерінің еңбек өнімділігін арттыру;

– клиентті несие түрлерімен таныстыру;

– клиенттерге қызмет көрсету сапасын арттыру;

– сонымен қатар клиенттің арызын қабылдап, кредит алуына келісім жасайтын, клиенттің төлемдерін қабылдайтын банктің несие бөлімінің автоматтандырылған ақпараттық қосымшасын құру болып табылады;

Ақпараттық қосымшаның негізгі орындайтын функциялары:

– клиентке кредит түрлері туралы ақпарат беру;

– жылдық несиеден түскен пайданы есептеу, мониторинг жасау.

Бұл қосымша несие алушыны несие туралы ақпаратпен қамтамасыз етіп, несие беру операцияларын жүзеге асырып, қағазбасты технологиядан құтылуға жағдай тудырады.

Дипломдық жоба нәтижесінде құрастырылған жүйе тұтынушы қауымға белгілі деңгейде пайдасын тигізеді.

А.1.2 Қолдану аумағы

Негізінен банк саласында қолдануға болады

А.2 Жалпы сипаттама

Әрбір үлкен не кіші банкте бөлімдер болады. Олардың әрқайсысы бір бірінен ерекшеленген әлем болып саналады, онда өзіне тән принциптер мен ережелер сақталады. Ұйымның жұмысы тиімді болу үшін бөлімдер өзара тиімді әрекеттесе алу қажет. Қайсы бөлім маңызды, қайсысы маңызды емес екенін анықтап отырудың қажеті жоқ. Олардың бәрі маңызды. Алайда олардың ішінен бірінші дәрежелі роль ойнайтындарын ерекшелеп көрсетуге болады.

Несиелік ұйымдардағы ең маңызды бөлімдерді атап көрсетейік:

- есептік-кассалық қызмет;
- валюталық операциялар;
- бағалы қағаздар;
- пластикалық карталар;
- несие бөлімі.

Бұл бөлімдер банктің экономикалық жағдайын анықтайды. Сондықтан осы бөлімдердің жұмысын ұтымды ұйымдастыра алу қажет.

Енді несие бөлімінің жұмысы үш кезеңнен тұрады:

- несие беру жөніндегі арызды рәсімдеу;
- арызды қарастыру және несие беру жөнінде шешім шығару;
- несие беру және қарыз сомасы жөнінде келісім шартқа қол қою.

А.2.1 Қолданушылар интерфейсі

Бұл жүйеде негізінен мынадай интерфейстер қарастырылады:

- а) деректер қордан мәліметті іздеп табуға арналған интерфейс;
- ә) деректер қорға мәлімет енгізуге арналған интерфейс;
- б) деректерді баспаға шығаруға арналған интерфейс;

А.2.2 Аппараттық интерфейстер

Сервер үшін:

- процессор типі – Intel Core i7-6500U;
- жедел жады – 8 Гб;
- қатты диск жадысы – 500 Гб;
- монитор - 17’’;
- тінтуір - PS/2 стандартты;
- пернетақта (орысша/қазақша);

Жүйенің қолданушылары және администраторы үшін:

- процессор типі – Intel Core i7-6500U;
- жедел жады – 8 Гб;
- қатты диск жадысы – 500 Гб;
- монитор - 17’’;

- тінтуір - PS/2 стандартты;
- пернетақта (орысша/қазақша).

А.2.3 Бағдарламалық интерфейс

Бағдарламалық қамтамасыздандыруды құру:

- а) Microsoft Windows операциялық жүйесі;
- ә) Visual Studio пакеті;
- б) Жобадауды модельдеу үшін, RationalRose құралы;
- в) Деректер қорын модельдеу үшін ERWin құралы;

А.2.4 Адаптация бойынша талаптар

Жүйе баспаға деректерді қазақ тілінде шығаруына мүмкіндік беруі керек.

А.2.5 Пайдалану шарттары

«Кредитный инспектор» жүйесін пайдалану шарттары, дербес компьютерлерді пайдаланудағы стандартты шарттар сияқты.

А.3 Бөлшектік талаптары

А.3.1 Функционалдық талаптар

Орындалатын функцияларға талаптар:

- а) Мәліметтерді сақтайтын деректер қоры қажет;
- ә) Клиентті ИИН нөмірі бойынша іздеп табу;
- б) Несие төлеу мерзімін кешіктірген клиентке хабарлама жіберу.
- в) Таңдалған клиентпен байланыс құру деректерін көруге мүмкіндік беретін арнайы бетті ашу;
- г) Барлық деректер қорындағы ақпараттарды өзгертуге мүмкіндік жасау;
- ғ) Қолданушы таңдаған ақпаратты баспаға шығаруына мүмкіндік жасалуы қажет;

А.3.2 Функционалдық емес талаптар

А.3.2.1 Сенімділігі мен қатынау

Осы бағдарламаның тұрақты түрде жұмыс істеуі үшін келесі нұсқауларды орындаған жөн:

А Қосымшасының жалғасы

- негізгі желілік қызметтерге қатынас құруды қамтамасыз ету;
 - жүйемен байланыс жұмыс басталу уақытынан бастап қосылуы керек.
- Жүйе жұмысы жұмыс уақытының аралығында тоқтап қалмауы тиіс. Профилактикалық жұмысты белгілі бір уақытта жүргізу керек;
- авариялық жағдайларды болдыртпау;

А.3.2.2 Қателерді өңдеу

Қателер болған кезде (қосымша генерацияламайтын) қолданушыны ескертетін хабарламалар шығып отыруы қажет.

А.3.2.2 Шектеулер

- а) жабдықтар және тілдерге шектеулер: Microsoft Windows ортасында өңделеді;
- ә) қолданылатын платформалар: Intel платформасының процессорларында құрылған компьютерлерде жұмыс істейді.

А.3.3 Қабылдау және бақылау реті

Бағдарламаны сәйкестікке тестілеу және қабылдау оны дербес компьютерге орнату кезінде жасалады. Бағдарламаның дұрыс жұмыс істеуін қамтамасыз ету үшін барлық қажетті техникалық және бағдарламалық құралдарды орнату іске асырылады. Сол сияқты қажетті өзгертулер мен толықтыруларды жүйені пайдалану үрдісі кезінде енгізуге болады.

Құрудың негізгі қадамдары А.1-кестеде көрсетілген.

А.1-кесте – Құрудың деңгейлері мен қадамдары

№	Қадам мазмұны	Жобалау құжаты	Құру нәтижесі
1	Талаптарды анықтау	Талаптардың мақұлдануы	
2	Техникалық тапсырма	Техникалық тапсырма	
3	Қойылған есептер	Қойылған есептер	
4	Жобалау	Кластар диаграммасы	
5	Іске асыру	Бағдарлама мәтіні	
6	Өңдеу және тестілеу	Тестілеу нәтижелері	
7	Бағдарлама сипаттамасы	Қолданушы нұсқауы	

Ә қосымшасы
(міндетті)

Программа листингі

Авторизациядан өту коды:

```
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;
using System.Data.SqlClient;
namespace Кредитный_инспектор
{
publicpartialclass Login : Form
{
publicstaticstringstr;
publicLogin()
{
this.StartPosition = FormStartPosition.CenterScreen;
InitializeComponent();
}
privatevoid button1_Click(object sender, EventArgs e)
{
inti;
str = textBox1.Text;
Reading login = newReading();
login.readf("Users.txt");
var logins = login.str.Split(new[] { "," }, StringSplitOptions.RemoveEmptyEntries);
intcountlogin = logins.Count();
Reading pass = newReading();
pass.readf("Passwords.txt");
var passwords = pass.str.Split(new[] { "," }, StringSplitOptions.RemoveEmptyEntries);
intcountpass = passwords.Count();
for (i = 0; i < countlogin; i++)
{
if (logins[i] == textBox1.Text && passwords[i] == textBox2.Text)
{
label1.Text = "Верно!"; label1.ForeColor = Color.Black;
MessageBox.Show(" Для входа в базу данных нажмите OK", " Верны данные.",
MessageBoxButtons.OK, MessageBoxIcon.Information);
this.Hide();
Main Main = newMain();
Main.FormClosed += (s, args) => this.Close();
}
}
}
}
```

Ә қосымшасының жалғасы

```
Main.Show();
break;
}
else{ label1.Text = "Логинилипароль НЕ верны!"; label1.ForeColor = Color.Red; }
}}
privatevoid button2_Click(object sender, EventArgs e)
{
this.Hide();
Form1 Form1 = new Form1();
Form1.FormClosed += (s, args) =>this.Close();
Form1.Show();
}}}
```

Пайдаланушыны тіркеу коды:

```
using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Threading.Tasks;
usingSystem.Windows.Forms;
namespaceКредитный_инспектор
{
publicpartialclassReg : Form
{
publicReg()
{
this.StartPosition = FormStartPosition.CenterScreen;
InitializeComponent();
}
privatevoid button2_Click(object sender, EventArgs e)
{
this.Hide();
Form1 Form1 = new Form1();
Form1.FormClosed += (s, args) =>this.Close();
Form1.Show();
}
privatevoid button1_Click(object sender, EventArgs e)
{
if (textBox1.Text.Contains(" ")) { textBox1.Text = textBox1.Text.Replace(" ", "_"); }
if (textBox1.Text != ""&& textBox2.Text != ""&& textBox3.Text != ""&& textBox2.Text ==
textBox3.Text)
{
MessageBox.Show(" Пользовательзарегистрирован.", " Регистрацияпрошлауспешно!",
MessageBoxButtons.OK, MessageBoxIcon.Information);
Writing login = newWriting();
}
```

Ә қосымшасының жалғасы

```
login.writetf("Users.txt", "," + textBox1.Text);
Writing pass = new Writing();
pass.writetf("Passwords.txt", "," + textBox2.Text);
this.Hide();
Form1 Form1 = new Form1();
Form1.FormClosed += (s, args) =>this.Close();
Form1.Show();
}
else{ label1.Text = "Неправильнозаполненыполя!"; label1.ForeColor = Color.Red; }
}}}
```

Мәліметтер қоры коды:

```
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 Кредитный_инспектор
{
publicpartialclassMain : Form
{
publicMain()
{
this.StartPosition = FormStartPosition.CenterScreen;
InitializeComponent();
if (Login.str.Contains("_"))
{
Login.str = Login.str.Replace("_", " ");
}
label1.Text = "Здравствуйте, " + Login.str + ", для продолжения выберите БД из списка:";
}
privatevoid button4_Click(object sender, EventArgs e)
{
this.Hide();
Employees Employees = newEmployees();
Employees.FormClosed += (s, args) =>this.Close();
Employees.Show();
}
privatevoid button3_Click(object sender, EventArgs e)
{
this.Hide();
Credit Credit = newCredit();
Credit.FormClosed += (s, args) =>this.Close();
Credit.Show();
}
```


Ә қосымшасының жалғасы

```
}
private void button5_Click(object sender, EventArgs e)
{
    this.Hide();
    Client Client = new Client();
    Client.FormClosed += (s, args) => this.Close();
    Client.Show();
}
private void changeUserToolStripMenuItem_Click_1(object sender, EventArgs e)
{
    this.Hide();
    Login Login = new Login();
    Login.FormClosed += (s, args) => this.Close();
    Login.Show();
}
private void exitToolStripMenuItem_Click_1(object sender, EventArgs e)
{
    this.Hide();
    Form1 Form1 = new Form1();
    Form1.FormClosed += (s, args) => this.Close();
    Form1.Show();
}
private void button1_Click(object sender, EventArgs e)
{
    this.Hide();
    CreditPrograms CreditPrograms = new CreditPrograms();
    CreditPrograms.FormClosed += (s, args) => this.Close();
    CreditPrograms.Show();
}}
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;
using System.Data.SqlClient;
namespace Кредитный_инспектор
public partial class Client : Form
{
    SqlCommands Command;
    SqlDataAdapter Adapter;
    SqlCommandBuilder Builder;
    DataSet Ds;
    DataTable Table;
    int selectedID = 0;
    int selectedIDcopy = 0;
```

Ә қосымшасының жалғасы

```
public Client()
{
this.StartPosition = FormStartPosition.CenterScreen;
InitializeComponent();
}
```

«Добавить клиента» батырмасының коды:

```
private void button10_Click(object sender, EventArgs e)
{
if (textBox3.Text == "" || textBox4.Text == "" || textBox5.Text == "" || textBox6.Text == "" ||
textBox7.Text == "" || textBox8.Text == "" || textBox9.Text == "" || textBox10.Text == "" ||
textBox11.Text == "" || textBox16.Text == "" || textBox13.Text == "" || textBox14.Text == "" ||
textBox15.Text == "")
{
MessageBox.Show("Не все данные введены верно!", "Ошибка!", MessageBoxButtons.OK,
MessageBoxIcon.Error);
}
else
{
if (MessageBox.Show("Вы действительно хотите добавить клиента?", "Добавление",
MessageBoxButtons.YesNo, MessageBoxIcon.Information) == DialogResult.Yes)
{
string connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" +
Environment.CurrentDirectory + "\\Kredit.mdf;Integrated Security=True";
string sql = "INSERT INTO Client (Фамилия,Имя,Отчество,Дата_рождения,Адрес_прописки,
Адрес_проживания,Домашний,Мобильный,Номер_УД,ИИН,Тип_клиента,Место_работы,
Адрес_работы)
values(@ Фамилия,@ Имя,@ Отчество,@ Дата_рождения,@ Адрес_прописки,@ Адрес_
проживания,@ Домашний,@ Мобильный,@ Номер_УД,@ ИИН,@ Тип_клиента,@ Место_
работы,@Адрес_работы)";
SqlConnection connection = new SqlConnection(connectionString);
SqlCommand cmd = new SqlCommand(sql, connection);
connection.Open();
cmd.Parameters.AddWithValue("@Фамилия", textBox3.Text);
cmd.Parameters.AddWithValue("@Имя", textBox4.Text);
cmd.Parameters.AddWithValue("@Отчество", textBox5.Text);
cmd.Parameters.AddWithValue("@Дата_рождения", numericUpDown1.Value.ToString()
+ " " + comboBox1.Text + " " + textBox6.Text);
cmd.Parameters.AddWithValue("@Адрес_прописки", textBox7.Text);
cmd.Parameters.AddWithValue("@Адрес_проживания", textBox8.Text);
cmd.Parameters.AddWithValue("@Домашний", textBox9.Text);
cmd.Parameters.AddWithValue("@Мобильный", textBox16.Text);
cmd.Parameters.AddWithValue("@Номер_УД", textBox10.Text);
cmd.Parameters.AddWithValue("@ИИН", textBox11.Text);
cmd.Parameters.AddWithValue("@Тип_клиента", textBox13.Text);
cmd.Parameters.AddWithValue("@Место_работы", textBox14.Text);
cmd.Parameters.AddWithValue("@Адрес_работы", textBox15.Text);
cmd.ExecuteNonQuery();
}
```

Ә қосымшасының жалғасы

```
connection.Close();
}}
RefreshTable();
}
```

«Очистить поля» батырмасының коды:

```
private void button9_Click(object sender, EventArgs e)
{
    ClearData();
}
```

«Сохранить изменения таблицы» батырмасының коды:

```
private void button13_Click(object sender, EventArgs e)
{
    if (textBox3.Text == "" || textBox4.Text == "" || textBox5.Text == "" || textBox6.Text == "" ||
        textBox7.Text == "" || textBox8.Text == "" || textBox9.Text == "" || textBox10.Text == "" ||
        textBox11.Text == "" || textBox16.Text == "" || textBox13.Text == "" || textBox14.Text == "" ||
        textBox15.Text == "")
    {
        MessageBox.Show("Не все данные введены верно!", "Ошибка!", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
    Else
    {
        if (MessageBox.Show("Вы действительно хотите обновить данные клиента?", "Обновление",
            MessageBoxButtons.YesNo, MessageBoxIcon.Information) == DialogResult.Yes)
        {
            if (selectedID == 0)
            {
                MessageBox.Show("Вы не выбрали клиента для изменения! Для выбора нажмите 'Показать
                данные'", "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            else
            {
                string connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" +
                    Environment.CurrentDirectory + "\\Kredit.mdf;Integrated Security=True";
                string sql = "Update Client set
                Фамилия=@Фамилия,Имя=@Имя,Отчество=@Отчество,Дата_рождения=@Дата_рождения
                ,Адрес_прописки=@Адрес_прописки,Адрес_проживания=@Адрес_проживания,Домашний
                =@Домашний,Мобильный=@Мобильный,Номер_УД=@Номер_УД,ИИН=@ИИН,Тип_кли
                ента=@Тип_клиента,Место_работы=@Место_работы,Адрес_работы=@Адрес_работы
                where Id=" + selectedID;
                SqlConnection connection = new SqlConnection(connectionString);
                SqlCommand sCommand = new SqlCommand(sql, connection);
                connection.Open();
                sCommand.Parameters.AddWithValue("@Фамилия", textBox3.Text);
                sCommand.Parameters.AddWithValue("@Имя", textBox4.Text);
            }
        }
    }
}
```

Ә қосымшасының жалғасы

```
sCommand.Parameters.AddWithValue("@Отчество", textBox5.Text);
sCommand.Parameters.AddWithValue("@Дата_рождения", numericUpDown1.Value.ToString() + " " +
comboBox1.Text + " " + textBox6.Text);
sCommand.Parameters.AddWithValue("@Адрес_прописки", textBox7.Text);
sCommand.Parameters.AddWithValue("@Адрес_проживания", textBox8.Text);
sCommand.Parameters.AddWithValue("@Домашний", textBox9.Text);
sCommand.Parameters.AddWithValue("@Мобильный", textBox16.Text);
sCommand.Parameters.AddWithValue("@Номер_УД", textBox10.Text);
sCommand.Parameters.AddWithValue("@ИИН", textBox11.Text);
sCommand.Parameters.AddWithValue("@Тип_клиента", textBox13.Text);
sCommand.Parameters.AddWithValue("@Место_работы", textBox14.Text);
sCommand.Parameters.AddWithValue("@Адрес_работы", textBox15.Text);
sCommand.ExecuteNonQuery();
connection.Close();
}}
selectedID = 0;
ClearData();
RefreshTable();
}
```

«Оформления кредита» батырмасының коды

```
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;
using System.Data.SqlClient;
using iTextSharp.text;
using iTextSharp.text.pdf;
using System.IO;
namespace Кредитный_инспектор
{
    public partial class Credit : Form
    {
        SqlCommandsCommand;
        SqlDataAdapter;
        SqlCommandBuildersBuilder;
        DataSetsDs;
        DataTablesTable;
        int selectedID = 0;
        int selectedIDcopy = 0;
        public Credit()
        {
            this.StartPosition = FormStartPosition.CenterScreen;
            InitializeComponent();
        }
    }
}
```

Ә қосымшасының жалғасы

```
private void Credit_Load(object sender, EventArgs e)
{
    comboBox1.Items.Clear();
    string connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" +
        Environment.CurrentDirectory + "\\Kredit.mdf;Integrated Security=True";
    string sql = "SELECT * FROM Client";
    SqlConnection connection = new SqlConnection(connectionString);
    SqlCommand command = new SqlCommand(sql, connection);
    connection.Open();
    using (SqlDataReader read = command.ExecuteReader())
    {
        string s = "";
        while (read.Read())
        {
            s = read["Фамилия"].ToString() + " " + read["Имя"].ToString() + " " +
                read["Отчество"].ToString();
            while (s.Split(' ').Length > 4)
            {
                s = s.Replace(" ", " ");
            }
            comboBox1.Items.Add(new ComboBoxItem() { Text = s });
        }
        /*ComboBoxItem item = new ComboBoxItem() { Text = read["Название"].ToString() };
        comboBox1.Items.Add(item);*/
        //comboBox1.Items.Add(new ComboBoxItem() { Text = s });
    }
    connection.Close();
}
```

«Добавить заемщика» батырмасының коды:

```
private void button10_Click(object sender, EventArgs e)
{
    if (comboBox3.Text == "" || numericUpDown1.Value == 0 || comboBox1.Text == "" ||
        numericUpDown1.Value == 0 || comboBox2.Text == "")
    {
        MessageBox.Show("Не все данные введены верно!", "Ошибка!", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
    else
    {
        if (MessageBox.Show("Вы действительно хотите добавить заемщика?", "Добавление",
            MessageBoxButtons.YesNo) == DialogResult.Yes)
        {
            string connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" +
                Environment.CurrentDirectory + "\\Kredit.mdf;Integrated Security=True";
            string sql = "INSERT INTO
                Credit(Программа,Размер,Валюта,Срок,Дата_открытия,Заемщик,Статус,Скидка)
                values(@Программа,@Размер,@Валюта,@Срок,@Дата_открытия,@Заемщик,@Статус,@Скидка)";
            SqlConnection connection = new SqlConnection(connectionString);
        }
    }
}
```

Ә қосымшасының жалғасы

```
sCommand = newSqlCommand(sql, connection);
connection.Open();
sCommand.Parameters.AddWithValue("@Программа", comboBox3.Text);
sCommand.Parameters.AddWithValue("@Размер", numericUpDown1.Value.ToString());
sCommand.Parameters.AddWithValue("@Валюта", comboBox2.Text);
sCommand.Parameters.AddWithValue("@Срок", numericUpDown2.Value.ToString());
sCommand.Parameters.AddWithValue("@Дата_открытия", dateTimePicker1.Value);
sCommand.Parameters.AddWithValue("@Заемщик", comboBox1.Text);
sCommand.Parameters.AddWithValue("@Статус", "Новый клиент");
sCommand.Parameters.AddWithValue("@Скидка", "0%");
sCommand.ExecuteNonQuery();
connection.Close();
}}
RefreshTable();
for (inti = 0; i< dataGridView1.RowCount; i++)
{
for (int j = dataGridView1.RowCount - 1; j >i; j--    {
try
{
if (dataGridView1[6, i].Value.ToString().Trim() == dataGridView1[6, j].Value.ToString().Trim())
{
int summa = Convert.ToInt32(dataGridView1[2, j].Value);
dataGridView1.Rows.RemoveAt(dataGridView1[6, j].RowIndex);
sAdapter.Update(sTable);
selectedID = Convert.ToInt32(dataGridView1.Rows[i].Cells[0].Value);
stringconnectionString = "Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=" +
Environment.CurrentDirectory + "\\Kredit.mdf;Integrated Security=True";
stringsql = "Update Credit set Размер=@Размер,Статус=@Статус,Скидка=@Скидка WHERE
Id=" + selectedID;
SqlConnection connection = newSqlConnection(connectionString);
SqlCommandCommand = newSqlCommand(sql, connection);
connection.Open();
sCommand.Parameters.AddWithValue("@Размер",
((((Convert.ToInt32(dataGridView1.Rows[i].Cells[2].Value)) * 85) / 100) +
(summa*85/100)).ToString());
sCommand.Parameters.AddWithValue("@Статус", "Постоянный клиент");
sCommand.Parameters.AddWithValue("@Скидка", "15%");
sCommand.ExecuteNonQuery();
connection.Close();
}}
catch (NullReferenceException) { }
}}
RefreshTable();
}
```

«Удалить выбранного заемщика» батырмасының коды:

```
privatevoid button3_Click(object sender, EventArgs e)
{
```

Ә қосымшасының жалғасы

```
selectedID = selectedIDcopy;
if (selectedID == 0)
{
    MessageBox.Show("Выделитесотрудника в таблице!", "Ошибка!", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}
else
{
    if (MessageBox.Show("Выдействительнохотитеудалитьвыбранногосотрудника?",
    "Удаление", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
    {
        dataGridView1.Rows.RemoveAt(dataGridView1.SelectedRows[0].Index);
        sAdapter.Update(sTable);    }
        RefreshTable();
    }
}
```

«Программы кредитования» батырмасының коды:

```
using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Threading.Tasks;
usingSystem.Windows.Forms;
usingSystem.Data.SqlClient;
namespaceКредитный_инспектор
{
    publicpartialclassCreditPrograms : Form
    {
        SqlCommandsCommand;
        SqlDataAdapterAdapter;
        SqlCommandBuildersBuilder;
        DataSetsDs;
        DataTablesTable;
        intselectedID = 0;
        intselectedIDcopy = 0;
        publicCreditPrograms()
        {
            this.StartPosition = FormStartPosition.CenterScreen;
            InitializeComponent();
        }
    }
}
```

«Добавить программу» батырмасының коды:

```
Private void button10_Click(object sender, EventArgs e)
{
```

Ә қосымшасының жалғасы

```
if (textBox1.Text == "" || numericUpDown1.Value == 0 || numericUpDown2.Value == 0 ||
numericUpDown3.Value == 0 || comboBox2.Text == "")
{
    MessageBox.Show("Невседанные введены верно!", "Ошибка!", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}
else
{
    if (MessageBox.Show("Вы действительно хотите добавить программу?", "Добавление",
    MessageBoxButtons.YesNo, MessageBoxIcon.Information) == DialogResult.Yes)
    {
        string connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" +
        Environment.CurrentDirectory + "\\Kredit.mdf;Integrated Security=True";
        string sql = "INSERT INTO CreditPrograms(Название_продукта,Процентная_ставка,Максимальная_сумма,
        Валюта,Максимальный_срок)
        values(@Название_продукта,@Процентная_ставка,@Максимальная_сумма,@Валюта,@Ма
        ксимальный_срок)";
        SqlConnection connection = new SqlConnection(connectionString);
        SqlCommand sCommand = new SqlCommand(sql, connection);
        connection.Open();
        sCommand.Parameters.AddWithValue("@Название_продукта", textBox1.Text);
        sCommand.Parameters.AddWithValue("@Процентная_ставка",
        numericUpDown1.Value.ToString());
        sCommand.Parameters.AddWithValue("@Максимальная_сумма",
        numericUpDown2.Value.ToString());
        sCommand.Parameters.AddWithValue("@Валюта", comboBox2.Text);
        sCommand.Parameters.AddWithValue("@Максимальный_срок",
        numericUpDown3.Value.ToString());
        sCommand.ExecuteNonQuery();
        connection.Close();
    }
}
RefreshTable();
}
```