

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ
«Компьютерлік технологиялар» кафедрасы

«Қорғауға жіберілді»
Кафедра меңгерушісі
ф.-м.ғ.д., проф. З.Қ. Құралбаев

_____ « _____ » _____ 2014 ж.
(КОЛЫ)

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: «КСГ компаниясындағы тауар айналымын есепке алуға және бақылауға арналған бағдарлама»

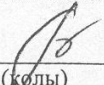
5В070400 – Есептеу техникасы және бағдарламалық қамтамасыз ету мамандығы бойынша

Орындаған: ВТк-10-1 Жакелеева Ж.Ж.

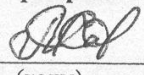
Жетекші: ф.-м.ғ.к, доцент Шайхин Б.М.

Кенесшілер :

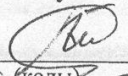
Экономикалық бөлім бойынша :

 «05» 06 доцент Боканова Г.Ш.
(КОЛЫ) 2014 ж.

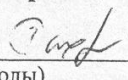
Өмір тіршілігі қауіпсіздігі бойынша:

 «09» 06 т.ғ.к., аға оқытушы Муташева Г.С.
(КОЛЫ) 2014 ж.

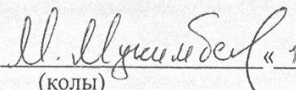
Есептеу техникасын қолдану бойынша :

 «09» 06 ф.-м.ғ.к, доцент Шайхин Б.М.
(КОЛЫ) 2014 ж.

Мөлшер бақылаушы:

 «11» 06 ассистент Тусупов Д.М.
(КОЛЫ) 2014 ж.

Пікір жазушы :

 «11» 06 ҚазҰУ доценті, т.ғ.д. Мукимбеков М.Ж.
(КОЛЫ) 2014 ж.

Алматы 2014

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

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

жобаны орындауға берілген

ТАПСЫРМА

Студент Жакелеева Жанар Жузжасаровна

Жоба тақырыбы «КСГ компаниясындағы тауар айналымын есепке алуға және бақылауға арналған бағдарлама»

Ректордың «24» 09 № 115 бұйрығы бойынша бекітілген.

Аяқталған жұмысты тапсыру мерзімі: «11» 06 2014 ж.

Жобаға қажетті алғашқы мәліметтер (талап етілетін жоба нәтижелерінің параметрлері) және нысананың бастапқы деректері:

MS SQL Server 2008 дерекқорларды басқару жүйесін және Microsoft Visual Studio 2012 ultimate бағдарламалау ортасын пайдалану арқылы күнделікті сатылған тауарды есептеуге ыңғайлы бағдарлама құрылды.

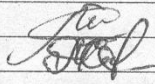
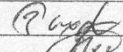

Диплом жобасындағы әзірленуі тиіс сұрақтар тізімі немесе диплом жобасының қысқаша мазмұны:

- Техникалық тапсыма.
- Ақпараттық жүйелер, дерекқорларды басқарудың жүйелері (ДҚБЖ) түсініктері.
- Бағдарламаның дерекқорын жобалау, жобалау құралдарын таңдау, ER диаграммасын жобалау, дерекқорды құру, бағдарламаның интерфейсін жобалау және оны дерекқормен байланыстыру.
- Техникалық-экономикалық негізделуі.
- Өндірістегі өміртіршілік қауіпсіздігі шараларын ұйымдастыру.
- Бағдарламаның жұмыс істеу принципі мен функционалдығын көрсететін графикалық материалдар.

Негізгі ұсынылатын әдебиеттер

- 1 Ник Рендольф, Дэвид Гарднер, Майкл Минутилло, Крис Андерсон Visual Studio 2010 для профессионалов = Professional Visual Studio 2010. — М.: «Диалектика», 2011.
- 2 Роберт Э. Уолтерс, Майкл Коулс SQL Server 2008: ускоренный курс для профессионалов = Accelerated SQL Server 2008. — М.: «Вильямс», 2008.
- 3 Карли Уотсон, Кристиан Нейгел, Якоб Хаммер Педерсен, и др. Visual C# 2008: базовый курс. Visual Studio® 2008 = Beginning Visual C# 2008. — М.: «Диалектика», 2009.
- 4 Роберт Виейра Программирование баз данных Microsoft SQL Server 2005. Базовый курс = Beginning Microsoft SQL Server 2005 Programming. — М.: «Диалектика», 2007.
- 5 Когаловский М. Р. Энциклопедия технологий баз данных. — М.: Финансы и статистика, 2002.

Жоба бойынша бөлімшелерге қатысты белгіленген кеңесшілер

Бөлім	Кеңесші	Мерзімі	Қолы
Негізгі бөлім	Шайхин Б.М.		
Тіршілік қауіпсіздігі	Муташева Г.С.		
Экономикалық бөлім	Боканова Г.Ш.		
Норма бақылаушы	Рахимжанова З.М.		
Есептеу техникасын қолдану	Шайхин Б.М.		

ДИПЛОМ ЖОБАСЫН ДАЙЫНДАУ

КЕСТЕСІ

№ р/с	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
1	Техникалық тапсырма	12.12.2013	
2	Ақпараттық жүйелер, дерекқорларды басқарудың жүйелері (ДҚБЖ) түсініктері.	17.02.2014	
3	Бағдарламаның дерекқорын жобалау, жобалау құралдарын таңдау, ER диаграммасын жобалау, дерекқорды құру.	23.04.2014	
4	Дерекқормен пайдаланушы интерфейсін байланыстыру, қосымша жасау.	29.05.2014	

Тапсырманың берілген уақыты « 22 » 10 2014ж.

Кафедра меңгерушісі _____

(қолы)

ф.-м.ғ.д., проф. Құралбаев З.К.

Жоба жетекшісі _____

(қолы)

ф.-м.ғ.к, доцент Шайхин Б.М.

Орындалатын тапсырманы қабылдаған студент Жакееева Жакееева Ж.Ж.
(қолы)

Андатпа

Осы дипломдық жұмыста «КСГ» компаниясындағы тауар айналымын есепке алуға және бақылауға арналған бағдарламаны дайындау жобасы көрсетілген. Компанияның жұмыс жасау сферасының анализі өткізілді. Сол зерттеулер нәтижесі негізінде өзін-өзі қысқа уақыт аралығында өтейтін, компания жұмысын автоматтандыратын және қызметші қателіктерін азайтатын бағдарлама дайындалды.

Бағдарлама MS Sql Server R2 және Microsoft Visual Studio 2012 ultimate бағдарламалау орталарының көмегімен дайындалды және барлық қойылған талаптарға сай.

Соңғы бөлімдерде жобаның экономикалық тиімділігі есептеліп, жоба пайдалы деп шешілді. Сонымен қатар еңбекті қорғау есебі шығарылды.

Аннотация

В данном дипломном проекте разработана программа для учета и контроля товарооборота в компании «КСГ». Была изучена работа компании. С учетом этих факторов была написана программа, позволяющая не только окупить себя в кратчайшие сроки, но и принести прибыль, при этом, автоматизировать работу компании и свести ошибки работника к минимуму.

Система разработана с использованием современных технологий таких как: MS Sql Server R2, Microsoft Visual Studio 2012 ultimate и отвечает требованиям, предъявляемым к подобным программам.

В последних главах рассматриваются вопросы безопасности жизнедеятельности, приводится технико-экономическое обоснование и рассчитывается цена разработки проекта.

Annotation

The development of the program for account and checking the trade turnover in "KCG" company is presented in the given degree project. The basic principles of work of this sphere are considered. Taking into account these factors and the received results was written the program, which allows not only to pay back itself in the shortest terms, but also to make profit, thus, to automate work of the company and to minimize mistakes of the employee.

This system is developed with the use of modern information technologies such as MS Sql Server R2, Microsoft Visual Studio 2012 ultimate, also meets the requirements imposed to similar programs.

In the last chapter are considered problems of safety of vital activity, the feasibility study are provided and the price of project development is counted.

Кіріспе

21-ші ғасыр инновациялар мен технологиялар ғасыры. Бұл заманда адам техника дауынан қалып қоймауы керек. Әр атылған таңмен бірге жаңа технология ойлап табылады десек артық болмайды. Осы технологилар адам өмірінің қызықты әрі жеңіл болуы үшін шығарылуда. Ал нарықта жаңа техника көмегімен қызмет көрсету саласының дәрежесі, жылдамдығы және сапасы артуда. Өндіріс саласы жоғарғы дәрежелі технологиялар қолданбаса даму екіталай. Шығындарды оңтайландыру, үнемдеуді арттыру, тұтынушылардың барлық талаптарына сәйкес қызмет ету шарттарына сәйкес өндірістің дауы қатал нарықтық бәсекелестік шартында іске асырылады. Кәсіпорындағы шығын орталықтарының барлығына жан-жақты бақылау жасау, талдауда күрделі математикалық әдістері қолдану, болжам жасау, параметрлер мен белгілердің есебіне және жинақтардың үйлесімді жүйесіне негізделген жоспарлау, қор жинау және ақпараттарды өңдеу қажет. Бұл мәселелерді басқару үшін өте көм қызметкер қажет, немесе заманауи бір ғана бағдарламамен алмастыруға болады. Бұл жолды таңдау кәсіпорын жұмысын әлде қайда жақсарттады.

Берілген дипломдық жобаның негізгі мақсаты компания жұмысын басқарудың ақпараттық жүйесін жобалау болып табылады.

Қойылған мақсатты орындау үшін бұл жобада төмендегідей тапсырмалар қойылады:

- ақпараттық жүйелерді жобалаудың жалпы сұрақтарын зерттеу;
- компания қызмет ету аясын зерттеу;
- компания тауар айналымын реттеу жүйесінің жобасын әзірлеу.

Үлкен көлемді ақпаратты күделікті өңдеу, сақтау және жинақтау тиімді жүзеге асырылған болуы, компания қызмет көрсетуінің басты шарттарының бірі. Қоғамның ақпараттық технологияларды көптеп қолдануы өндірісті автоматтандыруды әлдеқайда жеңілдетеді. Өндірісті автоматтандыру еңбек өнімділігін арттырады, құжаттармен жұмыс жасаудың тиімділігін көбейтеді және қызмет көрсетуге қажетті мәліметтермен жедел алмасуды қамтамасыз етеді. Соңғы жылдары көп көңіл бөлінетін тағы бір мәселе жұмыс жасауға ыңғайлы интерфейс, бағдарма мүмкіндіктерінің кеңдігі және жұмыс жасау шапшаңдығы.

Бұл дипломдық жоба компаниядағы жұмыс ұйымдастырылуын бақылауға және басқаруға арналған бағдарлама дайындауға арналған.

Бұл дипломдық жұмыс төмендегі бөлімдерден тұрады:

- кіріспе;
- компания жұмысының анализі;
- бағдарламаны әзірлеу құралдарын таңдау;
- бағдарламаны дайындау;
- қорытынды.

Мазмұны

Кіріспе	8
1 Деректер қорын құрудың негізгі принциптері.....	9
1.1 Кәсіпорынның бағдарламаланған жұмыс кеңістігі.....	9
1.1.1 Қызметкердің бағдарламаланған жұмыс кеңістігі	10
1.1.2 Бағдарламаланған жұмыс кеңістігінің тілдік құралдары	12
1.1.3 Бағдарламаланған жұмыс кеңістігінің жіктелуі.....	14
1.2 Деректер қорын ұйымдастырудың қанағаттандыратын талаптары.....	16
1.2.1 Көпжақты байланыс орнату.....	17
1.2.2 Өнімділігі	17
1.2.3 Минималды шығындары	17
1.2.4 Минималды артықтығы	18
1.2.5 Іздеу мүмкіндігі.....	18
1.2.6 Бүтіндігі.....	18
1.2.7 Қауіпсіздігі және құпиялығы	18
1.2.8 Өткенмен байланыс.....	19
1.2.9 Болашақпен байланыс.....	19
1.2.10 Пайдалану қарапайымдылығы.....	19
2 Бағдарлам құру құралдары.....	20
2.1 Бағдарлама интерфейсін дайындау құралы.....	20
2.1.1 Бағдарлама интерфейсін дайындау құралы. Microsoft Visual Studio ..	20
2.1.2 .NET Framework.....	22
2.1.3 C# тілі	26
2.2 Деректер қорын құру.....	28
2.2.1 Деректер қоры түсінігі.....	28
2.2.2 Microsoft SQL Server 2008 R2	31
2.2.3 SQL тілі	32
2.2.4 SQL операторлары	34
3 Дерекқорды жобалау және оны іске асыру	37
3.1 ДҚ құрудың бастапқы кезеңі.....	37
3.1.1 Қолдану облысы және тағайындалуы	37
3.1.2 Дерекқордағы шешілетін мәселелер	37
3.2 Концептуалды жобалау.....	37
3.2.1 Қажетті ақпарат: пайдаланушылардың талабы.....	37
3.2.2 ДҚ және қосымшаны жобалау құралдарын таңдаудың негіздемесі. 38	

3.2.3	ER моделін құру	39
3.3	Логикалық жобалау	39
3.4	Физикалық жобалау	40
3.5	Бағдарламаны сипаттау.....	41
4	Тіршілік қауіпсіздігі.....	47
4.1	Өндірістік ғимараттағы еңбек шартын талдау	46
4.2	Есептеу.....	52
4.2.1	Электр тогын есептеу	52
4.2.2	Вентиляцияны есептеу	56
4.3	«Тіршілік қауіпсіздігі» бөлімі бойынша қорытынды	59
5	Техникалық – экономикалық негізделуі.....	61
5.1	Жоба сипаттау және қажеттілігін негізду	61
5.2	Нарықты саралау және орнын анықтау.....	61
5.3	Бағдарламаның SWOT-анализі	62
5.4	Маркетинг-микс элементі.....	62
5.5	Бағдарламалық қамтаманың өңделуін еңбек өнімділігінің есептелуі	63
5.5.1	Еңбекақыға жұмсалатын шығындар	64
5.5.2	Қосымша шығындар статьясы.....	65
5.5.1	Өзіндік құн нәтижесінің кестесі мен диаграммасы.....	68
5.6	Бағдарламаның бағасына есептеу	69
5.6.1	Бағдарламалық өнімді иемденудегі кәсіпорындардың бірізгі шығындарын есептеу	70
5.6.2	Қолдану саласындағы жылдық шығындарды есептеу.....	70
5.6.3	Ақпараттық өнімді енгізуден алынған үнемдеу мөлшері мен табысты есептеу	71
5.6.3	Ақпараттық өнімді енгізуден алынған үнемдеу мөлшері мен табысты есептеу	71
5.7	Ақшалай құралдардың қозғалысы	72
5.8	Экономикалық тиімділікті есептеу.....	72
5.8.1	Таза ағымдағы құндылықты есептеу (Net present value, NPV)	72
5.8.2	Пайда индексын есептеу (Profitability index, PI)	73
5.8.3	Табыстың ішкі нормасын есептеу (Internal rate of return, IRR).....	73
5.8.4	Өтімділік периодын есептеу (Payback period, PBP)	74
	Қорытынды	75
	Әдебиеттер тізімі.....	76
	А қосымшасы.....	77
	Ә қосымшасы.....	88

1 Деректер қорын құрудың негізгі принциптері

1.1 Кәсіпорынның автоматтандырылған ақпараттық жүйесі

Даму прогресінің негізгі факторы кәсіп орынды басқарудың формаларын және әдістерін жетілдіру. Жетілдіру автоматтандырылған ақпараттық жүйенің материалды-техникалық базасын көрсететін есептеу техникасының және байланыс құралдарының негізінде іске асырылады. Ол басқарудың объектілері мен субъектілері арасындағы байланыстырушы бөлімше қызметін атқарады. Басқару бағдарламасы келесі функцияларды орындайды: пайдаланушы енгізген негізгі мәліметтер мен сұраныстарды қабылдайды, жүйеге енгізілген және сақталған мәліметтерді анықталған алгоритмдерге сәйкес өңдейді және қажетті шығатын ақпаратты құрады.

Компанияда басқару бағдарламасын қажетті ақпаратпен қамтамасыз ететін өзара байланысқан күрделі жүйелерден тұратын автоматтандырылған ақпараттық жүйе құрылады. Негізгі функционалды жүйелер мекемеде пайда болатын мәселелер шешімін қамтамасыз етеді. Олар: өндірістің техникалық дайындығы, перспективті жоспарлануы, өндіріс дамуының болжамы, маркетингтік зерттеу, материалды, еңбектік және қаржылық ресурстарды оперативті басқару, дайын өнімнің сатылуы және іске асуы, кәсіпорынның бухгалтерлік есебі және шаруашылық қызметтер талдауы. Шешілетін тапсырмалар жиынтығы бойынша кәсіпорынның автоматтандырылған ақпараттық жүйесі 80-ші жылдардағы кәсіпорынды басқарудың автоматты жүйесіне ұқсайды, бірақ басқа сапалы идеологиялық базада. Автоматтандырылған ақпараттық базада әрбірі өңделетін мәліметтің аса өзіндік ерекшелік құрылымдық ерекшеліктеріне ие жүйелер немесе «бизнес-процестер» қарастырылады. Бизнес-процесс негізінде кәсіпорынның өзара әрекет ететін бөлімдері арқылы өтетін және тұтынушы қалап таңдаған тапсырысын (тауарды дайындау немесе қызмет көрсету) орындауға бағытталған материалды ақпараттық, қаржылық ағындардың немесе жұмыстық ағындардың өзара байланысқан жиынтығы түсіндіріледі. Автоматтандырылған ақпараттық жүйе бизнес-процеске өзгеріс енгізу жөнінде уақытында шешім қабылдау мақсатында экономикалық объектілердің жағдайын тікелей көрсетуге, бизнес-процестердің тиімділігін болжауға және олардың ұйымдастырылуы бойынша сипаттама беруге міндетті.

Қазіргі заманауи автоматтандырылған ақпараттық жүйенің сипаттамалық ерекшелігі – компанияның өзара байланысқан мәліметтер қорын және бірыңғай ақпараттық жүйенің білім қорын пайдалану.

Кәсіпорынның автоматтандырылған ақпараттық жүйесі техникалық жағынан байланысқан локальды есептеу желілерінің (ЛЕЖ) жиынтығын көрсетеді. Ірі корпорацияларға, бірлестіктерге, холдингтерге арнап мыңдаған ЭЕМ-ді санайтын және күрделі құрылымы бар үлкен корпоративті желілер құрылады. Корпоративті желілер локальды есептеу желілерін және глобальді есептеу желілерін (ГЕЖ) қосады. Соңғылары әр түрлі қалаларда, елдерде және тіпті әр түрлі континенттерде болуы мүмкін территориялық бөлінген ЭЕМ-дерді біріктіреді. Масштабына байланысты бөлімдердің желілері мен жұмыстық топтардың желілері ажыратылады. Жұмыстық

топтардың желілері әдетте 10-нан 20-ға дейінгі ПЭЕМ қосады. Мұндай желілердің қарапайымдылық және біртектілік қасиеттері бар. Бөлімшелердің желісі 100-150 ПЭЕМ-ді қамтиды. ПЭЕМ басқарушы қызметкердің жұмыс орнына орнатылады және олардың қазметтерін автоматтандыру үшін, кәсіпорынның кеңейтілген ақпараттық базасына оперативті рұқсатты қамтамасыз ету үшін пайдаланылады. Мұндай жұмыс орны автоматтандырылған деп аталады және кәсіпорынның автоматтандырылған ақпараттық басқаруының элементі болып табылады.

1.1.1 Қызметкердің бағдарламаланған жұмыс кеңістігі

Бағдарламаланған жұмыс кеңістігі (БЖК) ретінде есептеу желілерінің құрамындағы секілді бір-бірінен тәуелсіз жұмыс жасайтын, соңғы пайдаланушының жұмыс орнында орнатылған ЭЕМ негізіндегі оның кәсіби қызметінің есеп-тапсырмасын шешудің техникалық және ұйымдастырушылық-әдістемелік қамтамасыз етілуін қосқанда, соңғы пайдаланушының техникалық құралдарының жиынтығы түсіндіріледі. Орындалатын функцияларына байланысты бағдарламаланған жұмыс кеңістігі үш түрі қарастырылады:

- басқарушы менеджердің БЖК;
- жетекші маманның БЖК;
- техникалық және қосалқы қызметкердің БЖК.

Басқарушының бағдарламаланған жұмыс кеңістігі келесі талаптар ұсынылады:

- үнемі жедел және ең анық ақпараттармен толтырылып отыратын бөлінген мәліметтер қоры мен ақпараттың болуы;
- компания мәліметтер қорына және корпаративтік құпия ақпаратқа немесе олардың жеке бөліктеріне тек шектелген тұлғалар, ал жеке элементтерге тек басқарушының ғана рұқсаты болуы мүмкін;
- ақпарат басқарушының психологиялық сипаттамасына бағытталған формадағы көрнекілігі;
- ақпаратты жедел және ыңғайлы іздеуді қамтамасыз ету;
- басқарушылық шешім қабылдаудың қамтамасыз бағдарламалық құралдарының болуы;
- жұмысты орындау қарапайымдылығы;
- басқарушылық шешім негізінде қызметкерлердің тәжірибе жинау мүмкіндігін қамтамасыз ету;
- кәсіпорынның немесе оның бөлімшелерінің ұйымдастырушылық құрылымы шеңберінде формацияның басқа көздерімен оперативті байланысын қамтамасыз ету.

Бұл типтің автоматтандырылған жұмыс орындарында ақпаратты енгізудің тілдік және сезімталдық мүмкіндіктері жиі пайдаланылады.

Маманның (жоспарлаушының, қаржыгердің, нормалаушының, технологтың, маркетингтің, есепшінің және т. б.) бағдарламаланған жұмыс

кеңістігі локальды мәліметтер қоры мен ақпараттар сияқты мәліметтер қоры мен қолданбалы бағдарламалық қамтаманың бөлінуі негізіндегі кәсіби қызмет тапсырмаларын шешуді қамтамасыз етеді.

Техникалық қызметкердің бағдарламаланған жұмыс кеңістігі мына функцияларды жүзеге асырады:

- ақпарат енгізу;
- мәліметтер қорын басқару;
- кіріс және шығыс құжаттарын өңдеу;
- атқарушылық қызметтерді басқару.

Жеке бағдарламаланған жұмыс кеңістігітерінің арасындағы ақпарат алмасу есептеу желілерінің байланыс каналы бойынша жүзеге асады.

Локальді есептеу желілерінде қысқа ара қашықтық үшін мәліметтер жіберудің қарапайым әдісін пайдалануға, мәліметтер алмасудың жоғары жылдамдығына (100 Мбит/с-қа дейін) мүмкіндік беретін қымбат жоғары сапалы байланыс желілері пайдаланылуы мүмкін. Осы қызметтерге байланысты жіберілетін локальды есептеу желілері кең алуан түрлілігімен ерекшеленеді және әдетте, on-line режимінде қосымшаның іске асырылуын қарастырады.

Егер компания (бірлестік, корпорация) бір қаланың әр түрлі аймағында, бір елдің түрлі облыстарында, елдерде және континенттерде территориялық оңашаланған бөлімшелер болса, онда жеке бағдарламаланған жұмыс кеңістігінің арасында ақпарат алмасу корпоративті есептеу желілерінің байланыс каналы арқылы жүзеге асырылады (КЕЖ).

Жойылған локальды есептеу желілері мен корпоративті есептеу желілеріндегі жеке дербес компьютерлерді жалғау үшін әр түрлі телекоммуникациялық құралдар, оның ішінде телефон каналдары, радиоканалдар, спутникті байланыс пайдаланылады.

Мұндай күрделі, үлкен және ауқымды көлемдегі желілердің құрамында компьютерлердің әр түрлі типтерінің – үлкен ЭЕМ-нен бастап (оларды мейнфреймдар деп атау қабылданған) персоналды компьютерге дейін, көптеген операциялық жүйелер және қосымша қамтамалар болуы оның өзіндік ерекшелігі болып табылады. Корпоративті желінің біртекті емес бөліктері пайдаланушыға барлық қажетті ресурстардан мүмкіндігінше қажетті деректерді алуға айқын рұқсат бере отырып, бүтін бір тұтас желі ретінде жұмыс жасауы керек.

Корпоративті желілерде және локальды есептеу желілерінде есептеу желілеріндегі ақпарат – видеосурет, суреттер, дауыстар өңделе бастады. Мұндай мультимедиялық ақпараттарды желі арқылы жіберу кезінде кідірістер туындайды. Ол ақпаратытардың кең көлемді болуы мәліметтер дестесін жіберу кезіндегі кідіріске сезімталдығымен байланысты. Кідірістер, көбінесе, желінің соңғы түйінінде осындай ақпараттың бұрмалануына – «эхо» эффектісінің туындауына, кейбір сөздердің үзілуіне, яғни талдану мүмкіндігінің болмауына, суреттердің дірілдеуіне және т. б. алып келеді. Сондықтан корпоративті желілерде жаңа желілік технологиялар және жаңа құрал-жабдықтар есебінен

файлдарды жолдау жылдамдығы локальды есептеу желілерінің қарапайым дәстүрлі желілеріндегі жылдамдығына жақындайды (frame relay желілерінде — 2 М бит/с, АТМ - 622 М бит/с).

1.1.2 Бағдарламаланған жұмыс кеңістігінің тілдік құралдары

Бағдарламаланған жұмыс кеңістігінің тілдік құралдары, ең алдымен, пайдаланушы әрекеті мен ПЭЕМ реакциясының мағыналық жағынан сәйкес келуі үшін қажет. Оларсыз оқу процетерін, диалог ұйымдастыру, қатені табу және жою мүмкін емес. Мұндай тілдерді өңдеудің қиындығы олардың әсіресе процедуралық тіл емес болуы аса қиындатады. Егер процедуралық тіл берілген жағдай қалай орындалып іске асырылатынын көрсетсе, онда процедуралық емес тіл – нақты орындау керек істі емес, ол үшін қандай әрекеттер қажет екендігін көрсетеді. Соңғы пайдаланушылар, яғни бағдарламалық өнімді сатып алушылар, ақпараттық қажеттілікті жүзеге асыру процесін нақты білмейді және білуге болмайтындықтан, БЖК-ның интеллектуалдығы неғұрлым жоғары болса, оның тілдерінде соншалықты процедуралық емес мүмкіндіктер болмауы керек.

БЖК дайындау тілдері пайдаланушыға бағытталған, сонымен бірге кәсіби бағытталған болуы керек. Бұл тек қана кәсіби құрал-жабдықтар емес, қызметтік жағдайдың сатысы, оқыту деңгейі, қолданылатын мәліметтер қоры және тағы басқасы бойынша бөлінетін пайдаланушы классификациясындағы өзгешеліктермен байланысты. Табиғи ауыз-екі тілді пайдалану, мұндай тәсілдің қарапайымдылығына қарамастан, кейде қиын емес нәтиже алу үшін пернетақта арқылы үлкен мәтіндік конструкция енгізудің қажеттілігі біршама кемшілік беретіндігін ескерген жөн.

Барлық тілдердегі сияқты, БЖК тілдерінің негізін алдын ала белгіленген термин сөздер, сонымен қатар жаңа терминдер орнату көмегімен бар нәрселерді өзгертіп немесе толықтыра жаңартып әдістерді сипаттау арқылы құру керек. Бұл БЖК жоспарлау кезінде БЖК-ның терминологиялық негізін анықталған түрде жіктеу қажеттілігіне алып келеді, яғни тілдің барлық негізгі синтаксистік құрылымын және терминдер мен олардың жиынтығы арасындағы семантикалық қарым-қатынасты анықтайды. Осыған байланысты БЖК-ның қарапайым жіктелуінде қажеттілік туындауы мүмкін, мысалы, мәліметтердің асмастырылып немесе өзгертіліп өнделуі пайдаланушылық режимінде көріністерінің мүмкіндіктері бойынша: сандық, мәтіндік, аралас. Күрделі жағдайларда БЖК-ның жіктелуі деректер қорының ұйымдастырылуымен анықталуы мүмкін. Көбіне тілдің мүмкіндіктері, қолданушы ақпараттық қажеттіліктің іске асуымен сәйкес келетін шартты сұлбалар тұрғызуы мүмкін болатын ережелер жиынтығын, тізімін анықтайды. Мысалы, кейбір БЖК-да барлық мәліметтер және конструкциялар кестелік түрде (кестелік БЖК) немесе арнайы түрдің операторлары (функционалды БЖК) түрінде бекітіледі.

Қолданушы тілдерін БЖК диалог түрі бойынша бөледі. Соңғы есепте диалогты қолдау құралдары қолданушыға қажетті тілдік конструкцияларды анықтайды.

БЖК-ны өңдеу кезінде болатын диалогтардың ішінде көбірек таңдалатыны немесе пайдаланылатыны: ПЭЕМ әсер ететін диалог, жаттықпаған қолданушы диалогы және ақпараттың бекітілген кадрларының көмегімен диалог. ПЭЕМ әсер ететін диалог кезінде БЖК қолданушысы мнемониканы және тіл конструкциясын зерттеуден толық босайды. Бұл әдістің басты модификациясының бірі – ұсынылған ПЭЕМ шығарылымдарының бірі немесе бірнешеуі таңдалатын мәзір әдісі болып табылады.

ПЭЕМ жұмыс атқаратын формаларды толтыру диалогы кезінде қолданушы олардың келесі түсіндірілуімен және өңделуімен терезеде арнайы таңдалған формаларды толтырады. Гибридті диалог қолданушыға да, ПЭЕМ-ға әсер етуі мүмкін.

Жаттықпаған қолданушымен, яғни күнделікті компьютер қолданушысы болғанмененде бағдарламамен жұмыс жасай алмайтын қызметкер, диалог кезінде оған не істеу керектігіне байланысты қолданушыда күмән қалдырмауы мүмкін ПЭЕМ жауаптарының толық анықтығы қамтамасыз етілуі керек.

Ақпараттың бекітілген кадрларының көмегімен диалог кезінде ПЭЕМ бар болған тізімнен жауап таңдайды және жауапты шығарады. Бұл жағдайда қолданушы тек өте қысқа жауаптарды ғана енгізуі қажет, ал негізгі ақпарат автоматты түрде беріледі.

Сонымен қатар, диалог типі БЖК-ның жіктелуін анықтауы мүмкін, мысалы жаттықпаған қолданушының диалогтық құралдары БЖК. Осындай белгілері бойынша БЖК-ның жіктелуі қолданушының кәсіби бағыты бойынша топтастырылуымен байланысты. Егер жұмыс орнын оларды іске асыратын бағдарламалық құрал-саймандар тұрғысынан қарастырса, онда БЖК-ның типтері немесе нұсқалары өте кең тараған болуы керек. Олар: бағдарламалау тілі бойынша, қолданушыға бағдарламалаудың процедуралық құралдарын көрсету мүмкіндігі бойынша, эксплуатация процесі кезінде бағдарламалық жүйенің құрылуын аяқтау мүмкіндігі бойынша, деректер қорын басқару жүйесінің, қолданушы тілі бар интерпретатордың немесе транслятордың, қателерді табу және түзету құралдарының болуы бойынша және т.б. жіктелуі мүмкін. БЖК-да қолданылатын қолданбалы бағдарламалардың дестесі жүйенің нақты қосымшаға қосылуын қамтамасыз ету үшін параметрлерін меншіктеуі мүмкін. Қолданбалы бағдарламалар дестесінің генераторлары пайдаланылуы қажет болады.

БЖК құрамына міндетті түрде негізгі есептік функцияларды және диалогтың ұйымдастырылуын қамтамасыз ететін әр түрлі бағдарламалық компоненттер, сонымен қатар деректер қорын басқару жүйесі, трансляторлар, анықтамалық жүйелер, негізгі мәліметтерден тұратын деректер қоры, диалог сценарийі, параметрді басқаратын нұсқаулықтар, қателер тізімі және т.б. БЖК-ның негізгі компоненттері оның құрамын анықтайды және әр түрлі белгілері бойынша БЖК-ның жіктелу мүмкіндігін қамтамасыз етеді.

БЖК шеңберінде пайдаланылуына байланысты БЖК-ның соңғы қолданушы арқылы дамуын қамтамасыз ететін БЖК құралдардың шеңберінде пайдалануға байланысты БЖК-ны екі үлкен класқа бөлеміз: қызмет ететін және интеллектуалды. Бұл екі класс та әр түрлі қолданушыларға арналуы мүмкін. Бірақ сонымен қатар, екі БЖК-ның да қолданушысы бола алмайтындығын алдын ала айтуға болатын қолданушылар да болады. Мысалы, қызмет көрсетуші тұлға (іс жүргізушілер, хатшылар) олар арқылы орындалатын қызметтердің ерекшеліктеріне байланысты интеллектуалды БЖК-ны (өзінің тікелей қызметтерінде) қажет етпейді.

Қызмет көрсететін БЖК ұйымдастырушылық басқаруда былай бөлінуі мүмкін:

- ақпараттық-анықтамалық;
- есептік;
- мәтін құратын.

Интеллектуалды БЖК-ны ең алдымен мәліметтерге бағытталған және білімге бағытталған (даталогиялық және фактологиялық) деп бөлуге болады. Ақпараттық-анықтамалық БЖК қандай да бір басқару процесіне қызмет етеді. Есептік БЖК өзінің мазмұны бойынша әр түрлі және қолданушының бірнеше категорияларымен қабылдануы мүмкін. Олардың көмегімен бір-бірімен байланысты және байланысты емес ұйымдастырушылық-экономикалық тапсырмалар, БЖК құру процесі кезінде анықталатын немес алдын ала анықталған мәліметтерді іздеу және өңдеу тапсырмалары қойылуы және шешілуі мүмкін. Мәтін құратын БЖК әр түрлі құрылымдағы мәтіндік ақпараттың құрылуына және өңделуіне арналған.

1.1.3 Бағдарламаланған жұмыс кеңістігінің жіктелуі

БЖК жеке, топтық, ұжымдық болуы мүмкін. ЭЕМ жүйесін тиімді құру мақсатында топтық және ұжымдық БЖК-ға байланысты – мамандарға (ұжымға) БЖК жұмысын ұйымдастыру талаптарын күшейту мұндай жүйедегі әкімшілік жүргізу функцияларын нақты анықтау керек. «Адам-машына» болып табылатын БЖК жүйесі ашық, икемді, үнемі пайдалануға және жетілдіруге бейімделген болуы керек. Мұндай жүйеде келесі аталғандар қамтамасыз етілуге тиіс:

- ақпаратты өңдеудің машыналық құралдарына мамандардың максималды жақындауы;
- диалогтық режимдегі жұмыс;
- БЖК-ның эргономикалық талаптарына сәкес жабдықталуы;
- компьютердің жоғары өнімділігі;
- бір типті процестердің максималды автоматтандырылуы;
- мамандардың өзін-өзі оқыту мүмкіндігі.

БЖК-да шешілетін тапсырмаларды шартты түрде ақпараттық және есептік деп бөлуге болады.

Ақпараттық тапсырмаларға ақпаратты кодтау, жіктеу, жинақтау, құрылымдық ұйымдастыру, түзету, сақтау, іздеу және беру жатады. Көбіне

ақпараттық тапсырмалар арифметикалық және мәтіндік сипаттаманың және қатынастың (байланыстың) күрделі емес есептік және логикалық процедураларын енгізеді. Ақпараттық тапсырмалар, ереже бойынша, еңбекті көп қажет ететін болып табылады және маманның жұмыс уақытының үлкен бөлігін алады.

Есептік тапсырмалар толығымен реттеліп жазылмаған болса да, тіркелеген болып табылады. Жазылған тапсырмалар негізгі алгоритмдер негізінде шешіледі және екі топқа бөлінеді: тура есеп тапсырмасы және математикалық модель негізіндегі тапсырмалар. Тура есеп тапсырмасы қарапайым алгоритмдер көмегімен шешіледі. Күрделірек тапсырмалар үшін әр түрлі математикалық модельдерді қолданылу талап етіледі.

Соңғы уақыттарда семантикалық деп аталатын толығымен рәсімделмеген тапсырмаларды шешу құралдарына үлкен мән беріледі. Осындай тапсырмалар экономикалық объектілерді жедел басқару кезінде, яғни толық емес ақпарат шартында шешім қабылдау кезінде әдетте жиі туындайды. БЖК құрылымы – оның жүйелері мен жекелеген элементтерінің топтық жиынтығы. Қамтамасыз ететін жүйелерге ең алдымен мыналар жатады: техникалық, ақпараттық, бағдарламалық және ұйымдастырушылық. Сонымен қатар, жүйелердің толық қатары бар.

Техникалық қамтамасыз ету маман жұмысын аралық мамансыз (бағдарламашылар, операторлар және т.б.) қарастыратын кәсіби персоналды компьютер қызмет ететін техникалық құралдардың тобының жиынтығын білдіреді. Топтық БЖК-да ондай компьютерлерді 4-6 адам пайдалана алады. Кәсіби персоналды компьютер жиынтығына процессор, дисплей, пернетақта, ақпараттың магнитті дискісі жазу құралдары және графикалық құрастырушылар жатады. Техникалық құралдар жиынтығына желідегі әр түрлі БЖК-ны байланыстыруға арналған коммуникация құралдарын, сонымен қатар телефондық байланыс құралдарын жатқызуға болады.

Ақпараттық қамтамасыз ету – локальды деректер қорында сақталатын ақпараттар массиві. Ақпарат негізінен магнитті дискілерде ұйымдастырылады, өңделеді және сақталады. Ақпаратты басқару оны жазуды, тапсырмаларды іздеуді, оқуды, түзетуді және шешуді жүргізетін деректер қорын басқарудың бағдарламалық жүйесінің көмегімен жүзеге асырылады. БЖК-да бірнеше деректер қоры болуы мүмкін.

Ұйымдастырушылық қамтамасыз ету БЖК-ның жұмыс істеуін, жетілдірілуін және дамуын ұйымдастырудың, сонымен қатар кадр біліктілігін дайындаудың және арттырудың әдістері мен құралдарын енгізеді. Топтық және ұжымдық БЖК үшін ұйымдастырушылық қамтамасыз ету жүйесіне БЖК әкімшілік іс жүргізу функциялары енгізіледі: жобалау, жоспарлау, есептеу, басқару, талдау, реттеу, инфражүйесі бар ұйымдастырушылық байланыстар және т.б. Ұйымдастырушылық қамтамасыз ету БЖК қолданушыларының құқықтары мен міндеттерін анықтауды және құжаттық рәсімдеуді қарастырады.

Бағдарламалық қамтамасыз ету жүйелік және қолданбалы бағдарламалық қамтамасыз етуден тұрады. Жүйелік қамтамасыз етудің негізі операциялық

жүйе және бағдарламалау жүйесі, мысалы Basic алгоритмдік тілі болып табылады. Жүйелік бағдарламалар ақпаратты өңдеудің рационалды технологиясын қамтамасыз етеді. БЖК желісіндегі ақпараттық байланысты және әр түрлі канал бойынша БЖК байланысын қамтамасыз ету үшін жүйелік бағдарламалауға жататын бағдарламалық құралдар қолданылады.

1.2 Деректер қорын ұйымдастырудың қанағаттандыратын талаптары

Жеке серіктестік немесе кәсіпкерлік – азаматтардың өз атынан **мүліктік жауапкершілікке** негізделе отырып жүзеге асыратын **дербес** бастамалық қызметі. Жеке кәсіпкер, әдетте, кәсіпорынды өзінің меншікті **капиталымен** немесе мемлекет көмегімен қаржыландырады, барлық тәуекелді өзінің мойнына алады және осы қызметке байланысты өзінің міндеттемелері бойынша барлық мүлкімен жауап береді. Жеке кәсіпкерлік жалдамалы еңбекті пайдаланып, не пайдаланбай жүзеге асыралады, сонымен қатар **заң** актілерімен шектелмеген шаруашылық қызметтің барлық түрін де жүргізеді.

Жеке кәсіпкердің мынадай құқықтары бар:

- мемлекеттік кәсіпорындардың және басқадай **меншік** түрлеріне негізделген кәсіпорындардың мүліктерін толық немесе жартылай иемдену;
- шаруашылық жүргізуші субъектілер қызметіне өз **мүлкімен** қатысу;
- **келісімшарт** негізінде жұмыскерлерді жалдау және жұмыстан босату;
- **еңбекақы** нысандарын, жүйесін, мөлшерін өздерінше белгілеу;
- шаруашылық қызметтің бағдарламасын қалыптастыру;
- жабдықтаушылар мен тұтынушыларды тандау;
- бағалар мен тарифтерді белгілеу;
- сыртқы **экономикалық қатынастарда** болу.

Жеке кәсіпкерлік өзінің құқықтық иелігіне жататын мүлік базасында өз бетінше қызмет ететін бір ғана жеке тұлғадан құрылады. Некеде тұратын жеке тұлғаның жеке кәсіпкерлікті жүзеге асыруы үшін жұбайынан келісім алуының қажеті жоқ. Тек жеке кәсіпкерлікті іске асыру үшін жеке тұлға жұбайы екеуінің ортақ мүлкін пайдаланар болса ғана, екіншісінің келісімін алуы қажет болады.

Жеке кәсіпкерліктің артықшылықтары:

- адамның еңбекпен қамтамасыз етілу және ақша табу проблемаларын шешеді;
- қымбат емес әрі тез арада тіркеуге болады;
- мемлекеттік реттеуге немесе тексеріске аз ұшырайды;
- тым көп құжаттар жүргізуді қажет етпейді.

Жеке кәсіпкерліктің кемшіліктері:

- мұраға қалдыру мүмкіндігі жоқ;
- алғашқы капитал жинау қиындығы;
- кәсіпкер жеке мүлкін тәуекелге қояды.

1.2.1 Көпжақты байланыс орнату

Әр түрлі бағдарламашыларға әр түрлі логикалық файлдар жиынтығы қажет болады. Бұл файлдар бір ғана деректер жиынтығынан алынады. Деректерді есте сақтайтын элементтер арасында әр түрлі байланыстар болуы мүмкін. Кейбір деректер қоры өзара байланыстың күрделі шиеленісуінен тұрады. Деректердің байланысын немесе басқаруын ұйымдастыру амалдары өте көп. Деректерді ұйымдастыру әдісі осы өзара байланысты және онда жылдам келісімді ыңғайлы көрсету мүмкіндігі енгізіліп, қамтамасыз етілетін болуы тиіс. Деректер қорын басқару жүйесі деректер мен олардың арасындағы байланыстың қажетті логикалық немесе жүйелік файлдарын алу мүмкіндігін қамтамасыз етуі керек. Логикалық файлдың қолданбалы бағдарламадағы көрінісі мен деректерді физикалық сақтау әдісі арасында аздаған ұқсастықтық болуы керек.

1.2.2 Өнімділігі

Компания қызметкері арқылы пайдалану үшін арнайы жасалған деректер қоры адамдар диалогы үшін қанағаттандырылатын кері жауап уақытын – терминалды қамтамасыз етеді. Одан басқа, деректер қорын басқару жүйесі сәйкес өткізу-өңдеу мүмкіндігін қамтамасыз етеді. Сұраныстардың аз көлемді ағынына есептелген жүйелерде өткізу мүмкіндігі деректер қорының құрылымына біренше шектеу қояды. Сұраныстың үлкен ағыны бар жүйелерде, мысалы авиабилеттерді резервтеу жүйесінде өткізу мүмкіндігі деректердің физикалық сақталуын ұйымдастыруды таңдауға маңызды әсер етеді.

Тек тақталық өңдеуге арналған жүйелерде кері жауап уақытына аса көп көңіл бөлінбейді және физикалық ұйымдастыру әдісі тиімді дестелік өңдеуді, өзгертуді қамтамасыз ету шартынан таңдалады.

1.2.3 Минималды шығындары

Деректер қорын құруға және пайдалануға жұмсалған шығындарды көп мөлшерде кеміту үшін ішкі жадыға талапты азйтатын ұйымдастыру әдістері таңдалады. Осы әдістерді пайдалану уақытында жадыдағы деректердің физикалық көрінісі қолданбалы бағдарламашы пайдаланатын көріністен әдеқайда ауқымды ерекшеленеді. Бір көріністің екіншісіне айналуы бағдарламалық қамтамасыз етілуді немесе, егер мүмкін болса, техникалық немесе шағын бағдарламалық құралдарды жүзеге асырады. Мұндай жағдайларда түрлендіру алгоритміне кеткен шығындар мен жадыны үнемдеу барасында екеуінің бірін таңдауға тура келеді.

1.2.4 Мининалды артықтығы

Деректер қорын басқару жүйесін қолдануға дейін пайда болған өңдеу жүйелерінде ақпараттық қорлар артықшылықтың өте жоғары деңгейіне ие

болды. Ленталық кітапханалардың басым бөлігінде көптеген артық мәліметтер бар. Тіпті біріктірілген деректер қорымен байланыстырылған ақпараттың өсуіне байланысты деректер қорының қолданылуы кезінде артық деректердің пайда болуының потенциалды мүмкіндігі үнемі біртіндеп артады. Жолдың артық деректері жадыдан көбірек орын алса, жаңарудың бірден көп операциясын талап етеді. Деректер қорын ұйымдастыру мақсатында артық ақпарат жойылуы керек.

1.2.5 Іздеу мүмкіндігі

Деректер қорын қолданушы оған сақталған деректер бойынша әр түрлі сұрақтармен баруына болады. Қазіргі заманғы коммерциялық қосымшалардың көпшілігінде сұраныстар типі алдын ала анықталған, және деректердің физикалық ұйымдастырылуы оларды қажетті жылдамдықпен өңдеу үшін жетілдіріледі. Жүйеге қойылатын өспелі талаптар осындай сұраныстарды өңдеуді қамтамасыз етуге немесе алдын ала жоспарланбаған осындай жауаптардың құрылуына негізделеді.

1.2.6 Бүтіндігі

Егер деректер қоры көптеген қолданушы қолданатын деректерден тұрса, деректер бірлік элементтері мен олардың арасындағы байланыстар бұзылмауы өте маңызды. Қателер мен әр түрлі кездейсоқ бұзылулар туындатуы мүмкін екендігін ескеру керек. Деректерді сақтау, оларды жаңарту, деректерді қосу процедуралары, жүйе бұзылулардың пайда болу жағдайында деректер жойылмастан қайта құрыла алатындай болуы керек. Есептеу жүйесі онда сақталған деректердің бүтіндігіне кепілдік беруі керек.

1.2.7 Қауіпсіздігі және құпиялығы

Деректер қоры жүйесіндегі деректер құпияда сақталуы керек. Есте сақталатын ақпарат кейде оны пайдаланатын мекеме үшін өте маңызды. Ол жоғалуға және ұрлануға тиісті емес. Деректер қорындағы ақпараттың төзімділігін арттыру үшін оны аппараттық немесе бағдарламалық бұзылулардан, катастрофалық және криминалдық жағдайлардан, компонентті емес немесе оны дұрыс пайдаланбайтын арам ниетті қолданушылардан қорғау маңызды.

Деректердің қауіпсіздігі ретінде оған құқығы жоқ тұлғалардың кездейсоқ немесе алдын ала жоспарланбаған қолжетімділіктен, деректердің авторластырылмаған модификациясынан немесе олардың жойылуынан деректердің қорғалуын қарастырады.

1.2.8 Өткенмен байланыс

Белгілі уақыт аралығында деректерді өңдеу жүйесін эксплуатациялайтын кәсіпорындар бағдарламаларды, процедураларды жазуға және деректерді сақтауды маңызды құралдарды жұмсайды. Фирма есептік орнатуда деректер қорын басқарудың жаңа бағдарламалық қамтамасын пайдалана бастаған кезде, ол осы орнатудағы бағдарламамен жұмыс жасай алады. Мұндай шарт бағдарламалық және ақпараттық үйлесімділіктің болуын талап етеді, және оның болмауы жаңа деректер қорын басқару жүйесіне өту кезінде тоқтататын негізгі фактор болуы мүмкін. Бірақ өткенмен байланыс мәселесі деректер қорын басқару құралдарының дамуын тоқтатпағаны маңызды.

1.2.9 Болашақпен байланыс

Болашақпен байланыс ерекше маңызды болып саналады. Болашақта пайда болатын орталарға бейімдеуге барынша жарамды болу керек. Болашақта ақпарат және олардың сақталу-өңделу ортасы көптеген бағыт бойынша өзгеруі әбден мүмкін. Кез келген коммерциялық ұйым уақыт өте өзгерістерге ұшырайды. Әсіресе, деректерді өңдеу жүйесінің қолданушысы үшін бұл өзгерістер күрделі болады. Өте қарапайым өзгерістерді іске асыруға қажетті үлкен шығындар осы жүйелердің дамуын қатты тежейді. Бұл шығындар деректердің қайта құрылуына, қайта жазылуына өзгерістер енгізу нәтижесінде пайда болған қолданбалы бағдарламаны ретке келтіруге жұмсалады. Уақыт өте мекемедегі қолданбалы бағдарламалардың саны артады, және сондықтан барлық бағдарламаларды қайта жазудың болашағы іске аспайды. Деректер қорын өңдеу кезінде ең маңызды тапсырмалардың бірі – деректер қорын оның өзгерісін қолданбалы бағдарламалардың модификациясынсыз орныдауға болатындай жоспарлау.

1.2.10 Пайдалану қарапайымдылығы

Деректерді жалпы логикалық және бағдарламалық сипаттаудың көрінісі үшін қолданылатын құралдар қарапайым және көрнекі болуы керек.

Бағдарламалық қамтаманың интерфейсі соңғы қолданушыға бағытталуы керек және қолданушы деректер қоры теориясы бойынша қажет етпейтін мүмкіндікті ескереді. Бағдарлама тапсырыс берушінің барлық талаптарына сай болуы керек. Сонымен қатар барлық тілектері мен ұсыныстары барынша қолайлы іске асырылуы керек.

2 Бағдарламаны құру құралдары

2.1 Бағдарлама интерфейсін дайыдау құралы.

2.1.1 Бағдарлама интерфейсін дайыдау құралы. Microsoft Visual Studio.

Microsoft Visual Studio – Microsoft компаниясының шығаратын бағдарламалар топтамасынан. Бұл бағдарлама дайындауға арналған интеграцияланған бағдарламалау ортасы және көптеген басқада құрал саймандары бар өнім. Осы орта графикалық интерфейсі бар консольдық қосымшалар жасауға мүмкіндік береді. Алынған консольдық қосымшаларды құрасыру кезінде Windows Forms технологиясын қолдануға болады. Сонымен қатар web-сайттар, web-қосымша, web-қызмет жасау мүмкіндігі бар.

Microsoft Visual Studio құрамында бастапқы кодты өзгерту редакторы және бағдарламаның сыртқы келбеті мен жұмысын қозғамай ішкі құрылымын өзгерту мүмкіндігі бар. Редактор IntelliSense технологиясын қолдайды. Іштей кірістірілген қалыпқа келтіруші (отладчик) машиналық кодты дұрыстаушы ретінде, бастапқы кодты қалыпқа келтіруші ретінде қызмет атқара алады. Басқа құралдар графикалық қосымша жасауды жеңілдетуге арналған.

Web-қосымша жасау құралдары: web-редактор, класстар-дизайнері және деректер сұлбасының дизайнері. Visual Studio бағдарлама дайындаудың әр кезеңінде қосымша кеңейтілу қосу немесе дайындау мүмкіндігі бар, сонымен қатар бастапқы кодты бақылау және қолдау жүйелерін қосу, жаңа құралдар жиынтыған алу немесе дайындау, визуалды жобалау мүмкіндігіде бар.

Visual Studio құрамында келесі компоненттер бар:

- Visual Basic .NET;
- Visual C++;
- Visual C#;
- Visual F#. Visual F# 2010 жылғы шығарылымда енгізілді;
- Microsoft SQL Server.

Visual Studio тарихына көз жүгіртіп өтейік. Visual Studio 97 – алғаш шығарылған өнім. Бұл бағдарламалық қамтама дайындауға арналған сан түрлі құралдар бір ортаға біріктіріліп алғашқы шығарылған орта. Visual Studio 97 екі жинақ ретінде шығарылды: Professional және Enterprise. Осы шығарылым құрамында келесі бағдарламалау құралдары болды: Visual Basic 5.0, Visual C++ 5.0, Visual J++ 1.1, Visual FoxPro 5.0, Visual InterDev. Олар Developer Studio деп аталатын бір бағдарламалау ортасын қолданды. Visual Studio 6.0 Win9x платформасында жұмыс жасайды. Бұл орта 1998 жылы шығарылғаныменен Visual Basic бағдарламалау тілін қолданатын мамандардың арасында әлі де пайдаланылуда. Windows терезелеріне ұқсатып жасалынатын қосымшаларды дайындауға арналған алғашқы бағдарлама болды.

Visual Studio .NET – Rainier жасырын атымен дайындалып 2002 жылдың ақпан айында шығарылды. .NET Framework 1.0 құралы қосылып шығарылды.

Visual Studio .NET 2003 – Everett жасырын атымен дайындалып, 2003 жылдың сәуір айында .NET Framework 1.1 қамтамасымен бірге шығарылды. 2005 жылдың сәуір айында осы қамтаманың жаңа Microsoft Visual Studio .NET 2003 Professional Special Edition деп аталатын нұсқасы шығарылды. Бұл қамтамаға SQL Server 2000 Developer Edition қосылып шығарылды.

Visual Studio 2008 – Orcas жасырын атпен дайындалды. .NET Framework 3.5 қамтамасымен бірге 2007 жылдың қараша айының 19 күні шығарылды. Windows Vista операциялық жүйесіне арнап бағдарламалар дайындауға арналғанымен, Windows XP операциялық жүйесінде қолдайды. Осы шығарылымдағы жаңалық орыс тілінде бағдарламаны орнатуға болатындығы.

Visual Studio 2010 – Hawaii жасырын атымен дайындалып, 2010 жылдың 12 сәуірінде шығарылды. с .NET Framework 4.0 нұсқасы қосылды. Сонымен қатар C# 4.0, Visual Basic .NET 10.0, және F# тілдері кірістірілді. Visual Studio 2012 Windows Store-ға арнап қосымшалар дайындауға мүмкіндік береді. 2012 жылғы шығарылым бес нұсқада таратылды. Барлық бағдарламалау тілдері қойылады және қатар пайдалану мүмкіндігі бар.

Орнатылымдарына тоқтала кетейік.

Visual Studio Express орнатылымында жеңіл версиялы бағдарламалау құралдары жинақталып шығарылады. Жеңіл версиялы құралдардың толық нұсқасынан айырмашылығы бір ғана платформаға арналуында. Яғни, кластар дизайнері, плагиндерді қолдау мүмкіндігі және деректер дизайнерінде қашықтықтан дерек қорын басқару мүмкіндігі болмайды. Сонымен қатар 64 битті код компиляторы бұл орнатылымда жоқ. Бұл орнатылымдар бағдарламалауды үйренушілерге арналған. Орнатқан кезде артық құралдар болмағандықтан дербес компьютер жадысынан көп орын алмайды. Қазіргі таңда төмендегідей Express орнатылымдар бар:

- Visual Basic Express;
- Visual C++ Express;
- Visual C# Express;
- Visual Web Developer Express.

Visual Studio 2012 бірге тағыда жана Express қойылымдар шығарылды.

Олар:

- Visual Studio Express 2012 for Web — web-бағдарламалаушыларға;
- Visual Studio Express 2012 for Windows 8 — modern-интерфейсімен бағдарламалар әзірлеу үшін (C#, VB.Net, C++, JavaScript – тілдері);
- Visual Studio Express 2012 for Windows Desktop — тақталы қосымшалар дайындау үшін (C#, C++, Visual Basic.Net – тілдері);
- Visual Studio Express 2012 for Windows Phone — Windows Phone 7.5 и 8.0 платформасына арнап бағдарлама дайындаушыларға;
- Visual Studio Team Foundation Server Express 2012.

Visual Studio Standard шығарылымы MSDN Library бөлімінің толық нұсқасын ұсынады. XML және XSLT кодтарын түрлендіруді қолдайды. Сонымен объекттерді тестіден өткізу мүмкіндігі бар. Бірақ Microsoft SQL Server серверлерін қарау және интеграциялау мүмкіндігі жоқ. Мобильді құралдарға арнап бағдарлама жазу мүмкіндігі 2005 жылғы осы орнатылымға қосылса, 2008 жылғы орнатылымда тек Professional орнатылымында ғана шығарылды, ал 2010 жылғы орнатылымдарда толық алынып тасталынды.

Visual Studio Professional орнатылымы Standard Edition мүмкіндіктерінің барлығын қолдайды. Microsoft SQL Server серверлерін қарау және интеграциялау мүмкіндігі де қол жетімді. Сонымен қатар қашықтықтан өңдеу мүмкіндігі қосылған.

2.1.2 .NET Framework

.NET Framework — бағдарламалық платформа. Microsoft компаниясымен 2002 жылы шығарылды. Платформа негізі ретінде Common Language Runtime (CLR) тілі жалпы тілдік орындау ортасы болып табылады. Бұл тілдік орындау ортасы көптеген бағдарламалау тілдерін қолдайды. CLR тілдік ортасының функционалдық мүмкіндіктері, осы ортаны қолданатын барлық тілдер үшін қол жетімді.

.NET Framework платформасы Sun Microsystems компаниясы шығарған Java платформасынан соң жарық көрді. .NET Framework Microsoft компаниясының жауабы ретінде шығарылған бағдарлама болып бағаланады. Sun Microsystems компаниясы қазіргі таңда Oracle компаниясына тиесілі.

.NET платформасы Microsoft компаниясының патенттелген технологиясы. Бірақ тек Microsoft Windows операциялық жүйелерімен жұмыс жасауға негізделген.

Платформаны даярлау 1999 жылы басталды. Бірақ бағдалама туралы ресми хабар тек 2000 жылдың 13-ақпанында жасалды. Осы күні Билл Гейтс Microsoft компаниясының бас директоры лауазымын Стив Балмерге тапсыратындығы туралы айтқан болатын. Сол жерде Microsoft компаниясының жаңа даму стратегиясы айтылды. Ол стратегия Microsoft компаниясы тарихында Next Generation Windows Services деген атау алды. Бұл стратегия бойынша Microsoft компаниясы бұрын шығарылған және болашақта шығарылатын бағдарлама арасында байланыс өткізу керек.

Сол күні өткізілген пресс конференцияда Стив Балмер корпорация шығаратын бағдарламалар ендігі болашақта дербес компьютерге емес мобильді құрылғыларға бағдарланатын болатындығын жария етті.

Мобильді құрылғылардың жұмыс жасау күшінің аздығына байланысты қосымшалардың сақталуы және тасмалдануы серверде болуы керек болды. Ал дербес компьютердегі ақпарат жадыда локалді сақталатын еді. Сол кездегі компаниялар клиент серверлі технология көшуді барлығы қолдаған болатын.

Microsoft компаниясы 2000 жылы жаңа стратегия таңдауының себептері көп болды. Компания операциялық жүйе және веб-браузерлер шығару аясында

басты орында болды, сонымен қатар интернетті пайдалануға арналған бағдарлама шығаруға арналған көптеген зерттемелері болды. Бағдарламалаушыларға арналған тілдік құралдар және орталар саласындада көптеген жаңа идеялары іске асырылып жатты. Ол кезде Visual Basic және C++ тілдерінің әр қайсысының өзіндік құралдары мен бағдарламалау ортасы болды, басты мақсат осы орталарды бір бірімен сәйкестендіру болды. Ал ең бастысы жаңа платформа объектіге ориентерленген болу керек еді.

Сол кезде жарияланған жоспарға сәйкес жаңа операциялық жүйе жасалу керек еді. Сонымен бірге бағдарламашыларға арналып жаңа бағдарламалау ортасы дайындалу керек болды. Жаңа орта веб-ориентерленген қосымшалар жазуға арналды. Жаңа платформаға Microsoft Office-те көшірілу керек болды.

2000жылы платформа Microsoft.NET деп аталып, компания өнімдерінің барлығы осы платформада қызмет атқаратын болып шешілді. 2000 жылдың 12 қарашасында платформаның бетта нұсқасы шығарылды. Платформа NET Framework SDK Beta 1 деген атпен шығарылды. Бірақ бағдарламада көптеген қателіктер болды. Шығарылған платформа тек Windows 2000, Windows NT 4.0, Windows 98 и Windows ME операциялық жүйелерінде ғана қызмет атқарды. Ал платформадан бұрын сол жылы шығарылған SQL Server 2000 бірге платформа дұрыс жұмыс жасамады. Тағыда айта кететін жайт платформаны орнату барысында да көптеген қиындықтар туындап отырды.

.NET Framework платформасында жазылған кез келген бағдарлама алдымен компилятор көмегімен .NET ортасына ортақ байт-код Common Intermediate Language (CIL) аралық кодына ауыстырылады. Содан соң код Common Language Runtime (CLR) виртуалды машинасымен орындалады немесе NGen.exe утилитасы көмегімен машиналық кодқа ауыстырылады. CLR виртуалды машинасы қолданылса платформаға кірістірілген JIT-компилятор аз уақыт аралығында байт-кодты машиналық кодқа түрлендіреді. CLR виртуалды машинасы базалық қауіпсіздікті қамтамасыз етеді, жадыны басқарады және ерекшеліктер жүйесін анықтайды.

.NET объект класстары барлық қолданатын бағдарламалау тілдері үшін қол жетімді және Framework Class Library (FCL) кітапханасында сақталған. Framework Class Library (FCL) құрамына Windows Forms, ADO.NET, ASP.NET, Language Integrated Query, Windows Presentation Foundation, Windows Communication Foundation кластары кіреді.

.NET платформасын қолдайтын бағдарламалау орталары:
Microsoft Visual Studio (C#, Visual Basic .NET, Managed C++, F#);
SharpDevelop;
MonoDevelop;
Embarcadero RAD Studio (Delphi for .NET);
панее Borland Developer Studio (Delphi for .NET, C#);
Zonnon;
PascalABC.NET.

.NET Framework 1.0 – 2002 жылы 5 қаңтарда Windows 98, NT 4.0, 2000 және XP операциялық жүйелеріне арнап шығарылды. Бұл платформаны қолдау 2007 жылдың 10-шы шілдесінде аяқталды

.NET Framework 1.1 – 2003 жылдың 1-ші сәуірінде шығарылды. Windows Server 2003 операциялық жүйесімен бірге автоматты түрде орнатылатын алғашқы платформа болды. Бұрын шығарылған операциялық жүйелер үшін .NET Framework 1.1 жеке орнату бағдарламасы ретінде қол жетімді болды. Бағдарламаны қолдау 2008 жылы аяқталды.

.NET Framework 2.0 Visual Studio 2005, SQL Server 2005 және BizTalk 2006 өнімдерімен бірге шығарылды. 64-биттік платформаларды толық қолдайтын болды. Бапталатын класстар қосылды. 2011 жылы бұл платформаны қолдау аяқталды.

.NET Framework 3.0 WinFX атымен дайындалды. Осы кеңейтілумен жасалды. 4 жаңа компонент қосылды. Олар:

- Windows Presentation Foundation (WPF) — XAML қолданатын презентациялық графикалық жүйе;

- Windows Communication Foundation (WCF) — платформа аралық байланысты қамтамасыз ететін біртекті бағдарламалық модель;

- Windows Workflow Foundation (WF) — жұмыс процесстерін анықтайтын, басқаратын және орындайтын технология;

- Windows CardSpace — унификацияланған идентификация технология.

Бағдарламаны қолдау 12 шілде 2011 жылы аяқталды.

.NET Framework 3.5 алдыңғы нұсқаларымен салыстығандағы жаңалықтары:

- C# 3.0 и VB.NET 9.0;

- LINQ тілі мен LINQ to Objects, LINQ to XML және LINQ to SQL провайдері қосылған;

- ASP.NET AJAX орнатылымдарға қосылға;

- WF және WCF функционалдығы кеңейтілген;

- System.CodeDom кеңістігі қосылған.

.NET Framework 4.0 платформасының бірінші бета нұсқасы 2009 жылы Visual Studio 2010 ортасының бета нұсқасымен бірге шықты. Платформаға енгізілген жаңалықтар:

- Parallel Extensions — PLINQ (Parallel LINQ) және параллелді есептер кітапханасы (Task Parallel Library). Кітапхана көппроцессорлы және таратушы жүйелерге арнап бағдарлама жазуды жеңілдетуге арналған;

- Visual Basic және C# тілдерінің соңғы нұсқалары енгізілді;

- Managed Extensibility Framework (MEF) технологиясы енгізілді;

- IronPython, IronRuby және F# толық қолдайды;

- .NET Framework және ASP.NET таралымдарын Server Core технологиясында қолдау;

- Code Contracts-ты қолдау;

- Средства моделирования Oslo моделдеу құралы және M бағдарламалау тілі қосылған. Құрал объектке-ориентерленген тілдері мен модельдерді

дайындауға арналған предназначенный для создания предметно-ориентированных языков и моделей.

.NET Framework 4.5 орнатқан кезде .NET Framework 4.0 платформасын өшіреді, Windows XP-дан төмен нұсқаларымен жұмыс жасамайды. Жаңа енгізілімдер:

- орамадан ашу кезінде компьютерді өшіріп қайта жүктеу санын азайту;
- 64-разрядты платформада екі гигабайттан асатын массивтерді қолдау;
- фондық режимде программа қалдықтарын жинап өшіріп отыруға байланысты бағдарлама өнімділігі арттырылған;

- JIT қажет еткен кезде фондық режимде компиляция жасау. Көп разрядты жүйеде қол жетімді;

- көп кездесетін өрнектерді өңдеушінің жұмыс уақытын басқару;

- қосымшалар домені үнсіз келісім бойынша әдебін анықтау;

- Юникод (UTF-16) кодтауын қолдау;

- Жолдарды реттеу және салыстыру;

- ZIP сығуды қолдау;

- айналық көшірмені баптау;

- 2008 нұсқаның стандарттарын, интернационалды домен аттарын, Windows 8 кластарын қолдау;

- Жолдарды салыстыруды операциялық жүйеде сатылы бөлу;

- Хэш-кодты кез келген қосымша домені негізінде есептеу мүмкіндігі.

.NET Framework 4.5.1 платформасы 2013 жылдың 17-қазан айында шығарылды. Бұл нұсқа Windows Vista SP2 немесе бұдан жаңа нұсқаны қажет етеді. Және Windows 8.1 и Windows Server 2012 R2 жүйелерімен бірге қойылады.

Microsoft .NET платформасының басты идеяларының бірі, әр түрлі тілде жазылған бағдарламалық бөліктердің сәйкестігі. Мысалы, С++ тілінде жазылған қызмет Microsoft .NET платформасының көмегімен Delphi тілінде жазылған бағдарлама кітапханасына жүгіне алады. С# тілінде Visual Basic .NET тілінде жазылған класстың мұрагер классын жазуға болады. Microsoft .NET платформасында әр бір кітапхана өзінің шығарылымы туралы ақпаратын сақтайды, ол ақпарат бұрынғы немесе кейінгі шығарылымдармен туындауы мүмкін қиындықтарды болдырмауға немесе түзетуге арналған.

Microsoft Visual Studio ортасымен бірге орнатылатын тілдер тізімі:

- С#;

- J# — соңғы рет Visual Studio 2005 бағдарламасында болды;

- VB .NET;

- JScript .NET;

- С++/CLI — жаңа нұсқасы Managed С++;

- F# —ML бағдарламалау тілдерінің отбасынан, Visual Studio 2010

қосылған.

2.1.3 C# тілі

C тілінің тарихы 1972 жылдан бастау алады, яғни, Деннис Ритчи мен Брайан Керниганның құрамында жоғарғы деңгейлі тілдің мүмкіндіктері бар, кейін Америка Ұлттық Стандарттау Институты (ANSI) нақтылаған программалау тілінің жасалған кезінде.

1980 жылы Бьярна Страуструптың қабілетіне байланысты ANSI – дің өкілі – C тілі пайда болды. Онда басқа да тілдердің жақсы жақтары көрініс тапты. C тілі программистке дәстүрлі құрылымдық және объектілік бағытталған программаларды жасауға мүмкіндік береді.

C тілін қолданып жасалған программалық қамсыздандыру идентификаторлар, кілттік сөздер, функциялар, тұрақтылар, препроцессорлар, құрылымдар, массивтер және т.б элементтерден тұрады.

C тіліндегі нәтижені экранға Hello, World қатарлары арқылы шығатын программаны қарастырайық.

Кез – келген программаның мақсаты, деректі енгізу, сақтау, модификациялау және шығару болып табылады.

Идендификатор – программаның құрылымы болып табылады.

Идендификатор арқылы айнымалы аты, функция және белгі аты түніледі. Програмада идендификатор жазбалы және қатарлы латын әріптерін, сандвр және регитрдің есебі бойынша кілттік сөзбен сәйкес келмеуге тиіс анықтайтын символдан немесе әріптен міндетті түрде басталатын анықтауыш символдан тұруы мүмкін. Жоғарыда көрсетілген мысалда a, b, c, d идендификаторлары көрсетілген.

Кілттік сөздер – бұл өзіндік арнайы тағайындаулары бар тілдік қор идендификаторлары.

C# — объектіге ориентирленген бағдарламалау тілі. 1998-2001 жылдар аралығында Андерс Хейлсберг бастауымен Microsoft корпорациясында жасалып шығарылды. Microsoft .NET Framework платформасына бағдарлама жазуға негізделіп жасалды. Сонан соң ECMA-334 және ISO/IEC 23270 стандарттарына сәйкестендірілді.

C# тілі C текті тілдер тобына жатады. Сол топтың ішінде C# тілінің синтаксисі C++ тіліне және Java тіліне жақын. Тіл статистикалық типизациясы бар, полиморфизмді қолдайды, операторлар қайта жүктелуін қарастырады, делегаттар, атрибуттар, сипаты, итераторлар, анонимді функциялар, LINQ, айрықшылық, комментарилер сынды функцияларды қолдайды.

Алдында ншығарылған тілдерден көптеген қасиеттерін жинастырған. C++, Pascal, Модула, Smalltalk және Java тілдерінен көптеген мүмкіндіктері алынған. Осы тілдердің практикасына сүйене отырып бағдарлама жазуда қиындық туындайтын тұстарын өзгертті. Мысалы, C++ тіліменен салыстырған кезде C# тілі класстардың көп түрлі мұралануын қолдайды.

C# тілі қолданбалы деңгейдегі CLR бағдарламалау тілі ретінде дайындаған еді. CLR бағдарламалау тілдік ортасының мүмкіндіктеріне байланысты. Бұл C# типті жүйелерге байланысты. Тілдің өзіндік күрделілігі

немесе жеңілдігінің болу болмауы әртүрлі бағаланады. Тілдің өз ерекшелігі CLR конструкциясына транслитерация жасалуы мүмкін. CLR 1.1-ші нұсқадан 2.0-ға дейін жетілдірілгенде C# тіліде көптеген өзгерістерге ұшылады. C# 3.0 тілі бұл үрдіске кірмейді, себебі бұл нұсқа өте ыңғайсыз әрі күрделі жасалған. C# тілі CLR бойынша .NET бағытталған тіл. Мысалы артық қалдықтарды жинастыру C# тіліменен емес CLR тілдік ортасымен іске асырылады.

C# тілі атауы музыкадан бастау алады, соның ішінде ноталық белгілеуден. «#» диез белгісі. Бұның астарлы мәні C# тілі C тіліне жоғары дегенді білдіреді.

Шығарылымдарына тоқталып өтелік.

C# 1.0 нұсқасы 1998 жылы бастау алған жоба нәтижесінде 2000 жылы шығарылды. Ал негізгі шығарылымы Microsoft Visual Studio .NET бағдарламасыменен 2002 жылдың ақпан айында шығарылды. Бірінші нұсқа Java 1.4 өте қатты ұқсады. Бірақ кеңейтілуі үлкенірек болатын. Сонымен қатар C# тілінде келесідей қасиеттер болды: индексаторлар, оқиғалар, циклдар, структуралар атрибуттар және тағы басқалары.

Сонымен қатар C# тіліне C++ тілінің кейбір қасиеттерін аударылды. Олар: белгісіз типтер, операторларды қайта жүктеу, параметрледі беру, айнымалы сан мәндері, goto операторы, нұсқағыштар ерекшеленген жұмыс жасау мүмкіндігі.

C# тілінің 2.0 нұсқасы. 2003 жылы корпорация 2.0 нұсқаны дайындау үстінде екенін хабарлады, ал 2004 жылы бета нұсқасы шығарылды. Ал соңғы нұсқасы Visual Studio 2005 және .NET 2.0 бағдарламаларымен бірге 2005 жылдың 7-ші қарашасында жарық көрді.

2.0 нұсқасындағы жаңа мүмкіндіктер:

- Ішінара типті (бір файлдан артық кластарды жүзеге асыруды бөлу)
- Жинақталған немесе параметрленген типті (generic). C++ үлгілерімен салыстырғанда, олар кейбір қосымша мүмкіндіктерді қолдайды және виртуальдық машина деңгейінде жұмыс істейді. Сонымен қатар, сөздер жинақталған типті параметрлер бола алмайды, олар толықтай немесе жартылай мамандындырылған бола алмайды, үнсіздік бойынша үлгі параметрлерді қолдамайды, үлгілік параметрлерден мұралануға болмайды және т.б.:

- Yield кілттік сөзінің көмегімен бағдарламаларды құруға мүмкіндік беретін итератордың жаңа түрі Python мен Ruby ұқсас.

- Тұйықталудың функциональдығын қамтамасыз ететін аноним әдістер.

- Нөлденетін (“nullable”) сөз-типтер, null мәнін қабылдай алады. Мұндай типтер SQL тілі арқылы деректер қорымен қарым-қатынасты жақсартуға жағдай жасайды.

- Сақталатын процедуралары, триггерлерді және деректер типтерін .Net тілдерінде, соның ішінде C#-та да, құруға мүмкіндік береді.

- 64-разрядты есептеулерді қолдайды, ол адресік кеңістікті арттыруға және 64-разрядты көне деректер типін қолдануға мүмкіндік береді.

3.0 нұсқасы. 2004 жылдың маусым айында Андерс Хейлсберг алғаш Microsoft сайтында C# 3.0 тіліндегі жоспарланып жатқан кеңейтілімдер туралы айтты. 2005 жылдың қыркүйегінде C# 3.0 айрықшалама жобасы және C# 3.0 бета-нұсқасы шықты. Ол пайдаланылып жүрген Visual Studio және 2005 Net 2.0 қосымшасы түрінде орнатылады. Бұл тілдің нұсқасы ақырында Visual Studio 2008 және .Net 3.5 кірді.

3.0 нұсқасындағы жаңа мүмкіндіктер

C# 3.0-да тілге келесі радикалдық қосымшалар пайда болды:

- Select, from, where кілттік сөздері SQL, XML, жинақтамалар және т.б.-дан сұраныстар жасауға көмектеседі.

- Объектінің инициализациясы және оның қасиеттерімен: `Customer c = new Customer(); c.Name = "James"; c.Age = 30;`

- Мұны былай жазуға да болады: `Customer c = new Customer { Name = "James", Age = 30};`

4.0 нұсқасы

C# 4.0 алғаш 2008 жылдың соңында Visual Studio 2010 CTP нұсқасымен бірге ұсынылды.

Visual Studio 10.0 және C# 4.0 Visual Studio 2010 шығарылымымен қатар 2010 жылдың сәуір айында шығарылды.

4.0 нұсқасындағы жаңа мүмкіндіктер

- Пайдалану үшін кейінгі байланыстыруды қолдану мүмкіндігі:

- динамикалық тұрпаттандыру тілдерімен (Python мен Ruby);

- COM-объекттерімен;

- көрсетілу (reflection);

- өзгертілетін құрылымды (DOM) объекттер. Dynamic кілттік сөзі шығады.

- аталған және опциональды параметрлер;

- COM interop жаңа мүмкіндіктері;

- коварианттылық пен контраварианттылық;

- келісімшарт код ішінде (Code Contracts).

2.2 Деректер қорын құру

2.2.1 Деректер қоры түсінігі

Осы бөлімді жазу үшін қолданылған әдебиеттерде дерек қоры терминіне нақты бір анықтама берілмеген. Менің жұмысыма сәйкес келетін, мағынасымен жуық анықтама Тайлер Коннолли мен Кролин Беггтің «Базы данных. Проектирование, реализация и сопровождение» кітабында берілен. Деректер қоры дегеніміз логикалық байланысы бар, бір тұтас қолданылатын, ақпараттық қажеттіліктерді қанағаттандыратын деректер жиынтығы деп жазылады. Барлық қаралған анықтамаларда байқалатын терминнің бірегей қасиеттері бар. Бұл терминнің қасиеттері Е. А. Мирошниченконың «К формальному определению

понятия «база данных» еңбегінде, меніңше, өте нақты, ұтымды көрсетілген. Автор келесі қасиеттерді ерекшелеп көрсетеді:

Деректер қоры есептеу техникасында сақталады және өңделеді. Бұл жердегі есептеу техникасы дегеніміз ақпаратты өңдеу процессін автоматтандыруға арналған құрылғы немесе құрылғылар жиынтығы.

Деректер қорындағы деректерге есептеу техникасында оңай қол жеткізу үшін оларды логикалық сипат бойынша топтастырады.

Деректер қорына деректер қорының логикалық құрылымын сипаттайтын сұлбалар мен метадеректер кіреді.

Деректер қорының түрлері өте көп. Мысалы М.Р. Когаловский жазған «Деректер қоры технологиясының энциклопедиясында» 50-ден астам дерек қорларының түрлері келтірілген. Дерек қорлары түрлі сипаттамалар бойынша ерекшеленеді.

Деректер моделі бойынша топтау:

- иерархиялық;
- объекттік және объектіге ориентирленген;
- объектті реляциялық;
- реляциялық;
- жүйелік;
- функциялық.

Сақталу орнына байланысты топталуы

- екінші дәрежелі жадыда немесе әдеттегі дәстүрлі жады;
- жедел жадыда;
- үшінші ретгі жады (қатқыл диск, магнитті лента және басқа құрылғылар).

Дерек қорларын тағы басқа көптеген сипаттамалар бойынша шексіз топтауға болады. Бұл сала зертеу объектісі ретінде салыстырмалы түрде аз уақыт қарастырылуда сондықтан болашақта анықтамалар, топтаулар бұның үстіне көбейе түседі деп болжауға болады.

Дерек қорының өзіндік қызмет ету циклі бар. Қызмет ету циклі дегеніміз дерек қорының жасалуынан бастап қолданыстан шығарылғанға дейін өтетін кезеңдерінің жиынтығы.

Дерек қорының өмір сүру циклі:

- дерек қоры жазылған кезде шешілетін мәселенің анализі;
- дерек қорының жалпы моделін жасау;
- дерек қорының бастапқы (негізгі) кестелерін алу;
- дерек қорының бүтіндігін тексеру;
- дерек қорыны сақтау пайдалану техникалық құралдарын таңдау;
- кіріс және шығыс формасының жобасын әзірлеу;
- қамтаманың интерфейсін әзірлеу;
- қамтаманы функционалды толықтыру;
- қамтаманың толықтыруларын тексеру;
- деректердің кіріс және шығыс формасын тестіден өткізу;

- қолданысқа енгізу. Қызметкерлерді оқыту;
- қажет болған жағдайда бағдарламаға жаңалықтар енгізу;
- бағдарламаны қолданыстан шығару. Деректерді жаңа дерек қорын басқару жүйесіне ауыстыру;

Деректер қорымен жұмыс жасауға арналған бағдарламаларды деректер қорын басқару жүйесі деп біріктіріп атайды.

Деректер қорын басқару жүйесінің негізгі қызметтері:

- сыртқы жадыдағы деректерді басқару;
- жедел жадыдағы деректерді басқару;
- өзгерістерді тіркеу, қосымша көшірме жасау және ақаулықтардан соң деректер қорын қайта қалпына келтіру;
- деректер қорының деректерін анықтау және өзгерту тілдерін қолдау.

Деректер қорын басқару жүйелерін топтастыру бір неше қасиеттерге байланысты, олар: деректер моделі, таралуына және деректер қорына қол жеткізу амалы. Деректер моделіне байланысты деректер қорының топтасуы жоғарыда жазылғандықтан, кең таралу қасиетіне байланысты топтасуды қарастырамын. Деректер қорын басқару жүйесі таралуына байланысты екіге бөлінеді: локальді (барлық бөліктер бір компьютерде орналасады) және бөлшектелген (екі немесе одан көп компьютерге таратылған). Деректер базасына қол жеткізу амалы бойынша жүйе үшке бөлінеді:

Файл-серверлік. Файл-серверлік дерек қорын басқару жүйесінде дерек файлдары центрлік файл-серверде орналасады. Ал басқару жүйесі әрбір клиент компьютерінде орналасады. Жүйе серверде сақталған деректерге локальді желі арқылы қол жеткізеді. Мұндай жүйе мысалы ретінде Microsoft Access, Paradox, dBase, FoxPro, Visual FoxPro бағдарламаларын алуға болады.

Клиент-серверлік. Клиент-серверлік дерек қорын басқару жүйесі дерек қорыменен бірге серверде орналатылады және дерек қорына монопольді режимде қол жеткізіп отырады. Барлық клиенттік өңделуге жолданған сұраныстар клиент-серверлік басқару жүйесімен орталықтандырылып қарастырылады. Бұл жүйенің ұтымды тұстары жоғары сенімділік, қол жетімділігі және қауіпсіздігі. Мысал ретінде Oracle, Firebird, Interbase, IBM DB2, Informix, MS SQL Server, Sybase Adaptive Server Enterprise, PostgreSQL, MySQL, Caché бағдарламаларын алуға болады.

Кірістірілмелі. Кірістірілмелі дерек қорын басқару жүйесі негізгі бағдарлама ретінде қойылмайды. Басқа бағдарламаның құрамдас бөлігі ретінде келеді. Бұл жүйе шағын ақпаратты локальді сақтауға арналған, жалпы желі арқылы қолданылмайды. Мысалы OpenEdge, SQLite, BerkeleyDB, Firebird Embedded, Microsoft SQL Server Compact бағдарламалары.

2.2.2 Microsoft SQL Server 2008 R2

Microsoft SQL Server - реляциялық деректер қорын басқаруға арналған жүйе. Бағдарлама Microsoft корпорациясымен жасалған. Сұраныстарды құрастыру тілі Transact-SQL, Microsoft және Sybase компанияларының серіктестігі нәтижесінде жасалынған. Transact-SQL тілі ANSI/ISO стандартына сәкес келеді. Microsoft SQL Server дербес және ірі кәсіпорын деректер қорын басқару үшін қолданылады.

Клиент-сервер технологияларының дамуы өткен ғасырдың 80 жылдарында басталды, оған дербес компьютер мен компьютерлік желі салаларының қарқынды дамуы себеп болды. Бастапқы кезде деректер қорын басқару жүйесі тек mainframe-де қолданылды.

1983 жылы oracle компаниясы Oracle v3 басқару жүйесін қолданысқа шығарды, бұл деректер қорын басқару жүйелерінің арасындағы дербес компьютерде қызмет атқаратын алғашқы бағдарлама. 1986 жыл бұл технологиялардың дамуында ең жемісті жыл болды. Осы жылдары бағдарлама әзірлеуші компаниялар көбейді, соладың ішінде көзге түскені Sybase компаниясы. Sybase компаниясы Sun Microsystems интеллектуалды жұмыс станцияларын Oracle деректер қоры серверімен шығаруды бастады. Бұл станцияларда клиент-сервер технологиясы ақпаратты өңдеу модульін интерфейс модулінен бөлуге мүмкіндік берді. Осы кезде иерархиялық деректер қорынан реляциялық деректер қорына көшу жүзеге асырылған болатын. Бұл үрдісті IBM компаниясы бастаған болатын. Жаңа басқару жүйелерінде ақпаратты өңдеу жылдамдығы төмен болғанымен, қызмет көрсету және бағдарламалау мүмкіндігі әлде қайда жеңілдетілді.

1986 жылдың қыркүйек айында Gupta Technologies компаниясы SQL Base бағдарламасын жарыққа шығарды, бұнда іске асырылған жаңалықтың бірі серверде сақталған деректерге тікелей қол жеткізу болды яғни серверде сақталған кестелерді және файлдарды клиенттік компьютерге көшіру қажет болмады. Осы жылдың ағында деректер қорын басқару тілі ретінде SQL тілі кеңіне қолданылатын болды. American National Standards Institute осы 1986 жылы SQL тілін деректерді өңдеу саласының әлемдік стандартты тілі ретінде бекітті.

1988 жылдың 13-ші қаңтарында Америка Құрма Штарының Нью-Йорк қаласында ресми пресс-конференция өткізілді. Бұл пресс-конференцияда Ashton-Tate және Microsoft компаниялары бірлесіп, Sybase технологиялары негізінде жаңа Ashton-Tate/Microsoft SQL Server бағдарламалық өнімін сол жылдың екінші жартысында қолданысқа шығаратынын хабарлаған болатын. Сол күні шығарылған пресс-релизға байланысты Ashton-Tate компаниясы деректер қорының әзірленуін бақылау керек болатын, Microsoft компаниясы локалды желі технологиясында ұқсас қызмет атқаратын болды. Басты рөл жаңа өнімде клиент-сервер қатынасына бөлінген еді. Бұл бағдарламаға келесі жаңалықтар енгізілді: сақталған процедураларды қолдау, көп қолданысты ортада деректердің бүтіндігін қолдау, қолданушы жұмысын тоқтатпай ядроға

қол жеткізу мүмкіндігі, резервті көшірмелер жасау, деректер қоры жіне онлайн-транзакциялар арасындағы байланысты қамтамасыз ететін кез келген технологияларды қолдау. 1989 жылдың 29-шы сәуірінде бағдарлама ресми сатылымға шығарылды. Infoworld журналы өткізген тестің қортындысы келесідей болды бағдарлама 24 станциямен жұмыс жасаған кезде көп қолданысты режимде жұмыс жасайтын қарапайым деректер қорын басқару жүйелерінен әлде қайде тез қызмет атқарды, ал сақталған процедураларды қолданған кезде жауап қайтару уақыты 2 секундтан кем болды. Журналисттердің тағы бір ұнатқан тұсы тесттік кодты жазу жеңілдігі мен қолайлылығы.

Бағдарлама бұдан әрі көптеген өзгерістер мен жетілдірулер енгізіп келесі нұсқаларын шығарды:

- 1990 жылы SQL Server 1.1;
- 1991 жылы SQL Server 1.11;
- 1991-1992 жылы SQL Server 4.2;
- 1992-1993 жылы SQL Server for Windows NT;
- 1993-1995 жылдары SQL Server 6.0;
- 1995-1996 жылдары SQL Server 6.5;
- 1996-1998 жылдары SQL Server 7.0;
- 1998-2000 жылдары SQL Server 2000;
- 2000-2005 жылдары SQL Server 2005;
- 2000-2008 жылдары SQL Server 2008;
- 2008-2010 жылдары SQL Server 2008 R2;
- 2010-2012 жылдары SQL Server 2012.

Бұл дипломдық жұмыста SQL Server 2008 R2 нұсқасы қолданылды. SQL Server 2008 R2 2010 жылдың 21-ші сәуірінде ресми түрде сатыла бастады.

SQL Server 2008 R2 нұсқасында жаңа басқару құралдары бар. Жаңа құралдар бір неше деректер қоры бар орталарды басқару мүмкіндігін ашады. Бағдарлама аппараттық инфраструктураны тиімді қолдануға мүмкіндік береді. Мониторинг терезесі ресурстардың қолданылуын бақылауды қамтамасыз етеді. Ресурстар регуляторы арқылы администраторлар ресурстарды қолдануды шектей алады. Деректерді сығу мүмкіндігі деректерді сақтау шығынын азайтады. SQL Server 2008 R2 256 логикалық процессорға дейін қолдайды. SQL Server StreamInsight технологиясы ағындық ақпаратты нақты уақыттағы режимде өңдеуге мүмкіндік береді. Деректерді сақтау қорының жадысын өзгерту мүмкіндігі бар.

2.2.3 SQL тілі

SQL тілі деректер қорындағы ақпаратты өңдеуге және алуға арналған құрал болып табылады. SQL – құрылымдық сұраныстар тілінің қысқартылғаны, ал толық атауы – Structured Query Language. Айтылғандай, SQL пайдаланушы мен деректер қоры арасындағы өзара әрекет-байланысын ұйымдастырып, орнату үшін пайдаланылатын бағдарламалау тілі. Негізінде SQL реляциялық

типті деректер қорымен жұмыс жасайды. Қабылданған жүйеге сәйкес, есептеу ортасында маңызды, құнды ақпарат сақталатын қоры болады. Егер есептеу жүйесі бизнес саласына қатысты болса, онда деректер қорында шығарылатын өнімнің материалдық құндылықтары, сатылым көлемі және жалақысы жайлы ақпарат сақталуы мүмкін. Персоналды компьютердағы деректер қорында жазып шығарылған чектар, телефондар және мекен-жайлар туралы ақпарат немесе үлкенірек есептеу жүйесінен шығарылған ақпараттар сақталуы мүмкін. Деректер қорын басқаратын компьютерлік бағдарлама деректер қорын басқару жүйесі деп аталады.

Егер пайдаланушыға деректер қорынан деректерді көру керек болса, ол оны деректер қорын басқару жүйесінен SQL тілінің көмегімен сұрайды. Деректер қорын басқару жүйесі сұранысты өңдейді, қажетті деректерді табады және оларды пайдаланушыға жолдайды. Деректерді сұрау процесі және нәтиже алу деректер қорына сұраныс деп аталады: осыдан оның атауы – сұраныс тілі.

Бірақ бұл атау шындыққа сәйкес келмейді. Біріншіден, қазір SQL қарапайым сұраныс құру құралына қарағанда үлкен бағдарлама, алайда ол басында осы сұраныстар құру үшін жасап шығарылған. Соған қарамастан, деректерді өңдеу бұрынғыша SQL-дің маңызды функцияларының бірі болып қала береді, қазір бұл тіл ДҚБЖ-нің пайдаланушыға мүмкіндік туғызатын барлық функционалдық мүмкіндіктерін іске асыру мақсатында пайдаланылады, олардың ішінде:

- деректерді ұйымдастыру. SQL пайдаланушыға деректер көрінісінің құрылымын өзгертуге, сонымен қатар деректер қоры элементтерінің арасында қарым-қатынас орнатуға мүмкіндік береді.

- деректерді оқу. SQL пайдаланушыға немесе қосымшаға деректер қорындағы деректерді оқуға және оны пайдалануға мүмкіндік береді.

- деректері өңдеу. SQL пайдаланушыға немесе қосымшаға деректер қорын өзгертуге, яғни оған жаңа деректер қосуға, сонымен қатар ондағы бар деректерді жоюға және жаңартуға мүмкіндік береді.

- рұқсат етуді басқару. SQL-дің көмегімен пайдаланушының оқу және өзгерту мүмкіндіктерін шектеуге және оларды мүмкіндік берілмеген рұқсат етуден қорғауға болады.

- деректерді бірлесіп пайдалану. SQL пайдаланушы мен жұмысшының деректерді бірлесіп пайдалануын бір-біріне кедергі жасамас үшін, қатар ұйымдастырады.

- деректердің бүтіндігі. SQL деректер қорын үйлестірілмеген өзгерістердің әсерінен бұзылудан және жүйенің істен шығуынан қорғай отырып, оның бүтіндігін қамтамасыз етеді.

Сондықтан, SQL ДҚБЖ-мен өзара әрекет үшін оңтайлы тіл болып табылады.

Екіншіден, SQL – COBOL, FORTRAN немесе C сияқты толыққанды компьютерлік тіл емес. SQL-де шартты тексеруге арналған IF операторы, өтулерді ұйымдастыруға арналған GOTO операторы және DO операторы

немесе циклдарды құруға арналған FOR операторы жоқ. SQL операторлары базалық тілде құрылады, мысалы COBOL, FORTRAN немесе C деректер қорына рұқсат алуға мүмкіндік береді. Одан басқа, C сияқты тілден SQL операторлары функцияның шақыру интерфейстерін пайдалана отырып, ДҚБЖ-сін анық түрде жіберуге болады.

Соңында, SQL – бұл C және Pascal сияқты мықты құрылымданған тілдермен салыстырғанда әлсіз құрылымданған тіл. SQL операторлары ағылшынша сөйлмді еске түсіреді және оператордың мағынасына әсер етпейтін, бірақ оның оқылуын жеңілдететін «бос сөздерден» тұрады. SQL-да қисынсыздық жоқ, сонымен қатар дұрыс сияқты көрінгенімен, мағынасы жоқ SQL операторларының құрылуын болдырмайтын арнайы ережелер атары бар.

Жанама аталуына қарамастан, қазіргі кезде SQL реляциялық деректер қорымен жұмыс жасауға арналған ерекше стандартты тіл болып табылады. SQL – үйрену үшін көп еңбекті қажет еткенімен, жеңіл бағдарламалау тілі болып табылады.

2.2.4 SQL операторлары

SQL тілі сұраныс жасауда, сонымен қатар реляциялық деректер қорын өзгерту және басқару үшін қолданылады. SQL толыққанды бағдарламалау тілі болып есептеледі, оған тек сұраныс операциялары ғана емес, DDL – Data Definition Language – деректерді сипаттау тіліне сәйкес келетін операторлар да қатысады. Бұдан басқа, бұл тілде деректер қорын басқаруға арналған операторлар да бар.

Объектілерді құру және өзгерту үшін деректерді анықтау операторлары DDL қолданылады:

- CREATE TABLE – кесте құру. Деректер қорында жаңа кесте құрады;
- DROP TABLE – кестені жою. Деректер қорындағы кестені жояды;
- ALTER TABLE – кестені өзгерту. Қолданылып отырған кестенің деректерін немесе берілген кестені өзгертеді;
- CREATE VIEW – көрініс құру. Кейбір SQL-сұраныстарға сәйкес келетін виртуалды кестелерді құрады;
- DROP VIEW – құрылған көріністі жояды;
- ALTER VIEW – көріністі өзгерту. Бұрын құрылған көріністі өзгертеді;
- CREATE INDEX – индекс құру. Кейбір кестелерге арнап индекстарға кірген атрибуттарға тез қолжетімділікті қамтамасыз ету үшін индекс құрады;
- DROP INDEX – бұрын құрылған индексті жояды.

Деректерді өзгерту үшін деректерді манипуляциялайтын операторлар пайдаланылады:

- DELETE – деректі өшіру. Негізгі кестедегі фильтрациялау шартымен сәйкес келетін бір немесе бірнеше қатарларды жояды. Оператордың қолданылуы бүтіндікті қолдау принципімен келіседі, сондықтан бұл оператор синтаксис тұрғысынан дұрыс жазылса да, кейде дұрыс орындалмауы мүмкін;

– INSERT – дерек енгізу. Негізгі кестеге бір қатар қояды. Бірден бірнеше қатар бір кестеден немесе сұраныстан базалық кестеге ауысуы мүмкін болғанда, оператор модификациясы болады;

– UPDATE – қатарды жаңарту. Фильтрациялау шартымен сәйкес келетін бір немесе бірнеше қатардағы бір немесе бірнеше бағандардың мәндерін жаңартады.

Деректер талдауын өңдеу үшін сұраныстар тілі Data Query Language (DQL) қолданылады:

– SELECT – қатар таңдау. Реляционды алгебраның барлық операторларын алмастыратын және сұранысқа сәйкес келетін нақты қатынасты құруға мүмкіндік беретін оператор;

– SELECT – кілттік сөз. Барлық сұраныстар осы сөзден басталып, бұдан соң бос орын қалдырылады. Одан соң сұрыптау әдісі - дубликаты өшірілген (DISTINCT) немесе өшірілмеген (ALL, үнсіз келісім бойынша) болады. Содан соң кестеден сұраныспен таңдалатын үтір арқылы жазылған бағандар тізімі немесе барлық қатардытаңдау үшін * (жұлдызша) символы орналасады. Бұл жерде аталмаған кез келген кестелер команданың орындалуына сәйкес келетін нақты қатынасқа қосылмайды. Бұл, әрине, олар жойылады немесе олардың ақпараттары кестеден өшіріледі деген сөз емес, өйткені сұраныс кестелердегі ақпаратқа әсер етпейді – ол тек деректерді көрсетеді;

– FROM – әр сұраныста болуы тиіс SELECT-ке ұқсас кілттік сөз. Ол бос орын және содан соң ақпарат көзі ретінде пайдаланылатын кестелердің атымен құрылады. Егер бірден көп кесте аты көрсетілген жағдайда, аталған кестелермен декарттық туынды операциясы іске асатыны анық байқалмайды;

– WHERE – соңынан предикаты – сұрыптауға түсу үшін ол қанағаттандыруы тиіс кестедегі жазбаларға қойылатын шарты болатын кілттік сөз. Бұл бөлімде нәтиже қатарының сұрыптау шарты және негізгі кестені біріктіру шарыт қойылады;

– WHERE бөлімінің шартын көрсетуде келесі предикаттар пайдаланылуы мүмкін:

Дәстүрлі мәні бар салыстыру предикаттары { =, <>, >, <, >=, <= }.

Between A and B предикаты A және B арасындағы мәнді береді. Салыстырылатын мән диапазон шекарасын қоса берілген диапазонға түсетін болса, предикат шындық. Сонымен қатар, стандартта салыстырылатын мән диапазон шекарасын қоса берілген диапазонға түспейтін болса, шындық болатын қарама-қарсы Not Between A and B предикаты берілген.

Жиынтыққа кіру предикаты IN салыстырылатын мән берілген мәннің жиынтығына кіргенде, шындық болады. Бұдан мәннің жиынтығы қарапайым аударыммен немесе құрылған сұраныстармен берілуі мүмкін. Сонымен қатар, салыстырылатын мән берілген жиынтыққа кірмеген жағдайда шындық болатын қарама-қарсы NOT IN (жиынтық) предикаты бар.

LIKE және NOT LIKE түріндегі салыстыру предикаттары. LIKE предикаты берілген мән салыстырылатын шаблонның берілуін талап етеді, егер

салыстырылатын мән шаблонға сәйкес келсе, предикат шындық, және қарама-қарсы жағдайда жалған. NOT LIKE предикаты қарама-қарсы мағынаға ие.

IS NULL анықталмаған мәні бар салыстыру предикаты. Анықталмаған мән уақыттың берілген сәтінде анықталмаған мән ретінде реляциялық моделде интерпретацияланады. Бұл мән қосымша ақпарат пайда болғанда уақыттың кез келген сәтінде кейбір нақты мәндерге алмастырылуы мүмкін. Анықталмаған мәндердә салыстыру кезінде салыстырудың стандартты ережелері әсер етпейді: бір анықталмаған мән ешқашан басқа анықталмаған мәнге тең болып есептелмейді. Кейбір атрибут мәнінің теңдігін анықтау үшін анықталмаған ретінде арнайы стандартты предикаттар пайдаланылады: <атрибут аты> IS NULL және <атрибут аты> IS NOT NULL. Егер берілген қатарда көрсетілген атрибут анықталмаған мәнге ие болса, IS NULL предикаты «Шындық» (TRUE) мәнін қабылдайды, ал IS NOT NULL предикаты – «Жалған» (FALSE), қарама-қарсы жағдайда IS NULL предикаты «Жалған» мәнін қабылдайды, ал IS NOT NULL предикаты «Шындық» мәнін қабылдайды.

GROUP BY бөлімінде топтау аймағының тізімі беріледі.

HAVING бөлімінде шарттары әрбәр группаға қойылатын предикаттар беріледі.

ORDER BY бөлімінде нәтижені реттеу аймақтарының тізімі, яғни соңғы қатынасында сұрыптау ретін анықтайтын аймақтар қатары беріледі.

3 Дерекқорды жобалау және оны іске асыру

3.1 ДҚ құрудың бастапқы кезеңі

3.1.1 Қолдану облысы және тағайындалуы

Бұл жүйе бағдарламаланған қызмет көрсетуге арналған автоматтандырылған орта болып есептеледі. Яғни, компанияда нақты уақыт режимінде жасалып жатқан жұмыстар туралы ақпарат алу, жұмысты ұйымдастыру, оңтайлысын таңдау және қажет болған жағдайда ақпарат енгізу. Бұл жүйе төмендегідей мәліметтерді өңдеп, қолданушының сұрақтарына жауап беруі қажет:

- компания жұмысы жайлы мәліметтер;
- жоспарлық ай сайынғы есебін анықтау.

3.1.2 Дерекқордағы шешілетін мәселелер

Қолданушы қызметкерлерге ыңғайлы сонымен қатар тиімді жұмыс жасауға көмек беретін, орталықтандырылған деректер қоры бар, тауарлар туралы реттелген ақпарат алу мүмкіндігі бар басқару жүйесін құру. Excel бағдарламасы арқылы гистограммамен көрнекі түрде көрсету. Тауар құнын есептеу мүмкіндігі болуы.

3.2 Концептуалды жобалау

3.2.1 Қажетті ақпарат: пайдаланушылардың талабы

Бағдарлама қолданушының барлық негізгі талаптарын қанағаттандыру керек. Бұл жүйенің негізгі пайдаланушылары кез келген қарапайым қолданушы, яғни сатушылар мен менеджерлер. Негізгі талаптар алынған ақпараттың ұсынылатын тиімділік деңгейінде негізделеді. Жүйеге қойылатын негізгі талаптардың тізімі:

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

3.2.2 ДҚ және қосымшаны жобалау құралдарын таңдаудың негіздемесі

Барлық талаптарды бұлжытпай қанағаттандыратын электронды информационалдық жүйені жүзеге асыру үшін бірінші кезекте реляциондық дерекқорын және серверлі архитектура клиентін қолдануды анықтайтын шешіміне жеткізетін орталық деректер сақтау, өңдеу және түрлендіру әдісін таңдау қажет.

Қазіргі кезде қойылған тапсырмалар талаптарын дерек қорын басқару жүйесінің мүмкіншіліктеріне қарай және тапсырыс берушінің таңдауына байланысты таңдауға болатын бірнеше нұсқа көрсетілген:

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

Microsoft SQL Server төменөндіргіштік серверлік қуаттарға есептелген, сонымен қатар қолданушы жұмыс станциясында, яғни клиенттік қосымшада, қолданушыға жұмыс өндіргіштігінің төмендеуі білінуінсіз жұмыс істеуге қаблетті. Сонымен бірге кең таралғандық сипатына және деректерді сақтау сенімдігі мен қауіпсіздігін қамтамасыз ету құралдары бар.

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

- деректерді ұйымдастыру. SQL қолданушыға деректердің көрсетілімі құрылымын өзгертуге, сонымен қатар дерекқордың элементтері арасында қатынас құруға мүмкіндік береді;
- деректерді оқу. SQL қолданушыға немесе қосымшаға дерекқордың ішіндегі деректерді оқуға және олармен қолдануға мүмкіндік береді;
- деректерді өңдеу. SQL қолданушыға немесе қосымшаға дерекқорды өзгертуге, яғни оған жаңа деректер енгізуге, оған қоса бұрыннан бар деректерді өшіруге немесе жаңартуға мүмкіндік береді;
- қатынас құруды басқару. SQL-дің көмегімен қолданушының деректерді оқу және өзгерту және рұқсат етілмеген қатынас құрудан қорғауға мүмкіндігін шектеуге болады;

– деректерді бірге қолдану. SQL параллельді түрде жұмыс істейтін қолданушыларға бір-біріне кедергі келтірмес үшін деректерді бірге қолдануын қадағалайды;

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

SQL – дерекқорды программалау тілі. Дерекқорға қатынас құру үшін бағдарлаушылар өздерінің бағдарламаларына SQL командаларын қояды.

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

Қазіргі кезде қойылған тапсырмалар талаптарын ДҚБЖ мүмкіншілігіне қарай және тапсырыс берушінің таңдауына байланысты таңдауға болатын бірнеше ДҚБЖ көрсетілген:

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

3.2.3 ER моделін құру

ДҚ құрудың бастапқы кезеңінде алғашқы нысандар жиынтығын анықтау қажет. Бұл ақырғы пайдаланушылар мен жобалаушы көзқарасы бойынша жүйе объектілері туралы ең маңызды ақпарат болып табылады. Электрондық веб-журналдың дерекқоры 3.1-кестеде көрсетілген нысандарды қолданады.

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

3.3 Логикалық жобалау

Логикалық жобалау кезеңінде дерекқор моделі жүйені іске асыру кезінде қолданылатын белгілі бір ДҚБЖ (MS SQL Server) үшін жарамды форматқа трансформаланады.

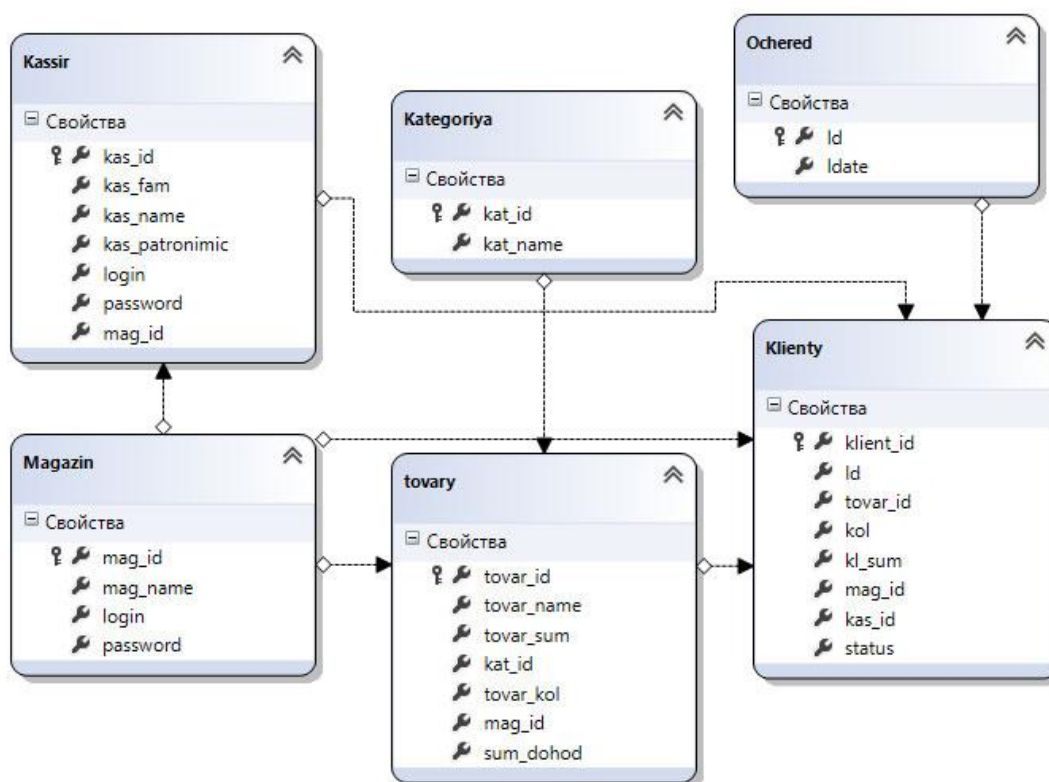
Логикалық жобалаудың мақсаты реляционды кестелік құрылымдарды құру болып табылады. Жобаланатын дерекқорда кестелердің және байланыстардың құрылымдарын MS SQL Server көмегімен жасаймыз.

Кестелер құру.

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

ДҚ диаграммасы.

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



Сурет 3.1 – ДҚ диаграммасы

3.4 Физикалық жобалау

Тапсырманың қойылуы.

Мәліметтер қорымен жұмыс істеу қосымшасын жасау әдістемесі кәдімгі программаны құру әдістемесінен ешқандай айырмашылығы жоқ. Формаға

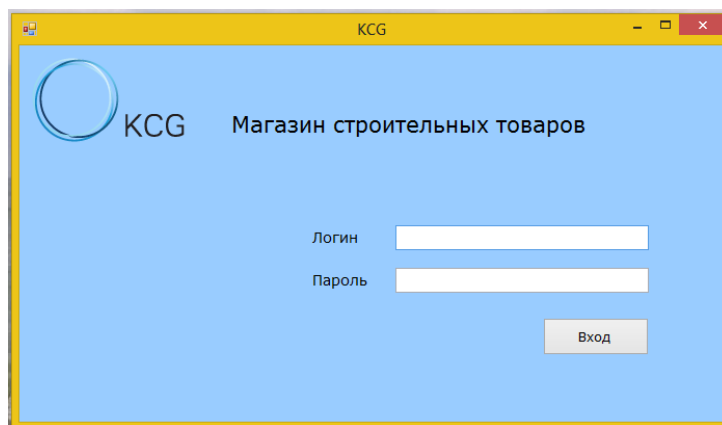
керекті компоненттер енгізіледі, компоненттердің қасиеттерінің мәндері қойылады, керекті оқиғаны өңдеу процедурасы жасалады.

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

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

3.5 Бағдарламаны сипаттау

Жоғарыда айтылғандай, жобаның интерфейсі Visual Studio 2012 объектіге-бағытталған бағдарламалау ортасында орындалды. Қосымшаның бағдарламалық коды Б қосымшасында келтірілген. Бағдарламаны қосу кезінде, алдымен бағдарламаның алғашқы беті ашылады.



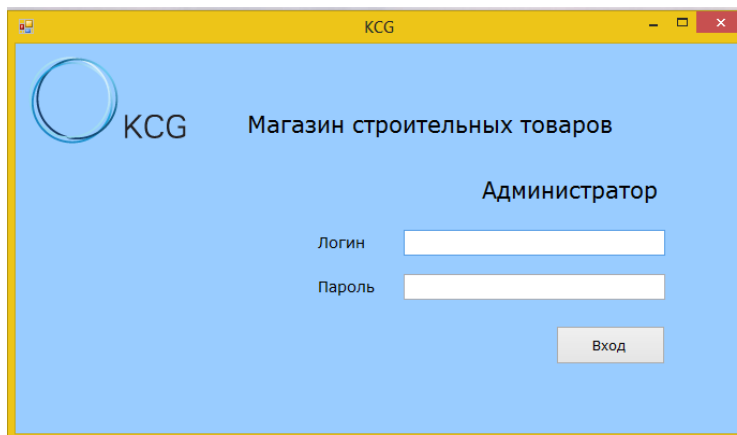
Сурет 3.2 – Бағдарламаның тіркелу терезесі

Қолданушылар:

- администратор;
- кеңесші-сатушылар;
- менеджер.

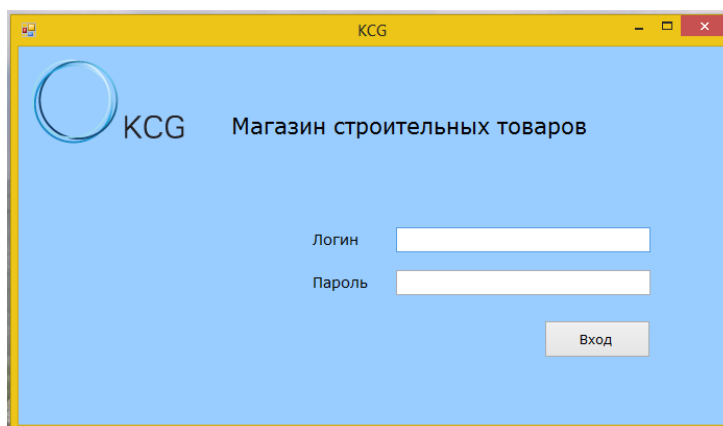
Мұнда тауралар туралы мәліметтер бар. Яғни тауар аты, тауар саны және тауар бағасы. Бұл жерде сатушы-кеңесшілер клиенттің таңдаған тауарларын енгізе алады. Тауарлар бойынша сатып алынатын заттардың жалпы бағасы

есептелініп шығарылады. Тауарлады таңдау оңай болу үшін тауар категориясы атты қосымшасы бар. Онда біртекті тауарлар біріктірілген. Таңдалған тауарды сатып алудан клиент айнитын болса, тауарды қайтадан базаға ауыстыруға болады. Тауар таңдалған соң арнайы қағаз дайындалып принтерден шығарылады.



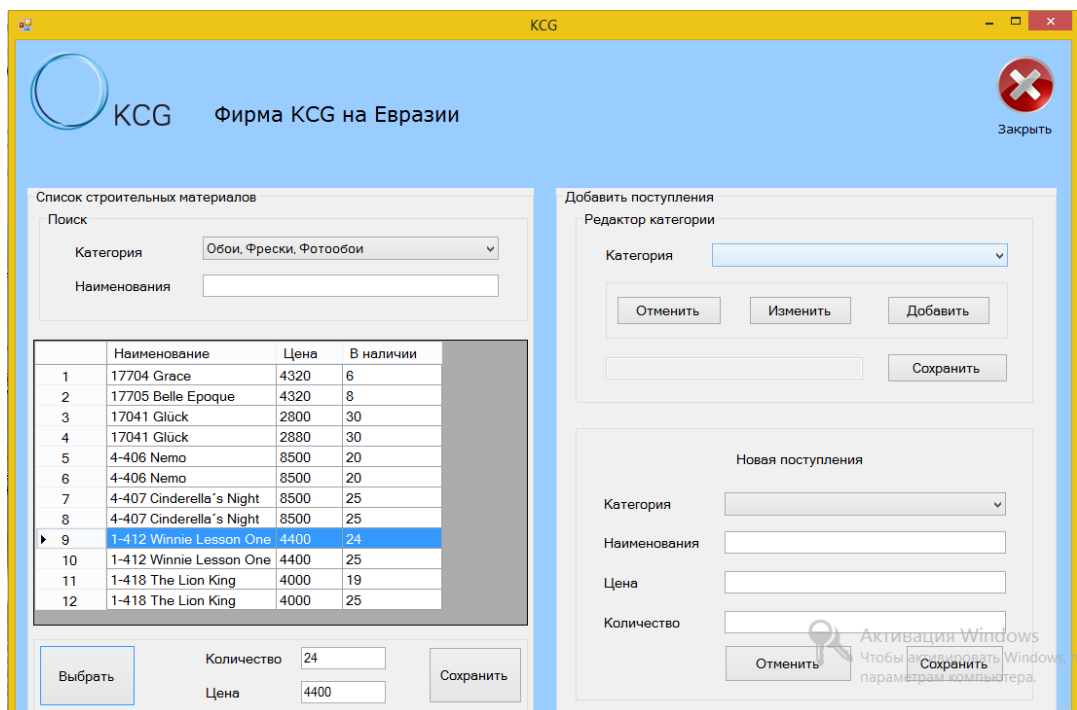
Сурет 3.3 – Жүйеге администратор арқылы кіру терезесі

Жүйеге администратор ретінде кіру арқылы жасалып жатқан жұмыстарды басқаруға болады. Магазиндегі тауар сатылуын бақылау мүмкіндігі бар. Есеп алуды бір күн үшін көруге болады немесе белгілі бір период үшін жүктеуге болады. Есеп алуды әр магазин үшін жеке іске асыруға болады немесе екі магазинді қатар қарауға болады. Жаңа жұмысшыларды қабылдау кезінде мәліметтерді енгізу мүмкіндігі бар. Жұмысшыларға кілт сөзін тағайындаушы қызметін атқарады.

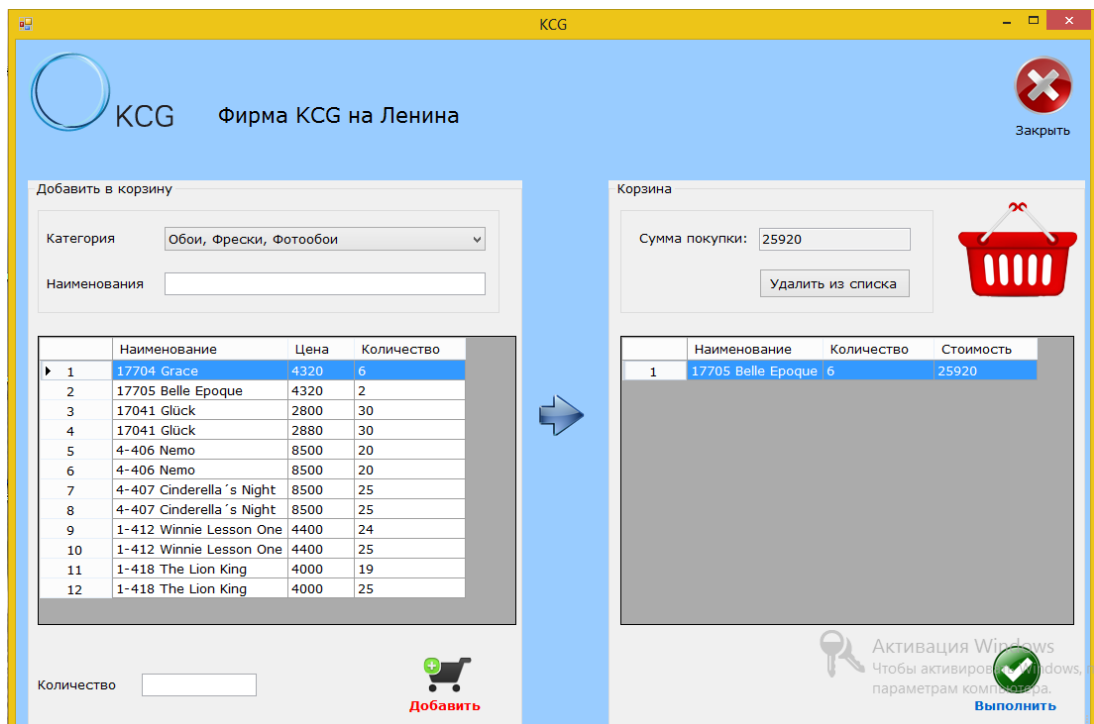


Сурет 3.4 – Жүйеге менеджер ретінде кіру

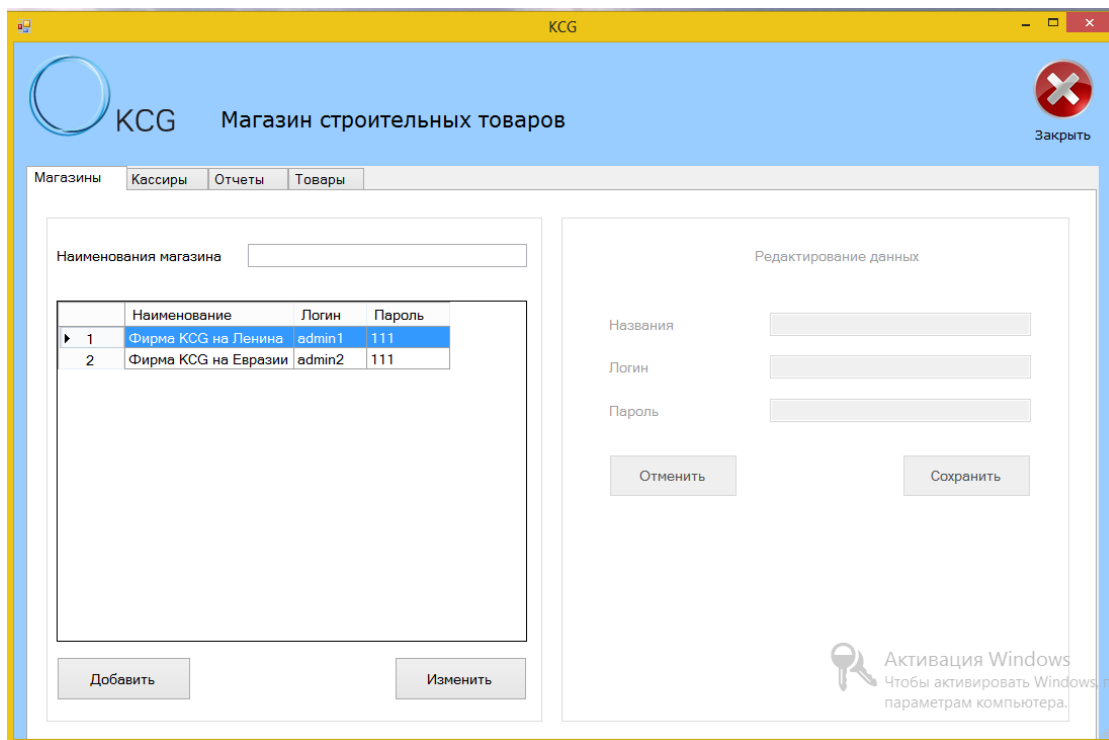
Магазин менеджері жүйеге өз атымен және индивидуалды кілт сөзімен жүйеге енеді. Менеджер тауар қабылдап алу қызметін атқарады. Жаңа категория жасау, тауар қосу, санын өзгерту функцияларын атқарады.



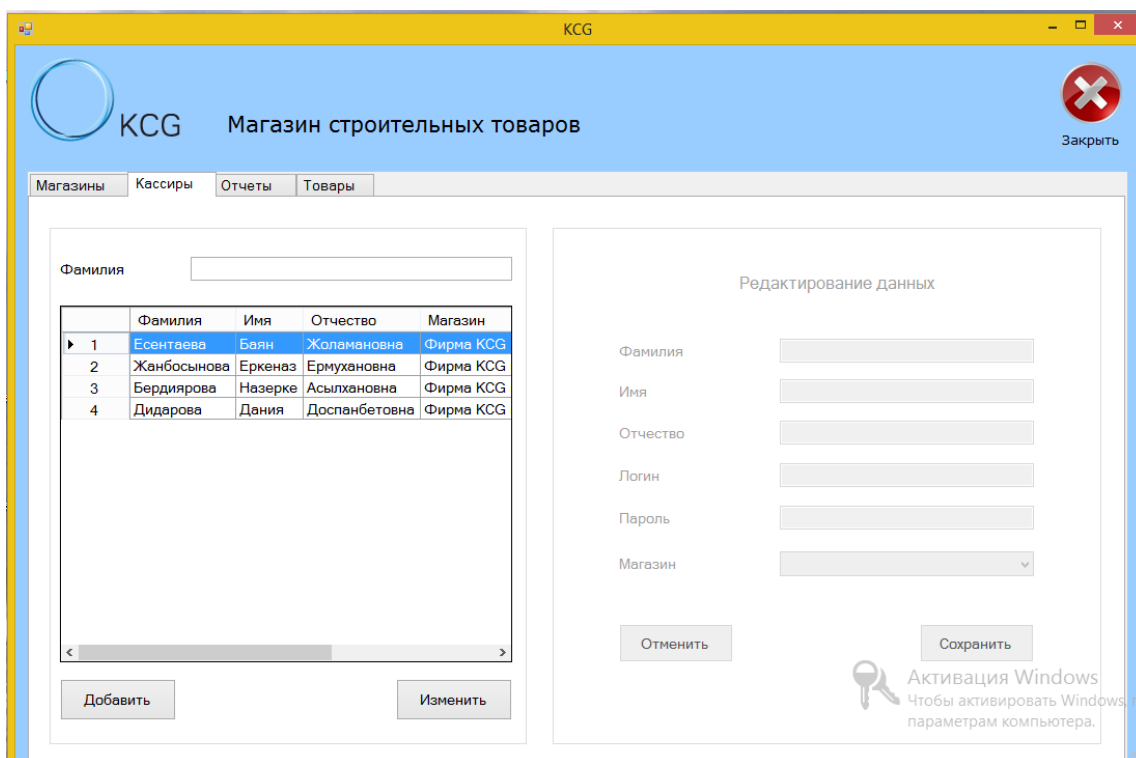
Сурет 3.3 – Тауар қабылдап алу бөлімі



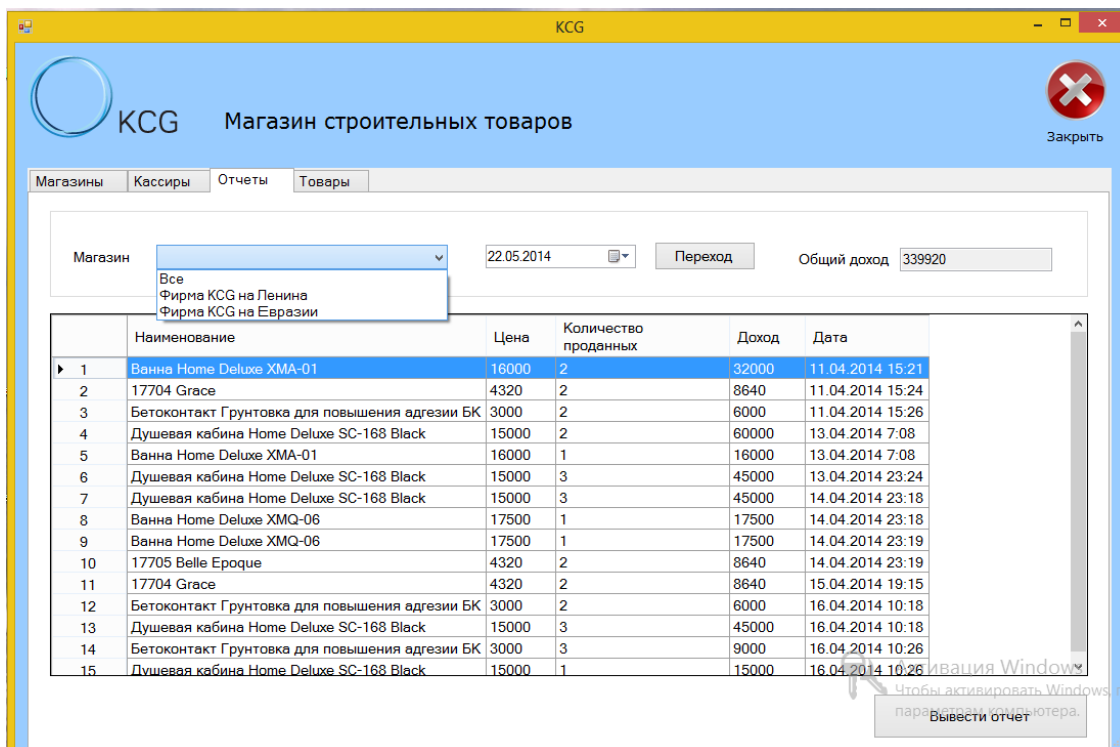
Сурет 3.4 – Таңдалған тауарды сату бөлімі



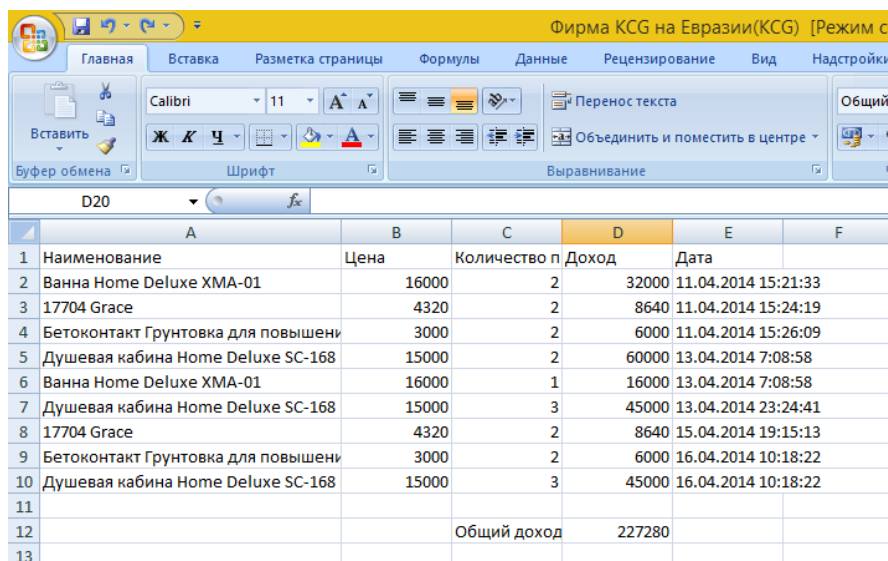
Сурет 3.5 – Компания менеджерлері туралы ақпарат



Сурет 3.6 – Магазин сатушы-кеңесшілері туралы ақпарат



Сурет 3.7 – Есеп алу бөлімі



Сурет 3.8 – Excel-ға экспорттау

КCG Магазин строительных товаров

Магазины Кассиры Отчеты Товары

Магазин Категория

	Наименование	Стоимость	Количество
1	Душевая кабина Home Deluxe SC-168 Black	15000	3
2	Душевая кабина Home Deluxe SC-168 Black	15000	1
3	Ванна Home Deluxe XMQ-06	17500	1
4	Ванна «CARIBA» 170x75 г/м 6ф(х) с/п(х)	14000	1
5	Ванна «CARIBA» 170x75 г/м 6ф(х) с/п(х)	14000	3
6	Ванна «GRENADA» 180x90 г/м 6ф(х) с/п(х)	24000	4
7	Ванна «GRENADA» 180x90 г/м 6ф(х) с/п(х)	24000	4
8	Бетоконтакт Грунтовка для повышения адгезии БК	3000	53
9	Бетоконтакт Грунтовка для повышения адгезии БК	3000	9
10	Грунтовка KF - морозостойкий концентрат Акрилатная	2600	4
11	Грунтовка KF - морозостойкий концентрат Акрилатная	2600	4
12	Добавки к Litokol Starlike	2000	3
13	Латексные добавки IDROSTUK - м	3000	5
14	Латексные добавки IDROSTUK - м	3000	3
15	Цементные затирки LITOCHROM 1-6 (2 кг)	750	22
16	Цементные затирки LITOCHROM 1-6 (2 кг)	750	20
17	Затирка для ремонта швов	600	15
18	Затирка для ремонта швов	790	14
19	Затирка для швов Fugelweiss	800	13

Активация Windows
Чтобы активировать Windows, перейдите к параметрам компьютера.

Сурет 3.9 – тауар айналымын бақылау терезесі

Қазіргі технологиялардың дамыған заманында барлық жұмысты бір орталықтан басқару адамдардың жұмысын жеңілдетті. Бұл жоба клиент-сервер технологиясына негізделіп жасалғандықтан кәсіпорын жұмысын оңтайландыруға аса ыңғайлы. Деректер қоры SQL-да жасалып, интерфейс ретінде Visual Studio пайдаланылды.

4 Тіршілік қауіпсіздігі

4.1 Өндірістік ғимараттағы еңбек шартын талдау

Бұл дипломдық жұмыста «КСГ» компаниясындағы тауар айналымын есепке алуға және бақылауға арналған бағдарлама жасалды. Бағдарлама Орал қаласында орналасқан компанияда орнатылды. Қарастырылып отырылған ғимарат аэропорт немесе темір жол магистралі сынды шу көздерінен алыс орналасқан, сондықтан сыртқы зиянды шу жұмысшыларға әсер етпейді. Бағдарламалық өнімді қолданысқа енгізген кезде қосымша компьютер қойып, ескі компьютерді ауыстыру қажет болды. Жылу бөлетін аппараттар көбейген соң кондиционерлеу жүйесін қайта есептеуге тура келді. Енгізілген өзгерістер жарықтандыру жүйесіне еш жаңалық енгізген жоқ, сондықтан бұл жүйені қайта есептеу керек емес. Алдымен, бағдарлама орнатылатын ғимараттың сипаттамасын қарастырамыз:

- орналасқан орны: Орал қаласы, Пролитарная көшесі 1;
- ғимарат типі: офис, бес қабатты ғимараттың бірінші қабатында орналасқан;
- жұмыс бөлмесінің өлшемі: ұзындығы 6 м, ені 4 м, биіктігі 3 м;
- бөлмедегі жұмыс орнының саны – 2 адам; есептеу орталығының қызметкерлері орналастырылған бөлме көлемі $19,5 \text{ м}^3$ /адамнан кіші болмауы керек;
- жұмыс күнінің ұзақтығы: 8 сағат, аптасына 5 күн;
- орындалатын жұмыс категориясы – жеңіл;
- микроклиматтың оптималды нормасы: $22-24^{\circ}\text{C}$, 40-60% ылғалдылық;
- жарықтандыруы: бөлме терезесі – өлшемі 2000x2000мм екі терезе, екі сатылы шыны;
- жасанды жарықтандырылуы – шамдар саны: 3 төбеге бекітілген шам, әр қайсысында 2 люминесценциялық лампалары бар (ПВЛМ-2x40);
- сыртқы шу факторлары: жоқ;
- өрт қауіпсіздігі: 3 құрама дачиктер (жылу түтін) қойылған;
- электр қауіпсіздігі: жедел әрекет ететін өшіргіш орнатылған;
- бөлме қабырғаларының сылануы – ашық сары түсті. ГОСТ 12.1.028-80 бойынша объекттерді ажырату 1 – 10 мм және одан жоғары болғандықтан, бөлме көздің көру жұмыс шарттары бойынша V разрядқа жатады;
- жұмысының разряды – V;
- нормаланған жарықтандыру – 400 лк.

Есептеу техникасы бөлмесіндегі өндірістік ортаның микроклиматы.

Компания қызметкерінің жұмысы тек компьютерде жасалады. Компьютермен жұмыс жасау кезінде бөлмедегі температура артады және

ауаның ылғалдылығы тура пропорционалды түрде төмендейді, ауаның иондық және сапалық құрамы нашарлайды: ауаның құрамында органикалық заттар және көміртегінің қостотығы артады. Осы себептерге байланысты микроклиматтың 4.1 – кестеде келтірілген оптималды параметрлерін сақтау керек. Сонымен қатар ауа райы жағдайына байланысты жұмыс орнын желдетіп отыру керек, ұзақтығы 10 минуттан кем болмауы керек.

К е с т е 4.1 – Жұмыс орнындағы микроклиматтың оптималды нормасы

Жыл мезгілі	Жұмыс категориясы	Ауа температурасы, С	Ауаның салыстырмалы ылғалдылығы, %	Ауаның қозғалыс жылдамдығы, м/с
Салқын	Жеңіл – 1	22–24	40–60	0,1
Салқын	Жеңіл – 1	21–23	40–60	0,1
Жылы	Жеңіл – 1	23–25	40–60	0,1
Жылы	Жеңіл – 1	22–24	40–60	0,2

Бөлмеде жыл бойы ауа температурасының, ауа ылғалдылығының және ауа жылдамдығының оптималды көрсеткіштері сақталады. Ол үшін бөлмеде сплит жүйесінде қызмет атқаратын кондиционер орнатылған. Бөлмеде сақталу керек ауа иондалу нормасы 4.2 – кестеде көрсетілген.

К е с т е 4.2 – Жұмыс бөлмесіндегі ауаның иондалу деңгейінің нормасы

Иондалу деңгейі	1см ³ ауадағы иондар саны	
	n +	n –
Минималды қажет	400	600
Оптималды деңгей	1500–3000	30000–50000
Максималды мүмкін	50000	50000

ЭВМ-нің эргономикалық визуалды өлшемдеріне келесі жолдармен жетуге болады:

- жоғары сапалы компьютер алу;
- компьютерді алдын ала стандарттарға сәйкестігін тексеру;
- анықтаған олқылықтарлы жою.

Визуалды эргономикалық өлшемдердің оптималды мәндері 4.3 кестеде келтірілген.

К е с т е 4.3 – Дербес компьютердің визуалды эргономикалық өлшемдері

Өлшем атауы	Өлшемнің шекті көрсеткіші	
	минималды шекті	максималды шекті

	көрсеткіш	көрсеткіш
--	-----------	-----------

Кесте 4.3-тің соңы

Белгі анықтығы (кд/м ²)	35	120
Ішкі жарықтандыру (лк)	100	250
Бұрыштық белгі өлшемі (угл/мин)	16	60

Жұмыс үстелінің эргономикалық өлшемдерін қамтамасыз ету.

ГОСТ 12.2.032–78 талаптарына сай «КСГ» компаниясында жұмыс тақтасының 725мм биіктігі бар жұмыс үстелдері және көтерілмелі-айналмалы механизмі бар орындықтар қолданылады. Креслоның көтерілмелі-айналмалы конструкциясы негізгі отыру биіктігін 400-500мм және алға еңкейу бұрышын 15 градуста, артқа еңкейу бұрышын 5 градусқа дейін өзгерту мүмкіндігін береді. Жұмыс кезінде қол буындарына зақым келтірмеу үшін әр бір креслоның қолды тіреуге арналған конструкциясы бар.

Техникалық эстетика нормаларының сақталуын қамтамасыз ету маңыздылығы жоғары, себебі бұл нормаға бөлме безендірілуі кіреді. Қабырға түсі – ақшыл сары, төбе түсі – ақ, еден – паркеттік тақталар, паркет түсі – ашық қоныр. Түстік безендіру СН–181–70 ұсынысы негізінде жасалған. СН–181–70 ұсынысы бойынша қабырға, төбе және еден түстері бір гаммада сонымен бірге үйлесімді болуы қажет. Түс терапия тұсынан қарасақ сары және қоңыр түстер адамның көңіл күйін көтереді, жүйке жүйесіне оң әсер етеді.

Аса қауіпті және қауіпті еңбек шартын ұйымдастыру жұмыс беруші компанияның еңбекті сонымен қатар жұмыс орнын дұрыс ұйыдастырмаумен немесе қорғаныс құралдарын пайдаланудың ережелерін және нұсқауларын қажетті мөлшерде сақтамаумен немесе бұзуымен байланысты.

Аса қауіпті және қауіпті еңбек шартын ұйымдастырудың техникалық факторлары: құрал-жабдықтардың жетілдірілмеуі, құрал-жабдықтардың жарамсыздығы, еңбекті қауіпсіз жүргізудің техникалық құралдарының болмауы (нашар оқшаулау жағдайының, қорғалған жерге тұйықтаудың болмауы, қоршау және блоктау).

Аса қауіпті және қауіпті еңбек шартын ұйымдастырудың санитарлы-гигиеналық нормаларды бұзуменде байланысты. Санитарлы-гигиеналық факторлар жарықтанудың және микроклиматтың санитарлық нормасын сақтамаумен, өндірістік ортада зиянды заттарды пайдаланумен, шудың және жоғары жиілікті сәулеленудің болуымен, статикалық электрлеудің жоғары деңгейімен, жоғары кернеуі бар электр қондырғыларын пайдаланумен байланысты.

Жұмыс орнының жарықтандыру жүйесі. Дербес компьютермен жұмыс жасайтын бөлмені дұрыс жарықтандыру өте маңызды. Дұрыс

жарықтандырмаудан көзге зақым келеді, көздің көру мүмкіндігі төмендейді. Ол көбіне жарықтық және желілік жағдайда анықталады.

Жарықтың жұтылуын азайту керек. Ол үшін төбе мен панельден жоғары қабырғалар (1,5-1,7 м), егер олар дыбыс шектейтін арнайы материалмен қапталмаса, ақ бояумен боялады (шағылысу коэффициенті 0,7-ден кем болмауы керек). Панель қабырғаларын бояу үшін ашық түсті бояуларды тағндаған жөн.

Дербес электронды есептеу машиналарында жұмыс жасау жарықтандырудың келесі түрлерінде жүзеге асуы мүмкін:

– компьютер мониторлары бөлменің периметрі бойынша жұмыс орнының екі қатарында орналасуы кезіндегі жалпы люминесцентті жарықтандыру;

– экран мен жұмыс үстелінің беткі қабаты жарық тасымалдаушы қабырғаға перпендикуляр орналасқанда жұмыс орнының тек бір немесе үш қатарында орналасуы кезіндегі қатар қолданылатын жарықтандыру (табиғи + жасанды).

Кесте 4.4 – Дербес компьютермен жұмыс жасау бөлмесін жарықтандыру параметрі

Көз көру жұмысының сипаттамасы	Разряды	Объект пен фон арасындағы контраст	Фон сипаттамасы	Жасанды жарықтандыру, лк	
				Қос жарықтандыру	Жалпы жарықтандыру
Орташа дәлдікте 0,5–1,0	IV в	үлкен	Ашық	400	200

Табиғи жарықтандыру персоналды электронды есептеу машинасы бар жұмыс орындары терезесі бар қабырғадан 0,8 – 1,0 м қашықтықта бөлменің ұзындығы бойынша бір қатарда орналасқанда, және экрандар осы қабырғаға перпендикуляр орналасқанда болады. Табиғи жарықтың негізгі ағыны сол жақтан болуы керек. Табиғи жарықтың негізгі жарықтық ағынының бағыты компьютерде жұмыс жасаған адамға оң жақтан, артқы жақтан және алдынан түсуге рұқсат етілмейді.

Жұмыс орнына шудың әсері.

Шу адам ағзасына зиянды әсерін тигізе отырып, жұмыс жағдайын нашарлатады. Адамға шу ұзақ уақыт әсер еткен кезде жағымсыз жағдайлар туындайды: көз көру, есту қабілеті нашарлайды, қан қысымы көтеріліп, көңіл бөлу нашарлай түседі. Қатты ұзақ шу жүрек-қан қысымы және жүйке жүйесі қызметінің бұзылуына себепін тигізеді.

Жұмыс орнындағы шу көзі болып техникалық құралдар табылады, олар компьютерлер, серверлер, желдеткіш қондырғылар, сонымен қатар сыртқы шу. Тәжірибеде шамасы 20-30 дБ шу зиянсыз деп есептеледі, ең жоғары шу

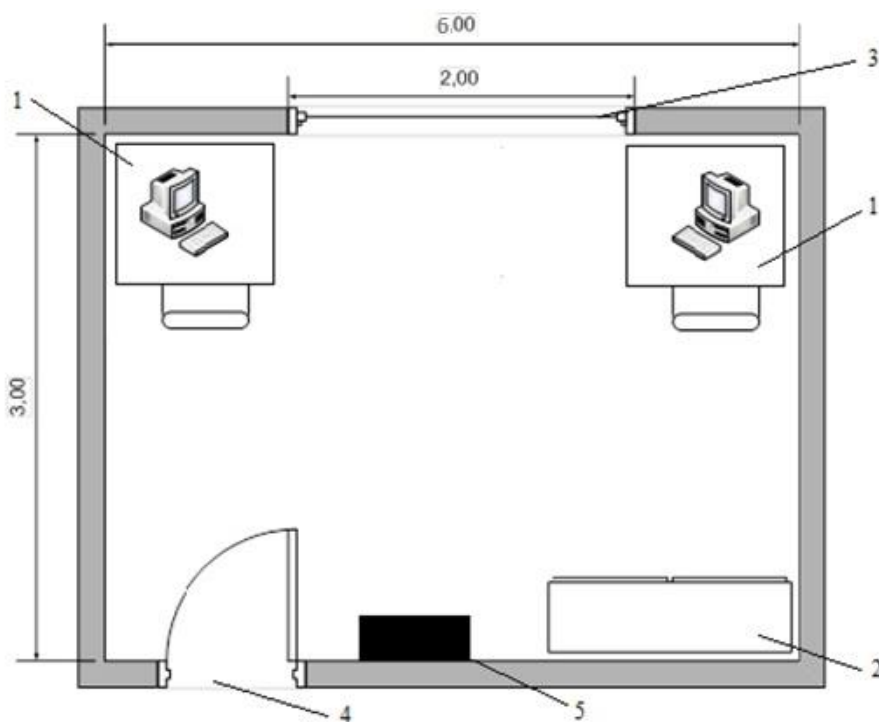
мөлшері 80 дБ-ден аспауы тиіс. Ал 130 дБ-ға жеткен кезде адам жайсыз сезінеді.

Шу деңгейін шу өлшегіштермен бақылайды. Ал қарапайым жағдайда сөйлеушіден 1,5 метр жерде тұрып тыңдағанда сөзі анық естілетін – естілмейтіндігін тексереді. Қазақстан өкіметінің территориясында қабылданған регламент бойынша аурухана мен санаторийларда 35 децибел, тұрғын үй кварталдары, класс бөлмелері және оқу аудиториялары үшін – 40, стадиондар мен вокзалдар үшін – 60 децибел. Сонымен бірге транспорт құралдары үшін сыртқы шудың шекті нормасы белгіленген – 82-85 децибел. Бөлмедегі шуды азайту үшін шу көзін жауып, қашықтатып отыруға болады. Шуды көп бөлетін технологиялық құрылғыларды айырбастау мүмкін болмағанда дыбыс шағылғыштар, дыбыс тұтқыштар қолданады. Мысалы, төбені және қабырғаны дыбыс жұтатын затпен қаптап тастаса, шу деңгейін 6-10 дБ төмендетуге болады. Шудың зиянды әсерінен қорғайтын жеке сақтық құралдары да бар.

МЕСТ 12.2.003-76 сәйкес жұмыс орнында және одан тыс жерде пайда болатын шудан қорғану үшін келесі әдістерді қолдану керек:

- шу шығатын көздерде оны азайту;
- ұжымдық және жеке қорғаныс құралдарын қолдану;
- жұмыс бөлмелерін ұтымды жоспарлау және акустикалық өндеуден өткізу.

Бағдарламалық өнімді жасау кезінде темір жол торабынан, әуежайдан алшақ орналасқан ғимарат ішіндегі жұмыс орны қарастырылған, сондықтан жұмыс процесіне әсер ететін сыртқы шу көздері жоқ.



Сурет 4.1 – Жұмыс орнының жоспары

мұндағы 1 – операторлардың жұмыс орны; 2 – кондиционер орны; 3 – терезе жақтауы; 4 – есік жақтауы; 5 – өрт сөндіргіш балоны.

Пайдаланылған құрал-жабдықтардың сипаттамасы:

- 1) Персоналды компьютер:
 - Intel® Core 2 Duo E7500 (2,93GHz, 3Mb, FSB800), 2048Mb DDR2 800MHz PC-6400, HDD Seagate Barracuda 7200.12 320Gb;
 - монитор 19" TFT Samsung SyncMaster 943N;
 - габариті: 1600x700x1050 мм (персоналды компьютер + үстел + орындық);
 - электр энергиясы: айнымалы кернеу 220-250 В, жиілігі 50 Гц, қуаты 350 Вт;
 - саны – 2 дана.
- 2) Модем:
 - D-Link DSL-2540U/BR, ADSL 2+, 4 портты коммутаторы 10/100 Мбит/с;
 - саны – 1 дана.
- 3) Бір терезелік Samsung SMG4590 кондиционері:
 - салқындату режимі – 2400 Вт;
 - жылыту режимі – 2400 Вт;
 - ағын – 404 м³/сағ;
 - шу деңгейі – 31 Дб;
 - орнату режимі – терезелік;
 - габариті - 754x272x176 мм;
 - салмағы – 39 кг;
 - ауданы 24 м² ғимаратқа есептелген.

4.2 Есептеу

4.2.1 Электр тогын есептеу

Бағдарламалаушының жұмысы электр приборларымен тікелей байланысты болғандықтан, оларды дұрыс пайдалаып және электр тоғынан зақымданудан қорғаныс шараларын сақтау керек.

Электр тоғынан сақтанудың бірнеше шаралары бар. Солардың бірі – қорғаныштық жерге қосу. Қорғаныштық жерге қосу – электрді жермен немесе оның металдық тоқ өткізбейтін бөлігінің эквивалентімен арнайы тұйықтау.

Қорғаныс құралдары шартты түрде 3 топқа бөлінуі мүмкін: изоляциялайтын, қоршайтын және алдын ала қорғайтын. Изоляциялайтын – адамды тоқ өткізетін немесе жерге қосылған бөліктерден, сонымен қатар жерден изоляциялайды.

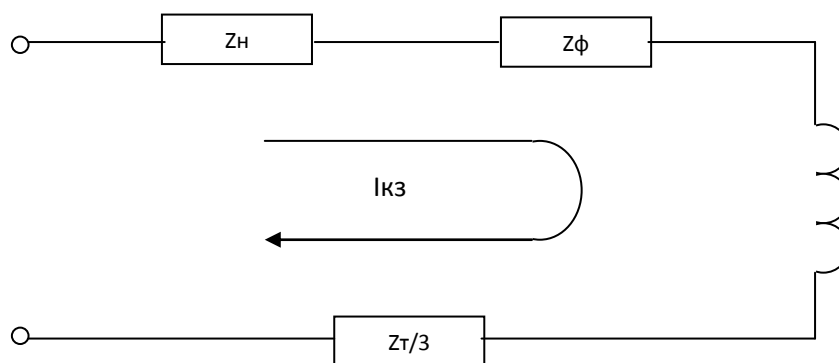
Электр тоғымен зақымданудан сақтану шарасының бірі – нөлдеу. Нөлдеу – металдық тоқ өткізбейтін бөліктің нөлдік қорғаныштық өткізгішімен

арнайы электрлік қосылуы. Бұл әдіс электрлік проводтың окшаулағышы зақымданған кезде қорғаныш қызметін атқарады.

Нөлдеуді 1кВ дейінгі желіде орындаған жөн. Желі жабық жерге қосылған нейтралі немесе жабық жерге қосылған бір фазалы токтыны шығаруы бар болуы керек. Сонымен қатар жабық жерге қосылған 380В айнымалы ток, 440 В тұрақты ток кернеуі бар үш өтімді желіде орта нүктелі болуы керек. 42В айнымалы және 110В тұрақты ток кернеуінде құрылғының металл қаптамасын және кабель мен проводтардың окшаулағышын нөлдеуді міндетті түрде орындау керек

Нөлдеу мақсаты – қысқа тұйықталу кезінде барлық электр құрылғыларын желіден тез ажырату. Авария кезінде адамның техника корпусына жақындауының қауіпсіздігін қамтамасыз ету. Жоғарыда талданған нөлденуді орындау есебі келесі факторларды талдауды қажет етеді: нөлдену кезіндегі электрондық құралдардың өшу ерекшеліктерін тексеру; адамды электр тогымен зақымдау мүмкіндігін бағалау. Жарылыс қаупі бар кез келген кластағы орындар міндетті түрде нөлдеуді орындауды қажет етеді. Барлық айнымалы және тұрақты токпен қызмет атқаратын электрлік приборлармен нөлдеуді орындау керек.

Желіні автоматты өшіргішпен қорғаған кезде таңдалынып алынған өткізгіштердің өтімділігі жедел әрекет ететін құрылғының өтімділігіндей токты қамтамасыз етуі қажет. Автоматты өшіргіш бұл жағдайда электромагниттік санағышты қолдануы қажет. Ал жедел әрекет ететін құрылғы өтімділігін зауыттық баптамаларына сай қор коэффициентіне(1,1 тең, егер зауыттық автоматты баутауларымен орнатылмаса 100А ток көлемі кезінде ауыстырылады), тарату коэффициентіне (1,25 тең) көбейту керек. Нөлдік қорғау өткізгіші фазалық өткізгіштің 50% құрауы қажет. Нөлдеу алюмини сымымен орындалады. Сым ұзындығы 50м тең. 4.1 суретте нөлдеудің қарапайым сұлбасы көрсетілген. (Есептеулер Долин П.А. – Электрқұрылғылардағы техникалық қауіпсіздік негіздері атты нұсқаулық бойынша жасалды).



Сурет 4.2 – Нөлдеу сұлбасы

Сұлбадан көретініміз, қысқа тұйықталу тогы $I_{кз}$ фазалық сым бөлігінде фазалық U_ϕ кернеуге және тізбектің толық кедергісіне тәуелді. Тізбектің толық

кедергісі бұл жерде трансформатордың орамасының толық кедергісінің, фазалық өткізгіштің кедергісінің, нөлдік қорғаушы өткізгішінің кедергісінің және сыртқы индуктивті катушка кедергісінің қосындысынан тұрады.

Қысқа тұйықталу тогының мәні келесі комплексті түрде жазылған формуламен есептеледі:

$$I_k = \frac{U_\phi}{Z_{\Gamma\Gamma} / 3 + Z_\phi + Z_n + jXX} \quad (4.1)$$

мұндағы U_ϕ – тізбектің кернеуі, вольтпен өлшенеді, В;

$Z_{\Gamma\Gamma}$ – трансформатор орамдарының толық комплексті кедергісі, оммен өлшенеді, Ом;

Z_ϕ – фазалық өткізгіштің толық комплексті кедергісі, оммен өлшенеді, Ом;

Z_n – нөлдік қорғаушы өткізгіштің толық комплексті кедергісі, оммен өлшенеді, Ом;

X_p – сыртқы индуктивті контурдың кедергісі, оммен өлшенеді, Ом;

Z_p – фаза тұйығының толық комплексті кедергісі, оммен өлшенеді, Ом.

Соңғы ақпаратты ескере отырып 4.1 формуланы келесідей түрге келтіреміз:

$$I_k = \frac{U_\phi}{(Z_{\Gamma\Gamma} Z / 3 + X_n)} \quad (4.2)$$

Фаза тұйығының толық кедергісі төмендегі формула көмегімен анықталады:

$$Z_n = \sqrt{(R_\phi + R_{\Gamma\Gamma}) + (X_\phi + X_n + X_n)^2} \quad (\text{Ом}) \quad (4.3)$$

Нөлдеу көп текті сыммен орындалып отырғанның ескеріп $X_\phi = X_n = X_n = 0$ теңдікті аламыз. 4.3 формуланы келесі түрге келтіреміз:

$$Z_n = \sqrt{(R_\phi + R_{\Gamma\Gamma})^2} \quad (\text{Ом}) \quad (4.4)$$

Қысқа тұйықталудан қорғау құралы ретінде автоматты ажыратқыштар қолданылады. Автоматты ажыратқыштар электр тізбегіне жалғанған.

Нәтижесінде қысқа тұйықталу тогының формуласы төмендегідей түрге келтіріледі:

$$Z_k = \frac{U_\phi}{Z_{\Gamma\Gamma} / 3 + \sqrt{(R_\phi + R_{\Gamma\Gamma})^2}} \quad (4.5)$$

Алюмини сымынан жасалған өткізгішінің белсенді кедергісінің мәні (4.6) формуласы бойынша анықталады:

$$R = \frac{g \times l}{S} \quad (4.6)$$

мұндағы $S = 6 \text{ мм}^2$ – өткізгіштің көлденең қимасы;

$l = 50 \text{ см}$ – өткізгіштің ұзындығы;

$g = 0,028 \text{ Ом мм}^2/\text{м}$ – алюминидің үлестік кедергісі.

Алюмини өткізгіштерінің белсенді кедергісінің R_{ϕ} және R_H мәнін анықтаймыз:

$$R_{\phi} = R_H, \quad (4.7)$$

$$R_{\phi} = 0,028 \times 50 / 6 = 0,23 \text{ Ом.}$$

Кондиционердің номиналды ток мөлшерін анықтаймыз:

$$I_H = \frac{P}{\sqrt{3} \times U_{\phi} \times \cos \alpha}, \quad (4.8)$$

$$I_H = \frac{6000}{1,73 \times 220 \times 0,8} = 19,7 \text{ А}$$

мұндағы P – кондиционер – 6000 Вт;

$\cos \alpha = 0,8$.

Қысқа тұйықталу тогын есептейміз:

$$I_K = \frac{220}{1,273/3 + \sqrt{(0,23 + 0,23)^2}} = \frac{220}{0,88} = 250 \text{ А}$$

Кедергі тогын анықтап, жұмыс тогы 20А тең автоматты таңдаймыз. Содан соң ток қатынасын (4.9) формуласы бойынша анықтаймыз:

$$I_K / I_{на} = K, \quad (4.9)$$

$$K = 250 / 20 = 12,5$$

мұндағы I – автоматтың номиналды тогы.

Шартымыз $I_K / I_{на} = K > 3$

Зақымдалған құрылғы қорабының потенциалдар айырымы (4,10) формуласы бойынша есептелінеді:

$$U_k = I_k \times Z_n, \quad (4.10)$$

$$U_k = 250 \times \sqrt{(0,23+0,23)} = 115 \text{ В}$$

Адам ағзасымен өтетін токты (4.11) формула бойынша есептейміз:

$$I_{ч} = \frac{U_k}{R_{ч}}, \quad (4.12)$$

$$I_{ч} = \frac{115}{1000} = 115$$

мұндағы $R_{ч}$ – адам денесінің кедергісі, қабылдау мүмкіндігі – 1000 Ом.

4.2.2 Вентиляцияны есептеу

Ауаны жасанды кондиционерлеудің талаптары өнеркәсіптің әр түрлі салаларындағы технологиялық процестерді жүргізуде, сондай-ақ компьютерлік жабдықтың, басқа құралдар мен аспаптардың және тағы басқа жұмыс қабілеттілігін қамтамасыз ету үшін ауа ортасының белгілі бір параметрлерін өндірістік нормаға сай сақтауға негізделеді. Бастапқы өзінің пайдалану аясының тағайыны бойынша кондиционер жүйесі қолайлы және технологиялық болып бөлінеді.

Аумағы 15-тен 140 м² дейінгі тұрғын және қоғамдық бөлмелерде сплит-жүйе кондиционерлерін пайдалану кең таралған. Олар сыртқы блоктан (компрессорлы-конденсаторлы) және ішкі блоктан (буландырғыш) тұрады. Сыртқы блок ғимарат қабырғасына, шатырға немесе шатыр астына, қосалқы бөлмеге, балконға, яғни ыстық конденсатор төменірек температурасы бар атмосфера ауасымен салқындатылатын жерге орнатылады.

Кондиционерді таңдау үшін алдымен артық жылудың қосындысын, сонымен қатар оған күннің радиациясынан бөлінетін жылу кіреді, өндірістік жарықтануды, жұмыс істейтін адамдар санын, электр техникасы және басқаларын есептеу қажет. Салқын өндіргіштік бойынша қосындысы сондай немесе үлкен мәнді, сонымен қатар қажетті ауа алмасу қамтамасыз ететін кондиционер моделі таңданылады.

Жылы не суық ауарайы кезінде, жұмыс жасап отырған адамға кедергі келтірмейтіндей, жылыту жүйесі және кондиционерлеу жүйесі орнатылуы тиісті. Жұмыс орнында микроклиматты белгілі деңгейге ауысуымен бірге тұрақты ұстауды ұсынады. Ауа температурасы бөлмеде адам денесіне қолайлы температурадан 5 градустан жоғары температураға ажырауын болдырмау керек. Вентиляция жүйесінің характеристикасын сипаттайтын басты параметрлер, еселік ауа алмасуы, ол ауаның бөлмеде қанша рет алмасқаны.

Ауа алмасуының еселегін 2000x2000мм екі терезесі бар бөлмені есептейік. Бөлмеде екі адам жұмыс істейді. 4.4-кестедегі көрсетілген электржабдығықтарды қолданамыз. (Есептеулер Баклашов Н.И. байланыс кәсіп орнындағы еңбекті қорғау атты нұсқаулығы көмегімен жүргізілді)

К е с т е 4.4 – Қолданылатын электржабдықытар

Атауы	Саны (дана)	Қолданылатын қуат, Вт
Дербес компьютер	2	600
Тоқтаусыз қоректендірі көзі	2	700

Ауа алмасуының еселегін мына формула бойынша есептейміз:

$$K = \pm \frac{V_{\text{вент}}}{V_{\text{бл}}} \quad (4.13)$$

мұнда $V_{\text{вент}}$ – бөлмеге берілетін ауа көлемі, м³;

$V_{\text{бл}}$ – бөлме көлемі, м³.

Ауысуға керекті ауа көлемін $V_{\text{вент}}$ жылулық тепе-теңдік балансы арқылы анықтаймыз.

$$V_{\text{вент}} = \frac{3600 \cdot Q_{\text{аж}}}{C \cdot (t_{\text{кт}} - t_{\text{кл}}) \cdot Y} \quad (4.14)$$

мұнда $Q_{\text{аж}}$ – артық жылу, Вт;

$C = 1000$ – ауаның сыбағалы жылуөткізгіштілігі, Дж/кг·К;

$Y = 1.2$ – ауаның тығыздығы, кг/м³.

Тыс кететін ауаның температурасы мына формула бойынша есептелінеді:

$$t_{\text{кт}} = t_{\text{км}} + (H - 2) \cdot t \quad (4.15)$$

мұнда t – бөлменің 1 метр биікке температуның өзгеруі, 1 ден 5 градусқа дейін мәндерін алады;

$t_{\text{км}}$ – жұмыс орындағы температурасы, 23 градусқа тең;

H – бөлменің биіктігі, 3 метрге тең;

$t_{\text{кл}}$ – келетін ауаның температурасы, 18 градусқа тең.

4.15-формула бойынша кететін ауаның температурасы мынағын тең:

$$T_{\text{кт}} = 23 + (3 - 2) \cdot 2 = 25.$$

Артық жылуды мына формула бойынша есептейміз:

$$Q_{\text{аж}} = Q_{\text{аж1}} + Q_{\text{аж2}} + Q_{\text{аж3}} \quad (4.16)$$

мұнда $Q_{аж1}$ – электр жабдықтардан және жарықтандырудан артық жылу, Вт;
 $Q_{аж2}$ – күннің сәулесінен келген жылу, Вт;
 $Q_{аж3}$ – адамның жылу беруі, Вт.

Электр жабдықтардан және жарықтандырудан артық жылуың мына формула бойынша есептейміз:

$$Q_{аж1} = E \cdot P \quad (4.17)$$

мұнда E – жылу бұру энергиясының шығын коэффициенті;

P – электр жабдықтар қуаты, Вт.

Электр энергияның шығын коэффициенті келесі мәндерді қабылдайды:

$E=0.55$ жарықтандыру үшін;

$E=0.4$ дербес компьютер үшін.

Келесі формула бойынша жарықтандырудың және электр жабдықтардың жалпы қуатын есептейміз:

$$P = P_{дана} \cdot K \quad (4.18)$$

мұнда $P_{дана}$ – бір дана жабдықтың қуат өлшемі, Вт;

K – жабдықтардың сан данасы.

Сөйтіп, 4.18- формула бойынша:

$$P_{жарықтандыру} = 200 \cdot 8 = 1600 \text{ Вт},$$

$$P_{компьютеры және т.б.} = 600 \cdot 8 + 700 \cdot 8 = 10400 \text{ Вт}.$$

4.17- формула бойынша электр жабдықтардан және жарықтандырудан артық жылуын есептейміз:

$$Q_{аж1} = 0.55 \cdot 1600 + 0,4 \cdot 10400 = 5040 \text{ Вт}.$$

Жасанды жарықтандырудан артық жылуын келесі формула бойынша есептейміз:

$$Q_{аж3} = m \cdot S \cdot k \cdot Q_T \quad (4.19)$$

мұнда m – терезе саны;

S – бір терезенің ауданы, м²;

k - әйнектеуді ескеретін коэффициент. Екі қабатты әйнектеудің коэффициенті – $k = 0.8$;

Q_T – терезеден жылу келуі, 127 Вт/м тең.

4.19-формула бойынша жасанды жарықтандырудан артық жылуын есептейміз:

$$Q_{аж2} = 3 \cdot 2 \cdot 0.8 \cdot 127 = 610 \text{ Вт.}$$

Адамның жылу беруін келесі формула бойынша есептейміз:

$$Q_{аж3} = n \cdot q \quad (4.20)$$

мұнда q – бір адамның жылуы бөлгіштігі, 80 Вт/ад тең;

n – адам саны, 8 тең.

4.20-формула бойынша адамның жылу беруін есептейміз:

$$Q_{аж3} = 8 \cdot 80 = 640 \text{ Вт.}$$

4.14-формула бойынша артық жылуды есептейміз:

$$Q_{аж} = 5040 + 610 + 640 = 6290 \text{ Вт.}$$

Сөйтіп, 4.16-формула бойынша ауысуға керекті ауа көлемін есептейміз:

$$V_{\text{вент}} = \frac{3600 \cdot 6290}{1000 \cdot (25 - 18) \cdot 1,2} = 2695,71$$

Демек, қолайлы шарттарды жасау үшін ауаның кондиционерлеуін ұйымдастыруды жүзеге асыру керек, бөлменің сыртында ауаның өзгеруіне және бөлменің ішінде өзгеруіне қармастан белгілі бір нұсқаулық шарты бойынша (берілген температура, ылғалдылық, ауаның ауысуы) тұрақты түрде ауаны сақтандыруды айтады.

4.3 «Тіршілік қауіпсіздігі» бөлімі бойынша қорытынды

Бұл бөлімде берілген офистік ғимараттағы еңбек шартының талдауы, оның ішінде электро техниканы нөлдеу есебі және вентиляцияның есептеулері келтірілген. Еңбек шартының деңгейі жақсы ұйымдастырылған деп танылды және есептеулерден алынған мәліметтер өмір тіршілігі қауіпсіздігі стандарттарының талаптарын қанағаттандырады.

Есептеулер табиғи жарықтандыру үшін ауданы 2000x2000 болатын бір ғана терезе жеткілікті болатынын көрсетті. Жұмыс орнын жарықтандыру үшін әрбір шамда әрқайсысының сәулеленуінің жарық ағыны стандартқа сай болатын 3 шам арқылы жалпы жарықтандыру жеткілікті, сондықтан бұл ғимаратта жарық емес тәулік уақытында да жұмыс жасауға болады.

Маңызды мәселелердің бірі, кондиционерді таңдау үшін алдымен артық жылудың қосындысын, сонымен қатар оған күннің радиациясынан бөлінетін

жылу, өндірістік жарықтандыруды, жұмыс істейтін адамдар санын, оргтехникаларды есептеу қажет. Есептеу барысында артық жылу: 6290Вт тең болса. Осы мәлеметтерге сүйене отырып, оператор бөлмесінің персоналы артық жылулық сәулеленуге шалдығатындығын көреміз. Бөлмеге қажетті ауа мөлшері $L = 2695 \text{ м}^3/\text{мин}$. Бөлмеде қажетті ауа ауысу мөлшерін қамтамасыз ету үшін бір кондиционер қойуымыз жеткілікті, ол Samsung сериясы HA85 максимальды ауаның шығыны $195 \text{ м}^3/\text{сағ}$, модель R22. Талаптарды толық қанағаттандыратын кондиционер таңдалып орнатылды.

Нөлдеу есебінің қорытындысы бойынша жаңадан орнатылған кондиционер жедел әрекет ететін механизмге қосылды. Жерге қосу адам тіршілік қауіпсіздігін қамтамасыз ету үшін маңызды, сондықтан жерге қосуда орындалды.

5 Технико – экономикалық негізделуі

5.1 Жұмысты сипаттау және қажеттілігін негіздеу

Орындалған дипломдық жұмысының тақырыбы – ЖСШ «КСГ» компаниясында тауар айналымын есепке алатын және бақылайтын бағдарлама жазу.

Айтылған бағдарламалық өнімді әзірлеудің негізгі мақсаты келесі мүмкіндіктердің болуы: деректерді енгізу, сақтау, бүтіндігін бұзбай өңдеу және өңделген ақпаратқа оңай қол жеткізу, ақпаратқа сырты ортадан қол жеткізе алмау. Ақпаратты уақытылы ала алмау қазіргі заманда негізгі қиыншылықтардың бірі болып табылады. ХХІ ғасырда технологиялар адам өмірінде маңызды орын алады. Осы технологиялардың көмегімен компания қызметкерлерінің күнделікті жұмысы жеңілдейді. Қызметкерлерге келесі мүмкіндіктер беріледі: тауралар туралы ақпараттарды өңдеу, слад жұмысы туралы ақпаратты алу, уақыт өтуіне байланысты есеп алу.

Берілген бөлімде осы жобаның жүзеге асырылуының экономикалық құрамының қарастырылуы келтіріледі. Ол уақыттық, еңбектік және қаржылай шығындарды көрсетеді. Бұл бағдарлама :

- 1) жаңа қызметкерлер туралы ақпаратты қағаз жүзінде емес, бағдарламаға оңай енгізіледі;
- 2) пайдаланушыда қызметкерлер деректер қоры архив сақталады;
- 3) тауар туралы барлық ақпарат сақталады.

5.2 Нарықты саралау және орнын анықтау

Бағдарламалық қамтаманың басты мақсаты мекемеде орнатылған бағдарлама көмегімен нақты уақыт режимінде тауар айналымын бақылап отыру.

Біздің өнімнің экономикалық көрсеткіштерін нарықта үш категорияға бөлуге болады:

- 1) өнім (Product) – бұл шифрланған деректер қорын қолданатын бағдарламалық қамтама;
- 2) құн (Price)– бұл бағдарламалық қамтаманы пайдаланатын мекемеге қойылған ақша сомасы;
- 3) дистрибуция - бағдарламалық қамтама жеткізілетін ұсыныс берген мекеме;
- 4) қозғалу, даму (Promotion) — мекеменің алдымен тәжірибе жүзінде бірнеше елді-мекенге бағдарламалық қамтаманы қоюы, кейіннен өзге мекендерге ақпарат таратуы мен сатып алуға үгіттейтін мүмкіндігі зор іс-шаралар.

5.3 Бағдарламаның SWOT-анализі

Бағдарламаны дайындамас бұрын жобаның SWOT-анализі өткізілді. SWOT-анализ төмендегі кестеде көрсетілген.

К е с т е 5.1 – Бағдарламаның SWOT-анализі кестесі

Артықшылықтары	Кемшіліктері
Бағдарламаның осы компанияға арналуында. Есеп алу мүмкіндігі бар болуында. Нақты уақыт аралығында жұмысты бақылау мүмкіндігі.	Қымбат баға. Бір ғана мекемеге арналуы
Мүмкіншілігі	Қауіпі
Мониторинг терезесі арқылы тауар айналымын бақылау мүмкіндігі	Хаккерлік шабуылдар

5.4 Маркетинг-микс элементі

Бағдарламаны енгізбес бұрын маркетингтік тексеру өткізілді. Маркетинг бойынша қорытынды төмендегі кестеде көрсетілген.

К е с т е 5.2 – Бағдарламаның маркетинг-микс кестесі

Маркетинг-микс элементтері	Қажетті істер мен талдаулар
Product	
Тауар айналымын бақылау бағдарламасы	Жоспар бойынша бірнеше компоненттер пайдаланылады: 1 Техникалық өнім; 2 Бағдарлама интерфейсі C# тілдерінде орындалмақ
Price	
Базалық бағасы	Техникалық құралдың түріне қарай бағдарламалық қамтаманың бағасы анықталады.
Promotion	
	Жарнама жоқ, сатылам тек тапсырыс бойынша
Place	
KCG компаниясы	Тауар саны көп мекемелер

5.5 Бағдарламалық қамтаманың өңделуінің еңбек өнімділігінің есептелуі

Еңбекке кеткен шығынның базалық көрсеткіштері келесі формула бойынша есептеледі:

$$Q=q \times c \quad (5.1)$$

мұндағы q - бағдарламалық өнімдегі (бастапқы команда) операторлар саны, осы жұмысымда операторлар саны 2500-ге тең, c -бағдарлама күрделілігінің коэффициенті, ол бұл жұмыста менде 1.26 ге тең. Осы әдіспен базалық көрсеткішті табамыз:

$$Q=2500 \times 1,26=3150$$

Келесі есептеу керек мөлшер бағдарламалық өнімді әзірлеуге кететін уақыт. Уақыт адам-сағатпен есептеледі, ал T_D нақты істелген уақытпен алынады, ал қалған кезеңнің уақыты Q командасының шартты санына байланысты есептік жолмен анықталады.

Бағдарламалық өнімін дайындауға кеткен әр кезеңнің уақытын анықтау төменде көрсетілген:

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

$$T_{ПО} = 24 \text{ адам / сағ.}$$

2) T_O (мақсатсипаттамасының уақыты) келесі формуламен анықталады:

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

$$T_O=3150 \times 1,5/50 \times 0,8=118,125 \text{ адам/сағ.}$$

мұндағы B – мақсат есебі өзгерісінің коэффициенті;

B коэффициенті мақсат күрделілігіне және өзгеріс санына тәуелді – 1,2-ден 1,5-ке дейін ;

K – бағдарлама жасаушы біліктілігін ескеретін коэффициент, менің берілген салада жұмыс істеу өтілім 2 жылға дейінгі уақытты құрағандықтан, K коэффициенті мәні – 0,8-ге тең болады.

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

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

$$T_A = 3150 / (50 \times 0,8) = 78,75 \text{ адам/сағ.}$$

4) T_{BC} (блок – сұлба құруға кеткен уақыт) T_A сияқты 3-формуламен есептеледі.

5) T_H (бағдарламаның тілінде жазуға кеткен уақыт) келесі формуламен анықталады:

$$T_H = Q \times 1,5 / (50 \times K), \quad (5.4)$$

$$T_H = 3150 \times 1,5 / (50 \times 0,8) = 118,125 \text{ адам/сағ.}$$

6) T_{II} (бағдарлама теру уақыты) келесі формуламен анықталады:

$$T_{II} = Q / 50, \quad (5.5)$$

$$T_{II} = 3150 / 50 = 63 \text{ адам/сағ.}$$

7) T_{OT} (бағдарламаны реттеу және тестілеу уақыты) келесі формуламен анықталады:

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

$$T_{OT} = 3150 \times 4,2 / 50 \times 0,8 = 330,75 \text{ адам/сағ.}$$

8) T_D (құжаттарды рәсімдеу уақыты), нақтылы деректер бойынша алынады және құрылады (3-тен 5 күнге дейін, күніне 8 сағат):

$$T_D = 24 \text{ адам / сағ.}$$

Еңбек шығындарының сомасы еңбек шығынының құрама сомасы ретінде 7 формуламен есептеледі:

$$T = T_{II} + T_{TO} + T_A + T_{BC} + T_H + T_{II} + T_{OT} + T_D, \quad (5.7)$$

$$T = 24 + 118,125 + 78,75 + 78,75 + 118,125 + 63 + 330,75 + 24 = 835,5 \text{ адам/сағ.}$$

5.5.2 Еңбекақыға жұмсалатын шығындар

Еңбек құны екі құрамнан құралады: негізгі еңбек құны және қосымша еңбек құны.

Негізгі еңбек құны:

$$Z_{осн} = t_{\Sigma} \times TC / (t_{cp} \times 8), \text{ теңге} \quad (5.8)$$

мұндағы t_{Σ} – (7) формуласы бойынша есептелетін жалпы еңбек шығыны;

t_{cp} – айдағы орташа күн саны, 21 күнге тең, жұмыс күніндегі 8 сағатқа көбейтіледі.

Тарифтік мөлшерлеме минималды өлшемдегі еңбек ақы 01.01.2014ж бастап ҚР-да 19966 теңгені құрады.

$TС$ – тарифтік қойылым = 19966 -ке тең болды.

t_{Σ} – (7) формуласы бойынша 473–ге тең.

Осылайша негізгі еңбек ақы:

$$З_{осн} = 835,5 \times 19966 / (21 \times 8) = 99265,49 \text{ теңге.}$$

Қосымша еңбек ақы негізгі еңбек ақының 20%-ын құрайды. Мына формула бойынша анықталады:

$$З_{доп} = 0,2 \times З_{осн}, \quad (5.9)$$

$$З_{доп} = 0,2 \times 99265,49 = 19853,1 \text{ теңге.}$$

Жалпы еңбек ақысы (еңбекті төлеу фонды) негізгі және қосымша еңбек ақының қосындысы ретінде анықталады (10):

$$ФОТ = З_{доп} + З_{осн}, \quad (5.10)$$

$$ФОТ = 99265,49 + 19853,1 = 119119,5 \text{ теңге.}$$

5.5.3 Қосымша шығындар статьясы

Бағдарламалық қамтамасыз етудегі қосымша шығындар: материалдар мен құрамаға кеткен шығындар (құрылғының өз құны, яғни компьютерлер есепке алынбайды), социалды сақтандыруға аударым, жүкқұжатқа кеткен шығындар, амортизациялық аударым, техникалық қондырғылар қызметіне кеткен шығын, компьютермен жұмыс істеу кезінде кеткен электроэнергиялық шығындар).

Қондырғының құны бағдарламалық қамтама өңделуінің өзіндік құнына кірмегенімен, қосымша шығындар статьясындағы есептеулерде қолданылады. ЭЕМ - на бағдарлама жазу кезінде құрылғы ретінде дербес компьютер қолданылады. Құны $C_{обор} = 140000$ теңге (дербес компьютердің қазіргі уақыттағы нарықтағы теңгемен берілген құны алынды)

Материалдар мен көмекші бөлшектер шығыны, бағдарламалық өнімді жазу барысында қолданылды ($C_{МжК}$), сонымен қатар техникалық қызмет көрсету шығыны ($C_{ТО}$), жабдықтың құнынан 1,5% және 2,5% құрайды және мына формулалар мен есептеледі:

$$C_{МжК} = 0,015 \times C_{обор}, \quad (5.11)$$

$$C_{МжК} = 0,015 \times 140000 = 2100 \text{ теңге,}$$

$$C_{ТО} = 0,025 \times C_{обор}, \quad (5.12)$$

$$C_{TO} = 0,025 \times 140000 = 3500 \text{ теңге.}$$

Амортизациялық аударым өнімнің физикалық және моральдық жарамсыздығына байланысты сол өнімді толық қалпына келтіру үшін жиналатын ақша қоры болып есептеледі. Амортизациялық аударым амортизация нормасы бойынша орнатылады, (13)-формуласы бойынша есептеледі:

$$A = \frac{C_{\text{обор}} \times H_A \times N}{100 \times 12 \times t} \quad (5.13)$$

мұндағы $C_{\text{обор}}$ – компьютер құны;

H_A – амортизация нормасы;

N – жұмыс орындалуына кеткен күннің саны;

t – дербес компьютерді қолдануға кеткен жалпы уақыт.

$$N = T/24 = 835,5/24 = 34,8 \text{ күн}$$

мұнда T еңбек шығынына тең, оның мәні – 473 адам/сағ тең болған.

Дербес компьютерде жалпы жұмыс істеу уақыты мына формуламен есептеледі:

$$t = T_A + T_{BC} + T_H + T_{II} + T_{OT} = 78,75 + 78,75 + 118,125 + 63 + 330,75 = 669,125 \text{ адам/сағ}$$

Амортизация нормасы (13)-формуласы бойынша анықталады:

$$H_A = \frac{C_{\text{обор}} - C_{\text{ликв}}}{T_{\text{норм}} \cdot C_{\text{обор}}} \cdot 100\%, \quad (5.14)$$

$C_{\text{ликв}}$ – таратылым құны, жабдықтың құнынан 5% құрайды.

$$C_{\text{ликв}} = 0,05 \times C_{\text{обор}}, \quad (5.15)$$

$$C_{\text{ликв}} = 0,05 \times 140000 = 7000 \text{ теңге.}$$

$T_{\text{норм}}$ – нормативті қызмет ету өтілі (дербескомпьютер үшін ҚР да $T_{\text{норм}} = 4$ жыл). Осыдан:

$$H_A = (140000 - 7000) / 4 \times 140000 \times 100\% = 23,75\%,$$

$$A = 140000 \times 23,75 \times 669,125 / 100 \times 34,812 \times 12 = 53273,65 \text{ теңге.}$$

Бағдарламалық қамтаманы жазу кезіндегі электроэнергияның шығынын қоса есептеу керек. Электроэнергияның құны мына формула бойынша есептеледі (15):

$$C_{ЭЭ} = M \times k_3 \times F_{эф} \times C_{кВт-ч} \quad (5.16)$$

мұндағы M – ЭВМ қуаты (450 Вт);

k_3 – жүктеу коэффициенті (0,8);

$C_{кВт.ч}$ – 1 кВт-сағ электроэнергияның құны, оның мәні ҚР – да 14,36 теңгеге тең;

$F_{эф}$ – жұмыс уақытының эффективті фонды, мына формула бойынша есептеледі (16):

$$F_{эф} = D_{ном} \times d \times (1 - f/100), \quad (5.17)$$

$$F_{эф} = 258 \times 8 \times (1 - 2,2/100) = 2019 \text{ сағ.}$$

мұндағы $D_{ном} = 258$ – бір жылдағы жұмыс күнінің номиналды саны;

$d = 8$ – жұмыс күнінің ұзақтығы [сағ];

$f = 2.2\%$ – ЭВМ – ды жөндеуге кеткен жоспарланған уақыт.

Берілген коэффициенттер мен параметрлер бойынша электроэнергияның құны:

$$C_{ЭЭ} = 0,45 \times 0,8 \times 2019 \times 14,36 = 10437 \text{ теңге.}$$

Дегенмен, алынған амортизациялық аударым мен электроэнергияға шығын – жылдық шығынның мәні, оны жалпы жылдық эксплуатациондық шығынмен анықталатын уақыт коэффициентімен дұрыстау керек (5.18):

$$\mathcal{E}_3 = t_{\Sigma'} \times C_{\mathcal{E}} / F_{эф}, \text{ теңге} \quad (5.18)$$

$C_{\mathcal{E}} = C_{ЭЭ} + C_{ТО} + A_{жыл}$ – ЭЕМ-ның жалпы жылдық эксплуатационды шығын құны:

$$C_{\mathcal{E}} = 10437 + 3500 + 53273,65 = 67210,65 \text{ теңге.}$$

$F_{эф}$ – жұмыс уақытының эффективті фонды, (17) формуласы бойынша анықталды, $t_{\Sigma'}$ – ЭВМ-нің (7) формуласы бойынша анықталған есепті шешу үшін жалпы қолданылған уақыты, бірақ бұл жерде тек компьютердегі жұмыс уақытын ескере отырып анықталған:

$$t_{\Sigma'} = t_{пр} + t_{отл} + t_d = 78,75 + 330,75 + 24 = 433,5 \text{ сағ.}$$

Сәйкесінше, ЭЕМ-ның жалпы жылдық эксплуатационды шығын құны:

$$\mathcal{E}_3 = 433,5 \times 67210,65 / 2019 = 14430,81 \text{ теңге}$$

Әлеуметтік салыққа кететін аударымдар бүкіл жалақының 11% құрайды [11%, ҚР ның Салық Кодексі], () формуласы бойынша анықталады:

$$CC=(\Phi OT-0,1\times\Phi OT)\times 0,11, \quad (5.19)$$

$$CC=(119119-0,1\times 119119)\times 0,11=11792,7\text{теңге}$$

Жүктеме шығындар қызмет көрсету мен басқаруға қатысты, құрылғының эксплуатациясы мен басқа да өндіріс процесін қамтамасыз ететін қосымша шығындар болып табылады, еңбек жалақысы фондының 50% құрайды, (5.20) формуласы бойынша анықталады:

$$C_{\text{накл}} = 0,5\times\Phi OT, \quad (5.20)$$

$$C_{\text{накл}}=0,5\times 119119= 5896,36\text{теңге.}$$

5.5.4 Өзіндік құн нәтижесінің кестесі мен диаграммасы

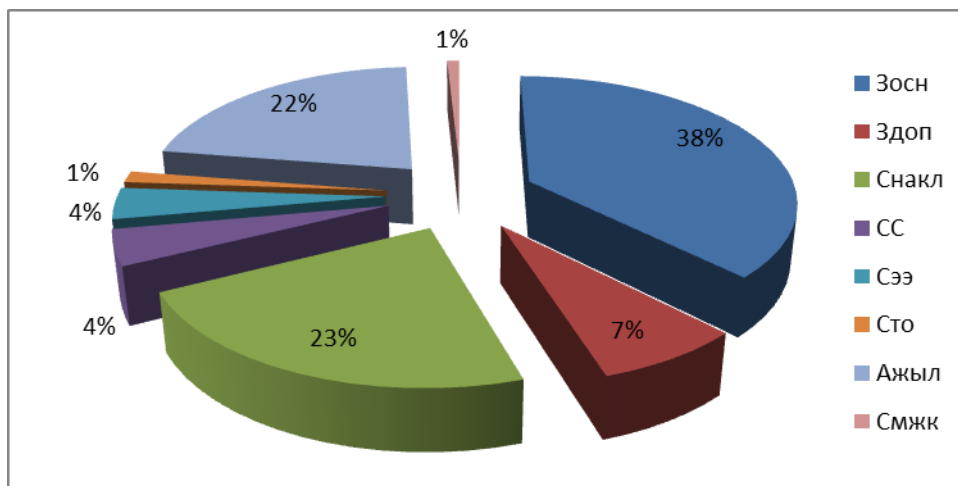
Бағдарламалық қамтаманы өңдеуге кеткен жалпы шығын еңбек жалақысы фондының, эксплуатационды шығынның, социалды сақтандыру, жүктеме шығындар мен материал және құрамалардың қосындысына тең болады.

Бағдарламалық қамтаманы өңдеудің қорытынды құны 1 кестеде көрсетілген.

К е с т е 5 . 3 – Өзіндік құн нәтижесінің кестесі

Шығын статьялары		Сомма, теңге	Жалпы соммадан, %
ФЗП	$Z_{\text{осн}}$	99265,49	38
	$Z_{\text{доп}}$	19853,1	7,5
Жүктеме шығындар, $C_{\text{накл}}$		5896,37	22,5
Әлеуметтік сақтандыру, CC		11792,7	4,4
Эксплуатационды Шығындар	$C_{\text{ээ}}^*$	10437	3,9
	$C_{\text{то}}$	3500	1,4
	$A_{\text{жыл}}^*$	53273,65	21,6
Материалдар мен комплектілер, $C_{\text{мик}}$		2100	1
Барлығы:		206116,65	100

Өзіндік құнының диаграммасы:



Сурет 5.1 – Өнімнің өзіндік құнының диаграммасы

5.6 Бағдарламаның бағасына есептеу

Өнімнің өзіндік құны 206116,65 теңге (жоғарыда кесте 3) болды. Өнімнің минималды бағасы келесі формуламен есептеледі:

$$Ц_{\min} = (C_{\text{пс}} + Ц_{\text{мд}} + Ц_{\text{з}}) \times (1 + r/100), \quad (5.21)$$

$$Ц_{\min} = (206116,65 + 200 + 150) \times 1,21 = 249401 \text{ теңге.}$$

мұндағы $Ц_{\text{мд}}$ - магнитті диск бағасы, қазіргі нарықтық бағасын қарастырып, 200 теңге деп алынды;

$Ц_{\text{з}}$ – жазба бағасы;

r - Өзіндік құнға қатынасымен алынған тиімділіктің жоспарланған пайызы=21%;

НДС ті есептегендегі минималды баға келесі формуламен есептеледі (ҚР да НДС 12%):

$$Ц_{\min_{\text{НДС}}} = Ц_{\min} + Н_{\text{ндс}} \times Ц_{\min}, \quad (5.22)$$

$$Ц_{\min_{\text{НДС}}} = 249401 + 0,12 \times 249401 = 279329 \text{ теңге.}$$

Сонда бағдарламаның құны 279329 теңге.

5.6.3 Бағдарламалық өнімді иемденудегі кәсіпорындардың бірмезгілдегі шығындарын есептеу

Кәсіпорындардың бірмезгілдегі шығындары келесі бөлімдерден тұрады:

- жүйе құны;
- көлік шығындары;
- оқуға кететін төлемдер;

Жүйе құны 279329 теңге (5.22 формула).

Көлік шығындары жүйе құнының 20% құрайды:

$$C_{\text{тр}}=279329 \times 0,2=55866 \text{ теңге.}$$

Жұмысшыларды жүйемен жұмыс істеуге оқыту шамамен екі сағатта консалтингті фирманың маманы көмегімен сағатына 3000 теңгедей болып табылады. Соған сәйкес оқу бағасы:

$$C_{\text{обуч}}=3000 \times 10=30000 \text{ теңге.}$$

Кәсіпорынның бірмезгілдегі шығындарын есептеулерінің нәтижесі 5.2-кестеде көрсетілген.

Кесте 5.4 – Ақпаратты жүйені енгізуге арналған бірмезгілдегі шығындардың есептелу нәтижесі

Шығындар түрлері	Сома, теңге
Жүйе құны	279329
Көлік шығындары	55866
Жұмысшыларды оқыту бағасы	30000
Қорытынды:	365195

5.6.4 Қолдану саласындағы жылдық шығындарды есептеу

Негізгі шығындар түрлері:

- жұмысшы еңбегіне төленетін ақша қоры;
- әлеуметтік салық;
- басқада шығындар.

ДұрыстауПК операторымен 70000 теңгежалақы көлемінде жүргізіледі.

Әлеуметтік салыққа жұмсалатын аударымдар(ҚР да әлеуметтік салық мөлшері 11% жұмысшы еңбегіне төленетін ақша қорынан, ҚР Салық Кодексі)

$$CC=(840000-840000 \times 0,1) \times 0,11=83160 \text{ теңге.}$$

Кесте 5.5 – Кәсіпорынның жылдық бірмезгілдегі шығындарын есептеу нәтижесі

Шығындар түрлері	Сома, теңге
Еңбекке төленетін қор	840000
Әлеуметтік салыққа аударымдар	83160
Қорытынды:	923160

5.6.5 Ақпараттық өнімді енгізуден алынған үнемдеу мөлшері мен табысты есептеу

Ақпараттық өнімді құрастыратын фирма үшін табыс көзі ретінде тапсырыс берушіге бағдарламалық өнімді сату болып табылады. Фирманың шығындары – ол құрастыруға кететін шығындар және жүйенің көбейтілуіне кететін шығындар. Қаржыландыру көзі ретінде құрастырушы-фирманың өзінің ақша қоры алынады.

Тапсырыс беруші-кәсіпорын үшін үнемдеу көзі есептеу уақытын біршама қысқартатын, қол еңбегін алмастыратын машина болып табылады. Кәсіпорын шығындары ақпараттық жүйені иемденудегі бірмезгілдік шығындардан, ақпараттық жүйені тасымалдау және енгізу шығындарынан, және де жүйеге талдау жүргізу мен бақылауға кететін шығындардан тұрады.

Жүйені енгізу қолмен есептеу шараларын автоматтандырылған шараға ауыстыру нәтижесінде өндірістің қорын біраз үнемдеуге алып келеді деп жоспарланады.

Үнемдеу мөлшерін есептейік. Ол үшін өткізу қабілетін есептеуге кететін кәсіпорынның шығындарын қолмен есептеу керек.

Қол еңбегімен келесі шығындар түрлері есептеледі:

- жұмысшы еңбегіне төленетін ақша қоры;
- әлеуметтік салық;
- басқа да шығындар.

Қолмен есептеу барысында еңбек өнімділігін талдау үшін 1500 теңге/сағ (ұйымның қойып отырған талабы) мөлшерінде жұмыс істейтін мамандандырылған инженер қажет. Өткізу қабілетін есептеу үшін инженер 21 күн жұмсайды. Қолмен есептеу барысында еңбектің жылдық төлем қоры:

$$EA=90000 \times 12=1080000 \text{ теңге.}$$

Әлеуметтік салыққа жұмсалатын аударымдар (ҚР да әлеуметтік салық мөлшері 11% жұмысшы еңбегіне төленетін ақша қорынан, ҚР Салық Кодексі):

$$CC=(1080000-1080000 \times 0,1) \times 0,11=106920 \text{ теңге.}$$

Кесте 5.6 – Ақпараттық жүйені қолданбай еңбек өнімділігін талдауға кететін шығындарды есептеу

Шығындар түрлері	Сома, теңге
Еңбекке төленетін қор	1080000
Әлеуметтік салыққа аударымдар	106920
Қорытынды:	1186920

Ақпараттық жүйені қолданғандағы жылдық шығындар жоғарыда есептелді және 21980 теңге болды.

Осыған сәйкес, шығындардың шартты үнемделуі мынаған тең болады:

$$C_{\text{экон}} = 1186920 - 923160 = 287520 \text{ теңге.}$$

5.7 Ақшалай құралдардың қозғалысы

Кесте 5.8 – Ақшалай тәсілдердің қозғалысы, теңге

Аты			
	2014	2015	2016
Бірмезгілдік шығындар	365195		
Операциялық кәсіпкерліктен пайда		287520	287520
Дисконттау коэффициенті (21% мөлшерінде)	1	0,826	0,683
Таза дисконтталған табыс (ТДТ)	-365195	217983	180152
ТДТәспелі нәтижесімен	-365195	-147211,7	32940

Таза дисконтталған табыс = Таза табыс * Дисконттау коэффициенті (5.23)

$$ТДТ_1 = 254968 \times 0,826 = 217983 \text{ теңге}$$

$$ТДТ_2 = 254968 \times 0,683 = 180152 \text{ теңге}$$

5.8 Экономикалық тиімділікті есептеу

5.8.1 Таза ағымдағы құндылықты есептеу (Net present value, NPV)

Шығындары бірмезгілде тек жобаның басында күрделі салымдар (C_0) ретінде іске асатын жобалар үшін NPV (24) формуламен есептеледі:

$$NPV = \sum_{i=1}^n \frac{B_i}{(1+r)^i} - C_0. \quad (5.24)$$

мұндағы B_i – i -ші жылдағы жобадан алынатын пайда;
 r – дисконттеу мөлшері.

$$NPV(21\%) = (287520(1/(1+0.21)^1 + 1/(1+0.21)^2 + 1/(1+0.21)^3) - 365195) = 32940 > 0$$

Сондықтан ұсынылып отырған жоба табысты. Сол себептен жобаны қажет және жобаны талдау мен оған баға беруді жалғастыру керек.

5.8.2 Пайда индекcін есептеу (Profitability index, PI)

PI жобаның салыстырмалы пайдасын көрсетеді. Бұл көрсеткіш келесі (25) формуламен есептеледі:

$$PI = \frac{NPV}{C_0} \quad (5.25)$$

$$PI = (398135/365195) = 1,09\%$$

5.8.3 Табыстың ішкі нормасын есептеу (Internal rate of return, IRR)

Табыстың ішкі нормасы (ТИН немесе IRR) r пайыздық мөлшерінің мәні болып табылады, мұндағы $NPV = 0$. Бұл нүктеде r^* қосылған шығындардың дисконтталған ағыны қосылған дисконтталған табыс ағындарына тең.

$$IRR = r_1 + \frac{f(r_1)}{f(r_1) - f(r_2)} * (r_2 - r_1) \quad (5.26)$$

$$r_b = 30\% : PV_1 = 287520 / (1 + 0,3) = 202892 \text{ теңге,}$$

$$PV_2 = 287520 / (1 + 0,3)^2 = 156071 \text{ теңге,}$$

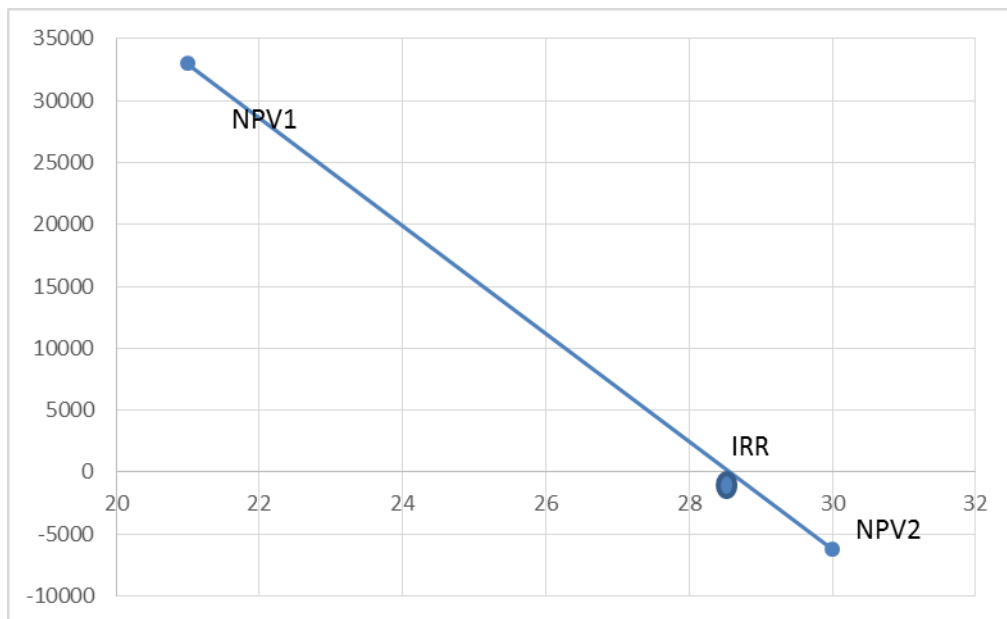
$$NPV(3\%) = (202892 + 156071) - 365195 = -6232 \text{ теңге}$$

$$NPV(21\%) = 32940 \text{ теңге,}$$

$$NPV(33\%) = -6232 \text{ теңге,}$$

$$IRR = 21 + 32940 / (32940 - (-6232)) * (30 - 21) = 28,57\% .$$

Табыстың ішкі нормасы сурет 2-де көрсетілген:



Сурет 5.2 - Іріктеу әдісі арқылы IRR табу

5.8.4 Өтімділік периодын есептеу (Payback period, PBP)

Өтімділік периоды мына формуламен есептеледі:

$$T_{\text{ок}} = t + \frac{C_0 - \sum_{i=1}^n B_i}{B_{i+1}} \quad (5.27)$$

мұндағы C_0 – жоба басындағы салым;

B_i – i -ші жылдағы жобадан алынатын пайда;

Сонда біздің өтімділік периодымыз:

$$T_{\text{ок}} = 1 + \frac{365195 - 217983}{180151} = 1.82 = 1 \text{ жыл } 10 \text{ ай.}$$

Бұл жобаның өтімділік периоды 1 жыл 10 айға тең, яғни жобамыз осы уақытта өз-өзін ақтайды.

Жалақысы 90000 тең компания есепшісін жұмыстан шығаруға болады. Бағдарламада барлық уақыт бойынша сатылған таур тізімі бар. Бағдарламадын тізімдерді шығарып алу мүмкіндігі бар. Компания бұл қызметкердің жұмысына мұқтаж емес.

Қорытынды

Осы дипломдық жобада барлық көзделген мақсаттар жүзеге асырылды. Компанияға арналып дерек қоры құрылды. Дерек қорын құру барысында барық ерекшеліктер мен қиындықтар ескерілді. Сонымен қатар компания тілектері бойынша бағдарлама функционалдығы арттырылды. Бағдарлама компания жұмысын ұйымдастыруға, оңай басқаруға және тауар айналымын бақылауға мүмкіндік береді. Компания жұмысы автоматтандырылды.

Деректер қоры Microsoft SQL Server деректер қорын басқару жүйесінде жасалды. Қамтама интерфейсі Microsoft Visual Studio 2012 бағдарламалау ортасында әзірленді. Авторизация қосымшасын кірістіру арқылы деректердің қауіпсіздігі қамтамасыз етілді. Тағы бір айта кететін жайт артық қажетсіз ақпарат енгізілмеу үшін қызметтерді бөлу арқылы деректі тек менеджер лауазымынан жоғары қызметкерлер немесе арнайы рұқсаты бар кеңесшілер енгізе алатын болды.

«Тіршілік қауіпсіздігі» бөлімінде берілген офистік ғимараттағы еңбек шарты талданды, оның ішінде электор тоғымен зақымданудан қорғану үшін құрылғыларды нөлдеу процесін орындаудың қолайлы параметрлер көрсетілген. Еңбек шартының деңгейі қолайлы деп бағаланды және есептеулерден алынған мәліметтер өмір тіршілігі қауіпсіздігі стандартты талаптарға сәйкес келеді. Есептеулер қысқа тұйықталу кезінде жедел әрекет ететін ажыратушы датчиктер орнату жеткілікті болатынын көрсетті.

Техника-экономикалық негізделуі бөлімінде жасалған есептеулер нәтижесінде бағдарламалық өнімді қолданысқа енгізген соң 1 жыл 10 ай мерзімінде бағдарлама өзін-өзі өтейтінгі анықталды. Жобаның құны жоғары болғанымен компанияда өтейтін шығындары көп болғандықтан жоба пайдалы.

Пайдаланған әдебиеттер тізімі:

1 Ник Рендольф, Дэвид Гарднер, Майкл Минутилло, Крис Андерсон Visual Studio 2010 для профессионалов = Professional Visual Studio 2010. — М.: «Диалектика», 2011.

2 Роберт Э. Уолтерс, Майкл Коулс SQL Server 2008: ускоренный курс для профессионалов = Accelerated SQL Server 2008. — М.: «Вильямс», 2008.

3 Карли Уотсон, Кристиан Нейгел, Якоб Хаммер Педерсен, и др. Visual C# 2008: базовый курс. Visual Studio® 2008 = Beginning Visual C# 2008. — М.: «Диалектика», 2009.

4 Роберт Виейра Программирование баз данных Microsoft SQL Server 2005. Базовый курс = Beginning Microsoft SQL Server 2005 Programming. — М.: «Диалектика», 2007.

5 Когаловский М. Р. Энциклопедия технологий баз данных. — М.: Финансы и статистика, 2002.

6 Базылов Қ.Б., Алибаева С.А., Нурмагамбетова С.С. Бітіруші жұмысының экономикалық бөлімі үшін әдістемелік нұсқаулар. — Алматы: АИЭС, 2009. — 25 б.

7 Иванов И.Н. Экономика промышленного предприятия: Учебник. ИНФРА-М, 2011. — 305 с.

8 Рофе А.И. Экономика труда: Учебник. — М.: КноРус, 2010. — 400 с.

9 Остапенко Ю.М. Экономика труда: Учеб.пособие. — М.: ИЦ РИОР, 2010. — 160 с.

10 Серебренников Г.Г., Тамбов. Экономические аспекты организации производства: Учебное пособие.—М.: Изд-во Эксмо, 2010. — 80с.

11 Козырев В.М. Основы современной экономики: учебник-3-е изд.

12 М.: Изд-во Финансы и статистика, 2008. — 98 с.

13 Манилов, Р. Г. Бизнес-план: Методические материалы / Под ред. Р.Г. Маниловского. — М.: Изд-во Финансы и статистика, 2008. — 124с.

А қосымшасы

Админ бөлігін дайындау коды:

```
namespace AdminKCG
{
    partial class EnterForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(EnterForm));
            this.button1 = new System.Windows.Forms.Button();
            this.textBox2 = new System.Windows.Forms.TextBox();
            this.textBox1 = new System.Windows.Forms.TextBox();
            this.label3 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.label1 = new System.Windows.Forms.Label();
            this.pictureBox1 = new System.Windows.Forms.PictureBox();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
            this.SuspendLayout();
            //
            // button1
            //
            this.button1.Font = new System.Drawing.Font("Verdana", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)204));
```

А қосымшасының жалғасы

```
this.button1.Location = new System.Drawing.Point(504, 262);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(102, 36);
this.button1.TabIndex = 13;
this.button1.Text = "Вход";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new System.EventHandler(this.button1_Click);
//
// textBox2
//
this.textBox2.Font = new System.Drawing.Font("Verdana", 10F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.textBox2.Location = new System.Drawing.Point(362, 214);
this.textBox2.Name = "textBox2";
this.textBox2.PasswordChar = '#';
this.textBox2.Size = new System.Drawing.Size(244, 24);
this.textBox2.TabIndex = 12;
//
// textBox1
//
this.textBox1.Font = new System.Drawing.Font("Verdana", 10F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.textBox1.Location = new System.Drawing.Point(362, 173);
this.textBox1.Name = "textBox1";
this.textBox1.Size = new System.Drawing.Size(244, 24);
this.textBox1.TabIndex = 11;
//
// label3
//
this.label3.AutoSize = true;
this.label3.BackColor = System.Drawing.Color.Transparent;
this.label3.Font = new System.Drawing.Font("Verdana", 10F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.label3.Location = new System.Drawing.Point(279, 217);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(61, 17);
this.label3.TabIndex = 10;
this.label3.Text = "Пароль";
//
// label2
//
this.label2.AutoSize = true;
this.label2.BackColor = System.Drawing.Color.Transparent;
this.label2.Font = new System.Drawing.Font("Verdana", 10F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));
this.label2.Location = new System.Drawing.Point(279, 176);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(52, 17);
this.label2.TabIndex = 9;
```


А қосымшасының жалғасы

```
this.label2.Text = "Логин";
//
// label1
//
this.label1.AutoSize = true;
this.label1.BackColor = System.Drawing.Color.Transparent;
this.label1.Font = new System.Drawing.Font("Verdana", 15.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)204));
this.label1.Location = new System.Drawing.Point(199, 63);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(353, 25);
this.label1.TabIndex = 8;
this.label1.Text = "Магазин строительных товаров";
//
// pictureBox1
//
this.pictureBox1.BackColor = System.Drawing.Color.Transparent;
this.pictureBox1.Image =
((System.Drawing.Image)resources.GetObject("pictureBox1.Image"));
this.pictureBox1.Location = new System.Drawing.Point(12, 12);
this.pictureBox1.Name = "pictureBox1";
this.pictureBox1.Size = new System.Drawing.Size(163, 87);
this.pictureBox1.TabIndex = 7;
this.pictureBox1.TabStop = false;
//
// EnterForm
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.BackgroundImage =
((System.Drawing.Image)resources.GetObject("$this.BackgroundImage"));
this.ClientSize = new System.Drawing.Size(679, 362);
this.Controls.Add(this.button1);
this.Controls.Add(this.textBox2);
this.Controls.Add(this.textBox1);
this.Controls.Add(this.label3);
this.Controls.Add(this.label2);
this.Controls.Add(this.label1);
this.Controls.Add(this.pictureBox1);
this.Name = "EnterForm";
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "KCG";
((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
this.ResumeLayout(false);
this.PerformLayout();

}

#endregion
```

А қосымшасының жалғасы

```
private System.Windows.Forms.Button button1;
private System.Windows.Forms.TextBox textBox2;
private System.Windows.Forms.TextBox textBox1;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.PictureBox pictureBox1;
}
}

using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data.SqlClient;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace AdminKCG
{
    static class Program
    {
        public static SqlConnection conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["KcgContext"].ConnectionString);
        public static kcgDataContext db = new kcgDataContext(conn);
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new EnterForm());
        }
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Drawing.Printing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

А қосымшасының жалғасы

```
using System.Windows.Forms;

namespace KCG
{
    public partial class OperationForm : Form
    {
        private static int mag_id;
        public OperationForm()
        {
            InitializeComponent();
            mag_id = (from i in Program.db.Kassirs where i.kas_id == EnterForm.kas_id select
i.mag_id).SingleOrDefault();
            var mag_name = (from i in Program.db.Magazins where i.mag_id == mag_id select
i.mag_name).SingleOrDefault();
            label1.Text = mag_name;
            a1 = "select t.tovar_id, t.tovar_name as 'Наименование', t.tovar_sum as 'Цена', t.tovar_kol
as 'Количество' from tovary t";
            connect();
            dataGridView1.Columns["tovar_id"].Visible = false;
            var kat = from i in Program.db.Kategoriyas select i.kat_name;
            foreach (var i in kat)
            {
                comboBox1.Items.Add(i);
            }
        }

        private void dataGridView1_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e)
        {
            int index = e.RowIndex;
            string indexStr = (index + 1).ToString();
            object header = this.dataGridView1.Rows[index].HeaderCell.Value;
            if (header == null || !header.Equals(indexStr))
                this.dataGridView1.Rows[index].HeaderCell.Value = indexStr;
        }

        private void dataGridView2_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e)
        {
            int index = e.RowIndex;
            string indexStr = (index + 1).ToString();
            object header = this.dataGridView2.Rows[index].HeaderCell.Value;
            if (header == null || !header.Equals(indexStr))
                this.dataGridView2.Rows[index].HeaderCell.Value = indexStr;
        }

        private string a, a1, b, b1, cn;
        public void connect()
        {
```

А қосымшасының жалғасы

```
SqlConnection cn = Program.conn;
a = a1;
DataSet ds = new DataSet();
SqlDataAdapter da = new SqlDataAdapter(a, cn);
da.Fill(ds);
dataGridView1.DataSource = ds.Tables[0];
}

public void connect2()
{
    SqlConnection cn = Program.conn;
    b = b1;
    DataSet ds = new DataSet();
    SqlDataAdapter da = new SqlDataAdapter(b, cn);
    da.Fill(ds);
    dataGridView2.DataSource = ds.Tables[0];
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    int kat_id = (from i in Program.db.Kategoriyas where i.kat_name == comboBox1.Text
select i.kat_id).SingleOrDefault();
    a1 = "select t.tovar_id, t.tovar_name as 'Наименование', t.tovar_sum as 'Цена', t.tovar_kol
as 'Количество' from tovary t where t.kat_id = " + kat_id;
    connect();
    dataGridView1.Columns["tovar_id"].Visible = false;
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    a1 = "select t.tovar_id, t.tovar_name as 'Наименование', t.tovar_sum as 'Цена', t.tovar_kol
as 'Количество' from tovary t where t.tovar_name like '" + textBox1.Text + "%'";
    connect();
    dataGridView1.Columns["tovar_id"].Visible = false;
}

private void button1_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(textBox3.Text))
    {
        MessageBox.Show("Введите количество.", "Внимание!", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
    else
    {
        try
        {
            SqlConnection sqlConnection1 = Program.conn;
            sqlConnection1.Open();
        }
    }
}
```

А қосымшасының жалғасы

```
SqlCommand och = new SqlCommand();
och.CommandType = System.Data.CommandType.Text;
och.CommandText = "select MAX(Id) from Ochered";
och.Connection = sqlConnection1;
var numb = och.ExecuteScalar();
sqlConnection1.Close();

int tov_id = Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value);
int kol = Convert.ToInt32(textBox3.Text);
int db_kol = (from i in Program.db.tovaries where i.tovar_id == tov_id select
i.tovar_kol).SingleOrDefault();
if (kol > db_kol)
{
    MessageBox.Show("Количество не может превышать количество из базы",
"Внимание", MessageBoxButtons.OK, MessageBoxIcon.Stop);
}
else
{
    int sum = Convert.ToInt32(dataGridView1.CurrentRow.Cells[2].Value);
    int kol_s = sum * kol;
    sqlConnection1.Open();
    SqlCommand cmd = new SqlCommand();
    SqlCommand cmd1 = new SqlCommand();
    cmd.CommandType = System.Data.CommandType.Text;
    cmd1.CommandType = System.Data.CommandType.Text;
    cmd.CommandText = "insert into Klienty(Id, mag_id, kas_id, tovar_id, kol, kl_sum,
status) values('" + numb + "', '" + mag_id + "', '" +
EnterForm.kas_id + "', '" + tov_id + "', '" + kol + "', '" + kol_s + "', 0)";
    cmd1.CommandText = "update tovary set tovar_kol = tovar_kol - '" + kol + "' where
tovar_id = '" + tov_id + "'";
    cmd.Connection = sqlConnection1;
    cmd1.Connection = sqlConnection1;
    cmd.ExecuteNonQuery();
    cmd1.ExecuteNonQuery();
    sqlConnection1.Close();
    MessageBox.Show("Товар добавлен", "Внимание", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    textBox3.Clear();
    a1 = "select t.tovar_id, t.tovar_name as 'Наименование', t.tovar_sum as 'Цена',
t.tovar_kol as 'Количество' from tovary t where t.tovar_name like '" + textBox1.Text + "%'";
    connect();
    dataGridView1.Columns["tovar_id"].Visible = false;
    b1 = "select k.Klient_id, t.tovar_name as 'Наименование', k.kol as 'Количество',
k.kl_sum as 'Стоимость' from Klienty k inner join tovary t on t.tovar_id = k.tovar_id where k.Id =
'" + numb + "'";
    connect2();
    dataGridView2.Columns["Klient_id"].Visible = false;

    sqlConnection1.Open();
```

А қосымшасының жалғасы

```
        SqlCommand sumc = new SqlCommand();
        sumc.CommandType = System.Data.CommandType.Text;
        sumc.CommandText = "select SUM(k.kl_sum) from Klienty k where k.Id = '" +
numb + "' and k.status = 0";
        sumc.Connection = sqlConnection1;
        var sum1 = sumc.ExecuteScalar();
        sqlConnection1.Close();
        textBox2.Text = sum1.ToString();
    }
}
catch (Exception)
{
    MessageBox.Show("Операция не выполнено", "Внимание",
MessageBoxButtons.OK, MessageBoxIcon.None);
}

}
}

private void button2_Click(object sender, EventArgs e)
{
    try
    {
        SqlConnection sqlConnection1 = Program.conn;
        sqlConnection1.Open();
        SqlCommand och = new SqlCommand();
        och.CommandType = System.Data.CommandType.Text;
        och.CommandText = "select MAX(Id) from Ochered";
        och.Connection = sqlConnection1;
        var numb = och.ExecuteScalar();
        sqlConnection1.Close();

        sqlConnection1.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = System.Data.CommandType.Text;
        cmd.CommandText = "update Klienty set status = 1 where Id = '" + numb + "'";
        cmd.Connection = sqlConnection1;
        cmd.ExecuteNonQuery();
        sqlConnection1.Close();
        MessageBox.Show("Операция выполнено успешно", "Внимание",
MessageBoxButtons.OK, MessageBoxIcon.Information);

        sqlConnection1.Open();
        SqlCommand ochu = new SqlCommand();
        ochu.CommandType = System.Data.CommandType.Text;
        ochu.CommandText = "update Ochered set Idate = GETDATE() where Id = '" + numb +
'''';
        ochu.Connection = sqlConnection1;
        ochu.ExecuteScalar();
```

А қосымшасының жалғасы

```
sqlConnection1.Close();

    sqlConnection1.Open();
    SqlCommand ochi = new SqlCommand();
    ochi.CommandType = System.Data.CommandType.Text;
    ochi.CommandText = "insert into Ochered(Idate) values(GETDATE())";
    ochi.Connection = sqlConnection1;
    ochi.ExecuteNonQuery();
    sqlConnection1.Close();

    var path = Application.StartupPath + @"..\..\\" + @"\check.txt";
    StreamWriter sw = new StreamWriter(path);
    int magazin_id = (from i in Program.db.Kassirs where i.kas_id == EnterForm.kas_id
select i.mag_id).SingleOrDefault();
    string magazin_n = (from i in Program.db.Magazins where i.mag_id == magazin_id
select i.mag_name).SingleOrDefault();
    sw.WriteLine(magazin_n);

    for (int i = 0; i <= dataGridView2.Rows.Count; i++)
    {
        if (i == dataGridView2.Rows.Count)
        {
            string kas_fam = (from t in Program.db.Kassirs where t.kas_id == EnterForm.kas_id
select t.kas_fam).SingleOrDefault();
            string kas_name = (from t in Program.db.Kassirs where t.kas_id ==
EnterForm.kas_id select t.kas_name).SingleOrDefault();
            string kas_patr = (from t in Program.db.Kassirs where t.kas_id == EnterForm.kas_id
select t.kas_patronimic).SingleOrDefault();

            sw.WriteLine("Дата и время: " + DateTime.Now);
            sw.WriteLine("Кассир: " + kas_fam + " " + kas_name + " " + kas_patr);
            sw.Close();
            PrintDocument D = new PrintDocument();
            D.DocumentName = path;
            try
            {
                D.Print();
            }
            catch (Exception)
            {
            }

            b1 = "select k.Klient_id, t.tovar_name as 'Наименование', k.kol as 'Количество',
k.kl_sum as 'Стоимость' from Klienty k inner join tovary t on t.tovar_id = k.tovar_id where k.Id =
'" + numb + "' and k.status = 0";
            connect2();
            dataGridView2.Columns["Klient_id"].Visible = false;
            textBox2.Clear();
        }
    }
}
```

А қосымшасының жалғасы

```
else
    {
        sw.WriteLine(dataGridView2.Rows[i].Cells[1].Value + ": кол-" +
dataGridView2.Rows[i].Cells[2].Value + " цена: " + dataGridView2.Rows[i].Cells[3].Value);
    }
}
}
catch (Exception)
{
    MessageBox.Show("Операция не выполнено", "Внимание", MessageBoxButtons.OK,
MessageBoxIcon.None);
}
}

private void button3_Click(object sender, EventArgs e)
{
    try
    {
        SqlConnection sqlConnection1 = Program.conn;
        sqlConnection1.Open();
        SqlCommand och = new SqlCommand();
        och.CommandType = System.Data.CommandType.Text;
        och.CommandText = "select MAX(Id) from Ochered";
        och.Connection = sqlConnection1;
        var numb = och.ExecuteScalar();
        sqlConnection1.Close();

        sqlConnection1.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = System.Data.CommandType.Text;
        cmd.CommandText = "delete from Klienty where klient_id = '" +
dataGridView2.CurrentRow.Cells[0].Value + "'";
        cmd.Connection = sqlConnection1;
        cmd.ExecuteNonQuery();
        sqlConnection1.Close();
        MessageBox.Show("Операция выполнена успешно", "Внимание!",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        b1 = "select k.Klient_id, t.tovar_name as 'Наименование', k.kol as 'Количество',
k.kl_sum as 'Стоимость' from Klienty k inner join tovary t on t.tovar_id = k.tovar_id where k.Id =
'" + numb + "' and k.status = 0";
        connect2();
        dataGridView2.Columns["Klient_id"].Visible = false;

        sqlConnection1.Open();
        SqlCommand sumc = new SqlCommand();
        sumc.CommandType = System.Data.CommandType.Text;
        sumc.CommandText = "select SUM(k.kl_sum) from Klienty k where k.Id = '" + numb +
'" and k.status = 0";
```


А қосымшасының соңы

```
sumc.Connection = sqlConnection1;
var sum1 = sumc.ExecuteScalar();
sqlConnection1.Close();
textBox2.Text = sum1.ToString();
}
catch (Exception)
{
    MessageBox.Show("Операция не выполнено", "Внимание", MessageBoxButtons.OK,
    MessageBoxIcon.None);
}
}

private void button4_Click(object sender, EventArgs e)
{
    this.Close();
}
```

Ә қосымшасы

Супер админ қосымшасын құру коды:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SuperAdminKCG
{
    public partial class OperationForm : Form
    {
        private static int rec1, rec2;
        public OperationForm()
        {
            InitializeComponent();
            a1 = "select mag_id, mag_name as 'Наименование', login as 'Логин', password as 'Пароль'
from Magazin";
            connect();
            dataGridView1.Columns["mag_id"].Visible = false;

            b1 = "select k.kas_id, kas_fam as 'Фамилия', k.kas_name as 'Имя', k.kas_patronymic as
'Отчество', m.mag_name as 'Магазин', k.login as 'Логин', k.password as 'Пароль' from Kassir k
inner join Magazin m on m.mag_id = k.mag_id";
            connect2();
            dataGridView2.Columns["kas_id"].Visible = false;

            comboBox2.Items.Add("Все");
            var mag = (from i in Program.db.Magazins select i.mag_name);
            foreach (var i in mag)
            {
                comboBox1.Items.Add(i);
                comboBox2.Items.Add(i);
                comboBox3.Items.Add(i);
            }
            c1 = "select t.tovar_name as 'Наименование', t.tovar_sum as 'Цена', k.kol as 'Количество
проданных', k.kl_sum as 'Доход', o.Idate as 'Дата' from Klienty k inner join tovary t on t.tovar_id
= k.tovar_id inner join Ochered o on o.Id = k.Id where k.status = 1";
            connect3();

            int sum = 0;

            for (int i = 0; i <= dataGridView3.RowCount; i++)
            {
                if (i == dataGridView3.RowCount)
```

Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
{
    textBox11.Text = sum.ToString();
}
else
{
    sum = sum + Convert.ToInt32(dataGridView3.Rows[i].Cells[3].Value);
}
}

groupBox2.Enabled = false;
groupBox4.Enabled = false;

comboBox4.Enabled = false;

var c4 = from i in Program.db.Kategoriyas select i.kat_name;
foreach (var i in c4)
{
    comboBox4.Items.Add(i);
}

d1 = "select t.tovar_name as 'Наименование', t.tovar_sum as 'Стоимость', t.tovar_kol as
'Количество' from tovary t where t.tovar_kol != 0";
connect4();
}

private void dataGridView1_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e)
{
    int index = e.RowIndex;
    string indexStr = (index + 1).ToString();
    object header = this.dataGridView1.Rows[index].HeaderCell.Value;
    if (header == null || !header.Equals(indexStr))
        this.dataGridView1.Rows[index].HeaderCell.Value = indexStr;
}

private void dataGridView2_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e)
{
    int index = e.RowIndex;
    string indexStr = (index + 1).ToString();
    object header = this.dataGridView2.Rows[index].HeaderCell.Value;
    if (header == null || !header.Equals(indexStr))
        this.dataGridView2.Rows[index].HeaderCell.Value = indexStr;
}

private void dataGridView3_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e)
{
    int index = e.RowIndex;
```

Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
string indexStr = (index + 1).ToString();
    object header = this.dataGridView3.Rows[index].HeaderCell.Value;
    if (header == null || !header.Equals(indexStr))
        this.dataGridView3.Rows[index].HeaderCell.Value = indexStr;
}
```

```
private void dataGridView4_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e)
{
    int index = e.RowIndex;
    string indexStr = (index + 1).ToString();
    object header = this.dataGridView4.Rows[index].HeaderCell.Value;
    if (header == null || !header.Equals(indexStr))
        this.dataGridView4.Rows[index].HeaderCell.Value = indexStr;
}
```

```
private string a, a1, b, b1, c, c1, d, d1, cn;
public void connect()
{
    SqlConnection cn = Program.conn;
    a = a1;
    DataSet ds = new DataSet();
    SqlDataAdapter da = new SqlDataAdapter(a, cn);
    da.Fill(ds);
    dataGridView1.DataSource = ds.Tables[0];
}
```

```
public void connect2()
{
    SqlConnection cn = Program.conn;
    b = b1;
    DataSet ds = new DataSet();
    SqlDataAdapter da = new SqlDataAdapter(b, cn);
    da.Fill(ds);
    dataGridView2.DataSource = ds.Tables[0];
}
```

```
public void connect3()
{
    SqlConnection cn = Program.conn;
    c = c1;
    DataSet ds = new DataSet();
    SqlDataAdapter da = new SqlDataAdapter(c, cn);
    da.Fill(ds);
    dataGridView3.DataSource = ds.Tables[0];
}
```

```
public void connect4()
{
```

Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
SqlConnection cn = Program.conn;
d = d1;
DataSet ds = new DataSet();
SqlDataAdapter da = new SqlDataAdapter(d, cn);
da.Fill(ds);
dataGridView4.DataSource = ds.Tables[0];
}

private void button4_Click(object sender, EventArgs e)
{
    this.Close();
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    a1 = "select mag_id, mag_name as 'Наименование', login as 'Логин', password as 'Пароль'
from Magazin where mag_name like '" + textBox1.Text + "%'";
    connect();
    dataGridView1.Columns["mag_id"].Visible = false;
}

private void textBox5_TextChanged(object sender, EventArgs e)
{
    b1 = "select k.kas_id, kas_fam as 'Фамилия', k.kas_name as 'Имя', k.kas_patronymic as
'Отчество', m.mag_name as 'Магазин', k.login as 'Логин', k.password as 'Пароль' from Kassir k
inner join Magazin m on m.mag_id = k.mag_id where k.kas_fam like '" + textBox5.Text + "%'";
    connect2();
    dataGridView2.Columns["kas_id"].Visible = false;
}

private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    if (comboBox2.Text == "Все")
    {
        c1 = "select t.tovar_name as 'Наименование', t.tovar_sum as 'Цена', k.kol as
'Количество проданных', k.kl_sum as 'Доход', o.Idate as 'Дата' from Klienty k inner join tovary t
on t.tovar_id = k.tovar_id inner join Ochered o on o.Id = k.Id where k.status = 1";
        connect3();

        int sum = 0;

        for (int i = 0; i <= dataGridView3.RowCount; i++)
        {
            if (i == dataGridView3.RowCount)
            {
                textBox11.Text = sum.ToString();
            }
            else
            {
```

Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
        sum = sum + Convert.ToInt32(dataGridView3.Rows[i].Cells[3].Value);
    }
}
else
{
    var mag_id = (from i in Program.db.Magazins where i.mag_name == comboBox2.Text
select i.mag_id).SingleOrDefault();
    c1 = "select t.tovar_name as 'Наименование', t.tovar_sum as 'Цена', k.kol as
'Количество проданных', k.kl_sum as 'Доход', o.Idate as 'Дата' from Klienty k inner join tovary t
on t.tovar_id = k.tovar_id inner join Ochered o on o.Id = k.Id where k.status = 1 and k.mag_id = '"
+ mag_id + "'";
    connect3();

    int sum = 0;

    for (int i = 0; i <= dataGridView3.RowCount; i++)
    {
        if (i == dataGridView3.RowCount)
        {
            textBox11.Text = sum.ToString();
        }
        else
        {
            sum = sum + Convert.ToInt32(dataGridView3.Rows[i].Cells[3].Value);
        }
    }
}

private void button10_Click(object sender, EventArgs e)
{
    c1 = "select t.tovar_name as 'Наименование', t.tovar_sum as 'Цена', k.kol as 'Количество
проданных', k.kl_sum as 'Доход', o.Idate as 'Дата' from Klienty k inner join tovary t on t.tovar_id
= k.tovar_id inner join Ochered o on o.Id = k.Id where k.status = 1 and (DATEPART(yy, o.Idate) =
" + dateTimePicker1.Value.Year + " and DATEPART(mm, o.Idate) = " +
dateTimePicker1.Value.Month + " and DATEPART(dd, o.Idate) = " + dateTimePicker1.Value.Day
+ ")";
    connect3();
    int sum = 0;

    for (int i = 0; i <= dataGridView3.RowCount; i++)
    {
        if (i == dataGridView3.RowCount)
        {
            textBox11.Text = sum.ToString();
        }
        else
        {
```

Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
sum = sum + Convert.ToInt32(dataGridView3.Rows[i].Cells[3].Value);
    }
}

private void button1_Click(object sender, EventArgs e)
{
    groupBox2.Enabled = true;
    groupBox1.Enabled = false;
    rec1 = 0;
}

private void button2_Click(object sender, EventArgs e)
{
    groupBox2.Enabled = true;
    groupBox1.Enabled = false;
    rec1 = 1;
    textBox2.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
    textBox3.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
    textBox4.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
}

private void button5_Click(object sender, EventArgs e)
{
    groupBox2.Enabled = false;
    groupBox1.Enabled = true;
    textBox2.Clear();
    textBox3.Clear();
    textBox4.Clear();
}

private void button3_Click(object sender, EventArgs e)
{
    if (rec1 == 0)
    {
        SqlConnection sqlConnection1 = Program.conn;
        sqlConnection1.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = System.Data.CommandType.Text;
        cmd.CommandText = "insert into Magazin(mag_name, login, password) values('" +
textBox2.Text + "', '" + textBox3.Text + "', '" + textBox4.Text + "')";
        cmd.Connection = sqlConnection1;
        cmd.ExecuteNonQuery();
        sqlConnection1.Close();
        MessageBox.Show("Операция выполнено успешно", "Внимание",
MessageBoxButtons.OK, MessageBoxIcon.Information);

        groupBox2.Enabled = false;
        groupBox1.Enabled = true;
    }
}
```

Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
textBox2.Clear();
textBox3.Clear();
textBox4.Clear();
a1 = "select mag_id, mag_name as 'Наименование', login as 'Логин', password as
'Пароль' from Magazin";
connect();
dataGridView1.Columns["mag_id"].Visible = false;
comboBox1.Items.Clear();
comboBox2.Items.Clear();
comboBox3.Items.Clear();
comboBox2.Items.Add("Все");
var mag = (from i in Program.db.Magazines select i.mag_name);
foreach (var i in mag)
{
    comboBox1.Items.Add(i);
    comboBox2.Items.Add(i);
    comboBox3.Items.Add(i);
}
}
else if (rec1 == 1)
{
    int mag_id = Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value);
    SqlConnection sqlConnection1 = Program.conn;
    sqlConnection1.Open();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = System.Data.CommandType.Text;
    cmd.CommandText = "update Magazin set mag_name = " + textBox2.Text + ", login =
" + textBox3.Text + ", password = " + textBox4.Text + " where mag_id = " + mag_id + """;
    cmd.Connection = sqlConnection1;
    cmd.ExecuteNonQuery();
    sqlConnection1.Close();
    MessageBox.Show("Операция выполнено успешно", "Внимание",
MessageBoxButtons.OK, MessageBoxIcon.Information);

    groupBox2.Enabled = false;
    groupBox1.Enabled = true;
    textBox2.Clear();
    textBox3.Clear();
    textBox4.Clear();
    a1 = "select mag_id, mag_name as 'Наименование', login as 'Логин', password as
'Пароль' from Magazin";
    connect();
    dataGridView1.Columns["mag_id"].Visible = false;
    comboBox1.Items.Clear();
    comboBox2.Items.Clear();
    comboBox2.Items.Add("Все");
    var mag = (from i in Program.db.Magazines select i.mag_name);
    foreach (var i in mag)
    {
```


Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
        comboBox1.Items.Add(i);
        comboBox2.Items.Add(i);
    }
}

private void button6_Click(object sender, EventArgs e)
{
    groupBox4.Enabled = true;
    groupBox3.Enabled = false;
    rec2 = 0;
}

private void button7_Click(object sender, EventArgs e)
{
    groupBox4.Enabled = true;
    groupBox3.Enabled = false;
    rec2 = 1;
    textBox6.Text = dataGridView2.CurrentRow.Cells[1].Value.ToString();
    textBox7.Text = dataGridView2.CurrentRow.Cells[2].Value.ToString();
    textBox8.Text = dataGridView2.CurrentRow.Cells[3].Value.ToString();
    comboBox1.Text = dataGridView2.CurrentRow.Cells[4].Value.ToString();
    textBox9.Text = dataGridView2.CurrentRow.Cells[5].Value.ToString();
    textBox10.Text = dataGridView2.CurrentRow.Cells[6].Value.ToString();
}

private void button9_Click(object sender, EventArgs e)
{
    groupBox3.Enabled = true;
    groupBox4.Enabled = false;
    textBox6.Clear();
    textBox7.Clear();
    textBox8.Clear();
    textBox9.Clear();
    textBox10.Clear();
    comboBox1.Items.Clear();
    var mag = (from i in Program.db.Magazines select i.mag_name);
    foreach (var i in mag)
    {
        comboBox1.Items.Add(i);
    }
}

private void button8_Click(object sender, EventArgs e)
{
    if (rec2 == 0)
    {
        int mag_id = (from i in Program.db.Magazines where i.mag_name == comboBox1.Text
select i.mag_id).SingleOrDefault();
```

Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
SqlConnection sqlConnection1 = Program.conn;
sqlConnection1.Open();
SqlCommand cmd = new SqlCommand();
cmd.CommandType = System.Data.CommandType.Text;
cmd.CommandText = "insert into Kassir(kas_fam, kas_name, kas_patronimic, mag_id,
login, password) values('" + textBox6.Text + "', '" + textBox7.Text + "', '" + textBox8.Text + "', '" +
mag_id + "', '" + textBox9.Text + "', '" + textBox10.Text + "')";
cmd.Connection = sqlConnection1;
cmd.ExecuteNonQuery();
sqlConnection1.Close();
MessageBox.Show("Операция выполнено успешно", "Внимание",
MessageBoxButtons.OK, MessageBoxIcon.Information);
groupBox3.Enabled = true;
groupBox4.Enabled = false;
textBox6.Clear();
textBox7.Clear();
textBox8.Clear();
textBox9.Clear();
textBox10.Clear();
comboBox1.Items.Clear();
var mag = (from i in Program.db.Magazins select i.mag_name);
foreach (var i in mag)
{
    comboBox1.Items.Add(i);
}
b1 = "select k.kas_id, kas_fam as 'Фамилия', k.kas_name as 'Имя', k.kas_patronimic as
'Отчество', m.mag_name as 'Магазин', k.login as 'Логин', k.password as 'Пароль' from Kassir k
inner join Magazin m on m.mag_id = k.mag_id";
connect2();
dataGridView2.Columns["kas_id"].Visible = false;
}
else if (rec2 == 1)
{
    int mag_id = (from i in Program.db.Magazins where i.mag_name == comboBox1.Text
select i.mag_id).SingleOrDefault();
    int kas_id = Convert.ToInt32(dataGridView2.CurrentRow.Cells[0].Value);
    SqlConnection sqlConnection1 = Program.conn;
    sqlConnection1.Open();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = System.Data.CommandType.Text;
    cmd.CommandText = "update Kassir set kas_fam = '" + textBox6.Text + "', kas_name =
'" + textBox7.Text + "', kas_patronimic = '" + textBox8.Text + "', mag_id = '" + mag_id + "', login =
'" + textBox9.Text + "', password = '" + textBox10.Text + "' where kas_id = '" + kas_id + "'";
    cmd.Connection = sqlConnection1;
    cmd.ExecuteNonQuery();
    sqlConnection1.Close();
    MessageBox.Show("Операция выполнено успешно", "Внимание",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    groupBox3.Enabled = true;
```

Ә ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
groupBox4.Enabled = false;
textBox6.Clear();
textBox7.Clear();
textBox8.Clear();
textBox9.Clear();
textBox10.Clear();
comboBox1.Items.Clear();
var mag = (from i in Program.db.Magazins select i.mag_name);
foreach (var i in mag)
{
    comboBox1.Items.Add(i);
}
b1 = "select k.kas_id, kas_fam as 'Фамилия', k.kas_name as 'Имя', k.kas_patronymic as
'Отчество', m.mag_name as 'Магазин', k.login as 'Логин', k.password as 'Пароль' from Kassir k
inner join Magazin m on m.mag_id = k.mag_id";
connect2();
dataGridView2.Columns["kas_id"].Visible = false;
}
}

private void comboBox3_SelectedIndexChanged(object sender, EventArgs e)
{
    comboBox4.Enabled = true;
    int mi = (from i in Program.db.Magazins where i.mag_name == comboBox3.Text select
i.mag_id).SingleOrDefault();
    d1 = "select t.tovar_name as 'Наименование', t.tovar_sum as 'Стоимость', t.tovar_kol as
'Количество' from tovary t where t.tovar_kol != 0 and t.mag_id = " + mi;
    connect4();
}

private void comboBox4_SelectedIndexChanged(object sender, EventArgs e)
{
    int mi = (from i in Program.db.Magazins where i.mag_name == comboBox3.Text select
i.mag_id).SingleOrDefault();
    int ki = (from i in Program.db.Kategoriyas where i.kat_name == comboBox4.Text select
i.kat_id).SingleOrDefault();
    d1 = "select t.tovar_name as 'Наименование', t.tovar_sum as 'Стоимость', t.tovar_kol as
'Количество' from tovary t where t.tovar_kol != 0 and t.mag_id = " + mi + " and t.kat_id = " + ki +
""";
    connect4();
}

private void button12_Click(object sender, EventArgs e)
{
    if (dataGridView3.Rows.Count == 0)
    {
    }
    else
}
```

Ә ҚОСЫМШАСЫНЫҢ СОҢЫ

```
{
    try
    {
        string myDocsPath =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);

        SaveFileDialog saveResults = new SaveFileDialog();
        saveResults.Filter = "Книга Excel (*.xls)|*.xlsx";
        saveResults.FilterIndex = 2;
        saveResults.RestoreDirectory = true;
        saveResults.InitialDirectory = myDocsPath;
        saveResults.FileName = comboBox2.Text + "(KCG)";
        saveResults.Title = "Экспорт результатов";

        Microsoft.Office.Interop.Excel.Application ExcelApp;
        ExcelApp = new Microsoft.Office.Interop.Excel.Application();
        ExcelApp.Application.Workbooks.Add(Type.Missing);
        ExcelApp.Columns.ColumnWidth = dataGridView3.Rows.Count + 3;

        ExcelApp.Cells[1, 1] = "Наименование";
        ExcelApp.Cells[1, 2] = "Цена";
        ExcelApp.Cells[1, 3] = "Количество проданных";
        ExcelApp.Cells[1, 4] = "Доход";
        ExcelApp.Cells[1, 5] = "Дата";

        ExcelApp.Cells[dataGridView3.Rows.Count + 3, 3] = "Общий доход";
        ExcelApp.Cells[dataGridView3.Rows.Count + 3, 4] = textBox11.Text;

        for (int i = 0; i < dataGridView3.ColumnCount; i++)
        {
            for (int j = 0; j < dataGridView3.RowCount; j++)
            {
                ExcelApp.Cells[j + 2, i + 1] = (dataGridView3[i, j].Value).ToString();
            }
        }
    }
}
```