

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТИ

Аппараттық жүйелер

кафедрасы

«Қорғауға жіберілді»

Кафедра меңгерушісі

Исмаишев М.И., Т.ғ.к., доцент
(аты-жөні, ғылыми дәрежесі, атағы)

« »

20/14 ж.

(колы)

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: Кондитерлік саласындағы жонашекерлік
жүйе динамикасының моделі іске асыратын
вазделаша камплексінің өңделу шығару

53070300- Аппараттық жүйелер

мамандығы бойынша

Орындаған

Дамхан Таужар

(аты-жөні)

ИЖ-10-1

(тобы)

Жетекші

Табубаев С.С.,

(аты-жөні, ғылыми дәрежесі, атағы)

Т.ғ.к., доцент

Кеңесшілер :

Экономикалық бөлім бойынша :

Э.С. Қасымов Б. Базенов
(ғылыми дәрежесі, атағы, аты-жөні)

Колтықов А. А. 20/14 ж.
(колы)

Өмір тіршілігі қауіпсіздігі бойынша:

Ала Оқотушев

(ғылыми дәрежесі, атағы, аты-жөні)

Табубаев С.С.
(колы)

« 20 » 05 20/14 ж.

Есептеу техникасын қолдану бойынша :

Қауіпбаева Б.К., Т.ғ.к., ала оқотушев

(ғылыми дәрежесі, атағы, аты-жөні)

Қауіпбаева Б.К.
(колы)

« 03 » 06 20/14 ж.

Мөлшер бақылаушы:

пер. Ғылы. маманы: ала оқотушев Аманжолбаева К.Б.

(ғылыми дәрежесі, атағы, аты-жөні)

Аманжолбаева К.Б.
(колы)

« 16 » 08 20/14 ж.

Пікір жазушы :

(ғылыми дәрежесі, атағы, аты-жөні)

« » 20 ж.

(колы)

Алматы 2014 ж.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТИ

Аппараттық технологиялар факультеті
Аппараттық жүйелер мамандығы
Аппараттық жүйелер кафедрасы

жобаны орындауға берілген

ТАПСЫРМА

Студент Бамаған Раухар (аты - жөні)

Жоба тақырыбы Кондитерлік салалындағы экономикалық жүйе динамикасының моделін іске асыратын базарлаша қалыпкеін өндір шығару
ректордың « 24 » қыркүйек 2013 № 115 бұйрығы бойынша бекітілген.

Аяқталған жұмысты тапсыру мерзімі: « 09 » маусым 20 14 ж.

Жобаға бастапқы деректер (талап етілетін жоба нәтижелерінің параметрлері және нысанның бастапқы деректері)

Инвестициялық саясаттың экономикалық жүйеде әсерінің бастапқы талдау модельдері мен әдістерін өңдеуге арналған экономикалық жүйенің дамуының бар бағыт модельдері мен әдістеріне сипаттама берілді. Экономикалық жүйеде негізінде кондитерлік салалындағы экономикалық жүйе динамикасының моделін іске асыратын базарлаша қалыпкеін өндір шығарылған.

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

Инвестициялық саясаттың экономикалық жүйеде әсер етуін бағалауға мүмкіндік беретін, айырмалық сәтталар мен ресурстық арақатынастар түрінде берілген модель өңделгенді. Өндірістің экономикалық жүйенің даму процесін талдаудың бар модельдеріне жеткізілген кезеңдеріне мазмұндық түсініктеме берілгенді. Экономикалық жүйенің дамуының экономикалық жүйенің қалыптасу даму бағытының математикалық тәсілі қарастырылған.

КЕСТЕСІ

№ р/с	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
1	Аналитикалық бағам Зерттеу объектісі „Бағам“ аккумулятивтік қозғалыс Кәсіпорын қозғалыс Өндірістің Экономикалық жүйенің даму процесін талдаудың бар модельдеріне және әдістеріне шығу Көзді қор мен еңбек ресурстарының динамикасын мөлшерлеу Бағам Өндірістің Экономикалық жүйенің динамикасының моделін өңдеу Мәселелер есептерге шығу тәсілдерін талдау		
2	Мобильдік бағам Кәсіпорын бағамы, сипаттамасы, көрсеткіш қозғалыс Мәліметтер базасын жасау Әріптестік Бағамнамаға тілін талдау жүйені визуалды модельдеу		
3	Экспериментальдық бағам Кейірақарға нұсқаулар Программистке нұсқаулар		
4	Өмір тіршілігі дәуірлері бағамы		
5	Экономикалық бағам Жұмыстың сипаттамасы мен қажеттілігін келіздесесі		

Тапсырманың берілген уақыты « 24 » қазан 20 13 ж.

Кафедра меңгерушісі

(қолы)

(аты-жөні, ғылыми дәрежесі, атағы)

Жоба жетекшісі

(қолы)

(аты-жөні, ғылыми дәрежесі, атағы)

Орындалатын тапсырманы қабылдаған студент

(қолы)

Баилхан Тауқар
(аты -жөні)

Аңдатпа

Берілген дипломдық жоба, инвестициялық саясаттың экономикалы жүйеге әсерінің бағасын талдау модельдері мен әдістерін өңдеуге арналады. Инвестициялық саясаттың экономикалық жүйеге әсер етуін бағалауға мүмкіндік беретін, айырымдық схемалар мен рекурренттік арақатынастар түрінде берілген модель өңделеді. Құрылған модель негізінде кондитерлік саласындағы экономикалық жүйе динамикасының модельін іске асыратын бағдарлама комплексі өңдеп шығарылды.. Бұдан басқа, дипломдық жобада есептеу техникасымен жұмыс істеу барысындағы еңбекті қорғау мәселесі және бағдарламаны өңдеудің мақсатқа сәйкестігінің экономикалық негізделу мысалы қарастырылады.

Аннотация

Данный дипломный проект посвящен разработке моделей и методов анализа оценки влияния инвестиционной политики на экономическую систему. Разработана модель, которая представлена в форме разностных схем и рекуррентных соотношении, позволяющие оценить влияние инвестиционной политики на экономическую систему. На основании разработана программный комплекс который моделирует динамику экономической системы в кондитерской сфере. Кроме того, в дипломном проекте рассмотрены вопросы охраны труда при работе с вычислительной техникой и пример экономического обоснования целесообразности разработки программы.

Annotation

This diploma project is about the real problems of the creation of information system of support of country economic system. A model that is presented in the form of peddling schemes and recurrence relations for evaluating the impact of the investment policy on the economic system . On the basis of elaboration software package that simulates the dynamics of the economic system in the confectionery sector. In addition, the capstone project addressed issues of safety when working with computer hardware and sample feasibility study of the development program..

Мазмұны

Кіріспе	8
1 Аналитикалық бөлім	9
1.1 Зерттеу объектісі. «Рахат» АҚ	9
1.2 Кәсіпорын қызметі	11
1.3 Өндірістегі экономикалық жүйенің даму процесін талдаудың бар модельдеріне және әдістеріне шолу	11
1.4 Өндірістегі экономикалық жүйенің даму процесінің негізгі факторларын табу және мағыналы баяндау	13
1.5 Негізгі қор мен еңбек ресурстарының динамикасын мағыналы баяндау	15
1.6 Өндірістегі экономикалық жүйе динамикасының моделін өңдеу	16
1.7 Өндірістегі экономикалық жүйенің негізгі қорының динамикасының модельдерін жетілдіру	24
1.8 Модельді есептерді шешу тәсілдерін таңдау	27
1.9 Құрылған модельдердің базасында экономикалық жүйенің қалыпты даму бағасының математикалық тәсілін жетілдір	30
1.10 Есептің қойылымы	33
2 Жобалық бөлім	35
2.1 Кәсіпорын бөлімі, сипаттамасы, көрсетілген қызмет	35
2.2 Мәліметтер базасын жобалау әдістемесі	38
2.3 Бағдарламалау тілін таңдау	43
2.4 Жүйені визуалды модельдеу	50
3 Эксперименталдық бөлім	57
3.1 Пайдаланушы нұсқаулығы	57
3.2 Программистке нұсқаулық	62
4 Ақпараттық қауіпсіздік	
4.1 Өндірістегі қауіпті және зиянды факторларды талдау	65
4.2 Есептеу бөлімі	67
5 Экономикалық бөлім	72
5.1 Жалпы жағдай	72
5.2 Экономикалық есептеулер	72
Қорытынды	80
Пайдаланылған әдебиеттер тізімі	81
А Қосымшасы	83

Кіріспе

Қазақстан Республикасының бәсекеге қабілетті 50 елдің қатарынан нық орын алу жөніндегі стратегиялық міндеттің орындалуы үшін әлемдік экономиканың өсімінен тұрақты түрде асып отыратын экономикалық даму қажет. Еліміздің жоғары деңгейде өзіндік жолы бар мемлекет ретінде танылуы, оның әлеуметтік-экономикалық үстемелеп дамуы, халықтың әлауқатын, тұрмыс жағдайын жақсартудың негізі болып табылады. Бұл үшін қаржылық және әлеуметтік тұрақтылықты сақтай отырып шаруашылық жүргізудің қолайлы жолдарын туғызу қажет, халықаралық еңбек бөлінісіне, ел экономикасының өсуіне тиімді жағдай орната отырып, белсене қатысу керек.

Экономикалық өсуді жеделдетудің және өндірістің тиімділігін арттырудың басты факторы - жоғары өнімді еңбекке уәждеме жасау, белсенді еңбекке экономикалық ынталандыру. Қазақстанда экономикалық реформаларды жедел жүргізу нәтижесінде өндірістің барлық факторы ырықтандырылды, тиімділікті көтеруге бағытталған ынталандыру уәждемелері де жүргізіліп келеді. Қазір, яғни әлеуметтік-экономиканың серпінді даму кезеңінде өндіріс тиімділігін экономикалық жағынан ынталандыру, еңбектің сандық және сапалық сипаттамаларын, макроэкономикалық ұдайы өндіріс процесіндегі ынталандырудың рөлін айқындау, еңбектің трансформациялануын жалпылау және нақтыланған түрде зерделеу өте маңызды сондықтан мәліметтер қоры және іс жүргізу өндірісі мен қол еңбегін тиімді ету үшін құрал ретінде пайдаланумен байланысты өзге де бағдарламалық қамтамасыз ету аса маңызды орын алады. Оны пайдалану нақты құжаттарды және өнеркәсіп жобаларын дайындауға қажетті уақытты қысқартуға, оларды жүзеге асыруда өндірістік емес шығындарды азайтуға мүмкіндіктер береді.

Микропроцессор базасында есептегіш техникаларының дамуы компьютерлер мен бағдарламалық қамтамасыз етуді жетілдіруге алып келеді. Көп ақпараттар ағынын өңдеуге қабілетті бағдарламалар пайда болады. Осындай бағдарламалардың көмегімен ақпараттық жүйелер құрылады. Кез-келген ақпараттық жүйенің мақсаты нысандар мен нақты әлемнің құбылыстары туралы мәліметтерді өңдеу мен олар туралы адамға қажетті ақпаратты ұсыну болып табылады.

Мәліметтер қорындағы потенциалды мүмкіндіктердің бәрін ашу үшін жұмыста алға қойған міндеттерге барынша сай келетін бағдарламалық және аппаратты құрал-жабдықтар кешенін пайдалану қажет.

Дипломдық жұмыстың өзектілігі. Қазіргі заманда кәсіпорынның инвестициалық шығындарын, қажетті шикізат көлемін, толық бюджет көлемін есептейтін, сақтайтын, мәліметке оңай қол жеткізетін және өзгертуге мүмкіншілік беретін заманға сай қажетті аса зор бағдарламма құрылады.

1 Аналитикалық бөлім

1.1 Зерттеу объектісі. «Рахат» АҚ

«Рахат» акционерлік қоғамы – 70 жылдық дара тарихы бар, Қазақстандағы кондитерлік өнім өндіретін ірі өндіріс орындарының бірі.

Кондитерлік өнімдер алғаш рет 1942 жылы ликер зауыты алаңында соғыс кезінде Мәскеу мен Харьковтан көшіріліп әкелінген жабдықтармен шығарылды.

«Рахат» АҚ-нда 3500-ге тарта адам жұмыс істейді. Өндірістік қуат Алматы және Шымкент қалаларындағы екі алаңда шоғырланған. Өндірістің жан-жақтылық сипаты, оның ауқымы мен какао бұршақтарын қайта өңдеу бойынша дербес желінің болуы компания үшін отандық өндірушілер арасында кондитерлік өнімдердің кең сұрыпталымын иеленуге мүмкіндік береді. Бүгінде «Рахат» АҚ-ның сұрыпталым қоржыны 10 түрлі топқа жататын алуан түрлі кондитерлік өнімнің 250-ден аса атауымен толыққан.

«Рахат» АҚ - сусамыр (диабет) ауруынан зардап шегетін адамдарға арнап арнай өнім шығаруды игерген Қазақстандағы бірден-бір кәсіпорын. Бұл сұрыпталымға қуаттылығы төмен әрі қантты тең дәрежеде табиғи шикізатпен ауыстыру арқылы шығарылған шоколад, кәмпит, печенье және вафлидің бірнеше атауы кіреді.

«Рахат» АҚ-ның қазақстандық басқа кондитерлік өнім өндірушілерден өзгешелігі – алуан түрлілігімен айрықша шоколад өнімдерін шығаруда басымдылық танытуы.

Кәсіпорында бірыңғай жауапкершілік пен сапаны бақылаудың кешенді жүйесіне негізделген ішкі сапалық жүйе ұйымдастырылып, іске асырылуда. Бұл жүйе өндірістің әр сатысында шикізат пен өнім сапасын бақылайды. Кәсіпорында ҚР ИСО 9001 - 2001 СТ стандартына сәйкес сертификатталған сапа менеджменті жүйесі енгізілген және іске қосылған.

«Рахат» АҚ-ның өнімдері отандық нарықта ғана емес, таяу және алыс шет елдер нарығында да танымал. «Рахат» АҚ өз өнімдерінің ішкі нарықта ең жоғары деңгейдегі қол жетімділік саясатын жүзеге асыруда. «Рахат» АҚ сатылым географиясы Қазақстанның бүкіл аумағын қамтиды, себебі компания 14 облыс орталығын қосқанда 18 қалада өнім тарату жүйесін жолға қойған. Алматы қаласы бойынша өнім тарату ірі көтерме саудамен сатып алушылар, сондай-ақ фирмалық бөлшек сауда дүкендері арқылы іске асырылады. Республиканың басқа өңірлерінде «Рахат» АҚ өнімдерінің қол жетімділігі фирмалық сапалы өнім сатумен айналысатын дистрибьюторлық желі (еншілес кәсіпорын) құрумен қамтамасыз етіледі. Қазіргі кезде «РАХАТ» АҚ өнімі Қазақстан нарығында ғана емес, республикадан шеткері аймақтарда – Ресей, Өзбекстан, Тәжікстан, Қырғызстан, Германия, Қытай, Ауғанстан аумағына да тараған. ТМД-ның басқа республикалары мен алыс шет елдер тарапынан «Рахат» АҚ өнімдеріне деген қызығушылықтың болуына орай, компания

келешекте өндіріс көлемін ұлғайтуды және экспорт көлемін арттыру жолымен өткізу нарығын кеңейтуді жоспарлауда.

«Рахат» АҚ мақсаты – кәсіпорынның дәмді өнімдерімен адамдарға қуаныш сыйлау. «Рахат» АҚ қызметі көне дәстүрді сақтай отырып, қазіргі заманның талабына сай барынша көп адамның қажеттілігін жоғары сапалы кондитер өнімдерімен жоғары деңгейде қанағаттандыруға бағытталған. Керемет кондитер өнімдерін қалай шығаруды біледі және сол мақсатта қорды арттырады. Қазақстан мен оның шет аймақтарындағы әр адамға өз еліміздің байлығы пен қонақжайлылығын бейнелей отырып, Қазақстандағы және Орталық Азия нарығындағы № 1 компания болуға ұмтылады.

Тұтынушыларға: Қазақстандық және шетелдік тұтынушылардың кең шоғырының қажеттілігін сапалы тауарлармен (көтерме, бөлшек) қанағаттандыру.

Акционерлерге: бизнестің тұрақты, ашық түрде дамуын қамтамасыз ету және табыс табу үшін компанияны бірлесіп және қаржылық басқарудың тиімді механизмін құру.

Қызметкерлерге: нақты қойылған міндеттер мен жауапкершілікті бөліп беру арқылы қызметкерлердің жұмыс тиімділігін арттыру. Қызметкерлердің алға қойған міндеттерін орындауы мен компанияның жұмыс нәтижесіне байланысты оларға ынталандыру жүйесін енгізу. Бірыңғай мәдениет қалыптастыру.

«Рахат» АҚ ИСО 9000 сериясының стандарттары негізінде сапа менеджменті жүйесі (СМЖ) әзірленіп, енгізілді. 2010 жылы сараптама және сертификаттау Ұлттық орталығы сапа менеджменті жүйесінің ҚР ИСО 9001-2009 СТ талаптарына сәйкестігін растаған қайта сертификаттау аудитін өткізді.

1.1.1 Сапа менеджменті жүйесі. Сапа менеджменті жүйесі – кәсіпорын процестерін басқару жүйесі. Қазіргі заманғы тұтынушы тауарға ғана емес, оны өндіретін компанияға да назар аударады. Ол тауарды таңдар кезде «Ұтымды ұйымдастырылып, дұрыс басқарылған компанияның өнімі де жақсы» деген қағиданы жиі ұстанады. Ісі жолға қойылып, өз қызметін жүзеге асыруда СМЖ:

- өнім сапасының тұрақтылығына;
- компаниядағы процестердің басқарылуына;
- компания қызметінің сапаны қамтамасыз етуге бағытталуына кепілдік береді.

Сапа менеджменті жүйесі тек өнім сапасына қатысты емес, ол кәсіпорын тіршілігінің барлық саласын қамтиды. Бұл - жабдықтар мен техниканы дұрыс пайдалану, өндіріс мәдениеті, кәсіпорынның әр жұмыскерінің өз міндетін сапалы орындауы, тұтынушының сұранысы мен пікірін ескеру, шикізатты кіргізу мен дайын өнімді шығару сапасын үнемі әрі үздіксіз бақылау.

1.1.2 Тарихи кезеңдер. 1942 жыл, наурыз көшіріліп әкелінген жабдықтар негізінде балалар үйлерін, госпитальдар, майданды қамтамасыз ету үшін Алматыда кондитер өнімдерін өндірудің басталуы.

1964 жыл шоколад және кәмпит цехтарын орналастыруға арналған жаңа өндіріс корпусының құрылысы.

1978-1979 жылдар цехтардың жұмыскерлеріне арналған асхана, дәрігерлік санитарлық бөлім, кеңейтілген тұрмыстық орынжайлары бар әкімшілік-тұрмыстық корпустың құрылысы.

1992 жыл мемлекеттік кәсіпорынның «Рахат» акционерлік қоғамы болып қайта құрылуы; «Рахат» сауда белгісінің тіркелуі.

2001 жыл Шымкент қаласындағы еншілес кәсіпорында өндірістің ұйымдастырылуы және алғашқы өнімнің өндірілуі.

2003-2009 жылдар № 3 корпусты қайта жабдықтау және игеру жұмыстары жүргізілді. Европалық сапа деңгейіндегі жоғары технологиялық өнімдер – салмасы бар шоколад кәмпиттер, жоғары сұрыпты карамельдер, «One shot» және басқа технологиялар бойынша шоколад кәмпиттер өндіру ісі жолға қойылды.

2007-2009 жылдар Кельн қаласындағы (Германия) ISM кондитер және бисквит өнімдерінің Халықаралық арнайы көрмесіне қатысуы.

2012 жыл Компания Алматы кондитер фабрикасының 70 жылдық мерейтойын және кәсіпорынның «Рахат» акционерлік қоғамы болып қалыптасуының 20 жылдығын атап өтті.

1.2 Кәсіпорын қызметі

Басты қызмет түрлері:

- тамақ өнеркәсібі;
- қантты кондитерлік өнімдердің сонымен қоса шоколад өнімдерінің жекелеген сауда;
- шоколад және шоколад кондитерлік өнімдерді өндіру;
- қанытталған жемістер, жаңғақтар, кептірілген жемістерден кондитерлік өнімдерді өндіру;
- сусамыр (диабет) ауруынан зардап шегетін адамдарға арналған
- өнімдерді өндіру;

1.3 Өндірістегі экономикалық жүйенің даму процесін талдаудың бар модельдеріне және әдістеріне шолу

Қазіргі кезеңде модельдердің көп түрі бар, оларда тұтыну мен өндіріс процестерінің әр алуан қарым - қатынастарының барлығы әртүрлі дәрежеде егжей-тегжейлі жазылған.

Өндірістің сызықтық модельдерінің көбісінің негізгі сала аралық баланс болып табылады. Осы әдістің шығу тарихына тоқталмай-ақ, 20-шы жылдары (кеңестік) отандық экономистердің еңбектеріне оның идеясы бірінші рет

айқын түрінде тұжырымдалған және сонан соң американ экономикасының құрылымын зерттеу бойынша В.В. Леонтьев еңбегінде дамыған.

В.В. Леонтьев өзінің моделінде халық шаруашылығының барлық өндірістік секторын n таза салаға бөліп тастады. «Таза» сала сөзі әрбір осы салалардың өнімі біркелкі болып ұйғарылатынын білдіреді. Таза сала нақты бар болуы міндетті емес, бір экономикалық абстракция. Сала аралық баланс схемасына тек таза салаларды енгізу, практикада жоспарлау мен есеп беру бар ұйымдастыру құрылымының шеңберінде жүзеге асырылатындықтан, оның тікелей қолданылуын сөз жоқ қиындатады. Бірақта осындай идеализация бір жағынан тарату мен қоғамдық өндірістің қалыптасқан технологиялық құрылымын бөлшектеп талдауға мүмкіндік беретіндіктен, ал екінші жағынан берілген жеңілдетілген схеманы зерттеген кезде жиналған тәжірибе, көбірек маңызды модельдерді құруға әкелгендіктен ақталды. Мысалы Нейман моделі.

Нейман моделі өзіне Леонтьев моделін жеке жағдай сияқты. Нейман моделінің Леонтьев моделінен айырмашылығы, әрбір технологиялық процесте (салада) бірнеше тауарлар түрлері бірлесіп өндірілуіне рұқсат етіледі. Осымен «таза» сала түсінігіне сүйене жасалған Леонтьев моделінің елеулі кемшілігі ойылады. Нейман моделінің толық мазмұндама теориясы мен егжей-тегжей библиографиясын табуға болады.

Нейман моделі сансыз зерттеулер туғызды. Осы зерттеулердің барысында бұл модельдің басты қасиеті матрица тіліндегі өндірістік сектордың спецификасымен байланысты емес екендігі анықталады. Басты рольді сызықтық және тұйықтық туралы жорамал атқарады. Сонымен бірге Нейман моделі осы бағыттағы методологиясы, айталық, сала аралық баланс модельдері сияқты жақсы өңдемегенімен практикалық қолданудың көзқарасымен көбірек перспективті болып саналады.

Осылайша Нейман моделі Леонтьев моделіне қарағанда алға елеулі қадам жасады. Сонымен бірге осы уақытқа дейін, осы модельдердің теориясында көңіл аударарлықтай нәтижелер баяғыша технологияның бір қалыптылығы туралы ұсынысында, оның уақыт бойынша өзгермейтіндігінен алынады. Қазіргі уақыттағы қарқынды технологиялық прогресте бұл кемшілік әрине маңызды. Егер уақытымен технологиялық тәсілдерінің өзгеруін алдын-ала болжамасақ, экономиканың дамуының ұтымды жолдары туралы маңызды нәтиже күту екіталай. Леонтьев пен Нейман модельдері көбінесе өндірістік құрылымды баяндайды және тек өте агрегирланған экономиканы баяндау үшін ғана жарамды.

Ал Вальрас моделі экономиканы толық дезагрегирланған түрде зерттейді. Тарихта Вальрас моделі экономикалық - математикалық модельдердің алғашқыларының бірі болып табылады, және оны өмірге әкелушінің арқасында қазіргі заманғы тым жақын формальді тілде жазылды. Бұл модель аса теориялық болғанына қарамай, бұған салынған идеялар экономикалық процестерді модельдеудің барлық аймақтарына пайдалы ықпалын тигізді және тигізуде. Кейбір авторлар, бұл модельде әрбір қатысушы тек жеке басының амандығына басшылық ете отырып, өзінің

әрекетін таңдауда толық еркіндігі болғандықтан, Вальрас моделін “децентрализациялаған” экономиканы баяндау үшін арналған деп есептейді.

Вальрастық схемасы экономикалық жүйенің әртүрлі модельдерінің прототипі болды, олардың әрқайсысы экономикалық шындықтың қандай да бір елін көбірек ынтамен зерттеуге бөледі.

Жалпы тепе - теңдіктің Вальрастық моделіне салынған күрделі идеялар оның дамуы мен жетілуіне кең мүмкіндік береді. Мысалы, жалпы тепе - теңдік моделін В.Л. Макаров зерттеген. Макаров моделінде экономикалық жүйенің қатысушылары тұтынушы мен өндірушіге бөлінбейді - олардың әрқайсысында екі қасиеттерде болуы мүмкін.

Вальрас моделінен айырмашылығы, мұнда біріншіден, қатысушының мүддесі осы жағдайда алатынына ғана емес, сонымен қоса оның өзінің қандай стратегияны таңдайтынына тәуелді. Екіншіден, бейнелеу сол қатысушысының жағдайына ғана емес, басқа экономикалық агенттердің жағдайына тәуелді. Басқаша сөзбен айтқанда, экономикалық жүйенің әрбір қатысушысы “басқаларға қарап” әрекет етеді.

70-ші жылдардың басындағы экономикалық дамудың қарқынының төмендегені және іскерлік белсенділіктің тоқырап қалуы, экономикалық жүйенің тағдырын ойлауға себеп болады. 1971 жылы шыққан Форрестердің “Әлемдік динамика” кітабы алаңдатушылықтың бейнелеуі болады. Д. Форрестер тәсілі болашақты болжауға мүмкіндік береді деген көпшілік пікір туды. Бірақ бұл иллюзия болып шықты. Форрестер моделінде экономикалық құбылыстың айқын талдауы тіптен жоқ. Құрылған модель осы көрнектілікті имитациялайды. Экономикалық жүйеде болып жатқан негізгі процестердің барабар математикалық баяндау туралы сұрақ тіптен қойылмады.

Менім ойымша, Форрестер моделінің мазмұнын талдау, барлық экономикалық процестердің бастапқы микробаяндауынан экономикалық макроара қатыстарды алу маңызды[1].

1.4 Өндірістегі экономикалық жүйенің даму процесінің негізгі факторларын табу және мағыналы баяндау

1.4.1. Сала. Жүйеде n түрлі товарлар өндіріледі. Олар өндіріс құралы мен тұтыну бұйымы сияқты қолданылады. Өндіріс құралы еңбек бұйымы мен еңбек құралына бөлінеді. Еңбек құралы (машиналар, құрал - жабдықтар, т.б) бірен - сарандап қана істен шығады, ал еңбек бұйымдары (шикізат, отын, т.б) толық тұтынылады.

Саланың функциялары көп жоспарлы. Алдымен ол ағымдағы өндірісті іске асырады. Өндірістің нақты көлемі қолдағы бар өндіріс құралдары мен жоспарланған көлемге тәуелді. Салалық жоспарлау механизмі болашақта өндіріс көлемін ұлғайтудан тұрады, егер қолдағы ақшалай қаржының көлемі ағымдағы баға бойынша қосымша өндіріс құралдарын сатып алуға мүмкіншілігі болса.

Сала жұмысшылар жолдайды, оларға еңбекақы төлейді, нарықта товарлар сатады, банктан қарыз ала - алады немесе қарыз төлей алады. Егер саланың жалпы пайдасының мөлшері қарыздың процентінен аз болса, онда сала қарыздан өзі бас тартады, саланың жалпы пайдасы саудадан түскен пайда мен іске асырылған өнімнің өндірісінің шығындарының арасындағы айырмашылықтан анықталады. Осы түскен пайдадан сала қарыз капиталы бойынша процент төлейді. Қалған пайда салыққа салынады. Салықты төлегеннен кейін, саланың пайдасында таза пайда қалады. Таза пайда саланың меншікті капиталын толтырады. Егер қандайда бір сала ауыр финанстық жағдайға душар болса, яғни берілген периодта ағымдағы ақшалай қаражатын жұмсауды тоқтатпау үшін, алдағы периодтың аяғындағы қолдағы ақшалай қаржы және алынған банктік қарыз жетіспесе, онда мемлекет ақша түріндегі дотациямен көмек көрсетеді. Жүйенің қалыпты жұмыс істеген кезінде дотация механизмі әрекет жасамайды.

1.4.2. Тұрғын халық. Модельдегі барлық қоғам еңбекшілерден тұрады. Берілген жұмыста, біз еңбекшілерді әрқашан өндіріс қажеті үшін жеткілікті деп есептейміз. Сондықтан еңбекшілердің жалпы санына ерекше көңіл аудармаймыз. Сонымен қоса жұмыс басты емес жұмысшылар бірдей еңбекақы алады. Жұмыс күшінің артық болған жағдайында еңбекақы мөлшері, ағымдағы баға бойынша тіршілікке қажетті құралдармен қамтамасыз етуге мүмкіндік беретіндей, әр кезеңде минималды деңгейде белгіленіледі. Еңбекшілер жағынан товарға нақты сұраным құрылымы тұрақты және солтіршілікке қажетті құралдардың жиынтығымен баяндалады деп болжаймыз. Еңбекшілердің табыстары табыс салығын шегергендегі жұмыс ақыларынан, акцияға дивиденттерден, банк салымдары бойынша проценттерден, трансферттік төлемдерден, т.б. жинақталады.

Аз дәрежеде қамтылған халықтың бөлігі табысты трансферттік төлемдер және акцияға дивиденттер түрінде алады. Алынған табыс келесі түрде үйлестірілуі мүмкін: бір бөлігі тұтынуға шығындалса, қалған бөлігі жинақ ақша банкісіне жатқызылады.

1.4.3. Мемлекет. Мемлекеттің табысы өндіріс пайдасының салығынан тұрады. Шығынның статьялары болып мемлекеттік сауда жасау, минималды мүмкін ақша қаражатын жұмсауды жүзеге асыру үшін қажетті салаларға дотация және ақшалай қаржыны босату шамасы бойынша мемлекеттік борышты өтеу болып табылады. Мемлекет жағынан тауарға сұраным құрылымы бір қалыпты деп есептеледі.

1.4.4. Банк. Банқтың активтері мемлекетке (мемлекеттік борыш) және салаларға қарыз беруден тұрады, ал пассивтері - банқтың меншікті капиталының айналымындағы ақша массасынан және тұрғын халықтың салымынан тұрады. Уақыттың әрбір периодында банқтың активтері мен пассивтерінің теңдігі сақталады. Сонымен қоса тағы бір жалпы жүйелік баланс орындалады: шығарылған ақша белгілері (банқтың пассивтерінің статьясы сияқты) банктан тыс айналымдағы (өндірістің қолдағы бар ақша

саласы, халық пен мемлекеттің ақшалай қаражаты, тұтыну үшін бөлінген) ақша массасына әрқашанда тең.

Банк саларға қарыз береді. Саланың несие капиталы қалауынша көп болмайды, егер сапаның активтеріне несие капиталының қарым - қатынасы белгілі бір деңгейден асса, онда қарыз берілмейді. Әрбір периодта саладағы бар қарызды бөліп - бөліп өтеуге болады. Сала үнемі берілген мөлшерге сәйкес қарыздың процентін жүйелі түрде төлеп тұрады. Процент төленетін салымды тек еңбекшілер жасайды.

Банктың пайдасы кіріс пен төлеу проценттерінің арасындағы айырмашылық сияқты құралады. Ол банкта, оның меншікті қорын өсіруге қалады[2].

1.5. Негізгі қор мен еңбек ресурстарының динамикасын мағыналы баяндау

Өндірістің объектісінің жағдайы оның негізгі қорының көлемімен сипатталады. Негізгі қор немесе капитал өзіне барлық өндірістік құралдарды кіргізеді, яғни аспаптардың, машиналардың, қойма үйлерінің, транспорт құралдарының барлық түрі. Негізгі қордың өзгеруі эксплуатация кезінде оның құқықтық және рухани тозуы есебімен болады, және де негізгі қорды бұрынғы қалпына келтіру қажеттілігі туады, міне сол үшін белгілі бір амортизациялық жарналар өндіріледі. Осы амортизациялық жарналар капиталының тұтынуына жарналар деп аталады. Олар өндіріс процесінде тұтынылатын, инвестициялық тауарларды сатып алудың қайнар көзі болып саналады.

Егер инвестиция амортизациядан асса, кәсіпорынның экономикасы өрлейді.

Фирманың экономикасының тоқырауы инвестиция мен амортизация тең болған жағдайда болады.

Инвестиция амортизациядан кем болған жағдайда экономиканың тоқырауын тудырады.

Осындайда негізгі қордың динамикасының математикалық модельдері экономикалық даму процесінің стратегиясына ғылыми - негізгі анықтамасынан нақты көмек көрсете алады.

Өндірістік емес сферада дүниеге келу мен өлімнің нәтижесінде халық саны өзгереді, еңбек ресурстарын ұдайы өндіру процестері, қоғам мүшелерінің элеуметтік - статусының өзгеруінің процестері, соның салдарынан қоғамның құрылымын өзгертеді, және де өндіріс технологиясын жетілдіру үшін білімді жинау процестері болып жатады. Еңбек ресурстарын ұдайы өндіру және білімді жинау процестерін сүйемелдеу, сәйкес мамандыққа еңбек ресурстарын жұмылдыруды және тұтынылатын өнедірістік өнімге қаражат қажет етеді. Қоғам екі класстан тұрады: меншік иелері мен еңбекшілер класы. Меншік иелері табиғи ресурстар мен өндіріс құралдарын иеленеді. Меншік иелері өздерінің табыстарын капитал мен табиғи ресурстардан алады. Еңбекшілер класы екі элеуметтік топқа бөлінеді:

өндірісте жұмыс бастылар және білім жинау сферасында жұмыс бастылар. Еңбекшілердің табысының қайнар көзі еңбекақы.

Еңбекшілер бос уақыттың саны мен алған тұтынушылық игілігінің саны арасындағы өздері үшін ең қолайлы арақатынасқа жетуге ұмтылады. Бұл өндіріске еңбек ресурстарын ұсыну мен өнім нарығындағы еңбекшілердің сұранымын анықтайды. Білім жинау сферасында жұмыс бастылар мемлекетке еңбек ресурстарын ұсынады. Мемлекет білім жинауда жұмыс бастылардың еңбекақы мөлшерін белгілеп, ғылыми - техникалық прогресті басқарады. Мемлекет еңбек нарығындағы өзінің сұранымының шамасымен нарықтық конъюнктураға (жағдаят) әсерін тигізеді. Барлық нарықтар осылайша бір-бірімен байланысқан болады. Нарықта сұраным мен ұсыным қарым - қатынасы, өндірісте жұмыс бастылардың еңбекақысын, капиталға проценттік нормасын, табиғи ресурстар мен өнімге баға белгілейді. Осы көрсеткіштер арқылы өнімдерді шығару, шаруашылықта қолданылатын табиғи және еңбек ресурстарының саны, өндірілген өнімдердің тұтыну үлесі мен жинау үлесі, әлеуметтік топтардың табысы және олардың тұтыну деңгейінен анықталады[3].

1.6. Өндірістегі экономикалық жүйенің динамикасының моделін өндеу

Тұйықталған экономикалық жүйе қарастырылады. Бұл жүйе өздігінше әрекет ететін қатысушылар жиынтығы ретінде ұсынылады. Қатысушылар болып есептелінетіндер: өндіріс, халық, банк, мемлекет. Бұлардың арасындағы ара - қатынас - бұл нарық.

1.6.1. Өндіріс. Экономикалық жүйенің n - саладан тұратын өндірісін қарастырайық. Олардың әрқайсысы бір түрлі өнім шығарады. Саланың нөмірі және сәйкесінше осы саланың параметрлерін 1-индекісімен белгілейік, ол 1-ден n -ге дейін өзгереді.

Әр уақыт k ($k=0,1,\dots$) периодындағы экономикалық жүйенің жағдайы модель айнымалыларымен сипатталады. Олар өндіріс процесін сипаттайтын T ($T=, T=t_{k+1} - t_k, T=const, k=t/T$) периодымен төменде келтірілген рекуренттік қатынастар жүйесіне сәйкес динамикалық өзгереді.

Өнім шығару:

$$y_e(k) = \gamma(x_e(k))^{\alpha_e} (L_e(k))^{\beta_e} \quad (1.1)$$

мұндағы, $x_e(k)$ - ақша көлеміндегі негізгі қор көлемі;

j -1- ші өнімді шығаратын кәсіпорынның индексі;

$L_e(k)$ - еңбекшілер саны;

$\alpha_e, \beta_e, \gamma_e$ - өндірістік функция параметрлері.

Негізгі қор динамикасы:

$$x_e(k+1) = x_e(k) + \Phi_e^I(k) - \mu_e x_e(k) \quad (1.2)$$

мұндағы, μ_e - негізгі қордың істен шығуының коэффициенті;

$$\Phi_e^I(k) = \sum_{j=1}^n b_{je} P_j(k) * \Delta y_e(k+1) -$$

өндірісті кеңейтуге бөлінген ақшалай қаражат (инвестициялар);

b_{je} - қор сымдылығының коэффициенті;

$\Delta y_e(k+1) - (k+1) -$ периодындағы өнім өндірудің өсуінің күтімі.

Табыс сметасы.

Жалпы пайда: мұндағы,

$$\Pi_e(k) = D_e(k) - R_e(k) \quad (1.3)$$

мұндағы, $D_e(k) = y_e(k) * P_e(k)$ - сатудан алынған ақша түсімі (выручка);

$P_e(k)$ - l - түріндегі өнімнің бағалары;

$R_e(k) = \Phi_e^A(k) + R_e^T(k) + R_e^\Phi(k)$ - өндіріске кететін шығындар мөлшері;

$\Phi_e^A(k) = (\sum_{j=1}^n P_j a_{je}) y_e(k)$ - шикізат сатып алуға кеткен шығындар;

a_{je} - материал сымдылығының коэффициенті;

$R_e^T(k) = W_e(k) * L_e(k)$ - еңбек ресурстарын жалдауға кеткен шығындар;

$W_e(k)$ - жұмыс ақы деңгейі.

$R_e^\Phi(k) = \gamma R_e^T(k)$ әлеуметтік сақтандыру қорына аударым;

γ - әлеуметтік сақтандыру коэффициенті.

l - ші саланың таза пайдасы:

$$\Pi_e^r(k) = \Pi_e(k) - N_e(k) \quad (1.4)$$

мұндағы, $N_e(k) = \eta_e \max(\theta \Pi_e^h(k))$ - пайдаға салынатын салық;

$\Pi_e^h(k) = \Pi_e(k) R_e^k(k)$ - салықтан түсетін пайда;

$R_e^k(k) = r_e^k(k) \Phi_e^k(k)$ - қарыз капиталы бойынша төлемдер;

($k+1$) периодындағы несие капиталының мөлшері:

$$\Phi_e^k(k+1) = \Phi_e^k(k) - R_e^k(k) + (1 - \pi) * \Delta \Phi_e^k(k) \quad (1.5)$$

мұндағы, $\Delta \Phi_e^k(k) = \Pi_e(k) * \frac{1 - \eta_e}{1 + r_e^k}$ - несие капиталының мүмкін болатын өсуі (өсімше);

π -қайтарылатын несиенің үлесі;
 η_e - пайдаға салынатын салықтың нормасы;
 r_e^k - несие процентінің мөлшерлемесі;

($k+1$) периодындағы өнімдерді өндірудің күтілетін өсімі:

$$\Delta y_e(k+1) = -\frac{\varphi_e^i(k)}{\sum_{j=1}^n b_{je} P_j(k) + a_e W_e(k)} \quad (1.6)$$

мұндағы,

$$\varphi_e^1(k) = (1 - \pi) \Delta \Phi_e^{k_1}(k) + \Pi_e^{r_1}(k) + G_e^{D_1}(k);$$

$$\varphi_e^2(k) = (1 - \pi) \Delta \Phi_e^{k_2}(k) + \Pi_e^{r_2}(k) + R_{Ae}(k);$$

$$\varphi_e^3(k) = (1 - \pi) \Delta \Phi_e^{k_3}(k) + \Pi_e^{r_3}(k);$$

сәйкесінше l -ші саланың мемлекеттік меншік, акционерлік меншік, жеке меншік формасындағы өндірісінің ақшалай қаражаттары;

a_e - еңбек сыйымдылығының коэффициенті;

$R_{Ae}(k) = (1 - \int_e^1 \Pi_e^{r_2}) \max[0, \Pi_e^{r_2}(k)]$ - акционерлік қоғамның дивиденд түріндегі төлемдер мөлшері;

$\int \Pi_e^{r_2} - l$ - ші саланың қорлануға иілгіштігінің функциясы;

$\Pi_e^{r_2}(k) = \Pi_e^2(k) - N_e^2(k) - R_{Ae}(k)$; - дивидендтерді төлегеннен кейінгі акционерлік қоғамның таза пайдасы;

$G_e^{D_1}(k) = \max[0, (\xi_e - 1) * \Pi_e^{r_1}(k)]$ - мемлекеттік мөлшері;

t - дотация реттегіші.

($k+1$) - периодындағы өнім шығарудың күтілетін көлемі:

$$y_e(k+1) = y_e + \Delta y_e(k+1).$$

1.6.2. Тұрғын халық. Модельде тұрғын халық 4-топқа бөлінеді: меншіктенушілер (жеке меншік кәсіпорын иелері), өндірістік сфераның еңбекшілері, аз дәрежеде қамтылған топ, олар: зейнеткерлер, студенттер, жұмыссыздар. Бұл топтардың табыстары мен шығындары әртүрлі.

Еңбекшілердің табыстары табыс салығын шегергендегі жұмыс ақыларынан, акцияға дивидендтерден, банк салымдары бойынша проценттерден, трансферттік төлемдерден, т.б. жинақталады.

Аз дәрежеде қамтылған халықтың бөлігі табысты трансферттік төлемдер және акцияға дивидендтер түрінде алады. Алынған табыс келесі түрде үлестірілуі мүмкін: бір бөлігі тұтынуға шығындалса, қалған бөлігі жинақ ақша банкісіне жатқызылады.

Тұрғын халық құрылымының динамикасы:

$$\bar{N}(m+1) = B\bar{N}(m) \quad (1.7)$$

мұндағы, $\bar{N}(m) = (N_1(m), N_2(m), \dots, N_n(m))$ өспелі топ бойынша тұрғын халық санының векторы;
 $t_m = t_0 + mT, m = 0, 1, 2, \dots, n; t_0 = 0, T = t_{m+1} - t_m$ - өспелі қабат ені;

$$B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \dots & 0 & b_{1n} \\ b_{21} & b_{22} & 0 \dots & 0 & 0 \\ 0 & b_{32} & b_{33} \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 \dots & b_{n,n-1} & b_n \end{pmatrix} \quad (1.8)$$

мұндағы, $b_{12}, b_{13}, \dots, b_{1n}$ - өспелі топтардағы туылу қарқындылығы;
 $b_{jj} (j = 1, 2, \dots, n)$ - иммиграциялық қарқындылығы;
 $b_{j+1, j} = 1 - \mu_j - m_j$ және m_j - сәйкесінше өлім мен иммиграция қарқындылығы.

Тұрғын халықтың еңбекке қабілетті бөлігі:

$$N^T(m) = \sum_{j=\tau_{min}}^{\tau_{max}} \Omega(j, 3) N_j(m), \quad (1.9)$$

τ_{max} және τ_{min} - ең кіші және ең үлкен еңбекке қабілетті жас. Еңбек ресурстарын ұсыну:

$$N_\zeta(k) = K^T N^T(k) \quad (1.10)$$

l -ші тауарға тұрған халықтың төлем қабілеті бар сұранымы:

а) Меншіктенушілер

$$\omega_e^0 = (k+1) = \frac{C_e^0}{\sum_{e=1}^n C_e^0 P_e(k)} * \Phi^0(k) \quad (1.11)$$

мұндағы, $\Phi^0(k) = \alpha^0 \sum_{e=1}^n P_e^{r^3}(k)$ — тұтыну шығындары;

ә) Банктік салымдар

$$H^0(k+1) = H^0(k) + (1 - \alpha^0) \sum_{e=1}^n P_e^{r^3}(k) \quad (1.12)$$

мұндағы, $\alpha_i^i (i = 0, 1, 2, 3)$ — тұтынуға иілгіштігінің коэффициенті;

$\bar{C}^i = (C_1^i, C_2^i, \dots, C_n^i), (i = 0, 1, 2, 3)$ - сұраным құрылымы;

i - халық тобының нөмері.

б) Өндірістік сфераның еңбекшілері.

$$\omega_e^1(k+1) = \frac{C_e^1}{\sum_{e=1}^n C_e^1 P_e(k)} * \Phi^1(k) \quad (1.13)$$

мұндағы, $\Phi^1(k) = \alpha^1 T(k)$ – тұтыну шығындары;

$$T(k) = V^1(k) + R_A^1(k) + D^1(k) + V_0^1(k) - N_H^1(k) -$$

табыстар;

$$V^1(k) = \sum_{e=1}^n \sum_{i=1}^3 W_e^i(k) * L_e^i(k) - \text{еңбекақы мөлшері};$$

$$R_A^1(k) = K_A^1 \sum_{e=1}^n R_{Ae}(k) - \text{акция дивидендтері};$$

K_A^1 – еңбекшілер қолындағы Акционерлік Қоғамның (АК) акцияға үлесі;

$$V_0^1(k) \omega V^1(k) - \text{трансферттік төлемдер};$$

ω - реттегіш;

$D^1(k) = r_B H^1$ – банктік салымдар бойынша проценттік төлемдер;

r_B – банктік салымдарға проценттер;

$$H^1(k+1) = H^1(k) + (1 - \alpha^1)T(k), \quad (1.14)$$

Табыс салығы

$$N_H^1(k) = \eta(V^1(k) + R_A^1(k) + D^1(k)), \quad (1.15)$$

мұндағы, η - табыс салығының проценті.

в) Өндірістік емес сфераның еңбекшілері.

$$\omega_e^2 = \frac{C_e^2}{\sum_{e=1}^n C_e^2 P_e(k)} * \Phi^2(k) \quad (1.16)$$

мұндағы, $\Phi^2(k) = \alpha^2 D^2(k)$ – тұтыну шығындары;

$$D^2(k) = V^2(k) + R_A^2(k) + D^2(k) + V_0^2(k) - N_H^2(k) -$$

табыстар;

$$V^2(k) = W^2(k)L^2(k) - \text{жұмыс ақы мөлшері};$$

$w^2(k)$ – жұмыс ақы деңгейі;

$L^2(k)$ еңбекшілер саны;

$D_1^2(k) = r_B H^2(k)$ – банктік салымдар бойынша проценттік төлемдер;

$$R_A^2(k) = K_A^2 \sum_{e=1}^n R_{Ae}(k) - \text{акцияға дивидендтер};$$

$$V_0^2(k) = \omega V^2(k) - \text{трансферттік төлемдер};$$

(k+1) периодындағы банктік салымдар:

$$H^2(k+1) = H^2(k) + (1 - \alpha^2)D_1^2(k), \quad (1.17)$$

Табыс салығы

$$N_H^2(k) - \eta \left(V^2(k) + R_A^2(k) + D_1^2(k) \right), \quad (1.18)$$

г) Тұрғын халықтың аз дәрежеде қамтылған тобы

$$\omega_e^3(k+1) = \frac{C_e^3}{\sum_{e=1}^n C_e^3 P_e(k)} * \Phi^3(k), \quad (19)$$

мұндағы, $\Phi^3(k) = D^3(k)$ – тұтыну шығындары;

$D^3(k) = V^3(k) + R_A^3(k) + V_0^3(k) - N_H^3(k)$ – табыстар;

$V^3(k) = W^3(k)L^3(k)$ – жұмыс ақы мөлшері;

$w^2(k)$ – жұмыс ақы деңгейі;

$L^2(k)$ – адам саны;

$V_0^2(k) = \omega V^3(k)$ – трансферттік төлемдер;

$R_A^3(k) = K_A^3 \sum_{e=1}^n R_{Ae}(k)$ – АҚ акциясына дивидендтер мөлшері;

K_A^3 – аз қамтылған топтың қолындағы АҚ акцияларының үлесі;

Табыс салығы

$$N_H^3(k) - \eta R_A^3(k), \quad (20).$$

1.6.3. Банк. Банктің негізгі функциялары: салым қабылдау және қарыз беру. Бұдан басқа банк халық салымдары бойынша процент төлейді, қарыз бойынша процент алады және өзінің тұтыну шығынына белгілі мөлшерде қаржы бөледі.

Нарықтағы төлемдер ағыны (шығындар)

$$\Phi(k) = \Phi^A(k) + \Phi^1(k) + \Phi^0(k) + \Phi^1(k) + \Phi^2(k) + \Phi^3(k) + \Phi^B(k) + \Delta G^3(k), \quad (21),$$

мұндағы, $\Delta G^3(k)$ - мемлекеттік сатып алуларға кететін шығын;

$\Phi^B(k)$ – банк шығыны;

$\Phi^\Phi(l) = \sum_{e=1}^n \sum_{i=1}^3 \Phi_e^{Ai}(k)$ – шикізат сатып алуға кететін шығын;

$\Phi^1(l) = \sum_{e=1}^n \sum_{i=1}^3 \Phi_e^{li}(k)$ – негізгі қорларды сатып алуға кететін шығын;

Өндірістің банктік шотесептері:

$$D^f(k) = \Phi(k) + \Delta \Phi^K(k) - \Phi^A(k) - \Phi^1(k) - R^\Phi(k) - R^K(k) - N(k) - R_A(k) + R^A(k), \quad (22),$$

мұндағы, $\Delta\Phi^k(k) = \sum_{e=1}^n \sum_{i=1}^3 \Delta\Phi_e^{ki}(k)$ – несие капиталының мөлшерінің мүмкінше көбеюі;
 $R^\Phi(k) = \sum_{e=1}^n \sum_{i=1}^3 R_e^{\Phi i}(k)$ – әлеуметтік сақтандыру қорына аударым (жарна);
 $R^k(k) = \sum_{e=1}^n \sum_{i=1}^3 R_e^{ki}(k)$ – қарызға берілген капитал бойынша төлемдер мөлшері;
 $N(k) = \sum_{e=1}^n \sum_{i=1}^3 N_e^i(k)$ – кәсіпорын пайдасына қойылатын салық мөлшері;
 $R_A(k) = \sum_{e=1}^n R_{Ae}(k)$ – экономикада барлық АҚ дивидендтерінің төлем мөлшері;
 $R^A(k) = \sum_{e=1}^n \sum_{i=1}^3 R_e^{Ai}(k)$ – өндірістің АҚ-нан алатын дивидендтерінің мөлшері.

Банктің жалпы пайдасы:

$$\Delta D^B = R^k(k) - \Delta\Phi^k(k) - r_B^f D^f(k) - r_B H^H(k), \quad (23),$$

мұндағы, $H^H(k) = H^0(k) + H^1(k) + H^2(k) + H^3(k)$ – тұрғын халықтың салымдары;
 r_B^f – өндірістік салым бойынша процент.

Таза пайда:

$$D_r^B(k) = \Delta D^B - N_0(k) - \Phi^B(k), \quad (24),$$

мұндағы, $N_0(k) = \eta_0 \max[0, \Delta D^B(k)]$ – пайдаға салынатын салық.
 Банктік жүйенің l -ші өнімге сұранымы:

$$\omega_e^B(k+1) = \frac{c_e^B}{\sum_{e=1}^n c_e^B P_e(k)} * \Phi^B(k), \quad (25),$$

мұндағы, $\vec{C}^B = (C_1^B, C_2^B, \dots, C_n^B)$ – банк сұранымының құрылымы;
 η_0 – банк пайдасына салынатын салықтың проценті.
 (к+1) периодында ақшаның айналымдағы массасы:

$$M_0(k+1) = \Phi_0(k) + \theta\Phi(k) + E(k), \quad (26),$$

мұндағы, θ – ақша айналымының жылдамдығы $\theta < 1$;
 $\Phi_0(k) = \xi H^H(k)$ – резервтік капитал көлемі;
 ξ – салымдарды резервілеу нормасы, $0 < \xi < 1$;
 $E(k) = K_3 \sum_{e=1}^n \sum_{i=1}^3 P_e(k) y_e^i(k)$ – эмиссиясы;
 K_3 – ақша эмиссиясының нормасы;

1.6.4. Мемлекет. Әр периодтағы мемлекет табысы:

$$\Delta D(k) = N(k) + N_H(k) + N_0(k) + \Delta D_G^A(k) + D\Pi_G^A(k) + \Delta D^G(k) + 3\Pi(k), \quad (27),$$

мұндағы, $\Delta D_G^A(k)$ – мемлекет алатын АҚ акцияларына дивидендтер;
 $D\Pi_G^A(k)$ – мемлекеттік акция пакеттерін сатудан келген түсімдер;
 $D^G(k)$ – мемлекеттік меншікті сатудан келген түсімдер;
 $3\Pi(k)$ - қарыз және т.б табыстар.

Шығындар:

$$\Delta G^P(k) = \sum_{i=1}^3 V_0^i(k) + \Delta G^3(k) + \Delta G^c(k) + \Delta G^D(k) + \Delta G^\Pi(k), \quad (28),$$

мұндағы, $V_0^i(k)$ – тұрғын халықтар тобы бойынша трансферттік төлемдер;
 $\Delta G^c(k)$ – ғылымға, мәдениетке, білімге кететін шығындар;
 $\Delta G^D(k)$ – мемлекеттік дотациялар;
 $\Delta G^\Pi(k)$ – басқа да шығындар;

Мемлекеттік сатып алуларға кететін шығындар:

$$\Delta G^3(k) = \Delta D(k) - \Delta G^c(k) - \Delta G^D(k) - \Delta G^\Pi(k), \quad (29),$$

Мемлекеттің $(k+1)$ периодында l -ші өнімге деген сұранымы:

$$\omega_e^G(k+1) = \frac{C_e^G}{\sum_{e=1}^n C_e^G P_e(k)} * \Delta G^3(k), \quad (30),$$

мұндағы, $\vec{C}^G = (C_1^G, C_2^G, \dots, C_n^G)$ – мемлекет сұранымының құрылымы.

1.6.5 Нарық. Еңбек нарығы

$$\begin{cases} L_e^i(k+1) = L_e^i(k) + kL_e^i(k) \left(1 - \frac{L_e^i(k) \sum_{e=0}^n \sum_{i=1}^3 W_e^i(k)}{N_s(k) W_e^i(k)}\right) \\ L_e^i(k+1) \leq L_{De}^i(k), i = 0,1,2,3; e = 0,1, \dots, n \end{cases} \quad (31),$$

мұндағы, $L_e^i(k+1) = L_{1e}^i(k) + L_{2e}^i(k) + L_{3e}^i(k)$ – еңбек ресурстарына сұраным;
 $L_{1e}^i(k), L_{2e}^i(k)$ және $L_{3e}^i(k)$ – сәйкесінше босатылып, қайта құралып және қысқартылып жатқан жұмыс орындарының саны;

Тауар нарығы. Өнім сұранымы мен ұсынымының баланысы:

$$\begin{aligned} \sum_{i=1}^3 y_e^i(k) = \sum_{i=1}^3 \sum_{j=1}^n a_{ej}^i y_j^i(k+1) + \sum_{i=1}^3 \sum_{j=1}^n b_{ej}^i \Delta y_j^i(k+1) + \\ + \omega_e^0(k+1) + \omega_e^1(k+1) + \omega_e^2(k+1) + \omega_e^3(k+1) + \omega_e^B(k+1) + \\ \omega_e^G(k+1), l = 1,2, \dots, n, \end{aligned} \quad (32)$$

Нарық өндірілген өнімдермен, ресурстармен айырбастау және табыстарды үлестіру процесін реттейді. Нарықта өнімдер тауарға айналады да, айырбасталады.

(k+1) периодындағы l-ші тауар құны:

$$P_e(k+1) = \sum_{i=1}^3 \left(\sum_{j=1}^n a_{je}^i P_j(k) \right) * \frac{y_e(k+1)}{y_e(k)} + \sum_{i=1}^3 \left(\sum_{j=1}^n b_{je}^i P_j(k) \right) * \frac{\Delta y_e^i(k+1)}{y_e(k)} + \frac{P_e(k)}{y_e(k)} * [\omega_e^0(k+1) + \omega_e^1(k+1) + \omega_e^2(k+1) + \omega_e^3(k+1) + \omega_e^B(k+1) + \omega_e^G(k+1)], \quad l = 1, 2, \dots, n, \quad (33),$$

Сонымен, тауар құнының орташа деңгейінің динамикасы:

$$P(k+1) = P(k) \frac{1}{\sum_{i=1}^n P_e(k) y_e(k)} [M_0(k+1) - M_0(k)], \quad (34),$$

Экономикалық жүйенің күрделілігі таза мақсатқа бағытталған әлеуметтік- экономикалық экспериментті қоюға мүмкіндік бермейді.

Модельдермен ЭЕМ-да математикалық имитациялық эксперимент мемлекеттік әлеуметтік-экономикалық саясатының мемлекет дамуының орнықтылығына әсерін тигізудің бірден бір құралы болып табылады[4]., [5]., [6].

1.7. Өндірістегі экономикалық жүйенің негізгі қорының динамикасының модельдерін жетілдіру

Негізгі қордың шағын τ арқылы, ал уақытты t арқылы белгілеуге келісіп алайық. Сірә, шақтың астарынан, өндіріске негізгі қорды енгізу кезеңінен, τ уақыт кезеңіне дейінгі уақыттың шамасын айтамыз, мұнда біз оның шағын “өлшейміз”. Негізгі қордың шақтық құрамының динамикасының моделін құрамыз.

Негізгі қорды шақтық топтарға бөлеміз, τ шағының шкаласы ұзақтығы T -ға тең бірдей интервалдарға бөлінген деп есептелінеді: 0-дан T -ға дейінгі негізгі қордың көлемін x_1 арқылы белгілейміз; x_2 - арқылы T -дан $2T$ -ға дейінгі шақта қордың көлемі, және т.б. Соңғы топқа жататын қордың көлемін x_n арқылы (n - шақтық топтардың саны) белгілейміз. t уақыт кезеңіндегі негізгі қордың шақтық құрылымы барлық шақтық топтардың x_1, x_2, \dots, x_n көлемінің жиынтығымен (векторымен) сипатталатын болады. t_n уақытында үзіліссіз, негізгі қордың шақтық құрылымын қарастырамыз. Осыдан әрі $(t_{k+1} - t_k)$ уақытының қадамы, T шақтық қабатының (шақтық тобының) еніне әрқашан тең деп санаймыз, яғни $t_k = t_0 + kT, k = 1, 2, \dots, n$.

Алғашқы t_0 уақыт кезеңінен алғашқы есеп беру уақыты ($t_0=0$), ал T шамасын уақыттың өлшем бірлігі деп алайық. Сонда қордың шақтық құрылымы уақыт кезеңінде келесі сандар жиынтығымен сипатталады.

$$x_1(k), x_2(k), \dots, x_n(k), \quad (34),$$

Енді негізгі қордың өсу және кему процестерін және олардың бір шақтық топтан басқа біреуіне өтуін қарастырамыз. Бір қадам уақыт ішінде барлық шақтық топтардың негізгі қорының көлемінің өзгеруін бақылайық, яғни $(k+1)$ уақыт кезеңіндегі негізгі қордың күйінің өзгеруі оның k уақыт кезеңіндегі күйіне қалайша тәуелділігін анықтайық. Алғашқы шақтық тобының $x_1(k+1)$ қайтадан енгізілген жаңа қуаттылық көлемі өзіне енгізеді, жаңа қорды қатарға енгізу мен әрбір шақтық қабатта бар қорды модернизациялау және осы құрамаларды, осы қабатты құрайтын негізгі қордың санына пропорционалды деп айта аламыз. Сонда $x_1(k+1)$ үшін төмендегіні аламыз.

$$x_1(k+1) = b_{11}x_1(k+1) + b_{12}x_2(k+1) + \dots + b_{1n}x_n(k+1), \quad (35),$$

мұндағы, $b_{11}, b_{12}, \dots, b_{1n}$ коэффициенттері алғашқы инвестиция нәтижесінде сәйкес келетін шақтық топтарының қалпына келтіру параметрлері (жаңа қорды қатарға енгізу және бар қорды модернизациялау). Әрине ойша барлық b_{ij} коэффициенттері теріс емес және $b_{ij} = 0$ тек, егер j -ші шақтық топ күрделі жөндеу мен модернизациялау нәтижесінде толық қалпына келтірілмесе. Тура осылайша $x_2(k+1)$ үшін келесі өсімді аламыз.

$$x_2(k+1) = b_{22}x_2(k+1) + \dots + b_{2n}x_n(k+1), \quad (36),$$

мұндағы, b_{22}, \dots, b_{2n} коэффициенттері екінші инвестиция нәтижесінде сәйкес келетін шақтық топтарының қалпына келтіру параметрлері

$$b_{22} = (b_{22}^1 + b_2^K + b_2^A).$$

b_{22}^1 – екінші топтың негізгі қорының екінші рет қалпына келтіру коэффициенттері;

b_2^K – екінші шақтағы сатып алынған қорды енгізу коэффициентті;

b_2^A – екінші топқа қатысты қолданылған қордың арендасы.

Осылайша негізгі қордың өсімі үшін төмендегідей өрнек аламыз:

$$\begin{cases} x_1(k+1) = b_{11}x_1(k+1) + b_{12}x_2(k+1) + \dots + b_{1n}x_n(k+1) \\ x_2(k+1) = b_{21}x_1(k+1) + b_{22}x_2(k+1) + \dots + b_{2n}x_n(k+1) \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ x_n'(k+1) = \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots b_{nn}x_n(k), \end{cases} \quad (37),$$

Шақтық құрамының толық моделін (37) бірге құрайтын қалған арақатынастарды алу үшін, барлық қорлар k уақыт кезеңінде j -ші шақтық тобында шақтық дискретизациясының кадамы мен уақытының үйлесуіне қарай, $k(k+1)$ уақыт кезеңінде $b(j+1)$ шақтық тобына толығымен өтетінін байқаймыз. Осы өту кезеңінде негізгі қордың көлемі тек негізгі қордың тозуына байланысты кемиді (олардың ескіруіне байланысты өнімділіктің төмендеуі). Егер $b_{i+1,j}$ – арқылы қордың іске аспауы коэффициентін белгілесек, оның тозуына байланысты (олардың ескіруіне байланысты өнімділіктің төмендеуі) қорды сату немесе оны арендаға өткізу. j -ші шақтық тобынан $b(j+1)$ -ші шақтық тобына өтетін негізгі қордың көлемі келесі өрнекпен анықталады:

$$x''_{j+1}(k+1) = b_{i+1,j} x_j(k), \quad (38),$$

($j=1,2,\dots,n-1$)

$n-1$ (38) сияқты табамыз.

Ойша барлық $b_{i+1,j}$ коэффициенттері 0-ден 1-ге дейінгі шекте анықталған. (37) мен (38) қосылғанда негізгі қордың көлемінің тұтас өсімшесін сипаттайтын келесі n айырым теңдігінің жүйесін білдіреді.

$$\begin{cases} x_1(k+1) = b_{11}x_1(k) + b_{12}x_2(k) + \dots + b_{1n}x_n(k) \\ x_2(k+1) = x'_2(k+1) + x''_2(k+1) \\ x_3(k+1) = x'_3(k+1) + x''_3(k+1) \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ x_n(k+1) = x'_n(k+1) + x''_n(k+1), \end{cases} \quad (39),$$

осыдан аламыз

$$\begin{cases} x_1(k+1) = b_{11}x_1(k) + b_{12}x_2(k) + \dots + b_{1n}x_n(k) \\ x_2(k+1) = b_{21}x_1(k) + b_{22}x_2(k) + \dots + b_{2n}x_n(k) \\ x_3(k+1) = b_{31}x_1(k) + b_{32}x_2(k) + \dots + b_{3n}x_n(k) \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ x_n(k+1) = b_{n1,n-1}x_{n-1}(k) + b_{nn}x_n(k), \end{cases} \quad (40),$$

немесе векторлы-матрициялық белгімен $\vec{x}(k+1) = B\vec{x}(k)$.

мұндағы, $\vec{x} = \{x_1, x_2, \dots, x_n\}$ – шақтық топтардың негізгі қорының векторы.

(40) дискреттік модельдегі барлық коэффициенттерін қарастырамыз. b_{11} – инвестиция есебімен өндіріске жаңа қорды енгізу коэффициентті, b_{ij} , $j=1,2,3, \dots, n$ үшін $i=j$ кезінде j -ші топтан j -ші топқа қорды қалпына келтірудің

жиынтық коэффициентін және сатып алу мен аренда жолымен j -ші тобына қатысты, қолданылған қорды өндіріске енгізуді сипаттайды

$$b_{ij} = (b'_{ij} + b_j^k + b_j^A)$$

мұндағы, $b'_{ij} - j$ - тобынан j - тобына негізгі қорды қалпына келтіру коэффициентін,
 $b_j^k - j$ -ші шақтың өндіріске сатып алынған қорды енгізу коэффициенті.

$i < j$ кезінде $b_{ij}(i, j=1, 2, \dots, n)$ - жөндеу мен модернизация нәтижесінде j тобына j негізгі қорды қалпына келтіру коэффициентін сипаттайды. $i > j$ кезінде, яғни $i=j+1, j=1, 2, \dots, n-1, b_{ij} -$ қордың k уақыт кезінде $j -$ тобына, $(k+1)$ уақыт кезінде $(j+1)$ тобына өту коэффициентін сипаттайды. Сонымен қоса өтетін негізгі қордың көлемі тозу, арендаға беру және сату есебімен кемиді

$$b_j + 1, j = 1 - \mu_j + b'_{ij} - b_j^\Pi - b_j^{CA}$$

мұндағы, $\mu_j, b_j^\Pi, b_j^{CA} - j$ тобының негізгі қорының тозу, сату, арендаға беру коэффициенттері

$$B = \begin{pmatrix} b_{11} & b_{12} & b_{13} & \dots & b_{1n} \\ b_{21} & b_{22} & b_{23} & \dots & b_{2n} \\ 0 & 0 & b_j, b_{j-1} & b_{jj}, b_{j, j+1} & b_{jn} \\ 0 & 0 & 0 & b_{n, n-1} & b_n \end{pmatrix} \quad (41),$$

B - қалпына келтіру және істен шығару коэффициенттерінің матрицасы. Матрицада нөл емес элементтер негізгі диагональдың астарында және жоғары орналасқан. Әрине B матрицасының барлық элементтері жалпы жағдайда экономикалық ортаның жағдайына байланысты тәуелді, сондықтан жалпылап айтқанда матрица уақытқа тәуелді[7].

1.8. Модельді есептерді шешу тәсілдерін таңдау

Стационарлық ортадағы, (41) дискретті модельдің негізінде негізгі қордың өзгеру динамикасын қарастырамыз. Стационарлық ортаның шарты $b_{ij}, i = 1, 2, \dots, n, j = 1, 2, \dots, n$ моделінің барлық параметрлері тұрақты шамалар болып табылады.

Модернизация мен қалпына келтірудің нөлдік коэффициентінің (42) бары, ақырғы шақтың топтардағы, зерттелетін теңдік жүйесінің өлшемін кемуіне әкеледі.

(41) жүйесінің сипаттамалық теңдігі:

$$|B - \lambda I| = \begin{vmatrix} b_{11} - \lambda & b_{12} & b_{13} & \dots & b_{n-1} & b_{1n} \\ b_{21} & b_{22} - \lambda & b_{23} & \dots & b_{2n-1} & b_{2n} \\ 0 & b_{32} & b_{33} - \lambda & \dots & b_{3n-1} & b_{3n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & b_{n,n-1} & b_{nn} - \lambda \end{vmatrix} = 0 \quad (42),$$

Өндіріс өз құралдарын тек жабдықтар алуға және күрделі жөндеуге ғана инвестициялайды деп жорамалдайық, нәтижесінде тозған жабдықтар толық қалпына келтіреді, яғни кәсіпорын қолданылған жабдықты сатып алмайды, $b_{ij} = 0$ $i=j$ үшін $j=2,3,\dots,n$ және $i=2,3,\dots,n$ кезінде $i < j$, және де ескі қорды арендаға алмайды. Сонымен бірге модернизация нәтижесінде негізгі қордың ресурстары толық қалпына келтіріледі. Сонда B матрицасының түрі төмендегідей болады:

$$B_1 = \begin{vmatrix} b_{11} & b_{12} & b_{13} & \dots & b_{1n-1} & b_{1n} \\ b_{21} & 0 & 0 & \dots & 0 & 0 \\ 0 & b_{32} & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & b_{n,n-1} & 0 \end{vmatrix} \quad (43),$$

және (44) теңдігі төмендегідей түрде енеді:

$$\vec{x}(k+1) = B_1 \vec{x}(k) \quad (44),$$

(41) жүйенің сипаттамалық теңдігі осы жағдайда төмендегідей түрге өзгереді

$$|B_1 - \lambda I| = \begin{vmatrix} b_{11} - \lambda & b_{12} & b_{13} & \dots & b_{n-1} & b_{1n} \\ b_{21} & -\lambda & 0 & \dots & 0 & 0 \\ 0 & b_{32} & -\lambda & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & b_{n,n-1} & -\lambda \end{vmatrix}$$

Анықтауышты ашып, аламыз

$$\lambda^n - b_{11}\lambda^{n-1} - b_{12}b_{21}\lambda^{n-2} \dots b_{1,n-1}b_{21}b_{32} \dots b_{n,n-1} = 0 \quad (45),$$

Алдын ала ортақтықты кемітпей, ақырғы шама тобының b_{1n} қалпына келтіру коэффициентінің шамасы нөлден өзгеше екендігін байқаймыз. Шынында да, $b_{1n} = 0$ болсын, сонымен қатар b_{1k} – күрделі жөндеу нәтижесіндегі қалпына келтірудің жатық жолындағы ақырғы нөлдік элемент, яғни $b_{1k} > 0, b_{k+1} = \dots b_{1n} = 0$.

Сонда (44) теңдік жүйесі екі жүйеге ыдырайды, біріншісі алғашқы k - тобының динамикасын баяндайды[8]. Ал екіншісі - келесі $(n-k)$ топтарының динамикасын. Сонымен бірге алғашқы k — өлшемді жүйе келесі түрге енеді.

$$A_1 = \begin{vmatrix} b_{11} & b_{12} & \dots & b_{1,k-1} & b_{1k} \\ b_{21} & 0 & \dots & 0 & 0 \\ 0 & b_{32} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & b_{k,k-1} & 0 \end{vmatrix}$$

ал екінші $(n-k)$ - өлшемді жүйе төмендегідей түрге енеді

$$\begin{cases} x_{k+1}(k+1) = b_{k+1,k}x_k(k) \\ x_{k+2}(k+1) = b_{k+2,k+1}x_{k+1}(k) \\ \dots & \dots & \dots & \dots \\ x_n(k+1) = b_{n,n-1}x_{n-1}(k) \end{cases} \quad (46),$$

Алғашқы теңдік жүйесін, сірә бөлінеді және екіншіден тәуелсіз зерттеледі. Келесі $(n-k)$ топтарының динамикасы, бірінші жүйенің шешімі екінші рет қойылғаннан кейін табылады, осылайша ақырғы топтағы қалпына келтірудің нөлдік коэффициенттерінің бары зерттеліп отырылған теңдік жүйенің өлшемінің кемуіне әкеліп соғады. n шақтық тобының саны ерікті, сондықтан $b_{1n} > 0$ жорамалы жалпылық талқылауды кемітпейді. $b_{1n} \neq 0$ болғандықтан, яғни (45) сипаттамалық теңдіктің бос мүшесі нөлден өзгеше болғандықтан, сипаттамалық теңдіктің нөлдік коэффициенттері жоқ. Сонымен қоса (45) сипаттамалық теңдіктің еселі түбірі жоқ деп жорамалдаймыз. Сонда (44) жүйесінің шешімі түрі бар

$$\vec{x}(k) = \sum_{i=1}^n C_i \vec{Z}_i^k \lambda_i^k \quad (46),$$

мұндағы, $\lambda_1, \dots, \lambda_n$ — B_1 матрицасының меншікті мәндері;

$\vec{Z}^1, \vec{Z}^2, \dots, \vec{Z}^n$ — B_1 матрицасының $\lambda_1, \dots, \lambda_n$ меншік сандарына сәйкес келетін меншік векторлары;

C_1, C_2, \dots, C_n — ерікті тұрақтылар;

Егер $\vec{x}(0) = \vec{x}_0$ алғашқы шарты берілсе ғана, C_1 тұрақтылықтар анықталады. Өйткені (44) теңдігінің B_1 , матрицасының теріс

болмағандығынан Фробеннус теориясының (45) сипаттамалық теңдігінің түбірінің арасында $\lambda = \lambda_{max}$ он түбірі болады, төмендегідей:

$$|\lambda_i| \leq \lambda_{max} \quad (i=1,2,\dots,n) \quad (47),$$

Моделінің өрескел пікірінен мынадай мағынаны жорамалдауға болады: іс жүзінде қатаң теңсіздік орын алатынын. Осылайша (46) шешімін, барлық қалған қосындылардан уақыт ағысымен алдын ала белгілі көп болатын басты мүшені бөлуге болады, яғни:

$$x(k) \approx C_1 \vec{Z}^1 \lambda_{max}^k + \sum_{i=2}^n C_i \vec{Z}^i \lambda_i^k \quad (48),$$

Үлкен k үшін асимптоттық жүріс-тұрыс түрі төмендегідей

$$x(k) = C_1 \vec{Z}^1 \lambda_{max}^k \quad (49),$$

Қосындысының қалған мүшелері негізгі қордың көлемінің құбылуы мен шақтық құрамын баяндайды. (48) түсінігіндегі қосындының мүшелерінің арасында апериодтық құрама жоқтығын көрсетеміз. Басқаша айтқанда (45) полиномында, λ_{max} жалғыз түбірді қоспағанда оң түбірі жоқ, яғни $\lambda_1, \lambda_2, \dots, \lambda_n$ сипаттамалық сандар не теріс, не комплексті. Мұның айғағы көп мүшенің оң түбірі саны туралы Декарт теориясының тікелей шығады. Шынында да көп мүшенің (45) коэффициенттерінің қатарында әрқашанда бір ғана белгі ауыстыру болады. Демек оның түбірлерінің арасында, алдын ала бар болуы және λ_{max} тең, біреуден артық оң түбір болмайды[9].

1.9. Құрылған модельдердің базасында өндірістің экономикалық жүйесінің қалыпты даму бағасының математикалық тәсілін жетілдіру

Шешімнің асимптоттық жүріс-тұрысын қарастырамыз. Негізгі қордың шақтық тобының шегі келесі өрнекпен өрнектеледі:

$$P_i(k) = \frac{x_i(k)}{\sum_{i=1}^n x_i(k)} = \frac{x_1(k)}{N(k)} \approx \frac{Z_i^1}{\sum_{i=1}^n Z_i^1}$$

$k \rightarrow \infty$ кезінде

мұндағы, $N(k)$ - k қадамындағы негізгі қордың жалпы көлемі,

\vec{Z}^1 — векторы B матрицасының меншікті векторы, меншікті санына сәйкес келетін.

λ_{max} — келесі теңдік жүйесінен анықталады:

$$\|B - \lambda_{max}I\|\vec{Z}^1 = 0$$

немесе скалярлы формада

$$\begin{cases} (b_{11} - \lambda_{max})Z_1^1 + b_{12}Z_2^1 + \dots + b_{1n}Z_n^1 = 0 \\ b_{21}\lambda_1^1 = \lambda_{max}Z_2^1 = 0 \\ b_{32}Z_2^1 - \lambda_{max}Z_3^1 = 0 \\ \dots \dots \dots \\ b_{n,n-1}Z_{n-1}^1 - \lambda_{max}Z_n^1 = 0 \end{cases} \quad (50),$$

Осы теңдік жүйесінің есебінің шешімі ерікті тұрақтылыққа дейін толығымен анықталады. Сондықтан $Z_1^1 = 1$ деп аламыз, сонда екінші рет алынған (50) теңдік жүйесінен алатынымыз:

$Z_2^1 = b_{21}\lambda_{max}^{-1}$, ал үшінші жүйенің теңдігінен $Z_3^1 = b_{21}b_{32}\lambda_{max}^{-2}$ т.б. Ақырында \vec{Z}^1 векторын (негізгі қордың шақтық құрамының шегі) келесі түрде аламыз.

$$\vec{Z}^1 = \{1, b_{21}\lambda_{max}^{-1}, b_{21}b_{32}\lambda_{max}^{-2}, \dots, b_{21}b_{32} \dots b_{n,n-1}\lambda_{max}^{-n+1}\} \quad (51),$$

Осыған жиілікті нормалау шартын қолданып, негізгі қордың шақтық құрылымының тепе-теңдік шегін алу оңай.

$$\lim_{k \rightarrow \infty} P_i(k) = \frac{1}{1 + b_{21}\lambda_{max}^{-1} + \dots + b_{21}b_{32} \dots b_{n,n-1}\lambda_{max}^{-n+1}} \quad (51),$$

$$\lim_{k \rightarrow \infty} P_n(k) = \frac{b_{21} \dots b_{n,n-1}\lambda_{max}^{-n+1}}{1 + b_{21}\lambda_{max}^{-1} + \dots + b_{21}b_{32} \dots b_{n,n-1}\lambda_{max}^{-n+1}}$$

Негізгі қордың асимптоттық өсуі мен кемуі λ_{max} шамасымен анықталады. Егер $\lambda_{max} > 1$ болса, онда негізгі қор өндіріске сәйкес өседі, егер $\lambda_{max} < 1$ болса, онда негізгі қордың көлемі кемиді, егер $\lambda_{max} = 1$ болса, онда негізгі қордың тұрақты шамаға асимптотикалық түрде ұмтылады.

Стационарлық немесе өсіп келе жатқан негізгі қорда шақтық топтардың көлемі тепе-теңдікте топтардың (көлемі) өсуіне байланысты өспейді.

Шынында да (52) аламыз.

$$\lim_{k \rightarrow \infty} \frac{x_{i+1}(k)}{x_i(k)} = b_{i+1,i}\lambda_{max}^{-1} \quad (i=1,2,\dots,n) \quad (52)$$

$\lambda_{max} \geq 1$ ал $\alpha_i \leq 1$ есепке алып, аламыз

$$\lim_{k \rightarrow \infty} \frac{x_{i+1}(k)}{x_i(k)} \leq 1 \quad (i=1,2,\dots,n)$$

Егер де негізгі қор тез көшірілсе, онда керісінше көріністе болуы мүмкін: келесі шама тобының көлемі алдыңғы шақтық тобының көлемінен асып немесе басым болады. Шынында да, $\lambda - b_{i+1,j}$ болсын, сонда аламыз

$$\lim_{k \rightarrow \infty} \frac{x_{i+1}(k)}{x_i(k)} = \frac{b_{i+1,i}}{\lambda} \geq 1$$

Қорытындысында негізгі қордың өсу немесе кему шартын аламыз, яғни шарт былай: λ_{max} бірден көп немесе аз. Осы үшін (41) сипаттамалық теңдікті λ^n бөлеміз және $\xi = 1/\lambda$ белгілейміз. Сонда ξ үшін келесі теңдікті аламыз

$$P(\xi) = b_{11}\xi + b_{12}b_{21}\xi^2 + \dots + b_{1n}b_{21} \dots b_{n,n-1}\xi^n = 1$$

$P(\xi)$ полиномының теріс немесе коэффициенттерінің күшіне қарай өз дәлелінің бір қалыпты шақтық функциясы бар. Сондықтан айғақты оң түбір бірден көп немесе аз екендігін анықтау үшін, $P(1)$ шамасы бірден басым болуын тексеру жеткілікті. Егер, мысалы, $P(1) \geq 1$ болса, онда $\xi_0 < 1$ (яғни $\lambda_{max} > 1$). Ақырында келесіні аламыз

$$b_{11} + b_{12}b_{21} + \dots + b_{1n}b_{21}b_{32} \dots b_{n,n-1} > 1 - \text{өсу } (\lambda_{max} > 1)$$

$$b_{11} + b_{12}b_{21} + \dots + b_{1n}b_{21} \dots b_{n,n-1} < 1 - \text{кему } (\lambda_{max} > 1)$$

$$b_{11} + b_{12}b_{21} + \dots + b_{1n}b_{21} \dots b_{n,n-1} = 1 - \text{стационарлық көлем } (\lambda_{max} > 1).$$

Мұнда тек (52) жүйенің шешімінің асимптотикалық жүріс-тұрысы зерттелген, яғни шақтық топтың құрамының шегін және негізгі қордың асимптоттық өсу (кему) жылдамдығын қарастырдық (41) сипаттамалық теңдікте теріс және комплексті түбірлер болғандықтан, әлбетте негізгі қордың шақтық құрамы мен көлемнің құбылуы орын алады.

Мысалы негізгі қор екі шақтық топқа бөлінген жағдайды қарастырайық, сонымен бірге бірінші және екінші топтағы қалпына келтіру коэффициенті нөлден өзгеше. Айырым теңдігінің жүйесін келесі түрде аламыз

$$\begin{cases} x_1(k+1) = b_{11}x_1(k) + b_{12}x_2(k) \\ x_2(k+1) = b_{21}x_1(k) + b_{22}x_2(k) \end{cases} \quad (53),$$

(53) жүйесінің сипаттамалық теңдігі

$$\lambda^2 - b_{11}\lambda - b_{21}b_{12} = 0 \quad (54),$$

екі сипаттамалық саны бар

$$\lambda_{12} = \frac{1}{2}(b_{11} \pm b_{11}^2 + 4b_{21}b_{12}) \quad (55),$$

Екі түбірдің де айғақты екенін көру оңай, сонымен бірге біреуі оң, ал екіншісі теріс. Оң сипаттамалық санды λ_{max} арқылы, ал терісті - $(-\lambda)$ арқылы, (53) жүйесінің шешімін төмендегідей түрде жазамыз.

$$\lambda(k) = C_1 Z^1 \lambda_{max}^k + C_2 Z^2 (-1)^k \lambda^k \quad (56),$$

мұндағы, C_1, C_2 – ерікті тұрақтылар,

Z^1 және $Z^2 - \lambda_{max}$ мен $-\lambda$ меншікті мәндеріне сәйкес, меншік векторлары;

$-\lambda < \lambda_{max}$ болғандықтан, λ_{max} көрсеткішті геометриялық прогрессияны асимптотикалы аламыз. Шешімнің екінші қозылғышы (56) k уақытының кіші мән кезінде байқалатын, негізгі қордың шақтық құрамының мүмкін құбылыстарын баяндайды. Негізгі қордың өсу, кему, стационарлығы келесі өрнектің белгісіне қарай анықталады.

$$V = b_{11} + b_{21}b_{12} = 1$$

Егер $V > 0$ болса, онда негізгі қордың (өндірістің) көлемі өседі, $V < 0$ - өндіріс қысқарылады, $V = 0$ кезінде көлем қандай да бір тұрақты деңгейге ұмтылады.

Шектік шақтық көлемі (57) формуласы бойынша анықталады.

$$P_1 = \frac{1}{1 + b_{21}\lambda_{max}^{-1}}$$

$$P_2 = \frac{b_{21}\lambda_{max}^{-1}}{1 + \lambda_{max}^{-1}b_{21}} \quad (57),$$

Мысалдағы барлық пікірлер b_{21} мен $b_{21} \neq 0$ деген жорамалмен жүргізілді. $b_{11} = 0$ кезінде $|\lambda| = \lambda_{max}$ болса құлдырау жағдайына әкеледі [10].

Осындай жағдайда (56) жүйесінен негізгі қордың шақтық құрамы мен көлемнің өшпейтін құбылысын алуға болады

$$(|\lambda| = \lambda_{max} = 1)$$

1.10 Есептің қойылымы

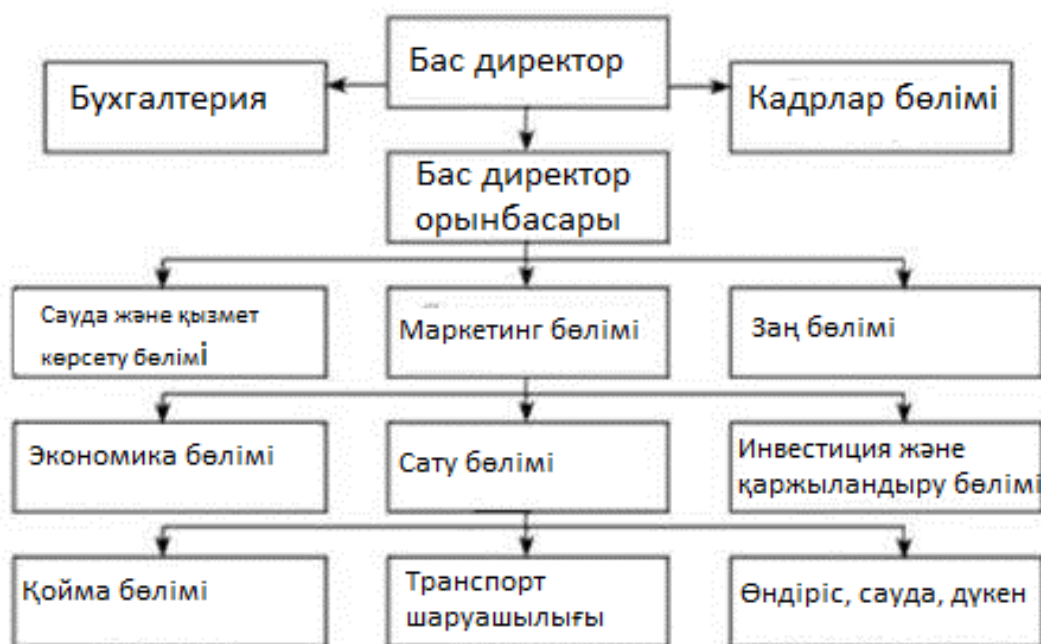
Берілген дипломдық жоба төмендегідей кезеңдерден тұрады:

- дипломдық жоба үшін белгілі бір объектіні, сол объектінің бөлімін анықтау, сипаттама беру.
- өндірістің экономикалық даму процесінің әдістері мен модельдеріне талдау және шолу жүргізу.
- өндірістің экономикалық жүйе динамикасының моделін жетілдіру.
- негізгі қор динамикасының моделін жетілдіру.
- бөлімнен қолданушыны таңдап алып, қандай қызмет корсетуді анықтау.
- құрылған модельдердің базасында экономикалық жүйенің қалыпты даму бағасының математикалық тәсілін жетілдіру.
- экономикалық жүйе динамикасының моделін іске асыратын бағдарламалар комплексін өңдеп шығару.
- программа алгоритмін құру, моделін жасау, тілін таңдау
- программалық кодтарға кесте түрінде түсініктеме беру.
- бағдарламалық комплексті іске асыру кезіндегі еңбекті қорғау шараларын өңдеу.
- бағдарламалық комплексті пайдаланудың экономикалық тиімділігін және оның өзін-өзі өтеу уақытын есептеу.
- техника құралдарының комплексін таңдау.

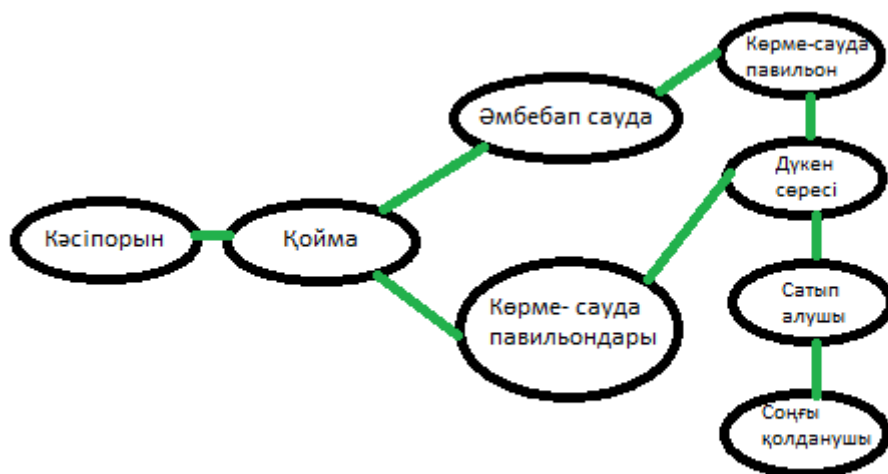
2 Жобалық бөлім

2.1 Кәсіпорын бөлімі, сипаттамасы, көрсетілген қызмет

Кәсіпорынның қызмет көрсетудегі басты бағыты дәмді өнімдерімен адамдарға қуаныш сыйлау, көне дәстүрді сақтай отырып, қазіргі заманның талабына сай барынша көп адамның қажеттілігін жоғары сапалы кондитер өнімдерімен жоғары деңгейде қанағаттандыруға бағытталған. Керемет кондитер өнімдерін қалай шығаруды біледі және сол мақсатта қорды арттырады. Қазақстан мен оның шет аймақтарындағы әр адамға өз еліміздің байлығы пен қонақжайлылығын бейнелей отырып, Қазақстандағы және Орталық Азия нарығындағы № 1 компания болуға ұмтылады. Қазақстандық және шетелдік тұтынушылардың кең шоғырының қажеттілігін сапалы тауарлармен (көтерме, бөлшек) қанағаттандыру.



2.1 сурет - Кәсіпорын құрылымы.



2.2 сурет- Кәсіпорынмен жұмыс жасау қызметі.

2.1.1 Кәсіпорын бөлімдері:

- сауда және қызмет бөлімі;
- маркетинг бөлімі;
- заң бөлімі;
- инвестиция және қаржыландыру бөлімі;
- сату бөлімі;
- транспорт Қауіпсіздік жүйе бөлімі. Қауіпсіздік жүйесін жобалау және құру;
- электромонтаждау бөлім. Қоректендіру көзі желілерін, электро жабдықтау және найзағайдан қорғау жүйесін жобалау және құру, зертханалық өлшемдер және сынақтан өткізу;
- өндірістік-техникалық бөлім. Техниканы -құрастыру және іске қосуды реттейтін жұмыстар;
- сервистік бөлім. Жабдықтарға сервисті және кепілдікті қызмет көрсету. Клиенттердің шаруашылығы бөлімі;
- сұрақтарына кеңес беру;
- логистика бөлімі. Нарықты саралау, талдау. Шикізатты және техниканы -сатып алу және жеткізіп беру;

2.1.2 Өндірістік бөлімі. Кәсіпорынның негізін салушы бөлім болып өндірістік бөлім саналады. Жоспарланған капиталды бастапқы мәліметтерге байланысты дұрыс өңдеу аталған бөлімнен басталады. Жұмыстың қорытындысы тапсырыс берушіге келісім бойынша дайын өнімді тапсыру.

Өндірістік бөлімнің басты қызметі болып оның бағыты табылады:

- тапсырыс берушімен жұмысты жоспарлау.
- өнімнің дайын болатын күнтізбелік кестесін құрастыру.

- жұмыстың тізбекті орындылуына мониторинг жасау.
- өзара талаптардың орындауын бақылау.
- атқарушылық өндірістік құжаттама қозғалысын бақылау.
- шикізаттық-материалдық құндылықтарды жөндеу және есептен шығару.

- бөлімді басқару өндірістік бөлімнің бастығы өкілеттілігінде, яғни басты қаржыландыру менеджері.

2.1.3 Басты қаржыландыру менеджері және жарнамалаушы қызметкер. Негізгі қаржыландыру менеджер қызметі – шикізат, өндірісшығындарын есептеу негізінде құрылады және зақымданған ошақта паналау ғимараттары мен орындарын салу, кірме және өтпе жолдарына, үйлер мен ғимараттардың осал құрастырмаларын құлатып түсіру немесе бекіту, үйінді басып қалған паналау ғимаратары мен орындарын ашуға, үйінділерді тазартуға, үйінділердің, құлаған паналау ғимараттары мен тасалану орындарының астынан адамдарды алып шығу жұмыстарына кеткен шығындарды ұйымдастыру және есептеу.

Бұл өндірістегі қаржыландыру менеджерінің қызметі:

- өндірістік бөлімді қаржылай қамтамасыздандыру;
- бөлімнің қызмет бағытының орындалуын толықтай қадағалау;
- өндірістің техникалық дайындағын керекті деңгейде болуын қаржылай қамтамасыз ету;
- жаңа техниканы енгізу жоспарын қаржылай ұйымдастыру және т.б.

Бұл барлық жұмыстар өндірістің қарқынды дамуына үлкен үлесін қосады. Сондықтан осы жұмыстарды атқарумен қоса, барлығын жарнамалау өте маңызды. Себебі жарнамалай отырып, фабрика көптеген тапсырыс берушілерді жинақтай алады. Бұл дегеніміз жұмыстың гүлденуіне әкеледі.

2.1.4 Көрсетілген жұмыс. Жоғарыда айтылғандай қаржыландыру менеджері кәсіпорынның барлық өндірістік, техникалық жұмыстарын қаржыландырады. Сол себептен барлық қаржылай жұмыстар қаржыландыру менеджерінің қарамағында. Сондықтан қаржыландыру менеджерінің қолдауымен кәсіпорынның жеке шығындарын, кірістерін, барлық қаржылай жұмыстарын есептейтін бағдарлама құруды ұйғардым. Бағдарламаның пайдалы жақтары:

- кәсіпорынның инвестициалық шығындарын толық есептейді;
- қажетті шикізат көлемін анықтайды;
- толық бюджет көлемін есептейді, сақтайды;
- енгізілген мәліметке оңай қол жеткізетін және өзгертуге мүмкіншілік беретін заманға сай қажетті аса зор бағдарламма;
- аймағын есептеу, құрал-жабдықтармен қамтамасыздандыру мөлшері, барлық капиталдық салым мөлшері, диаграммасын құру;
- сату мерзімі мен инвестициялық шығын мөлшерін есептеу;

- жоғарыда айтылғандарды эксел жұмыс істеу ортасында көрсетуге мүмкіндік берілген;
- бұл бағдарлама жобасы кәсіпорынның жұмыс тиімділігін көрсетуге ыңғайлы программа;
- кәсіпорында бар шикізатты білуге, олардың көлемін анықтауға, сатып алынған уақытын, жарамдылық кезеңін білуге мүмкіндік береді.

2.2 Мәліметтер базасын жобалау әдістемесі

Мәліметтер қорын жобалау және құрастыру өнерлілікті, іскерлікті талап етеді. Қолданушының талаптарын түсініп, оны тиімді мәліметтер қорының жобасына айналдыру шығармашылықты процесс болып табылады. Ал осы шығармашылық процесті толық-қанды және жоғары өнімді қосымшалар көмегімен физикалық тұрғыдағы мәліметтер қорына айналдыру – инженерлік процесс болып табылады. Екі процесс те бас ауыртарлық, қызықты интеллектуалдық күрделілікке толы. Деректер қорын қолдану арқылы үлкен ақпараттық жүйе де құруға болады. Ақпараттық жүйені құрмас бұрын онда қолданылатын деректер қорын жобалау арқылы ақпараттық жүйенің үлгісін жасап алған дұрыс. Ақпараттық жүйені жобалау жалпы бірнеше кезеңнен тұрады. Олардың негізгілері төменде келтірілген. Концептуалдық жобалау - программада қарастырылатын негізгі мәселелер мен функциялар анықталып, өңделетін алғашқы информацияны талдау мен оның құрылымын анықтау. Логикалық жобалау - ER–диаграмма негізінде реляциялық жүйелердің мүмкіндігі ескерілген мәліметтердің логикалық моделі, мәліметтер қоры кестелері арасындағы байланыс түрлерін анықтау. Физикалық жобалау - мәліметтер қорын басқару жүйелері құралдарымен орындалады: мәліметтерді сипаттау тілі (МСТ), мәліметтерді манипуляциялау тілі (ММТ). Физикалық модельдеудің соңғы этабында ЭЕМ жадысында мәліметтер қоры файлдары және іздеу структурасы файлдары пайда болады. Пәндік аймақты модельдеу төмендегідей кезеңдерден тұрады: инфологиялық модельді құрастыру, даталогиялық және физикалық жоспарлау, концептуальды жобалау. Инфологиялық модельді құрастыру кез келген ақпараттық жүйе саласымен байланысты. Пәндік аймақ дегеніміз сол еңбек саласындағы негізгі объектілер жиынтығы болып табылады. Жұмыс жасау аймағын модельдеу үрдісінде қарастырылып жатқан ақпараттық жүйедегі негізгі объектілер олардың қасиеттері және объектілер арасындағы байланыстар анықталады. Ақпараттық жүйелерді жобалау жұмысы барысында құрылған инфологиялық модель негізінде даталогиялық модель құрылады. Оның инфологиялық модельден айырмашылығы жүйедегі құбылыстар уақыт мерзімі бойынша нақтыланады. Әрі қарай физикалық модель жасалады. Бұл жұмыс кезеңінде ақпараттық жүйеде қолданылатын мәліметтер қандай құрылғыларда сақталып, қандай жадтың түрлерін талап ететіні анықталады[11].

2.2.1 Мәліметтердің реляциялық базалары. Реляциялық модель ([ағылш. relational model](#)) [дерекқор](#) басқару жүйелерінде [дерекқор моделі](#), 1969 жылы [Эдгар Кодд](#) ұсынған бірінші-ретті [логика](#) негізінде жұмыс істейді.

Деректердің реляциялық модель негізі алғашқы рет 1970 ж Е.Кодд мақаласында мазмұндалды. Бұл жұмыс реляциялық модель өркендеу алған үлкен санды мақала және кітап үшін стимул болды. Деректердің реляциялық моделінің ең көп тараған трактовкасы К.Дейтке жатады. Дейттің атықанына сәйкес реляциялық модель үш бөліктен тұрады:

- құрылымдық бөлімнен.
- тұтастық бөлімнен.
- манипуляциялық бөлімнен.

Құрылымдық бөлім, реляциялық модельде қандай объектілер қарастырылады, соны сипаттайды. Реляциялық модельде қолданылатын деректердің жалғыз моделі болып қалыпты n-арлы қатынас табылады.

Тұтастық бөлім кез келген реляциялық деректер қорында кез келген қатынас үшін орындалатын арнайы түрдегі шектеуді сипаттайды. Бұл мән тұтастығы және сыртқы кілт тұтастығы.

Манипуляциялық бөлім реляциялық деректерді манипуляциялаудың екі эквивалентті тәсілін сипаттайды - реляциялық алгебраны және реляциялық есептеуді.

Осы бөлімде реляциялық модельдің құрылымдық бөлігі қарастырылады.

Деректердің реляциялық моделін (ДРМ) деректер қоры саласын зерттеуші белгілі маман Е.Ф.Кодд (Dr.E.F.Cood) ұсынған, ол өзара байланыстар арқылы нақты қыбылысты және ақпараттар ағымын әрнектеуге мүмкін екенін ашып көрсетті. Ең алғаш рет бұл модельдің тұжырымдамасы 1970 жылы «A Relation Model of Data for Large Shared Data Banks» //ACM,1970, v.13№6 мақаласында жарияланған. Реляциялық модель деректерді қолданбалы программадан тәуелсіз етіп бөлектеуге, яғни деректер базасының басқарудың маңызды мәселесін шешуге мүмкіндік береді. Бұл деректер базасын жобалау мен программалауды оңайлатуға әсерін тигізді. Деректердің реляциялық моделі физикалық мазмұны екі әлшемді кесте болып табылатын қатынас деп аталатын математикалық ұғымға негізделген.

Негізгі кілт (PRIMARY KEY) - мәндері кесте жазбаларының ерекшеліктерін қамтамасыз ететін бір немесе бірнеше жиек. Кестенің кілтіне қойылатын негізгі талап – бір кестеде бірдей жазбалардың болмауын қадағалау. Кесте кілтінің құрамындағы ешбір жиек NULL мәнін қабылдауы мүмкін емес; негізгі кілт кестенің категориялық біртұтастығын (integrity) қамтамасыз етеді. Әр кестенің тек бір ғана негізгі кілті анықталуы мүмкін. Кестенің кілті жазбалардың ерекшеліктерін бақылаумен қатар басқа кестелермен байланыс орнату үшін де қолданылады. Негізгі кілт бойынша деректер базасын басқару жүйесі әздігінен керекті жазбаларды теріп алуды

жылдамдататын индекс анықтайды, сондықтан кестенің кілті болып тұрған жиек (жиектер) бойынша арнайы индекс анықтауға болмайды. Кейде кесте жиектерінен негізгі кілт бола алатын жиекті немесе жиектерді белгілеу қиын болуы мүмкін, мындайда кестеге бәрінен мәнді жасанды негізгі кілт енгізіледі. Бүтін мәнді негізгі кілттің мәнін генератор арқылы енгізуге болады.

Деректер қоры(ДҚ)-құрылымы берілген тәсіл бойынша ұйымдастырылған арнайы форматтағы файл. ДҚ – да сақталатын берілгендер кесте түрінде дайындалады. Яғни дискілік файл – деректер қоры, оған енгізілетін кесте – ДҚ құрамында сақталушы объекті. Деректерді және олардың арасындағы байланыстарды ұйымдастырудың түрлі типтері бар: иерархиялық, реляциялық және тораптық. Иерархиялық ДҚ – ында жазба элементтері реттеліп жазылады да, бір элементі негізі, қалғандары бағыныңқы элементтер деп есептеледі. Реляциялық ДҚ – кесте түрінде дайындалған берілгендер. Ол ең көп таралған берілгендер қоры, кестелер арасындағы байланыстар – жиі пайдаланылатын ең маңызды ұғым. Мәліметтерді кестелер немесе реляциалар түрінде көрсететін мәліметтер моделі мәліметтердің реляциялық моделі болып табылады. Реляция дегеніміз жатық және тік жолдардан тұратын екі олшемді кесте. Код моделінде кестедегі мәліметтермен жұмыс жасауға екі тілді қолдануды ұсынылады: реляциялық алгебра мен реляциялық есептеу. Бұл теориялар математикалық логиканың бөліктері. Оларда кестені relation (қатынас) деп атайды. Осы тілдердің екеуінде мәліметтермен физикалық көрсеткіштер негізінде емес, олардың логикалық сипаттамалары арқылы жұмыс істеуді қамтамасыздандырады. Кесте түсініктемесі реляциялық модельдің негізі болып табылады. Мәліметтер қорын кестелер жиыны деп түсінуге болады. Бірақ бұл кестенің түсініктемесі тек қана мәліметтерді көрсету түріне қатысты, дискіде немесе жадыда жазбаларды физикалық іске асыруымен еш қандай байланысы жоқ. Реляциялық МҚБЖ-де ең кемінде екі шарт орындалады: 1) пайдаланушылар мәліметтерді кесте түрінде қабылдайды; 2) пайдаланушы қана кестелерден бар болатын операторларды қолданып жаңа кестелерді жасауларына болады. Кестеге операцияны қолданған кезде нәтижесінде кестені алатынымыз өте маңызды қасиет (реляциялық бекітілу қасиеті). Басқа сөзбен айтқанда бір түрлі объектке операцияны қолдансақ нәтижесінде бірдей типті объекті аламыз. Реляциялық модель негізінде келесідей түсініктемелер болады: мәліметтер типі, атрибут, домен, бастапқы кілт, қатынас. Реляциялық модельдегі мәліметтер типі деген түсініктеме программалау тілдердегі осындай түсініктемемен толығымен сәйкес. Әдетте символдық, сандық, календарлы, ақшалы, және биттік типтер қарастырылады. Ақпаратты сақтау үшін атрибут қолданылады. Реляциялық модельдегі атрибут түсініктемесінің мағынасы ЕК-модельдегі объект қасиетімен бірдей. Атрибуттың аты мен типі болады. Қатынас құрылымы атрибуттар жиыны мен олардың типтерінен құрастырылады. Атрибуттар саны

реляция дәрежесі (рангі) болып табылады. Реляциялық деректер қорын құру және онымен жұмыс істеуді басқару үшін көптеген арнайы программалар дайындалған: dBase, FoxBase, FoxPro, Access. Оларды Деректерді қорын басқару жүйелері (ДҚБЖ) деп аталады[12].

2.2.2 Жоғары деңгей тілі. Мәліметтер қорының дамуы ақпаратты сақтай алатын және қажетті ақпаратқа жету жылдамдығы тез болатын. Мәліметтер қорының тарихы 30 жылды құрайды. 1968 жылы қолданысқа алғашқы IBM фирманың IMS жүйесі енді. 1975 жылы мәліметтер қорының басқару жүйесінің бірінші стандарты пайда болды. Ол CODASYL мәліметтерді өңдеу ассоциациясымен жасап шығарылды. Аталған стандарт мәліметтер қоры жүйесінің негізгі түсініктер теориясын анықтайды, олар қазіргі кезге дейін де желі моделінің негізі болып келеді. 1981 жылы Э.Ф.Кодд реляциялық модельді құрып, оған реляциялық алгебра операцияларын қолданды. Мәліметтер қорының келесі даму кезеңдері бар: Файлдар және файлдық жүйелер. Үлкен электронды есептеуіш машиналардағы мәліметтер қоры. Алғашқы МҚБЖ. Персоналды компьютер кезеңі. Үлестірілен мәліметтер қоры. Жалпы мәліметтер қорының даму кезеңдерінің арасында белгіленген шекара жоқ. Олар бір біріне еніп отырады, тіпті параллельді түрде дамып келеді. Берілген кезеңдердің әр даму сатысындағы технологиялардың өзгеруі мен олардың ерекшеліктерін анықтауға мүмкіндік береді[13].

2.2.3 Деректер қорының логикалық моделі. Логикалық деңгей – бұл мәліметтерге деген абстрактылы көзқарас, мұнда мәліметтер нақты өмірде қалай көрінсе солай көрінеді. Логикалық деңгейде көрсетілген моделдер объектілері мәндер мен атрибуттар деп аталады. Мәліметтердің логикалық моделі басқа логикалық моделдің негізінде жасалуы мүмкін, мысалы, процестер моделдерінің негізінде. Мәліметтердің логикалық моделі әмбебап болып табылады және нақты жүзеге асып тұрған деректер қорын басқару жүйесімен (ДҚБЖ) байланыспайды. Мәліметтердің логикалық моделі оны тіркеуге жататын фактілер мен объектілерді сипаттайды. Мәндер арасында байланыс орнатылады, атрибуттар анықталады және мәліметтерді қалпына келтіру жүргізіледі.

ДҚ логикалық жобалауы AllFusion Erwin Data Modeler (Erwin) құралының CASE көмегімен жасалып, «Entity-Relationship» моделі құрылды. Бұл сызба жоба үлгісін береді және әсіресе пайдаланушылар арасында ой алмасу үшін пайдалы. AllFusion Erwin Data Modeler (ERwin) қорды қолдау және құру, деректерді сақтау үшін деректер қорын моделдеуге арналған жетекші шешім болып табылады. ERwin моделі деректердің басқаруды және ұйымдастыруды жеңілдету үшін, деректердің өзара күрделі байланысын ықшамдау үшін деректердің құрылымын визуалды қалыптастырады, сонымен қатар деректер қорын құру технологиясы және өрістеу ортасы. Оның

арқасында деректер қорын жасау үрдісін жылдамдатады және ықшамдайды, ал оның сапасы мен сенімділігі айтарлықтай жақсарады. ERwin жетекші вендрлердің деректер қоры үшін автоматты түрде кестелерді және мыңдаған жолдар кодын генерациялайды.

Логикалық жобалау – ER-диаграмма негізінде реляциялық жүйелердің мүмкіндігі ескерілген мәліметтердің логикалық моделі, мәліметтер қоры кестелері арасындағы байланыс түрлерін анықтау. ER-диаграмманы тұрғызу үшін заттық аумақтың объектілерін, атрибуттық құрамын анықтау керек. Ғылыми–зерттеу бөлімінің деректер қорының логикалық моделі ERwin-да ортасында жасалып араларына байланыстар жүргізілді [14].

2.2.4 Мәліметтер базасын жобалау. Мәліметтер базасы көрсетілген физикалық нұсқаның кестесін немесе мәліметтер базасы физикалық моделін қарастырайық. Кестені физикалық түрге келтірген кезде құрылымы инвариантты болып табылады, яғни кесте құрылымы өзгеріссіз қалады. Бірақ кесте параметрі нақты мәнді қабылдайды.

Мәліметтер базасы физикалық кестесінің құрылымы бір қалыпты болып табылады.

Мәліметтер базасы жоспарының жалпы бөлігі болып табылатын мәліметтер базасы жоспарын құру.

Тапсырманы шығаруға мәліметтер базасы құру.

Мәліметтер базасы екі кестеден тұрады, оның біріншісі – мастер кесте, екіншісі – деталь. Олардың арасындағы қатынас бірді – көпке.

Осыдан мәліметтер базасы құрудың келесі жоспары пайда болады.

1- қадам. Құрудың технологиясын таңдау

- консолді режимде

-RAD режимде

Мәліметтер базасы құру жоспары. Мәліметтер базасы құру келесі қадамнан тұрады.

2- қадам. Мәліметтер базасын сақтау орнын тіркеу және анықтау

- мәліметтер базасының орналасу орнын анықтау

- мәліметтер базасының типін анықтау

- мәліметтер базасының атын анықтау немесе алиас

- алиасты тіркеу және мәліметтер базасының адресінің орналасуы

Мәліметтер базасы құру және конструкторлар, ол үшін келесіне орындау керек.

3- қадам. Мәліметтер базасы кестесінің құрылымын құру.

- кесте типін анықтау

- кестенің физикалық моделін анықтау. Кестенің бірінші параметрінің қимылын орнату.

- құрылған кестені таңдалған алдрес бойынша сақтау.

Мәліметтер базасы деталь-кесте құрылымын құру. Мәліметтер базасының екінші кестесін құру бірінші кестеге сәйкес келеді. Оның айырмашылығы тек қана екілік индекс скрыптау параметрін құру.

4- қадам. Екі кесте жолы арқылы мәліметтер базасы формалдау.

Мәліметтер базасын құру құрылған кесте бойынша іске асырылады:

- екілік индексацияны орындау.
- құрылған кесте арасында бірді көпке қатынасын құру.

Delphi - де мәліметтер базасын тіркеу.

Мәліметтер базасын анықтау. Бұл файлдар жиынтығы, кесте, триггер және индексте көрсетілген кестелер арасындағы байланыс.

Мәліметтер базасы келесіден тұрады:

- мәліметтер базасы кестесі
- индекс
- ескертулер

Бір кестені сақтау үшін жеке файл құрылады. Осындай жеке файлдар кесте индексімен Мемо өрісті сақтауға құрылады. Бұл файлдар бірге сақталып орналасу қажет, әйтпесе Мәліметтер базасы қате көрсетеді. Сондықтан мәліметтер базасы компоненттері бірге сақталынады.

Мәліметтер базасы орналасуының тапсырмасы. Барлық жағдайда мәліметтер базасы нақты компьютерде, директорияда сақталу қажет. Егер мәліметтер базасы сол компьютердің дискіде сақталынса, онда мәліметтер базасы локальді аталады. Бұны директорияны анықтау арқылы жүргізуге болады[15].

2.3 Бағдарламалау тілін таңдау

2.3.1 Жалпы мәліметтер. Программа атауы – Кондитер зауытының тиімділік бағасы.

Программаның белгіленуі - C:\Users\qauhara\Documents\Балхан Гаухар\Оценка эффективности \оценка эффективности.exe

Кондитер зауытының тиімділік бағасының бағдарламасының ішкі жүйесі Delphi 7.0 ортасында өңделген. Ішкі жүйені тиімді функциялау үшін Windows 9x/7/8/XP операциялық жүйелері болуы тиіс.

2.3.2 Функциялық белгілену. Берілген программалық өнім келесі программалық құжатнамаларды қосуы керек: техникалық тапсырма; техникалық жоба; программа мәтіні; программа сипаттамасы;

Функционалдық белгіленуі деректер қорына (қосу, өшіру, жазбаны өзгерту, жұмысты ұйымдастыру және жұмыс процестеріндегі қажет құжаттарды баспаға шығарудан) тұрады.

Жүйені тиімді функциялау үшін Windows 95/98/NT/2000/XP/Vista операциялық жүйелері мен Borland Database Engine (BDE) бар болуы қажет.

Техникалық құралдардың параметріне және құрамына талап А қосымшасында сипатталған Техникалық тапсырма.

Программа компьютердің қатты дискісінде орналасқан және 37,9 Мб орын алады.

Жүйеде қолданылатын базалар 870 кб-тан және одан да көп орын алады (ол ақталатын ақпарат мөлшеріне тәуелді). Жүйеде қолданушының келесі әрекеттері қарастырылған:

–шикізаттың, негізгі бағаның, инвестициялық шығынның, бюджет туралы мәліметтерді алу;

–әрқайсысының диаграммаларымен Excel ортасында қарау мүмкіндігі.

2.3.3 Қолданылатын техникалық құрал. Программаны орындау үшін ақпаратты қамтамасыздандыруға минималды қажет етулерге мыналар жатады:

- бос кеңістіктің жинақтаушы қатты дисктегі бастапқы орнатудың көлемі 20Мб. Онымен ары қарай жұмыс кезінде және көлемін үлкейтуде сақталынған ақпаратты үлкейту үшін бос кеңістіктің көлемін үлкейтілген ДҚ көлеміне пропорционалды түрде үлкейту керек.

- оперативті сақтау құрылғысының көлемі 64Мб төмен емес.

- ішкі жиілігі 500МГц-ден кем емес орталық процессор;

- 4Мб-тық көлемді бейнеадаптер;

- деректерді беру жылдамдығы 10/100 Мб/сек желілік адаптер (желіде бірнеше компьютер мен жұмыс кезінде).

- аппаратты қамтамасыздандыруға қажет етілетін талаптар:

- көлемі 2 Гб жинақтаушы қатты диск бос кеңістіктің көлемі;

- 256 Мб көлемнен төмен емес оперативті жадтайтын құрылғысы;

- ішкі жиілігі 2 ГГц кем емес орталық процессор;

- Мб бейне жады көлемді бейне адаптер;

- беру жылдамдығы 10/100/1000 Мб/сек желілік адаптер.

- программалық қамсыздандыруға қажет етулер:

- Windows 9x/2000/XP операциялық жүйе;

- MS Excel қосымшасының болуы.

2.3.4 Шығару және жүктеу. Программаны шығару Win 32 операциялық ортада жүзеге асады. Ішкі жүйені жүктеу Оценка эффективности.exe файл атын тышқанның сол жағын екі рет басу арқылы жүзеге асырамыз.

Программа жүктелгеннен кейін экранда программаның бас беті ашылады. Қолданушы алдымен өзіне қажет бөлімді қарайды. Әр бөлімнің аты үстінде жазылған, ішінен шикізат түрін және олардың бағасын, олардың бағаларын өсуіне байланысты бағаларын өзгерте және көре алады. Шикізаттарды қарап болған соң олардың барлық суммасы мен диаграмма түріндегі көрінісін «Диаграмма» батырмасын басу арқылы қарауға болады. Және жоғарғы оң жақ бұрышта тұрған «Excel» белгісін басуға арқылы барлық бөлім шикізат, негізгі құнның, инвестициялық шығынның, бюджеттің көрінісін Excel жұмыс ортасында көрсетіп береді. Дәл осылайша шикізаттың

ғана емес негізгі құн мен инвестициялық шығынды, бюджеттің де мәндерін өзгертіп суммасын тексеріп, диаграммаларын шығара алады.

2.3.5 Кіріс мәліметтер. Кондитер зауытының тиімділік бағасы бағдарламасы жүйесінің ішкі жүйесі үшін кіріс мәліметтер болып кілттік сөздер және диаграмма құру позициясы болып табылады.

2.3.6 Шығыс мәліметтер. Шығыс мәліметтер кілттік сөздерден табылған мәлімет және соған байланысты ақпарат болып табылады. Сонымен қатар кондитер зауытындағы бағдарламадағы керекті позициясы болып табылады.

2.3.7 Программалау ортасын таңдау. Программалау тілінің арасында ең көп тараған тілдердің бірі Си универсалды программалау тілі болып табылады. Оның системалық программаларда, трансляторларда, операциялық системаларды жазуда қолдану жақсы жетістіктерге жеткізеді.

1972 жылы Деннис Ритчи (Bell Laboratories, АҚШ) UNIX операциялық жүйесін жасау кезінде жұмыс инструменті ретінде Си тілін құрастырған болатын. Тілдің өте ыңғайлы болуы, әртүрлі операциялық жүйелер (мысалы, UNIX, MS DOS, OS/2, Windows және т.б.) үшін компиляторы жұмыс істей алатындығы оның тез арада пайдаланушылар арасында кең тарауына себепші болды. Басқа тілдер, мысалы, Алгол, Ада, ПЛ/1 және т.б. халықаралық келісім бойынша жасалынған болса, Си тілі өздігінен тарап кетті; қазіргі кезде ең көп тараған тілдің бірі.

Си тілінің ерекшелігі - программаның тексеруге жеңіл бөліктерге оңай бөлінетіндігі, жаңа пайдаланушы типтері мен кластардың (сыныптардың) енгізілуі, олардың көмегімен деректерді ашу, жадты басқару және операциялар мен функцияларды қайта жүктеу мүмкіншіліктерінің болуы.

Си тілінің тағы бір ерекшелігі – оның негізінде объектіге бағдарланған С++ тілінің құрастырылғандығы. С++ тілі қазіргі кезде программашылар арасында ең кең тараған тіл; оның негізінде С# тілі құрастырылған. Сонымен бірге, С++ тіліне арналған интегралдық орталарда Си тілінің программаларын компиляция жасауға және орындауға болады. Мысалы, төменде келтірілген Си тіліндегі барлық программалар Turbo С++ интегралдық ортасында қарастырылып, орындалған. Сонымен бірге, бұл тіл үшін Borland С++ интегралдық ортасын да қолдануға болады.

Си тіліндегі программа әрқайсысы жеке есептерді шешуге арналған бір немесе бірнеше функциялардан құралуы мүмкін. Мұндай мүмкіншілік программаларда дайын функцияларды пайдалануға жағдай жасайды. Бұл Си тілінде программа құрастыруды жеңілдетеді әрі оның көлемін азайтуға жағдай жасайды.

Кез-келген тіл белгілі бір белгілеулер мен таңбалардың жиынынан тұратыны белгілі. Оны тілдің әліппесі деп атайды. Алгоритмдік тілдердің көшілігінің әліппелері бір - біріне өте ұқсас, тек кейбір айырмашақтары ғана болады. Си мен Паскаль тілдерінің әліппелерінде де ұқсастық өте көп.

Си тілінің әліппесінің құрамына енетін негізгі белгілеулер:

- латын әліппесінің бас және кіші әріптері (A,B,C, . . . ,Z, a,b,c, z);
- цифрлар (0,1,2,3,4,5,6,7,8,9);
- арнайы белгілер: “ , { } []. () | + - / % \ ; ‘ . : ? < = > _ ! # & *
- көрінбейтін таңбалар: ақ жол, табуляция, жаңа жолға өту.

Си тілінде лексемалардың алты сыныбы қолданылады:

- идентификаторлар,
- қызметтік (кілттік) сөздер,
- константалар,
- жолдар (жол константалары),
- операциялар,
- бөлгіштер.

Идентификаторлар. Си тілінің идентификаторы дегеніміз пайдаланушының қалауы бойынша құрастырылатын, латын әрпімен басталатын әріптердің, цифрлардың және астын сызу белгісінің тізбегі. Мысалы,

ALMATY, TU204, Time_Out, X15

Идентификатордың ұзындығы компиляторға байланысты, кейбіреуі 31 таңбаға дейін қабылдаса, ал кейбіреуі тек 8 алғашқы таңбаны ғана қабылдайды., ал қалған таңбаларды ескермейді. Идентификатор константаларды, айнымалыларды, функцияларды және т.б. белгілеу үшін қолданылады. Идентификатор ретінде тілдің арнайы қызметтік (кілттік) сөздерін пайдалануға болмайды.

Қызметтік (кілттік) сөздер. Пайдаланушы идентификатор ретінде пайдалануына болмайтын резервтегі сөздер қызметтік сөздер деп есептеледі. Олар деректердің типтерін, жад сыныптарын, типтердің квалификаторларын, модификаторларды, псевдоайнымалыларды және операторларды анықтайды.

Деректердің типтерін белгілеу үшін типтердің спецификаторлары (char, double, enum, float, int, long, short, struct, signed, union, void, unsigned, typedef) мен квалификаторлары (const, volatile) қолданылады. Типтің спецификаторлары Паскаль тіліндегідей қызмет атқаратын болса, ал типтің квалификаторлары объектіге байланысты ерекше өндеудің қажет екендігін компиляторға ескертеді.

Жад кластарын (сыныптарын) белгілеу үшін auto (автоматты), extern (сыртқы), register (регистрлік), static (статикалық) қызметтік сөздер қолданылады.

Операторларды белгілеу үшін келесі қызметтік сөздер қолданылады: break, continue, do, for, goto, if, return, switch, while. Бұл қызметтік сөздердің де Паскаль тілінің қызметтік сөздерімен ұқсастықтары бар екендігіне көз жеткізуге болады. Оларды келесі амалдарды орындауға пайдаланады:

- break – циклдан немесе ауыстырғыштан шығу;
- continue – циклдағы итерацияны (қайталауды) аяқтау;
- do – орындау (шарт соңында берілетін циклдың басын көрсетеді);

- for – үшін (параметрлік циклдың басы);
- goto – шартсыз өту операторын белгілейді;
- if – егер (шартты өту операторында қолданылады);
- return – қайту (функция орындалғаннан кейін кері қайтуды қамтамасыз етеді);
- switch – ауыстырғыш (тармақталған алгоритмді программалауда қолданылады);
- while - әзірше (циклдық операторды құрастыруда қолданылады).

Қызметтік сөздерге келесі идентификаторларды да жатқызуға болады:

- default–switch операторында қажетті вариант жоқ болғанда орындалатын амалды анықтайды;
- case-switch операторында вариантты анықтайды;
- else–if операторына кіретін, тармақталған алгоритмнің альтернативті немесе қарсылас бұтағын анықтайды;
- sizeof – операнданың байт бойынша көлемін анықтайды. Операнда дегеніміз операцияға қатынасатын шамалар.

Жоғарыда аталынған қызметтік сөздердің атқаратын функциялары мен орындалу тәртіптері туралы мәліметтер кейінірек, оларды қолдану барысында, толығырақ беріледі.

Ескеретін жай, егер идентификатор астын сызу белгісімен басталса, онда ол кітапхана мен компиляторларда қолданылады, сондықтан оларды пайдаланушы идентификаторы ретінде пайдалануға тыйым салынған. Сонымен бірге, қызметтік сөздердің тізімі компилятордың типіне байланысты кеңейтілуі мүмкін. Сондықтан белгілі бір компиляторды пайдаланғанда осы ерекшелікті ескеруі қажет.

Константалар. Анықтамасы бойынша, константалардың мәндері программаның орындалу барысында өзгермейді. Си тілінің синтаксисі бес түрлі константаларды анықтайды:

- символдар;
- нақты және бүтін сандар;
- саналатын типтегі константалар;
- нольдік көрсеткіш («null- көрсеткіш»).

Константалардың аттарын үлкен әріптермен жазған ыңғайлы.

Символдар. Ішкі кодтары бар жеке таңбаларды бейнелеу үшін символдық константалар қолданылады. Әрбір символдық константа апострофпен шектелген таңба. Апостроф ішіне мәтіндік режимде дисплейге немесе принтерге шығаруға болатын кез-келген таңбаны жазуға болады. Бірақ ЭЕМ-де, дисплейде, клавиатурада немесе принтерде графикалық бейнесі жоқ кодтар да қолданылады. Мысалы, курсорды жаңа жолға немесе жолдың басына өткізу кодтары. Программада осыларға сәйкес символдық константаларды бейнелеу үшін графикалық бейнелері бар бірнеше символдардың комбинациясы қолданылады. Олар \ таңбасынан басталады.

Оларды Си тілінде басқарушы тізбектер деп атайды. Төменде олардың тізімі келтірілген:

- '\n' - жаңа жолға өту;
- '\t' - горизонтал бағыттағы кестелеу;
- '\r' - курсорды жол басына шығару;
- '\|' - кері қисық;
- '\"' - апостроф;
- '\"' - тырнақша;
- '\0' - нольдік символ;
- '\a' - сигнал-қоңырау;
- '\b' - курсорды бір символға кері қайтару;
- '\f' - бетті ауыстыру;
- '\v' - вертикал бағытта кестелеу;
- '\?' - сұрақ белгісі.

Символдық константа бүтін типке жатады; программаларда символдар бүтін мәнді операндалар ретінде пайдаланылады.

Операция белгілері. Өрнектерді құрастыру үшін және олардың мәндерін табу үшін операциялар қолданылады. Бір операцияны бейнелеу үшін бірнеше таңбалар қажет болады. Сондықтан олардың орындалу тәртібін орнату үшін олардың приоритеттерін (басымдылықтарын) білу қажет. Бірдей рангтағы операциялардың орындалу тәртібі жазылу тәртібімен анықталады.

Операциялардың приоритеттері (басымдылықтары) келесі кестеде көрсетілген:

анг	Операциялар	Ассоциативтігі
	() [] . ^ →	→
	! ~ + - ++ -- & * (тип) sizeof	←
	* / % (мультипликативті бинарлы)	→
	+ - (аддитивті бинарлы)	→
1	2	3
	<< >> (разряд бойынша жылжу)	→
	< <= >= > (қатынастар)	→
	== != (қатынастар)	→
	& (разряд бойынша конъюнкция “ЖӘНЕ”)	→
	^ (разряд бойынша болғызбау “НЕМЕСЕ”)	→
0	(разряд бойынша дизъюнкция “НЕМЕСЕ”)	→
1	&& (конъюнкция “ЖӘНЕ”)	→
2	(дизъюнкция “НЕМЕСЕ”)	→

3	?: (шартты операция)	→
4	= *= /= %= += -= &= ^= = <<= >>=	←
5	, (“үтір” операциясы)	→

Кестеде келтірілген кейбір терминдерге түсініктер берілсін. Бинарлық дегеніміз екі операнда қатынасатын, ал унарлық дегеніміз бір операнда қатынасатын операция. Мысалы, екі шаманың қосындысын табу $a+b$ - аддитивті, ал $a*b$ - мультипликативті бинарлы операция деп аталынады.

Операция белгілерінің көпшілігі жеке лексемалар ретінде қарастырылады. Кейбір белгілер әртүрлі мағыналы болуы мүмкін. Мысалы, $\&$ белгісі бинарлық операцияда разряд бойынша конъюнкцияны (және) көрсетсе, ал унарлық операцияда объектінің адресін анықтайды. Осы сыяқты, $-$ (минус белгісі) унарлық операцияда теріс таңбаны көрсетсе, ал бинарлық операцияда алу амалын орындайды.

Кейбір операциялардың орындалуы туралы қысқаша мәлімет берілсін:

а) инкремент операциясы $++$ операнданың мәнін бірге өзгертеді; оның екі формасы болады: префикс және постфикс операциялары. Префикс операциясы операнданың мәнін оны пайдаланғанға дейін бірге үлкейтеді, ал постфикс операциясы - пайдаланғаннан кейін үлкейтеді;

ә) декремент операциясы $--$ инкремент операциясына $++$ ұқсас, тек ол операнданың мәнін бірге кішірейтеді;

б) меншіктеу белгісі $=$ меншіктеу операциясында қолданылады;

в) қосылғаннан кейін меншіктеу операциясының белгісі $+=$; мысалы, $A+=B$ дегеніміз $A=A+B$;

г) алу операциясы орындалғаннан кейін $-=$ меншіктеу; мысалы, $A-=B$ дегеніміз $A = A-B$.

Ал қалған операциялардың кейбіреулерінің орындалу тәртіптері математика пәні мен басқа алгоритмдік тілдердегідей. Мысалы, Паскаль тіліндегідей. Ал кейбіреулерінің ерекшеліктері оларды пайдалану барысында көрсетілетін болады[16].

Си тілінің стандартты функцияларының кітапханасы туралы қысқаша мәлімет кітаптың соңындағы қосымшада берілген (Қосымша В). Стандартты функциялар туралы түсінік жоғарыда берілген болатын. Толық мәліметтерді Си тіліне арналған анықтамалардан алуға болады.

Си тілінің стандартты кітапханасы өте бай, ол бірнеше файлдарда сақталынатын келесі бөлімдерден тұрады:

- математикалық функциялар (math.h файлында орналасқан);
- символдарды тексеру және өзгерту функциялары мен макростары (type.h файлында орналасқан);

- стандартты ағымдарды енгізу-шығаруға арналған функциялар (stdio.h файлында орналасқан);
- жолдармен жұмыс істеуге арналған функциялар (файлдар: string.h, stdlib.h);
- жадты бөлу және босатуға арналған функциялар (файлдар: alloc.h, stdlib.h);
- мәтіндік режимде терминалдармен жұмыс істеуге арналған функциялар (conio.h файлында орналасқан);
- арнайы функциялар (әртүрлі файлдарда орналасады);
- файлдармен жұмыс істеуге арналған функциялар (stdio.h файлында орналасқан);

Егер программада осы функциялардың кейбіреулері пайдаланылатын болса, онда олардың мәтіндерін программаға енгізу үшін препроцессорлық директива (#include) қолданылады. Кейбір программа стандартты функцияларды енгізетін директивалардан ғана тұруы мүмкін.

Стандартты функциялар кітапханасы барлық алгоритмдік тілдерде қарастырылған; оларды пайдаланудың тәртіптері жоғарыда, Паскаль тілі туралы мәліметтерде, берілген болатын. Си тілінде стандартты функцияларды пайдалану әдістері де осыған ұқсас, әрине олардың әрқайсысында өздеріне тән ерекшеліктері бар[17].

2.4 Жүйені визуалды модельдеу

Объектілі – бағытталған программалық жүйелерді жобалау және талдау модельдерін құру үшін визуалды моделдеу тілдерін қолданады. Салыстырмалы түрде алғанда жақында ғана, 1989-дан 1997 жылдар аралығында пайда болған бұл тілдер дамуының көрнекті тарихына ие.

Қазіргі уақытта визуалды модельдеу тілдерінің үш ұрпағы бар. Бірінші ұрпақ 10 тілден құралған болса, екінші ұрпақтың саны 50 тілден асқан. 2-ші ұрпақтың анағұрлым әйгілі тілдерінің арасынан мыналарды айқындауға болады: Буч тілі, Рамбо тілі, Шлеер Меллор тілі, Джекобсон тілі, Коад-Иордан тілі және т.б әрбір тілі өзінің айқындылық құралдарын енгізді, өзіндік синтаксисі мен семантикалық негізге алды, басқа сөзбен жалған және қайталанбас тіл рөліне ие болды. Нәтижесінде өндірушілер бір-бірін түсінбей бастады. Тілдерді унифицирлеу (сәйкестендіру) қажеттілігі пайда болады.

Унифицирлеу жетістіктері 3-ші ұрпақ тілдерінің пайда болуына әкелді. Үшінші ұрпақ тілінің стандарты ретінде Unified Modeling Language (UML) қабылданған, ол 1994-1997 жылдарда (негізгі өндірушілер – үш “amigos” Г.Буч, Дж. Рэмбо, Джекобсон) құрылған. Қазіргі уақытта UML 1.4 версиясы өндірілген[18].

2.3.1 Модельдеудің бірыңғай тілі. UML – объектілі-бағытталған программалық жүйелерді жобалау, тал-дау және жүзеге асыру модельдерін

жазуға арналған стандартты тіл. UML программалық жобалар нәтижелерін құжаттау, құрастыру, спецификациялау және көрнекті ету үшін қолданылуы мүмкін. UML – бұл программалаудың көрнекті емес тілі, бірақ оның модельдері (Java, C++, Visual Basic, Ada 95, Object Pascal) программалау тілдерінде мәтінге және реляциялық деректер қоры үшін кестелерге де тікелей тасымалданады.

UML сөздігі үш түрлі құрылыс блоктарын құрайды: заттар, қатынастар, диаграммалар.

Заттар модельде негізгі элемент болып табылатын абстракциялар, қатынастар арқылы байланысқан заттар модельдің негізгі элементтері, ал диаграммалар заттар жиынтығын топтастырады[19].

2.3.2 UML-дегі диаграммалар. Диаграммалар – көптеген элементтердің графикалық көрінісі, заттар төбелерінен және қатынастар доғаларынан байланыстырушы графтар ретінде бейнеленеді. Диаграммалар түрлі көзқарас жағынан жүйелерді көрнекті ету үшін салынады, содан соң олар жүйеден көрініс табады. Әдетте диаграмма жүйені құрайтын элементтердің толық емес көрінісін береді. Бір элемент барлық диаграммаларда қолданылса да, тәжірибе жүзінде ол тек бірнеше диаграммаларда ғана пайда болады. Теория жағынан диаграмма кез келген заттар мен қатынастар жиынтығынан құрылуы мүмкін, тәжірибе жүзінде программалық жүйе архитектурасының бес көрінісіне сәйкес келетін жиынтықтардың аз ғана санымен шектеледі. Осы себептен UML-де тоғыз түрлі диаграмма бар:

Кластар диаграммалары кластар, интерфейстер, бірлесіп орындау және олардың қатынастарының жиынын көрсетеді. Объектілі-бағытталған жүйелерді модельдеу кезінде кластар диаграммалары өте жиі қолданылады. Кластар диаграммалары жүйенің статистикалық жобалық көрінісін қамтамасыз етеді. Активті кластарды қосатын кластар диаграммалары жүйе процестерінің статистикалық көрінісін қамтамасыз етеді.

Объектілер диаграммасы объектілер мен олардың қатынастарының жиынын көрсетеді. Объектілер диаграммасы кластар диаграммаларында орналасқан заттар даналарының статистикалық “сол мезеттегі түсірімін” береді. Кластар диаграммалары сияқты бұл диаграммалар жүйе процестерінің статистикалық жобалық немесе статистикалық көрінісін (нақты немесе фототипті жағдайлар көзқарасы жағынан) қамтамасыз етеді.

Use Case диаграммасы (прецеденттер диаграммасы) Use Case элементтері, актерлері мен олардың қатынастары жиынын көрсетеді. Use Case диаграммалары көмегімен жүйе үшін Use Case-тің статикалық көрінісі құрылады. Бұл диаграммалар жүйе тәртібін модельдеу мен ұйымдастыруда, тапсырыс берушінің жүйеге қоятын талаптарын беруде өте маңызды.

Тізбектілік диаграммасы және бірлесіп орындау диаграммасы – бұл өзара әрекеттесу диаграммаларының түрлері.

Өзара әрекеттесу диаграммасы объектілер мен олардың арасындағы қатынастар, сондай-ақ объектілер арасындағы бір-біріне жіберілетін хабарламалар жиынын қосатын өзара әрекеттесулер. Өзара әрекеттесу диаграммалары жүйенің динамикалық көрінісін беруді қамтамасыз етеді.

Тізбектілік диаграммасы – уақыт бойынша хабарламалардың реттеуін айқындайтын өзара әрекеттесу диаграммасы.

Бірлесіп орындау (прецеденттер) диаграммасы – бұл хабарламаларды жіберетін және қабылдайтын объектілерді құрылымдық ұйымдастырудың өзара әрекеттесу диаграммасы. Тізбектілік диаграммалары мен бірлесіп орындау диаграммалары изоморфты, яғни бұл бір диаграмманы басқа диаграммаға тасымалдауға болатынын білдіреді.

Күйлер схемаларының диаграммалары ақырлы автоматты көрсетеді, күйлер, ауысымдар, оқиғалар мен әрекеттерді көрсетеді. Күйлер схемаларының диаграммалары жүйенің динамикалық маңыздылығы. Бұл диаграммалар реактивті жүйелерді модельдеу кезінде өте пайдалы оқиғалармен басқарылатын объект тәртібін айқындайды.

Іскерлік диаграммасы- жүйе ішінде әрекеттен әрекетке дейінгі ағынды көрсететін күйлер схемалары диаграммаларының арнайы түрі. Іскерлік диаграммалары жүйенің динамикалық көрінісін қамтамасыз етеді. Олар жүйенің функционалдығын модельдеу кезінде өте маңызды және объектілер арасындағы басқару ағынын айқындайды.

Компоненттік диаграмма компоненттер жиынының ұйымдастырылуын және компоненттер арасындағы тәуелділіктерді көрсетеді. Компоненттік диаграммалар жүйенің жүзеге асырылуының статикалық көрінісін қамтамасыз етеді. Олар кластар диаграммаларымен бір немесе бірнеше кластар, интерфейстер немесе кооперациялар арқылы бейнеленетін компонентпен байланысады.

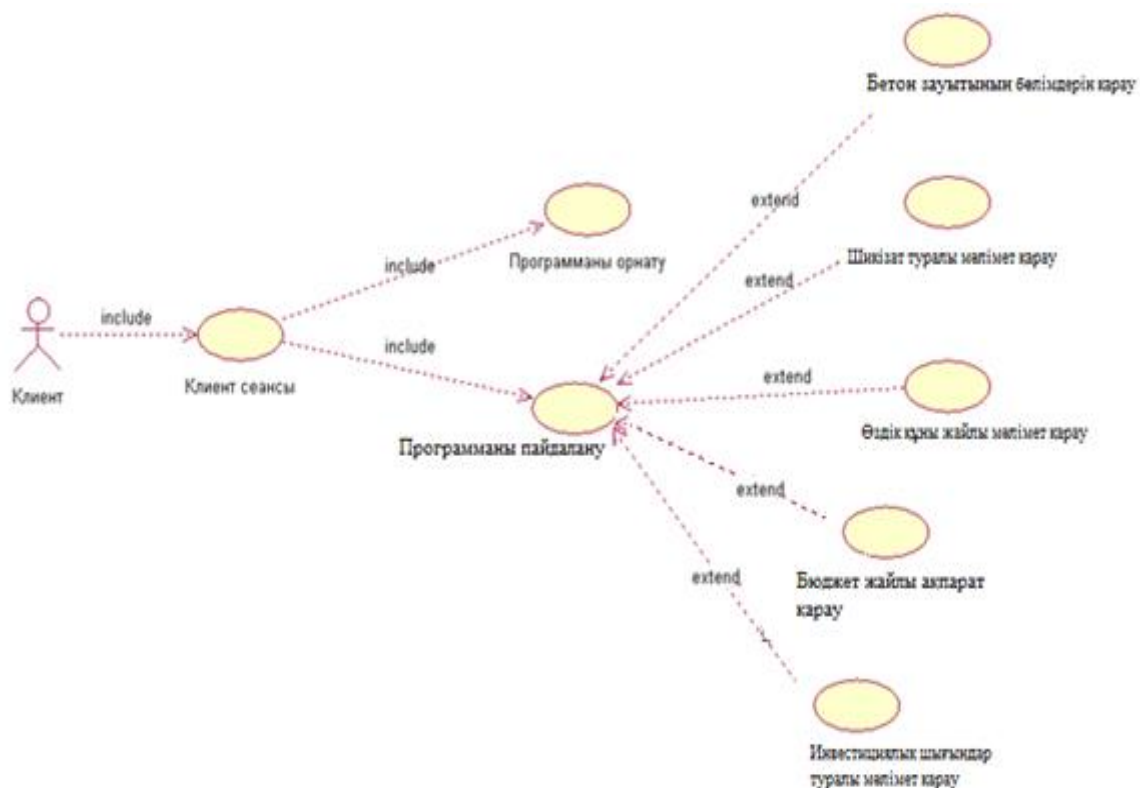
Орналастыру диаграммасы (ашу диаграммасы) орындау кезеңінің өңдеуші түйіндердің конфигурациясын, сондай-ақ оларда өмір сүретін компоненттерді көрсетеді.

Орналастыру диаграммалары жүйенің орналасуының статикалық көрінісін қамтамасыз етеді. Олар түйінге бір немесе бірнеше түйін қосылу мағынасында компоненттік диаграммалармен байланысады[20].

2.3.3 Прецеденттерді модельдеу. Прецеденттер диаграммасында жұмыс қалай жүретіндігі анық көрсетіледі. Жүйенің жүргізуі – сыртқы қолданушыға көрсетілгендіктен – прецеденттер түрінде бейнеленеді. Прецеденттер моделін абстракцияның әртүрлі деңгейінде құруға болады. Талдау сатысында прецеденттер жүйе не істеу керек және не істейтіндігіне сүйеніп, өздеріне жүйелік талаптарды алады.

Прецедент сыртқы субъект қадағалайтын және кейіннен құрастыру процесі кезінде жеке тестіленетін бизнес-функцияны орындайды. Субъект (актер) – бұл соңында пайдалы қорытынды алуды көздейтін прецедентпен

қатынасатын біреу немесе бірнәрсе. Прецеденттер диаграммасы 2.1 суретте көрсетілген.



2.3.3 сурет– Прецеденттер диаграммасы

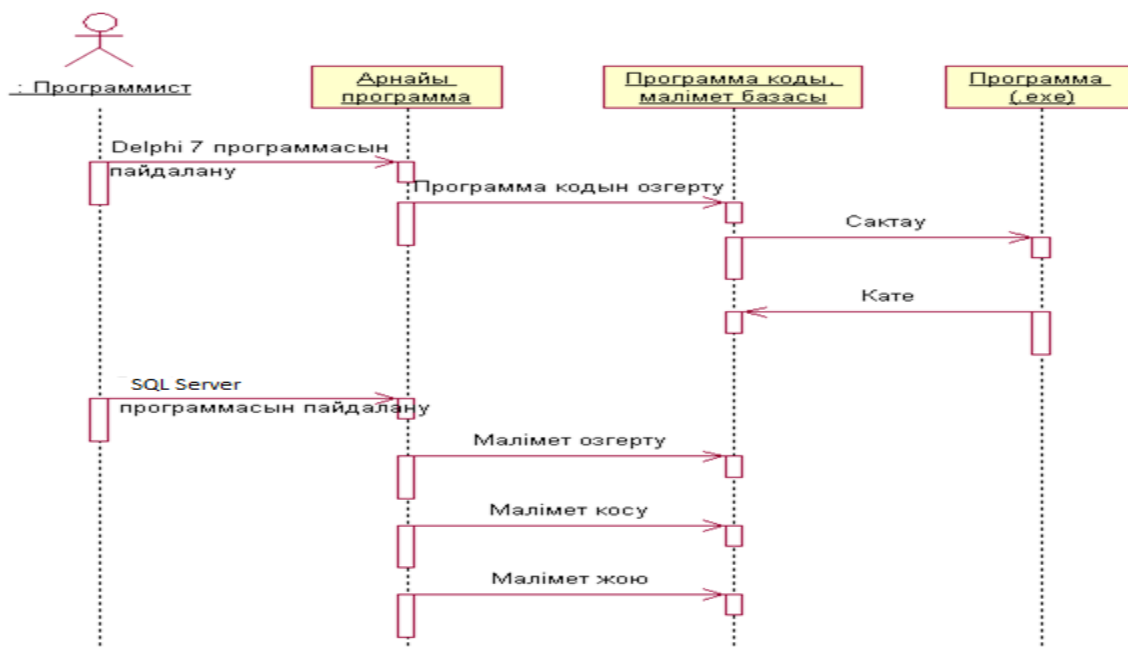
2.3.4 Тізбек диаграммасы. Жарамдылықтың талдаудың мақсаты (алдын ала жобалау) – объектілерді шығару. Бұл сатыда бөлшектеп жобалау объектілер арасындағы программаның функцияларын тарату жүреді. Тізбек диаграммасы бөлшектеп жобалаудың негізгі элементі болып табылады (объектілік модельдің динамикалық бөлігі). Тізбектелген диаграмма объектілер арасындағы хабарламалардың ауысуын уақытша тізбек түрінде реттеп көрсетеді.

Тізбек диаграммасы – жобалаудың негізгі жұмысшы өнімі болады. Әр прецедент үшін негізгі және альтернативті тізбектелген іс - әрекетті сипаттайтын диаграммалар тұрғызылады. Нәтижесінде динамикалық модельдің ядросын аламыз. Онда жүйенің іс – қимылының уақыты және осы іс – қимылдың жасалуы анықталған. Тізбек диаграммасы мынадай негізгі элементтерден тұрады:

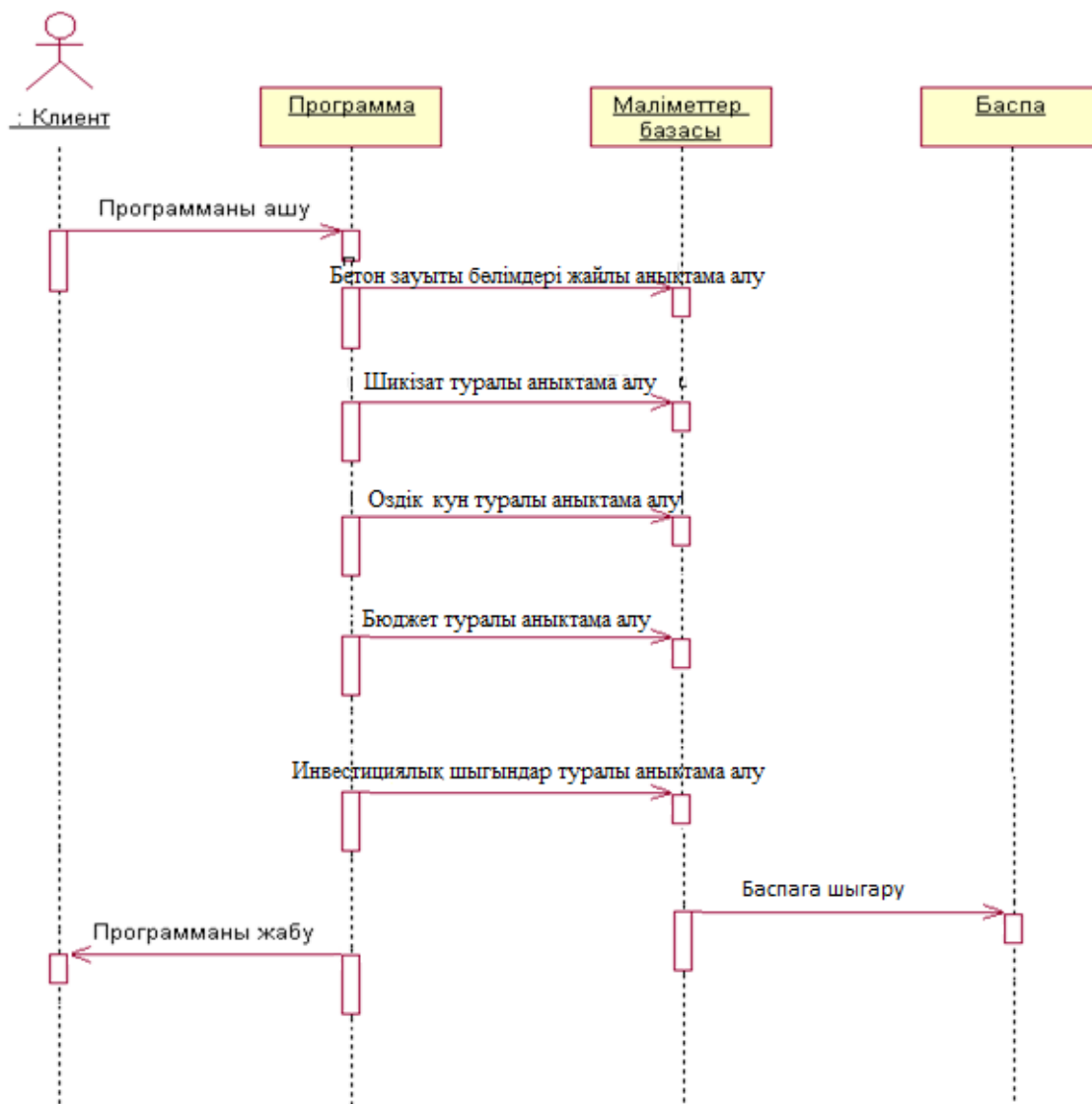
- а) прецеденттердегі іс-қимыл тізбегінің мәтіні, онда сол жақта, жоғарыдан төмен қарай жазылады;
- б) бағытшамен сипатталатын хабарлама, онда бар объектіден келесі объектіге бағытталған;

в) тікбұрыштар түріндегі операция тәсілдері, олар үздік сызықтарда орналасқан объектілерге қатысты тәсілдерді жатқызамыз. Тік бұрыштың ұзындығын тізбекті басқарудағы көрінісін көрсетуде пайдаланамыз;

Тік бұрыш біткен нүктеге дейінгі басқаруға ие тәсіл. Тізбек диаграммасы 2.3.4.2-суретте көрсетілген[21].



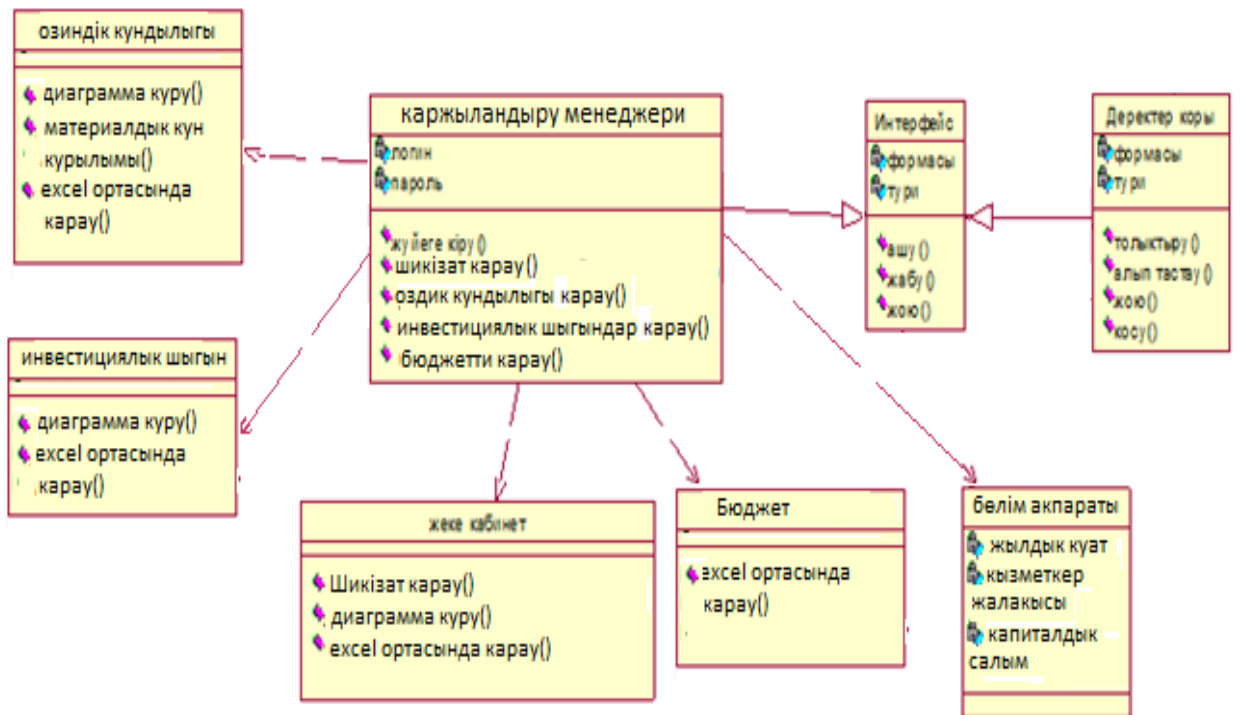
2.3.4.1 сурет- Программист үшін тізбектілік диаграммасы



2.3.4.2 сурет- Қолданушы үшін тізбектілік диаграммасы

2.3.5 Кластар диаграммасы. Класстар диаграммасы деп көптеген класстар және интерфейстер көрсетілген диаграммаларды атайды.

Жобаның класстар диаграммасында класстар аттары және олардың атрибуттары мен негізгі жасайтын операциялары 2.3.5-суретте көрсетілген:

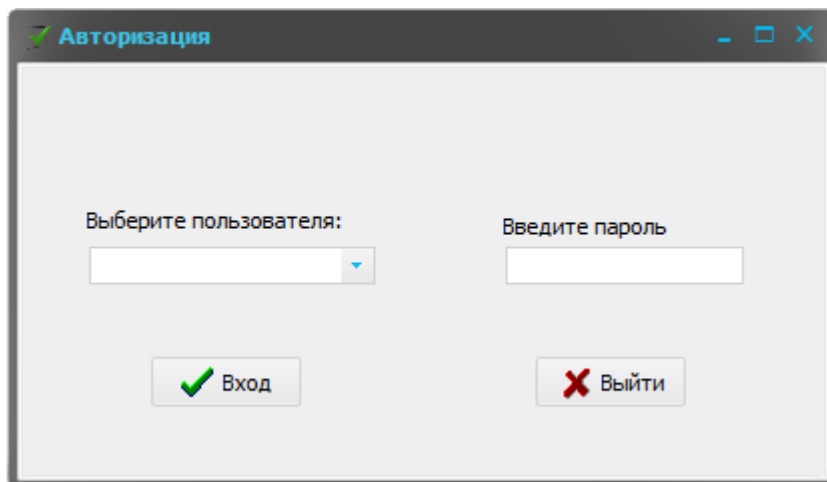


2.3.5 сурет – Кластар диаграммасы

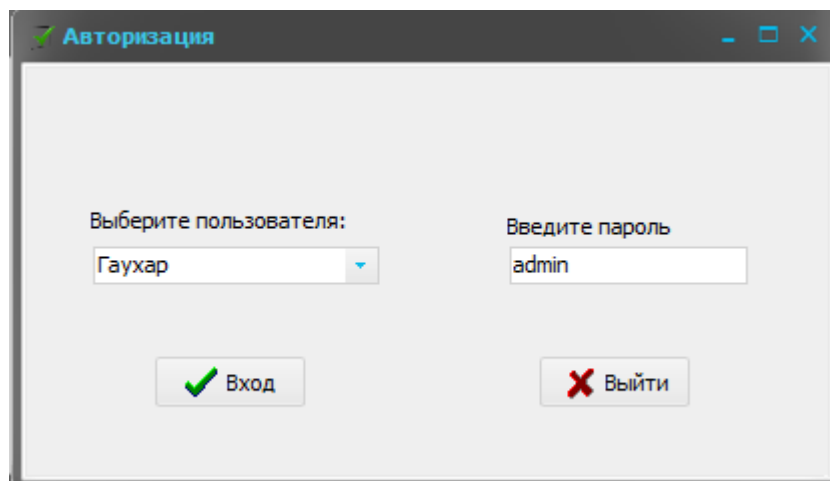
3 Эксперименталдық бөлім

3.1 Пайдаланушы нұсқаулығы

Жүйенің бағдарламалық коды А қосымшада көрсетілген.



3.1 сурет- Бағдарламаны ашқан кезде Пайдаланушы мен Құпия сөз енгізу керек



3.1.1 сурет- Есімі мен құпия сөзді енгізгеннен кейін бағдарламаның басты беті ашылады

Оценка эффективности

Сырье Себестоимость Инвестиционные затраты Бюджет

Код	Наименование	Количество, тн	Количество, %	Годовая потребность,
	Сахар	1,30	63	130,00
3	Каранель	0,20	10	20,00
6	Молочные снеси	0,01	0	1,00
11	Желе	0,05	2	5,00
12	Загустители	0,50	24	50,00
		2,06	99	206,00

Годовая мощность, тонн

3.1.2 сурет- Бағдарламаның басты беті

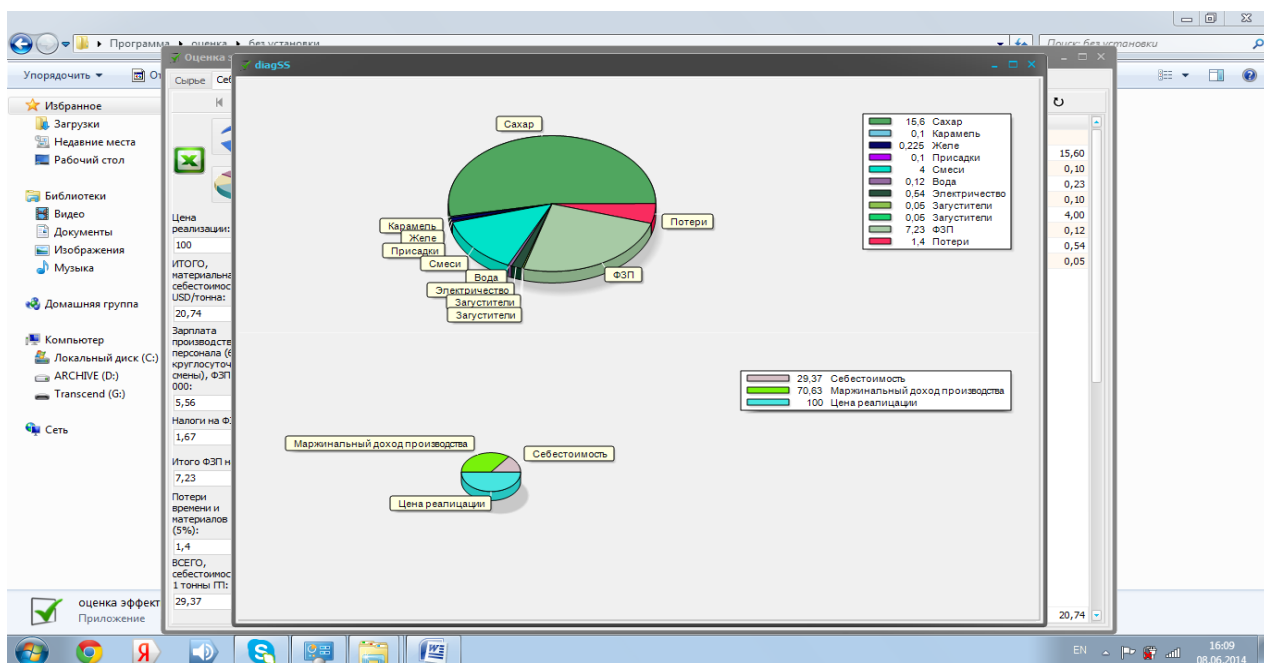
Оценка эффективности

Сырье Себестоимость Инвестиционные затраты Бюджет

Код	Наименование	Количество, тн	Количество, %	Годовая потребность,
	Сахар	1,30	4	780,00
3	Каранель	0,20	1	120,00
6	Молочные снеси	0,01	0	6,00
11	Желе	0,05	0	30,00
12	Загустители	0,50	2	300,00
13	Какао	2,00	6	1200,00
14	Масло	1,00	3	600,00
15	Мука	1,00	3	600,00
16	Разрыхлитель	1,20	4	720,00
17	Ванилин	5,00	15	3000,00
18	Корица	1,30	4	780,00
19	Орехи	2,10	6	1260,00
20	Цукаты	1,50	5	900,00
21	Сода	1,60	5	960,00
22	Кавосы	2,60	8	1560,00
23	Мед	2,00	6	1200,00
24	Куржун	3,00	9	1800,00
25	Яйцо	2,00	6	1200,00
26	Масло	1,90	6	1140,00
27	Молоко	2,30	7	1380,00
Всего		32,56	100	19536,00

Годовая мощность, тонн

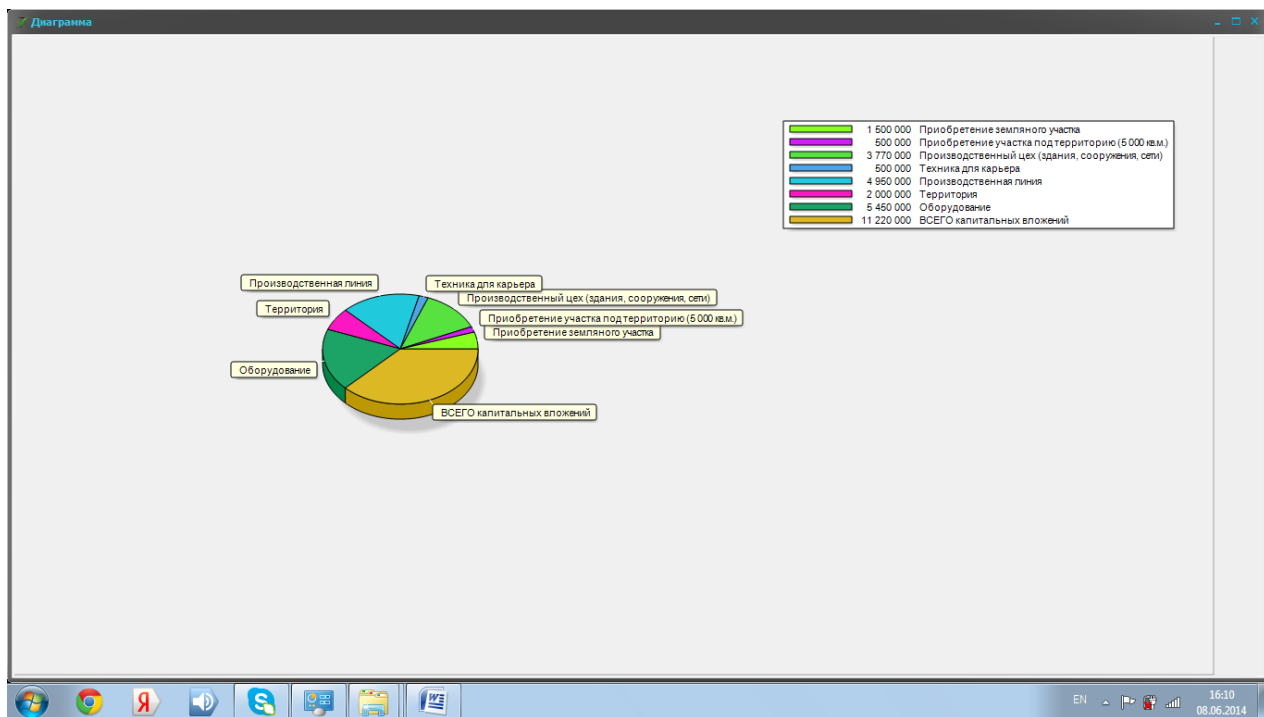
3.2 сурет- Кондитер зауыттың шикізаттары жайлы ақпарат «Сырьё» деген мәзірінде шикізат санын қосып, азайтуға болады. Бағасының мәндерін өзгерту арқылы жылдық өндірімділікті біле аламыз. «Годовая мощность» батырмасын басып жылдық қуатты өзгертуге болады



3.5 сурет- Өзіндік құнның диаграмма түріндегі сызбасын «Построить диаграмму» батырмасын шерту арқылы алуға болады. Екінші астыңғы диаграммада Қосымша пайда (маржинальный доход), Өзіндік құн, Құрастыру құны көрсетілген

Код	Наименование	Сумма, USD
1	Приобретение земельного участка	1500000
3	Приобретение участка под территорию (5 000 кв.м.)	500000
4	Производственный шах (здания, сооружения, сети)	3770000
6	Техника для карьера	500000
7	Производственная линия	4950000

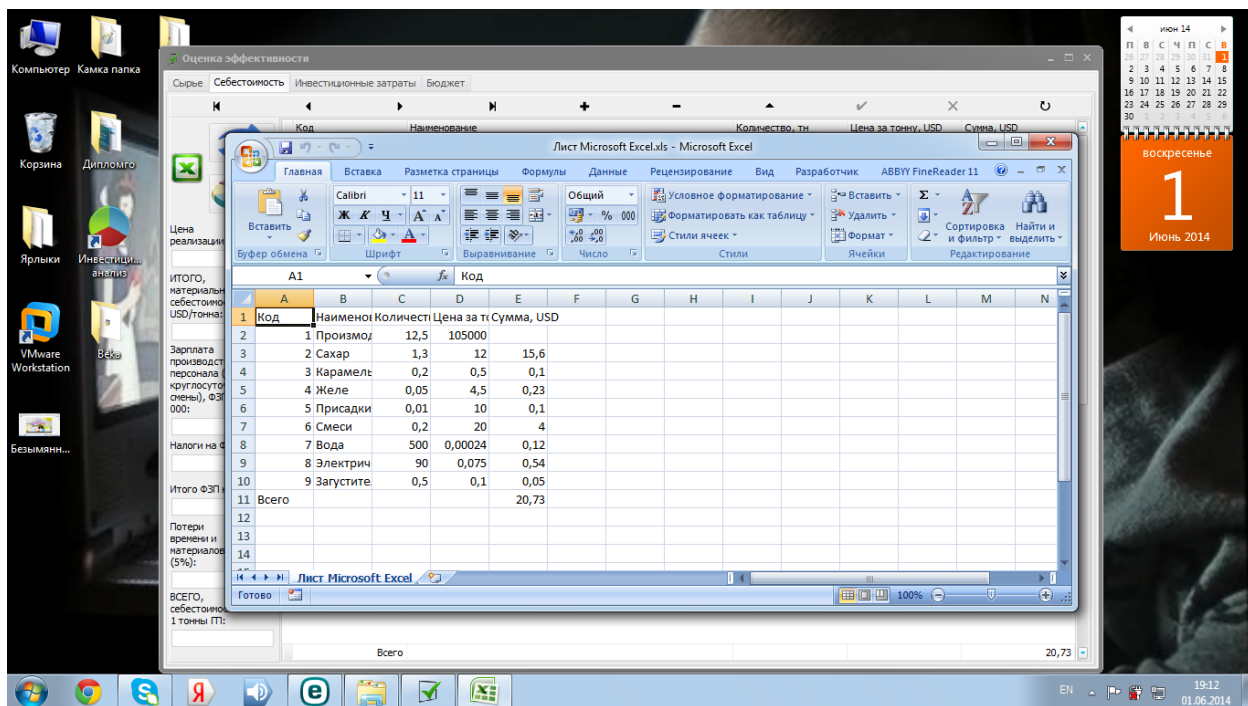
3.6 сурет- Инвестициялық шығын түрлерінің көрсетілімі. «Обновить» батырмасын басып Территория, Құрал-жабдықтар үшін, Барлық капиталдық салымдарды көруге болады



3.7 сурет- «Инвестиционные затраты» диаграммасы

код	Наименование	2013	2014	2015	2016	2017	2018	2019
1	Потребность, тонн	130000,00	130000,00	120000,00	120000,00	120000,00	120000,00	120000,00
2	Реализация, USD (без НДС)	13000,00	13000,00	12000,00	12000,00	12000,00	12000,00	12000,00
3	Себестоимость USD (без НДС)	3796,00	3796,00	3504,00	3504,00	3504,00	3504,00	3504,00
4	Затраты на логистику (16 USD/ton)	960,00	960,00	960,00	960,00	960,00	960,00	960,00
5	ФОТ непроизводственного персонала	120,00	120,00	120,00	120,00	120,00	120,00	120,00
6	Налоги С ФОТ	36,00	36,00	36,00	36,00	36,00	36,00	36,00
7	Накладные расходы	379,60	379,60	350,40	350,40	350,40	350,40	350,40
8	основного производства (10%)	379,60	379,60	350,40	350,40	350,40	350,40	350,40
10	Командировочные расходы	5,00	5,00	5,00	5,00	5,00	5,00	5,00
11	Услуги связи	3,00	3,00	3,00	3,00	3,00	3,00	3,00
12	Канцтовары, хозтовары	2,00	2,00	2,00	2,00	2,00	2,00	2,00
13	Спецдекада	10,00	10,00	10,00	10,00	10,00	10,00	10,00
14	Электроэнергия офиса	4,00	4,00	4,00	4,00	4,00	4,00	4,00
15	Другие административные расходы	10,00	10,00	10,00	10,00	10,00	10,00	10,00
29	Административные расходы	34,00	34,00	34,00	34,00	34,00	34,00	34,00
30	Аренда помещений	0,00	0,00	0,00	0,00	0,00	0,00	0,00
31	Аренда помещений	0,00	0,00	0,00	0,00	0,00	0,00	0,00
32	ЧП до вычета налогов, амортизации и процентов	7708,40	7708,40	7029,60	7029,60	7029,60	7029,60	7029,60
33	Чистая прибыль, %	0,00	50,91	55,15	49,49	49,49	49,49	49,49
34	Расходы по вознаграждению за кредит (15%)	168,30	168,30	168,30	168,30	168,30	168,30	168,30
35	Амортизация	922,00	922,00	922,00	922,00	922,00	922,00	922,00
36	Прибыль до налогообложения от обычной деятельности	6618,10	6618,10	5939,30	5939,30	5939,30	5939,30	5939,30
37	Чистая прибыль, %	50,91	50,91	49,49	49,49	49,49	49,49	49,49
38	КПН	1323,62	1323,62	1187,86	1187,86	1187,86	1187,86	1187,86
39	Прибыль	5294,48	5294,48	4751,44	4751,44	4751,44	4751,44	4751,44
40	Рентабельность деятельности	40,73	40,73	39,60	39,60	39,60	39,60	39,60
41	Прибыль с нарастающим	5294,48	10588,96	15340,40	20091,84	24843,28	29594,72	34346,16

3.9 сурет- Бюджеттің мәзірі. «Обновить» батырмасын басса инвестициялық шығындарды АҚШ доллары (\$) түрінде шығарып береді.



3.10 сурет- Excel ортасында барлық есептеулерді шығарып беру үшін



батырмасын басу арқылы шығара алады.

3.2 Программистке нұсқаулық

Программаға енетін нұсқауларды (командаларды) Си тілінде оператор дейді.

Си тілінің операторларын қарапайым және күрделі (құрама) операторлар деп екі топқа бөледі.

Қарапайым операторлардың құрамына басқа оператор енбейді. Ал күрделі операторлардың ішіне бірнеше қарапайым операторлар енуі мүмкін. Си тілінде қарапайым операторларға: меншіктеу, көшіру, бос, енгізу және шығару операторлары жатады.

Күрделі операторларға: шартты көшіру, қайталау (цикл), таңдау, жалғастыру және т.б. операторлар жатады. Бұл операторлар тұтас бір оператор қызметін атқарады. Программада операторлар орналасқан реті бойынша (көшіру операторы болмаса) орындалады. Си тілінің тағы бір ерекшелігі – оның негізінде объектіге бағдарланған C++ тілінің құрастырылғандығы. C++ тілі қазіргі кезде программашылар арасында ең кең тараған тіл; оның негізінде C# тілі құрастырылған. Сонымен бірге, C++ тіліне арналған интегралдық орталарда Си тілінің программаларын компиляция жасауға және орындауға болады.

Программистке нұсқаулық:

- a % b – бүтінді бөлгеннен қалған қалдық
- ++a – префиксті инкремент
- a++ - суффиксті инкремент
- --a – префиксті декремент

- a-- - суффиксті декремент
- a == b - тең
- a != b - тең емес
- a > b –үлкен
- a < b - кіші
- a >= b – үлкен немесе тең
- a <= b – кіші немесе тең
- !a – логикалық жоқ
- a && b – логикалық және
- new typeи - жадты белгілеу
- new type[n]. – жаңа жадты босату
- delete a - жадты босату
- :: - көріну аймағын өзгерту
- ++ - суффиксті инкремент
- -- суффиксті декремент
- () – функцияны шақыру
- []. – массивқа элемент алу
- . – ссылақ бойынша элемент алу
- ->- бағдарша ьбойынша элемент алу
- break – циклдан немесе ауыстырғыштан шығу;
- continue – циклдағы итерацияны (қайталауды) аяқтау;
- do – орындау (шарт соңында берілетін циклдың басын көрсетеді);
- for – үшін (параметрлік циклдың басы);
- goto – шартсыз өту операторын белгілейді;
- if – егер (шартты өту операторында қолданылады);
- return – қайту (функция орындалғаннан кейін кері қайтуды қамтамасыз етеді);
- switch – ауыстырғыш (тармақталған алгоритмді программалауда қолданылады);
- while - әзірше (циклдық операторды құоастыруда қолданылады).
- Қызметтік сөздерге келесі идентификаторларды да жатқызуға болады:
- default–switch операторында қажетті вариант жоқ болғанда орындалатын амалды анықтайды;
- case-switch операторында вариантты анықтайды;
- else–if операторына кіретін, тармақталған алгоритмнің альтернативті немесе қарсылас бұтағын анықтайды;
- sizeof – операнданың байт бойынша көлемін анықтайды.
- ‘\n’ - жаңа жолға өту;
- ‘\t’ - горизонтал бағыттағы кестелеу;
- ‘\r’ - курсорды жол басына шығару;
- ‘\|’ - кері қисық;
- ‘\’ - апостроф;
- ‘\”’ – тырнақша;
- ‘\0’ – нольдік символ;

- '\a' – сигнал-қоңырау;
- '\b' – курсорды бір символға кері қайтару;
- '\f' – бетті ауыстыру;
- '\v' – вертикал бағытта кестелеу;
- '\?' – сұрақ белгісі.
- #include - мәтінді файлдан енгізу;
- #undef – макросты немесе препроцессорлық идентификаторды анықтауды болдырмау;
- #if - шарт-өрнекті тексеру;
- #ifdef - идентификатордың анықталғандығын тексеру;
- #ifndef - идентификатордың анықталмағандығын тексеру;
- #else - #if үшін альтернативтік тармақтың басы;
- #endif - #if шартты директивасының аяқталуы;
- #elif - #else#if құрама директивасы;
- #line - кейінгі төменгі жолдың номерін ауыстыру;
- #error - трансляция кезіндегі қате туралы хабардың мәтінін дайындау;
- pragma - жүзеге асыруда алдын ала белгіленген амалдар;
- # - бос директива.
- assert.h –Программаларды тексеру (диагностика)
- ctype.h – Символдарды өзгерту және тексеру
- errno.h – Қателерді тексеру
- float.h – Нақты деректермен жұмыс істеу
- limits.h – Бүтін мәнді деректердің шектік мәндері
- locate.h – Ұлттық ортаны қолдау
- math.h – Математикалық есептеулер
- setjump.h – Локальді емес өтудің мүмкіншіліктері
- signal.h – Ерекше ситуацияларды (жағдайларды) өңдеу
- stdarg.h – Параметрлердің айнымалы санын қолдау
- stddef.h – Қосымша анықтамалар
- stdio.h – Енгізу-шығару құралдары
- stdlib.h – Жалпы қызмет функциясы (жадпен жұмыс істеу)
- string.h – Символдар жолдарымен жұмыс істеу
- time.h – Даталар мен уақытты анықтау

4 Өмір тіршілік қауіпсіздігі бөлімі

4.1 Еңбек қорғаудың ұйымдастыру шаралары

Есептеу орталығы темірбетонды, бес қабатты үйдің 3-ші қабатында орналасқан. Машина залында 15 IBM PC/AT 80286 машиналары, әртүрлі принтерлер бар. Залдың көлемі (10*8*3,5) 80 шаршы метр. Бөлменің терезелері оңтүстік жаққа қараған. Бөлме іші әртүрлі түспен әшекейленген: төбесі - ақ, едені - қызғылт, қабырғалары - көгілдір. Габариттік өлшемдері 1,8*1,1 метр есік, және де 1,5*2 метрлік 3 терезе. Терезе екі қабатты әйнектен тұрады. Екі желдету жүйесі бар.

ДЭЕМ-мен жұмыс істеу барысында қызметшілер өндірістік ортаның мынадай қауіпті және зиянды факторларына ұшырайды:

- жарықтандырудың жетіспеушілігі;
- электр тогының қаупі;
- технологиялық процестегі өрт қаупі; -электромагниттік өріс.

Жұмысқа қабылданған әрбір жұмысшылармен техникалық қауіпсіздік, өндіріс тазалығы және өрт қауіпсіздігі бойынша нұсқаулар өткізіледі. Нұсқаулардың сипаты мен уақытына байланысты мынадай түрлері бар.

Нұсқау “Қауіпсіздік техникасы нұсқауына тіркеу журналына” жазылады. Қоғамдық бақылау жалпы еңбек қорғау заңдылығын бұзбау үшін пайдаланылады. Мемлекеттік қадағалау ЕС өкілдерімен жүргізіледі.

Техникалық қауіпсіздік шаралары

ЭЕМ-да жұмыс істейтін пайдаланушылар жұмыс орнында және маңайында тәртіп пен тазалық сақтауы тиіс. Компьютерлердің сыртындағы пломбыларды бүлдіруге, қозғалып немесе айналып тұрған құрылғыларға қол тигізуге, сыртқы қабығы ашық машиналармен жұмыс істеуге, ЭЕМ-дер қойылған бөлмеде тез тұтанатын сұйықтар мен газдарды, қышқылдар мен сілтілерді пайдалануға, экранды әртүрлі агрессивті сұйықтармен (ацетон, еріткіштер) сүртуге, ЭЕМ құрылғыларына бөгде нәрселерді қоюға тыйым салынады.

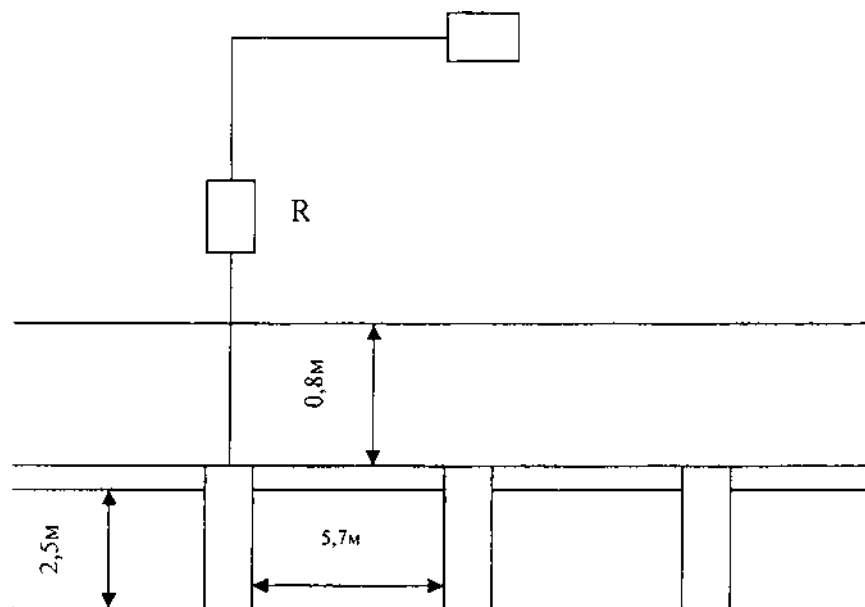
ЭЕМ-мен жұмыс істеуші пайдаланушылар:

- ЭЕМ-ның тораптық кабелін тоқ көзіне қосқанда, ажыратқанда тек штепселвилкасының корпусынан ұстауы керек;
- құрылғыны тоқ көзіне қосып қойып қараусыз тастап кетпеу керек;
- ЭЕМ-ның тораптық кабелін тоқ көзіне қосқанда, ажыратқанда ЭЕМ-ның процессоры және басқа да сыртқы құрылғылар өшірулі тұруы керек.

ЭЕМ-мен жұмыс істеу барысында дисплейдің бетінде статикалық электр тоғы пайда болады. Осы статикалық электрдің потенциалын төмендету үшін құрылғының электрленуі мүмкін барлық бөліктері жермен тұйықталады.

Құрылғының электр өткізбейтін бөліктерін, мониторды жермен тұйықтау үшін олардың бетіне жағылған өткізгіш қабықтар пайдаланылады.

Сақтану шарасы ретінде жерге қосылғыштың есептеуін 1000В-қа дейінгі кернеумен жұмыс істейтін еңбек құралын қабылдағандықтан 40м-нан аспайтынын ескеру арқылы $R_H=3,5$ Ом-ға тең екендігін есептеу арқылы ұсынамын.



4.2 сурет - Жерге қосу схемасы

Санитарлық - гигиеналық шаралар

Есептеу орталығының бөлмесінің өлшемі (ауданы, көлемі) сонда орналасқан техникалық құралдар мен жұмысшылар санына сәйкес болуы тиіс. Осыған байланысты температура, жарықтандыру, ауаның тазалығы, өндірістік шуды изоляциялау анықталады.

Санитарлық норма бойынша бір адамға 15 текше метрден кем емес өндірістік кеңістік керек. ЕО - бөлмелері ерекше талаптар қойылады. Залдың биіктігі, яғни технологиялық еден мен ілінбелі төбенің арасы 3-3,5 метр болуы тиіс. Біздің ЕО - бұл талаптарға сай. Ілінбелі төбе мен негізгі төбенің аралығы 0,5 метр. Еден астындағы кеңістіктің биіктігі 0,8 метр. Бұл норма ауа жүретін құдық пен 0,2 метр кабельді жүргізу үшін әдемі жасалған. Төбелердің арасындағы кеңістік вентиляциялық каналдар, өртке қарсы жарақтар мен жарықтандыру құрылғылар қоюға пайдаланылады. Құралдарды тасуды ескере отырып есік 1,8* 1,1 метр өлшемде жасалған. Таңдау бірқатар факторлармен анықталады: үйдің конструкциясы, істелетін жұмыстың сипаты, жарықтандыру және жұмысшылардың саны. Қабырғалар көгілдір түске боялған. Бұл түс көзді тынықтырады, шаң қонбайды әрі тазаланғанда қолайлы.

4.3 кесте- Бөлмедегі ылғалдылық нормасы

Сипаттама	Үйлесім	Мүмкін
1. Ауаның температурасы	20-23°C	19-25°C
2. Қалыпты ылғалдылық	40-60%	≤70%
3. Ауаның қозғалыс жылд.	0,2м/с	0,2 м/с

4.4 кесте- Дыбыс қысымының көрсеткіштері

Бөлме	Дыбыс қысымының деңгейлері								Дыбыс деңгейі, дб.
	3	25	50	00	000	000	000	000	
Машина Залы	1	1	4	9	5	2	0	8	50

4.5 кесте- Жарықтандыру

Бөлме	Нормаланған жарық беті	Жарық нормалары		K _п
		Біріккен	Жалпы	
ЭЕМ-дер үшін	Г – 0,8 Г – 0,8	750	400	15
Қызметшілер үшін		750	400	15

4.2 Есептеу бөлімі

Бөлмені желдету жүйесі есебі

Бөлмедегі ауа-райын анықтаймыз. Ол үшін кестеден t мен f - мәндерін табамыз: $t = 19 - 24^\circ\text{C}$ және $f = 50 - 70\%$ 4 нүкте алдық: 1 (18 : 50), 2'(18 : 70), 3'(24 : 50), 4'(24 : 70).

Жұмыстың шарты (бөлме екінші топқа жатады) бойынша (1-ші жұмыс категориясы) кестеден $t = 19-28^\circ\text{C}$, $f = 55-75\%$ аламыз.

Осылайша бөлменің ыңғайлылық шартын ескере отырып ауа-райы аймағын жасаймыз. Бұл аймақтар беттеспейді, бірақ қиылысады. Жұмыс аймағы (1-2 6 3-4) координаттарымен анықталады. Диаграммада жаз маусымындағы сыртқы ауа-райының нүктесін көрсетеміз: $t = 30^{\circ}\text{C}$ және $f = 60\%$.

Машина залына әртүрлі көздерден келіп түсетін жылулардың қосындысын есептейік.

Адамдардан бөлінетін жылуды есептеу

Бөлмеде жұмыс істейтін адамдардың максимал саны 15 адам. Кестеден $125 \text{ ккал/адам} = 525 \text{ кДж/адам}$.

$$Q_a = 15 * 525 = 7875 (\text{кДж}) \quad (4.1)$$

Күн радиациясының жылуын есептеу.

Зал 3-ші қабатта орналасқандықтан жылу ішкі терезе жақтаулары арқылы кіреді.

$$Q_{жк} = q_0 * F_0 * A_0 = 672 * (2 * 2 * 2) * 1.15 = 6182.4 (\text{кДж/сағ}) \quad (4.2)$$

мұндағы, q_0 - кестеден алынған мән,

F - әйнектелген беттің ауданы,

A - әйнектің қосарланғандығын ескеретін коэффициент.

Жасанды жарықтың шығаратын жылуын есептеу.

Бөлмеде әрқайсысы 60 Вт үш шамнан тұратын 6 шамшырақ бар.

$$N_{oy} = 1.08 (\text{кВт})$$

$$Q_0 = N * 3612 * N_{oy} = 1 * 3612 * 1.08 = 3900,9 (\text{кДж/сағ}). \quad (4.3)$$

мұндағы, N - электр энергиясының жылуға айналуын ескеретін коэффициент.

Дербес ЭЕМ-нан бөлінетін жылуды есептеу.

$$Q_g = N * 3612 * N_1 * N_2 * N_3 * N_4 = (1 * 0.5 + 2 * 0.6) + 35 / 2 * 0.25 = 2167.2 (\text{кДж/сағ}) \quad (4.4)$$

мұндағы, N - құрылғының жалпы қуаты,

3612 - жылуэлектрлік эквивалент,

$N_1, N_2, N_3, N_4 = 0.25$ арнайы суытылмайтын құрылғылар үшін.

$$Q_{\text{макс}} = Q_a + Q_{\text{жк}} + Q_o + Q_{\text{д}})$$

$$Q_{\text{макс}} = 5250 + 6182,4 + 3900,9 + 2167,2 = 17500,5 (\text{кДж/сағ}) \quad (4.5)$$

Желдету жүйесін таңдау.
Қажетті суықтық мөлшері:

$$Q_{\text{кс}} = 17500,5 \text{ кДж/сағ};$$

Жалпы өнімділігі 1000 текше метр/сағ және суықтық өнімділігі 17800 кДж/сағ болатын екі БК-2000 кондиционерлерін таңдап аламыз.

Өндірістік жарықтандыру есебі.

Бөлмеде жалпы жарықтандыру құралдары - қыздыру шамдарынан жасалған шамшырақтар пайдаланылады.

Есептегенде бөлменің жарықтануы біркелкі болғандықтан жарық ағынының коэффициентін пайдалану әдісі қолданылады.

Қыздыру шамдарына қажетті жарық ағыны

$$F_A = E_n * K_Z * S * Z / (N * O)$$

- формуласы бойынша есептеледі. Мұндағы E_n - жұмыс орнындағы нормаланған жарық; K_Z - пайдалану процесінде жарық өткізгіш материалдардың тозуы мен кірленуін ескеретін коэффициент; S - жарықталынатын беттің ауданы; Z - өте төменгі жарықтан орташа жарықталуға өту коэффициенті; N - шамшырақтар саны; O - пайдалану коэффициенті; яғни S бетіне түсетін жарықтың ықтималдылығы, ол төбеден ($P_{\text{төбе}}$) және бөлме қабырғасынан ($P_{\text{қаб}}$) шағылысу және бөлме индексі (i) коэффициенттерінен таңдап алынады.

$$i = A * B / (h * (A + B)) \quad (4.6)$$

мұндағы, A, B бөлменің ұзындығы мен ені, h - шамшырақтардың ілінуінің есептеу бетінен биіктігі.

Табылған F_l бойынша кестеден жарық ағынының сан мәніне қарай стандартты шам таңдалып алынады.

Төбе мен қабырғалардың шағылу коэффициенттері:

$$P_{\text{төбе}} = 50\% \quad P_{\text{қаб}} = 30\%$$

$$H=3,5 - 0,8=2,7$$

$$i=9*5/(2,5 *(9+5))=45/37,8=1,25 \quad (4.7)$$

кесте бойынша

$$O=0,75, E_n=100\text{лк}; K_z=1,3, Z=1,2,$$

$$F_{л100} * 1,3 * 45 * 1,2 / (6 * 0,73) = 1140(\text{лм}); \quad (4.8)$$

Кестеден жарық ағынының мәні 1100(лм) болатын “Әмбебап” типті шамшырақ тандалып алынады.

Өртке қарсы қауіпсіздік шаралары

ЭЕМ-дер өртке төзімділігі 2-ші дәрежеге жататын бөмеде орналастырылған.

Жұмыс кезінде қоректену тізбегіндегі пайда болатын қысқа тұйықталу мен монитор экранында жинақталатын статикалық электр өрттің шығуына себеп болуы мүмкін.

ЭЕМ орналасқан бөлмелерде - ОУ-2, ОУ-5 типті көмірқышқыл өртсөндіргіштерінің бірнешеуі болуы керек.

Өрт шығу жағдайында қызметшілер мен материалдық заттарды алып шығу үшін эвакуациялық жоспар болуы тиіс;

Өрт сақшылары мен техникаға қызмет көрсетушілердің телефондарының нөмірлері жазылған тақтайшалар ілінуі тиіс;

Өрт шыққанда өртті сумен, көбікті және химиялық өртсөндіргіштермен немесе басқа да ток өткізгіш сұйықтармен өшіруге тыйым салынады.

Өрт шыққанда:

- өртенген құрылғыны тоқтан ажырату керек;
- өрт сақшыларына өрт жайлы хабарлау керек;
- өртті қолда бар құралдармен (ОУ-2, ОУ-5) сөндіру керек.

Өртеніп жатқан, тоқтан ажыратылмаған құрылғыларына сумен сендіруге болмайды.

5 Экономикалық бөлім

5.1 Жұмыстың сипаттамасы мен қажеттілігінің негіздемесі

Қазақстан Республикасының бәсекеге қабілетті 50 елдің қатарынан нық орын алу жөніндегі стратегиялық міндеттің орындалуы үшін әлемдік экономиканың өсімінен тұрақты түрде асып отыратын экономикалық даму қажет. Еліміздің жоғары деңгейде өзіндік жолы бар мемлекет ретінде танылуы, оның әлеуметтік-экономикалық үстемелеп дамуы, халықтың әлауқатын, тұрмыс жағдайын жақсартудың негізі болып табылады. Бұл үшін қаржылық және әлеуметтік тұрақтылықты сақтай отырып шаруашылық жүргізудің қолайлы жолдарын туғызу қажет, халықаралық еңбек бөлінісіне, ел экономикасының өсуіне тиімді жағдай орната отырып, белсене қатысу керек.

Қазіргі заманғы ақпараттық жүйелерді жобалау құрылған программалық өнімнің экономикалық тиімділігінің есептеуінен тұрады. Ақпараттық жүйелер және компьютерлік технологиялардың көлемінің тұрақты өсуі нақтылы облыстың автоматтандыру және жобалауына заттық және еңбек шығындарын үлкеюіне алып келеді. Өз кезегінде ақпараттық жүйелер қымбат бағалы және қатаң талапты болып барады. Автоматтандырудың мәселенің шешімдері үшін жаңа ақпараттық жүйенің құрастыруынан пайданың дәл экономикалық есептеуін қолдану өте маңызды.

Осы таңда ақпараттық технологиялар нарқы тез қарқынмен дамып келеді. Әсіресе, күннен күнге жаңа бағдарламалар, ыңғайлы ақпарат жүйелері, электронды тіркеу автоматтандырылған жүйелері пайда болуда. Осы жүйелердің пайда болуы мен оны пайдаланудың адамның күнделікті өмірін оңайлануына алып келеді.

5.2 Атқарылған жұмысты бағалау

Берілген дипломдық жобада кодирерлік зауыдтың экономикалық жүйе динамикасының моделін іске асыратын бағдарламалар комплексін өңдеп шығару қажет. Сондықтан инвестиция процесінің экономикалық жүйеге әсер етуінің тиімділік бағасын талдау модельдері зерттелген. Осы модельдердің базасында экономикалық жүйенің даму процесіндегі бағаның математикалық тәсілі қолданылады.

Экономикалық өсуді жеделдетудің және өндірістің тиімділігін арттырудың басты факторы - жоғары өнімді еңбекке уәждеме жасау, белсенді еңбекке экономикалық ынталандыру. Қазақстанда экономикалық реформаларды жедел жүргізу нәтижесінде өндірістің барлық факторы ырықтандырылды, тиімділікті көтеруге бағытталған ынталандыру уәждемелері де жүргізіліп келеді. Қазір, яғни әлеуметтік-экономиканың серпінді даму кезеңінде өндіріс тиімділігін экономикалық жағынан ынталандыру, еңбектің сандық және сапалық сипаттамаларын, макроэкономикалық ұдайы өндіріс процесіндегі ынталандырудың рөлін

айқындау, еңбектің трансформациялануын жалпылау және нақтыланған түрде зерделеу өте маңызды сондықтан мәліметтер қоры және іс жүргізу өндірісі мен қол еңбегін тиімді ету үшін құрал ретінде пайдаланумен байланысты өзге де бағдарламалық қамтамасыз ету аса маңызды орын алады. Оны пайдалану нақты құжаттарды және өнеркәсіп жобаларын дайындауға қажетті уақытты қысқартуға, оларды жүзеге асыруда өндірістік емес шығындарды азайтуға мүмкіндіктер береді.

Микропроцессор базасында есептегіш техникаларының дамуы компьютерлер мен бағдарламалық қамтамасыз етуді жетілдіруге алып келеді. Көп ақпараттар ағынын өңдеуге қабілетті бағдарламалар пайда болады. Осындай бағдарламалардың көмегімен ақпараттық жүйелер құрылады. Кез-келген ақпараттық жүйенің мақсаты нысандар мен нақты әлемнің құбылыстары туралы мәліметтерді өңдеу мен олар туралы адамға қажетті ақпаратты ұсыну болып табылады.

Мәліметтер қорындағы потенциалды мүмкіндіктердің бәрін ашу үшін жұмыста алға қойған міндеттерге барынша сай келетін бағдарламалық және аппаратты құрал-жабдықтар кешенін пайдалану қажет.

Дипломдық жұмыстың өзектілігі. Қазіргі заманда кәсіпорынның инвестициалық шығындарын, қажетті шикізат көлемін, толық бюджет көлемін есептейтін, сақтайтын, мәліметке оңай қол жеткізетін және өзгертуге мүмкіншілік беретін заманға сай қажетті аса зор бағдарламма құрылады. Жобалап отырған дипломдық жұмысымың пәндік аймағы ретінде «Рахат» АҚ кондитерлік зауытын алдым. «РАХАТ» АҚ өнімі Қазақстан нарығында ғана емес, республикадан шеткері аймақтарда – Ресей, Өзбекстан, Тәжікстан, Қырғызстан, Германия, Қытай, Ауғанстан аумағына да тараған. ТМД-ның басқа республикалары мен алыс шет елдер тарапынан «Рахат» АҚ өнімдеріне деген қызығушылықтың болуына орай, компания келешекте өндіріс көлемін ұлғайтуды және экспорт көлемін арттыру жолымен өткізу нарығын кеңейтуді жоспарлауда.

«Рахат» АҚ мақсаты – өздерінің дәмді өнімдерімен адамдарға қуаныш сыйлау. Біздің қызметіміз көне дәстүрді сақтай отырып, қазіргі заманның талабына сай барынша көп адамның қажеттілігін жоғары сапалы кондитер өнімдерімен жоғары деңгейде қанағаттандыруға бағытталған. Біз керемет кондитер өнімдерін қалай шығаруды білеміз және сол мақсатта қорды арттырамыз. Қазақстан мен оның шет аймақтарындағы әр адамға өз еліміздің байлығы пен қонақжайлылығын бейнелей отырып, Қазақстандағы және Орталық Азия нарығындағы № 1 компания болуға ұмтыламыз.

Біз өз қызметкерлерімізге серіктес болғымыз келеді, сол себепті әр қызметкерімізді бағалаймыз, жұмысқа әлеуметтік бағытта жағдайлар жасаймыз, компанияның толық қанағаттанарлық әрі беделді жұмыс орны болуын қалаймыз.

Тұтынушыларға: қазақстандық және шетелдік тұтынушылардың кең шоғырының қажеттілігін сапалы тауарлармен (көтерме, бөлшек) қанағаттандыру.

Акционерлерге: бизнестің тұрақты, ашық түрде дамуын қамтамасыз ету және табыс табу үшін компанияны бірлесіп және қаржылық басқарудың тиімді механизмін құру.

Қызметкерлерге: нақты қойылған міндеттер мен жауапкершілікті бөліп беру арқылы қызметкерлердің жұмыс тиімділігін арттыру. Қызметкерлердің алға қойған міндеттерін орындауы мен компанияның жұмыс нәтижесіне байланысты оларға ынталандыру жүйесін енгізу. Бірыңғай мәдениет қалыптастыру.

Экономикалық жүйенің даму процесі үшін оның әртүрлі параметрлерінің үміт еткен сандық мәнін болжаудың өте үлкен маңызы бар.

Экономикалық процестерді модельдеу компьютерлеумен тығыз байланысты және экономикалық ғылымның шапшаң дамитын бағыты мен маңызды қосымшасы болып табылады.

Берілген дипломдық жобадағы бағдарламаның атқаратын негізгі қызметі кәсіпорынның жұмыс тиімділігін бағалау. Жұмыс түрлері ретінде мыналарды алуға болады:

а) шикізат санын есептеу, пайыздық жүйемен шикізат санын анықтау, шикізаттың жылдық мөлшерін, оның диаграмма құрылысын шығару;

ә) өзін-өзі ақтауды есептеу, материалдық жағынан өзін-өзі ақтауы, өндірістегі қызметкердің айлық ақысы, пайда түскен айлық ақы қоры, айлық ақы қоры салығы, жеке қызметкерлерге бөлінетін айлық ақы қоры, уақыт және материал шығыны, мемлекеттік кәсіпорынның 1 тонна деп есептегендегі өзін-өзі ақтауы, диаграмма құрылысын шығару;

б) инвестициялық шығындарды есептеу, аймағын есептеу, құрал-жабдықтармен қамтамасыздандыру мөлшері, барлық капиталдық салым мөлшері, диаграммасын құру;

в) бюджетті есептеу, сатып алу мерзімі мен инвестициялық шығын мөлшерін есептеу;

г) жоғарыда айтылғандарды эксел жұмыс істеу ортасында көрсетуге мүмкіндік көрсетілген.

ғ) бұл бағдарлама жобасы кондитерлік зауыдтың өндірістік комплексінің яғни, кәсіпорынның жұмыс тиімділігін көрсетуге ыңғайлы программа деп алсақ болады.

Бағдарламаны жасауға келесі компьютер конфигурациясы қолданылады:

Intel Core i3 2,2GHz, 500 Gb HDD, LG Full HD 23"

Осы бағдарламаны енгізуде қандай экономикалық тиімділік болатынын есептелік.

Жүйені құруға кеткен шығынның сметасын құрамыз.

Бағдарламаны құру барысында жасалған шығындардың сметасы негізгі, қосымша және басқа шығындардан тұрады.

Кез-келген бағдарламалық өнімнің өзіндік құны келесі шығындардан құралады:

- негізгі бағдарламалық өңдеушілердің жалақысы (3п);

- қосымша жалақы (бонус ретінде төленеді) (Здоп);
- еңбек ақы төлеу қоры (ФОТ = Зп + Здоп);
- әлеуметтік салық (Ос);
- амортизациялық төлемдер (А);
- шығын материалдары (қағаз, картридж, канцеляриялық құрал-жабдықтар және т.б.) (М);
- интернет шығындары (Ри);
- арендалық төлемдер (Ра);
- сертификация мен лицензиялау шығындары (Рл);
- басқа өндірістік шығындар (барлық шығындардың 20%-ы) (Рпр).

5.3 Экономика есептері

Бағдарламаны, жүйені немесе құралды жасағанда кеткен шығындардың барлығын күрделі (бірреттік) шығындарға жазуға болады. Шығындар – бағдарламаның жобалау, жасау және енгізуге кететін шығындарды есептейді.

5.3.1 Жобалауға кететін бір реттік шығындар. Жалпы бұл жүйені жобалауға бес маман қатысады. Олар: меңгеруші, үш программист және аудармашы.

5.3.1.1 Әдебиеттік зерттеуді өткізу. Жүйенің жобалауы және әдебиеттік зерттеуіне қатысатын мамандар тізімі және шығындары

3.1 кесте-Әдебиеттік зерттеуді өткізу

Мамандар	Айл.жал.мың.тг	Зерт.кет. күн	Жал.жал.мың.тг
Меңгеруші	80	10	80
Программист	60	5	30
Дизайнер	60	5	30
Барлығы			140

5.3.1.2 Бағдарламаны жазу үшін кеткен шығындар

3.4 кесте- Жүйені жасауға қатысатын мамандар тізімі және шығындары

Мамандар	Жалақы мың.тг	Жаз. кет. уақыт.ай	Бар. жал.мың.тг
Меңгеруші	80	4,5	360
Дизайнер	60	4	240
Программист	60	4	240
Программист	60	4	240
Барлығы			1080

Жобалау мен жүзеге асыру барысында мамандардың ережеден артық жасағаны үшін әдебиеттік зерттеу, патенттік зерттеу, іссапарға бару мен жүйені жасауға кететін шығындардың сомасынан 2% алуға болады.

5.3.1.3 Ережеден артық істегені үшін жалақы

$$D=(A+B)*2\%,$$

(5.1)

мұндағы, А – әдебиеттік зерттеу үшін кеткен шығын

Б – жүйені жасауға кеткен шығын. $D=(140+1080)*2\%=640$ мың тг.

5.3.1.4 Әлеуметтік салық. Алдымен $(A+E)$ сомасынан жеке зейнеткерлер қорына аударымдар шегеріледі, ол 10% тең. Қалған сомадан 11% әлеуметтік салық есептелінеді. Сондай-ақ міндетті әлеуметтік жинау есептелінеді.

$$J_c=0,11*[(A+B+D)-0,1*(A+B+D)].$$
$$J_c=0,11*[(140+1080+640)-0,1*(140+1080+640)].=0,11[2319-231,9].=271 \text{ мың тг.}$$

(5.2)

5.3.1.5 Жобалау және жүйені жасау тобына арендаға үй алу үшін төлем. Әрбір маманға 6м^2 таза жұмыс ауданы керек. Коридорды, дәретхананы, қоймаларды санағанда, бір маманға шамамен 10м^2 аудан керек. Сол себепті біз мамандарға байланысты жалпы ауданы 40м^2 екі бөлмелі пәтерді жалға аламыз. Пәтерді жалға алуда жеті ай мерзім алынады. Өйткені, ол сол бағдарламаны зерттеу, жобалау, жүзеге асыру, орнату үшін кететін уақытты толығымен қамтиды. Айына 70 мың теңге төленеді.

$$Ш_{\text{жалғ.ал.}}=B*Y,$$

(5.3)

мұндағы, B – жалға алатын пәтердің ауданы 40м^2 болған кездегі оның бағасы,

Y – пәтерді жалға алу мерзімі.

$$Ш_{\text{жалғ.ал.}}=70 \text{ мың тг.} * 3 \text{ ай}=490 \text{ мың тг.}$$

(5.4)

Қолданылатын техникалық жабдықтардың бағасы мемлекетіміздің қазіргі нарықтық бағасына сәйкес арнайы журналдан алынған.

5.3.2 Жабдықтарды сатып алу және амортизациялық аударым шығындары. Бұған типті және нақты бағдарламалық қамтаманы әзірлеуге қажетті қосалқы, арнайы мақсаттағы техникалық және бағдарламалық құралдар дайындауға, оларды жобалауға, дайындауға, дұрыстауға, орнатуға және пайдалануға жұмсалатын шығындарды қосқандағы шығындар жатады. Сатып алынатын жабдықтар атауы, саны және бағасы 3.5-кестеде келтірілген.

5.5 кесте- Жабдықтарды сатып алуға кеткен шығындар

Жабдықтар	Саны	Бағасы, мың тг.	Жалпы төлем, мың тг.
Компьютер	5	90	450
Принтер	2	17	34
Телефон	2	5	10
Сканер	1	15	15
Модем	2	7	14
USB Flash	4	5	20
UPS	5	7	35
Қорытынды			578

Амортизацияның $A_a=40\%$ дербес компьютерге белгіленген мөлшерінде амортизациялық аударымдар жылына:

$$A_a=0,4*578=231 \text{ мың тг.} \quad (5.5)$$

5.3.2.1 Қажетті материалдарға кететін шығындар. Бұған бағдарламалық құралдарды әзірлеу барысында қажетті магнитті дерек тасушыға, қатты дискілер мен басқа да кеңсе тауарларына жұмсалған шығындар есептелген. Олардың атауы, саны және бағасы 3.6-кестеде келтірілген. Бағалары «Abdi» фирмасының прайс парағынан алынған яғни 2013 жылдың деректерімен салыстырылған.

5.6 кесте- Материалдар шығыны

Материалдар	Өлшем бірлігі	Саны	Бағасы, мың тг.	Жал. төл,мың тг.
Қағаз	Бума	2	0.5	5
Қалам	Дана	2	0.05	0.75
Қорытынды				

5.3.3 Электроэнергия үшін төлем

5.7 кесте- Жабдықтар қажет ететін электроэнергия шығыны

Жабдық	Эл.қуаты	Бір айд.жұм.уақ.	кВт.сағ	Эл.эн.бағ.	Айл.шығ.
Компьютер	0,800кВт	8сағ*90күн=200сағ	160	13,45	2,08
Принтер	0,600кВт	3сағ*90күн=75сағ	45	13,45	0,59
Барлығы					3,5≈4

Электрожарық үшін жұмсалатын шығын:

$$Ш_{эж}=A*Э*У*Б, \quad (5.6)$$

мұндағы, А – жарықталатын аудан, м²,
 Э – бір шаршы метрге қажет электроэнергияның қуаты,
 У – жарық жанатын уақыт, сағ.,
 Б – электроэнергияның бағасы.

$$\text{Ш}_{\text{эж}}=40\text{м}^2*0,025\text{кВт/м}^2*125\text{сағ}*13,45\text{мың тг}=1,625\text{мың тг}.\approx 2$$

$$\text{Ш}_{\text{эл}}=(4+2)\text{ мың тг}*7\text{ ай}=42\text{ мың тг}.$$

5.3.4 Интернет үшін төлем

$$\text{Ш}_{\text{интернет}}=\text{ОБ}+\text{Б}*У, \quad (5.7)$$

мұндағы, ОБ – Megaline жоғары жылдамдықты интернет желісін орнату үшін кететін төлем (бір реттік төлем),
 Б – интернет желісі үшін айлық бағасы,
 У – интернет желісін пайдаланатын уақыт, ай.

$$\text{Ш}_{\text{интернет}}=3,6\text{ мың тг}+4,3\text{ мың тг}*3\text{ ай}=33,7\text{ мың тг}.$$

5.2.5 Бағдарламаны жарнамалау. Бағдарламаны жарнамалаумен айналысатын арнайы мекемелерге газет, плакат және кітапша арқылы таратып, жарнамаға ұсынады. Бұл баптың сомасын алдағы баптардың сома қорытындысын 10% алуға болады. Ол келесідей есептелінеді:

$$\text{Ш}_{\text{жарнама}}=(\text{А}+\text{Б}+\text{В}+\text{Г}+\text{Д}+\text{Е})*10\% \quad (5.8)$$

$$\text{Ш}_{\text{жарнама}}=(450+300+410+1080+45+34)*10\%=231,9\text{ мың тг}\approx 232\text{ мың тг}.$$

5.3.6 Барлық шығындар (жүйенің өз құны). Барлық бір реттік шығындар келесідей болады:

$$\text{Ш}=\sum 5.1\div 5.3$$

$$\text{Ш}=4573,2\text{ мың тг}\approx 4,6\text{млн.} \quad (5.9)$$

Фирмалық баға:

$$\text{Б}_{\text{ф}}=\text{Ш}+\text{П}, \quad (5.10)$$

мұндағы, Ш – барлық шығындар қосындысы;
 П – шығынның 20%-ы болып табылатын пайда.

$$\text{П}=4573,2\text{ мың тг}*20\%=914,6\text{ мың тг}\approx 915\text{ мың тг};$$

$$B_{\phi}=4573,2 \text{ мың тг}+915 \text{ мың тг}=5488,2 \text{ мың тг};$$

Жалпы баға:

$$B_{ж}=B_{\phi}+0,12B_{\phi}, \quad (5.11)$$

мұндағы, B_{ϕ} – фирмалық баға.

$$B_{ж}=5488,2 \text{ мың тг}+0,12*5488,2 \text{ мың тг}=6146 \text{ мың тг}.$$

$$B_{ж}=6202 \text{ мың тг}$$

Қорытынды: жоғарыдағы есептеулер бойынша мұндай дерекқорды енгізуде экономикалық тиімді нәтиже алуға мүмкіндік беретіндігіне көз жеткіздік. Бағдарламаны енгізудің қажеттілігі мол, себебі кіріс көрсеткіштері жоғарғы деңгейді көрсетеді. Сонымен, кітапхана дерекқорын құру және енгізу экономикалық жағынан тиімді және жарнама, маркетингтік саясат факторларына байланысты ұтымды.

Қорытынды

Дипломдық жұмысты қорытындылай келе кәсіпорынның жұмыс тиімділігін бағалайтын бағдарлама кәсіпорынның дамуына үлкен үлесін қосады. Сонымен қоса экономикалық жүйе динамикасының барлық модельдеріне шолу жасалды. Өндірістегі экономикалық жүйенің даму процесін талдаудың бар модельдеріне жекеленген кезеңдеріне мазмұнды түсініктеме беріледі. Кондитер кәсіпорында жұмыс тиімділігін бағалаудың модельдері жасалады. Сонымен қатар өндірістегі экономикалық жүйенің негізгі қорының динамикасының модельдерін жетілдірілді. Модельді есептерді шешу тәсілдерін қарастырдық. Құрылған модельдердің базасында экономикалық жүйенің қалыпты даму бағасының математикалық тәсілін құрдық.

Программаны жүзеге асыратын техникалық құралдар комплексі таңдалынып алынады. Қазіргі заманға сай кәсіпорынның инвестициялық шығындарын, қажетті шикізат көлемін, толық бюджет көлемін есептейтін, сақтайтын, мәліметке оңай қол жеткізетін және өзгертуге мүмкіншілік беретін заманға сай қажетті аса зор бағдарламма құрылды. Жобаның экономикалық тиімділігі есептелді және еңбекті қорғау жөнінде шаралар ұйымдастырылды.

Пайдаланылған әдебиеттер тізімі

1. Ни А.Г., Имангалиев Ш.И., Жумақанова Р.Ж. Методические указания к выполнению дипломных работ (проектов) для студентов специальности 5В070300 «Информационные системы». – Алматы: АУЭС, 2012 – 39 бет.
2. <http://www.rakhat.kz/index.php?p=data&lang=3&sct=6>
3. Палашарчук А. С. Экономика предприятия: учебник- М.: ИНФРА- М, 2011- 459с.
4. Липунцов Ю.П. управление процессами: Методы управления предприятием с использованием информационных технологий. М.:ДМК-Пресс: Компания АйТи,2003г.
5. Скляренко В.К., Прудников В.М. Экономика предприятия: Учебник. М.:ИНФРА-М, 2005г.
6. Экономика предприятия (фирмы): Учебник/ Под ред. О.И. Волокова и О.В. Девяткина. 4-е изд. М.: ИНФРА- М, 2007г.
7. Аманбаев У.А. Экономика предприятия. Учебное пособие У.А. Аманбаев- Алматы: Издательство «Бастау», 2012-432с.
8. «О стратегическом плане развития Республики Казахстан до 2020 года», от 1 февраля 2010 года.
9. Краснощеков П.С., Петров А.А Принципы построения моделей. М.: Изд-во МГУ. 1983 ж.
10. Грегори Мэнкью Н. Макроэкономика. Изд-во Московского университета. 1994г.
11. Алиманов. Ш. А, « Введение в математическую экономику » 1984 ж.
12. Экланд. И. «Элементы математической экономики » Мир.1983
13. Фуфаев Э.В. Базы данных.-М.,2006,2007
14. Селко Дж. SQL для профессионалов. Программирование.-М.,2004
15. Плю Р., Стефенс Р., Райан К. Освой самостоятельно SQL за 24 часа. – М.: Издательский дом «Вильямс», 2000
16. Базы данных, учебник для высших учебных заведений под редакцией Хомоненко А.Д С – П Корона 2000.
17. Джек Л. Харрингтон Проектирование реляционных баз данных просто и доступно. - Изд. «Лори», 2000г.
18. Хомоненко А., Гофман В., Работа с базами данных в Delphi, 3-е издание .БХВ-Петербург .2005 г. ISBN: 5-94157-361-8
19. Паченко К. Delphi 5. Руководство разработчика Т.1. Основные методы технологий программирования- М.:000«И.Д. Вильямс», 2002г.
20. Культин Н. Основа программирования в Delphi 7., - Санкт-Петербург,- 2009, 640с.
21. Е. В. Санников. Курс практического программирования в Delphi. – Практикум- Москва, Салон –Пресс, 2013 – 188с.
22. Стив Тексейра и КсавьеПачеко. Delphi 5 Руководство разработчика Т 1,2 . Вильямс 2000г.

23. Бобровский С. [Технологии Delphi 2006. Новые возможности](#) .Питер 2006г 288стр
24. Культин Н.Основы программирования в Delphi 7., БХВ-Петербург. 2007 г. ISBN: 978-5-94157-269-7
25. Орлов С. Технологии разработки программного обеспечения: учебник-СПБ.: Питер, 2002г.
26. Холзлер С., Visual C++ 6. Учебный курс. 1С. Холзлер – СПб.: Питер,2007.
27. Свердлов С. 3. Языки программирования и методы трансляции: Учебное пособие. СПб.: Питер,2007-638 с.
28. 4.Холингворт, Джаррод, Сворт и др. BorlandC++Builder6. Руководство разработчика.: Пер с англ. –М. : ИД «Вильямс», 2004-976 с.
29. Кватрани Т. Rational Rose 2000 и UML. Визуальное моделирование: Пер. с англ. – М.: ДМК Пресс, 2001. – 176 бет.
30. Д. Иванов, Ф. Новиков. Моделирование на UML. Учебно-методическое пособие. – Спб.: СПбГУ ИТМО 2010г
31. М. К Дюсебаев, Т. Е. Хакимжанов, Ж.С. Абдимуратов. Еңбекті қорғау және тіршілік қауіпсіздігі. Оқу құралы. АУЭС, - 2013. 84 б.
32. Санатова Т.С., Мананбаева С.Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет зануления» в выпускных работах для всех специальностей. Бакалавриат – Алматы: АИЭС, 2009. – 26 бет.
33. Абдимуратов Ж.С., Мананбаева С.Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат – Алматы: АИЭС, 2009. – 20 бет.
34. Хакімжанов Т.Е. Еңбек қорғау. Жоғары оқу орындары үшін оқу құралы. – Алматы: «Эверо», 2008 – 240 бет.
35. Базылов К.Б., Алибаева С.А., Бабич А.А. Методические указания по выполнению экономического раздела выпускной работы бакалавров для студентов всех форм обучения специальности 050719 – Радиотехника, электроника и телекоммуникации – Алматы: АИЭС, - 2008. -19 бет.

А Қосымша

```
unit Unit1;
interface
uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
    System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ExtCtrls,
    Vcl.ComCtrls,
    Data.DB, Data.Win.ADODB, DBGridEhGrouping, ToolCtrlsEh,
    DBGridEhToolCtrls,
    DynVarsEh, GridsEh, DBAxisGridsEh, DBGridEh, Vcl.DBCtrls,
    Vcl.Buttons, math;
type
    TOcenka = class(TForm)
        PageControl1: TPageControl;
        TabSheet1: TTabSheet;
        TabSheet2: TTabSheet;
        TabSheet3: TTabSheet;
        TabSheet4: TTabSheet;
        ADOTable1: TADOTable;
        DBGridEh1: TDBGridEh;
        DBGridEh2: TDBGridEh;
        DBGridEh3: TDBGridEh;
        DBGridEh4: TDBGridEh;
        Panel1: TPanel;
        Panel2: TPanel;
        Panel3: TPanel;
        Panel4: TPanel;
        Panel5: TPanel;
        DBNavigator1: TDBNavigator;
        Panel6: TPanel;
        DBNavigator2: TDBNavigator;
        Panel7: TPanel;
        DBNavigator3: TDBNavigator;
        BitBtn1: TBitBtn;
        LabeledEdit1: TLabeledEdit;
        BitBtn2: TBitBtn;
        DataSource1: TDataSource;
        LabeledEdit2: TLabeledEdit;
        BitBtn3: TBitBtn;
        BitBtn4: TBitBtn;
        LabeledEdit3: TLabeledEdit;
        LabeledEdit4: TLabeledEdit;
```

```

    LabeledEdit5: TLabeledEdit;
    LabeledEdit6: TLabeledEdit;
    LabeledEdit7: TLabeledEdit;
    LabeledEdit8: TLabeledEdit;
    LabeledEdit9: TLabeledEdit;
    BitBtn5: TBitBtn;
    BitBtn6: TBitBtn;
    LabeledEdit10: TLabeledEdit;
    LabeledEdit11: TLabeledEdit;
    LinkLabel1: TLinkLabel;
    BitBtn7: TBitBtn;
    LinkLabel2: TLinkLabel;
    Label1: TLabel;
    procedure PageControl1Change(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure DBGridEh1DrawColumnCell(Sender: TObject; const Rect:
TRect;
        DataCol: Integer; Column: TColumnEh; State: TGridDrawState);
    procedure BitBtn1Click(Sender: TObject);
    procedure LabeledEdit1KeyPress(Sender: TObject; var Key: Char);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure BitBtn5Click(Sender: TObject);
    procedure BitBtn6Click(Sender: TObject);
    procedure BitBtn7Click(Sender: TObject);
    procedure DBGridEh2DrawColumnCell(Sender: TObject; const Rect:
TRect;
        DataCol: Integer; Column: TColumnEh; State: TGridDrawState);
    procedure DBGridEh3DrawColumnCell(Sender: TObject; const Rect:
TRect;
        DataCol: Integer; Column: TColumnEh; State: TGridDrawState);
    procedure DBGridEh4DrawColumnCell(Sender: TObject; const Rect:
TRect;
        DataCol: Integer; Column: TColumnEh; State: TGridDrawState);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    Ocenka: TOcenka;
implementation
{$R *.dfm}

```

```

uses Unit2, Unit3, Unit4;
procedure TOcenka.BitBtn1Click(Sender: TObject);
var i:integer;
begin
  ADOTable1.First;
  if Length (LabeledEdit1.Text)>0 then
  begin
    for I := 0 to ADOTable1.RecordCount-1 do
      Begin
        ADOTable1.Edit;
        DBGridEh1.Fields[3].Value :=Round
(DBGridEh1.DataSource.DataSet.Fields.Fields[2].Value /
dbgrideh1.GetFooterValue(0, Dbgrideh1.Columns[2]) * 100);
        DBGridEh1.Fields[4].Value := Roundto
(DBGridEh1.DataSource.DataSet.Fields.Fields[2].Value * StrToInt
(LabeledEdit1.Text),-2);
        ADOTable1.Post;
        DBGridEh1.DataSource.DataSet.Next;
      End;
    ADOTable1.First;
  end
  else ShowMessage('Не введена "Годовая мощность"');
end;
procedure TOcenka.BitBtn2Click(Sender: TObject);
begin
diagSirie.ShowModal;
end;
procedure TOcenka.BitBtn3Click(Sender: TObject);
var i: integer;
proiz: Extended;
begin
ADOTable1.First;
  proiz := DBGridEh1.Fields[2].Value;
  DBGridEh1.DataSource.DataSet.Next;
  for I :=0 to ADOTable1.RecordCount-2 do
    Begin
      ADOTable1.Edit;
      DBGridEh2.Fields[4].Value
:=DBGridEh2.DataSource.DataSet.Fields.Fields[2].Value *
DBGridEh1.DataSource.DataSet.Fields.Fields[3].Value;
      if DBGridEh2.Fields[1].Value='Электричество' then
        DBGridEh2.Fields[4].Value
:=DBGridEh2.DataSource.DataSet.Fields.Fields[2].Value *
DBGridEh1.DataSource.DataSet.Fields.Fields[3].Value / proiz;
    end;
  end;
end;

```

```

        ADOTable1.Post;
        DBGridEh2.DataSource.DataSet.Next;
    End;
    ADOTable1.First;
    LabeledEdit2.Text := dbgrideh2.GetFooterValue(0,
Dbgrideh2.Columns[4]) ;
    LabeledEdit3.Text := FloatToStr(RoundTo( 50000/30/24/ proiz, -2) );
    LabeledEdit4.Text := FloatToStr(RoundTo(
StrToFloat(LabeledEdit3.Text) * 0.3, -2));
    LabeledEdit5.Text := FloatToStr( StrToFloat(LabeledEdit3.Text)+
StrToFloat(LabeledEdit4.Text));
    LabeledEdit6.Text := FloatToStr( RoundTo
((StrToFloat(LabeledEdit2.Text) + StrToFloat(LabeledEdit5.Text))*0.05, -2));
    LabeledEdit7.Text := FloatToStr( StrToFloat(LabeledEdit2.Text)+
StrToFloat(LabeledEdit5.Text) + StrToFloat(LabeledEdit6.Text));
    end;
    procedure TOcenka.BitBtn4Click(Sender: TObject);
    begin
    if length(LabeledEdit8.Text) > 0 then
    begin
    BitBtn3.Click;
    diagSS.ShowModal;
    end
    else
    ShowMessage('Введите "Цена реализации "');
    end;
    procedure TOcenka.BitBtn5Click(Sender: TObject);
    var ch, ceh: Extended;
    begin
    ADOTable1.First;
        ch:= DBGridEh3.Fields[2].Value;
        ADOTable1.Next;
        LabeledEdit9.Text := ch + DBGridEh3.Fields[2].Value;
        ADOTable1.Next;
        ceh := DBGridEh3.Fields[2].Value;
        ADOTable1.Next;
        ch := DBGridEh3.Fields[2].Value;
        ADOTable1.Next;
        LabeledEdit10.Text := ch + DBGridEh3.Fields[2].Value;

        LabeledEdit11.Text:= Floattostr( Strtofloat(LabeledEdit10.Text) +
Strtofloat(LabeledEdit9.Text) + ceh);
        ADOTable1.First;
    end;
end;

```

```

procedure TOcenka.BitBtn6Click(Sender: TObject);
begin
  BitBtn5.Click;
  diaginvest.ShowModal;

end;
procedure TOcenka.BitBtn7Click(Sender: TObject);
var i,j, srok: integer;
    pribil, potr : Extended;
    masch :array[0..100] of Extended;
    okup : array [0..10] of integer;
begin
  ADOTable1.First;
  pribil := 0;
  for I := 2 to 8 do
  begin
    potr := DBGridEh4.Fields[i].Value;
    for j := 0 to ADOTable1.RecordCount-1 do
      Begin
        ADOTable1.Next;
        if (DBGridEh4.Fields[1].Value = 'Реализация, USD (без НДС)')
then
          begin
            ADOTable1.Edit;
            DBGridEh4.Fields[i].Value := potr * 100 / 1000;
            ADOTable1.Post;
          end;
          if (DBGridEh4.Fields[1].Value = 'Себестоимость USD (без
НДС)') then
            begin
              ADOTable1.Edit;
              DBGridEh4.Fields[i].Value := potr * 29.20 / 1000;
              ADOTable1.Post;
            end;
            if (DBGridEh4.Fields[1].Value = 'Налоги С ФОТ') then
            begin
              ADOTable1.Edit;
              //ShowMessage(Floattostr(masch[3]));
              // ShowMessage(Inttostr(j));
              DBGridEh4.Fields[i].Value := masch[3] * 0.3;
              ADOTable1.Post;
            end;
            if (DBGridEh4.Fields[1].Value = 'Накладные расходы') then
            begin

```

```

        ADOTable1.Edit;
        DBGridEh4.Fields[i].Value :=roundto( masch[1] * 0.1, -2);
        ADOTable1.Post;
    end;
    if (DBGridEh4.Fields[1].Value = 'основного производства
(10%)') then
        begin
            ADOTable1.Edit;
            DBGridEh4.Fields[i].Value := masch[1] * 0.1;
            ADOTable1.Post;
        end;
        if (DBGridEh4.Fields[1].Value = 'Административные
расходы') then
            begin
                ADOTable1.Edit;
                DBGridEh4.Fields[i].Value := masch[7]+
masch[8]+masch[9]+masch[10]+masch[11]+masch[12] ;
                ADOTable1.Post;
            end;
            if (DBGridEh4.Fields[1].Value = 'ЧП до вычета налогов,
амортизации и процентов') then
                begin
                    ADOTable1.Edit;
                    DBGridEh4.Fields[i].Value := masch[0]- masch[1]- masch[2]-
masch[3]- masch[4]- masch[5]- masch[14]- masch[15] ;
                    ADOTable1.Post;
                end;
                if (DBGridEh4.Fields[1].Value = 'Чистая прибыль, %') then
                    begin
                        ADOTable1.Edit;
                        if masch[0]=0 then
                            DBGridEh4.Fields[i].Value := 0
                        else
                            DBGridEh4.Fields[i].Value := masch[16]/masch[0] * 100;
                        ADOTable1.Post;
                    end;
                    if (DBGridEh4.Fields[1].Value = 'Расходы по вознаграждению
за кредит (15%)') then
                        begin
                            ADOTable1.Edit;
                            DBGridEh4.Fields[i].Value := 0.15 * 1122000/1000 ;
                            ADOTable1.Post;
                        end;

```

```

        if (DBGridEh4.Fields[1].Value = 'Прибыль до
налогообложения от обычной деятельности') then
        begin
            ADOTable1.Edit;
            DBGridEh4.Fields[i].Value := masch[16]- masch[19]-
masch[18] ;

            ADOTable1.Post;
        end;

        if (DBGridEh4.Fields[1].Value = 'Чистая прибыль, %') then
        begin
            ADOTable1.Edit;
            if masch[0]=0 then
                DBGridEh4.Fields[i].Value := 0
            else
                DBGridEh4.Fields[i].Value := masch[20]/masch[0] * 100;
            ADOTable1.Post;
        end;
        if (DBGridEh4.Fields[1].Value = 'КПН') then
        begin
            ADOTable1.Edit;
            DBGridEh4.Fields[i].Value := masch[20]* 0.2 ;
            ADOTable1.Post;
        end;
        if (DBGridEh4.Fields[1].Value = 'Прибыль') then
        begin
            ADOTable1.Edit;
            DBGridEh4.Fields[i].Value := masch[20]- masch[22] ;
            ADOTable1.Post;
            if i=2 then pribil := DBGridEh4.Fields[i].Value
            else pribil := pribil + DBGridEh4.Fields[i].Value;
        end;
        if (DBGridEh4.Fields[1].Value = 'Прибыль с нарастанием')
then
        begin
            ADOTable1.Edit;
            DBGridEh4.Fields[i].Value := pribil ;
            ADOTable1.Post;
        end;
        if (DBGridEh4.Fields[1].Value = 'Рентабельность
деятельности') then
        begin
            ADOTable1.Edit;

```



```

        if masch[0]=0 then
            DBGridEh4.Fields[i].Value := 0
        else
            DBGridEh4.Fields[i].Value := masch[23]/masch[0] * 100;
            ADOTable1.Post;
        end;
    if length(DBGridEh4.Fields[i].Value)>0 then
        masch[j] := DBGridEh4.Fields[i].Value;
    End;  //////////////////////////////////////
//      case i of
//      2 : if (pribil/11220 <1) then edit11.Text := '0' else edit11.Text
:= '1';
//      3: if (pribil/11220 <1) then edit12.Text := '0' else edit12.Text := '1';
//      4: if (pribil/11220 <1) then edit14.Text := '0' else edit13.Text := '1';
//      5: if (pribil/11220 <1) then edit15.Text := '0' else edit14.Text := '1';
//      6: if (pribil/11220 <1) then edit15.Text := '0' else edit15.Text := '1';
//      7: if (pribil/11220 <1) then edit16.Text := '0' else edit16.Text := '1';
//      8: if (pribil/11220 <1) then edit17.Text := '0' else edit17.Text := '1';
////      end;
        case i of
            2: if (pribil/11220 <1) then okup[i] := 0 else okup[i] :=1;
            3: if (pribil/11220 <1) then okup[i] := 0 else okup[i] :=1;
            4: if (pribil/11220 <1) then okup[i] := 0 else okup[i] :=1;
            5: if (pribil/11220 <1) then okup[i] := 0 else okup[i] :=1;
            6: if (pribil/11220 <1) then okup[i] := 0 else okup[i] :=1;
            7: if (pribil/11220 <1) then okup[i] := 0 else okup[i] :=1;
            8: if (pribil/11220 <1) then okup[i] := 0 else okup[i] :=1;
        end;
        ADOTable1.First;
        // if (pribil/11220 <1) then
        end;
        //      srok :=10 -( StrToInt(edit11.Text) + StrToInt(edit12.Text) +
StrToInt(edit13.Text) + StrToInt(edit14.Text) + StrToInt(edit15.Text)
        //      + StrToInt(edit15.Text) + StrToInt(edit16.Text) +
StrToInt(edit17.Text));
        for I := 2 to 8 do
            srok :=srok + okup[i];
            srok := 10 - srok;
            LinkLabel1.Caption := IntToStr(srok);
        end;
        procedure TOcenka.DBGridEh1DrawColumnCell(Sender: TObject; const
Rect: TRect;
            DataCol: Integer; Column: TColumnEh; State: TGridDrawState);
        const

```

```

    clser=Tcolor($F0F8FF);
begin
if ( ADOTable1.RecNo mod 2<>0 ) then
    DBGridEh1.Canvas.Brush.Color := clser;
    DBGridEh1.DefaultDrawColumnCell(Rect, DataCol, Column, State);
end;
procedure TOcenka.DBGridEh2DrawColumnCell(Sender: TObject; const
Rect: TRect;
    DataCol: Integer; Column: TColumnEh; State: TGridDrawState);
const
    clser=Tcolor($F0F8FF);
begin
if ( ADOTable1.RecNo mod 2<>0 ) then
    DBGridEh2.Canvas.Brush.Color := clser;
    DBGridEh2.DefaultDrawColumnCell(Rect, DataCol, Column, State);
end;
procedure TOcenka.DBGridEh3DrawColumnCell(Sender: TObject; const
Rect: TRect;
    DataCol: Integer; Column: TColumnEh; State: TGridDrawState);
const
    clser=Tcolor($F0F8FF);
begin
if ( ADOTable1.RecNo mod 2<>0 ) then
    DBGridEh3.Canvas.Brush.Color := clser;
    DBGridEh3.DefaultDrawColumnCell(Rect, DataCol, Column, State);
end;
procedure TOcenka.DBGridEh4DrawColumnCell(Sender: TObject; const
Rect: TRect;
    DataCol: Integer; Column: TColumnEh; State: TGridDrawState);
const
    clser=Tcolor($F0F8FF);
begin
if ( ADOTable1.RecNo mod 2<>0 ) then
    DBGridEh4.Canvas.Brush.Color := clser;
    DBGridEh4.DefaultDrawColumnCell(Rect, DataCol, Column, State);
    //if ADOTable1.TableName= 'Budjet' then
    with DBGridEh4.Canvas do
        if ((ADOTable1.RecNo = 1) or (ADOTable1.RecNo = 5) or
(ADOTable1.RecNo = 10) or (ADOTable1.RecNo = 11)
            or (ADOTable1.RecNo = 12) or (ADOTable1.RecNo = 13) or
(ADOTable1.RecNo =14)
            or (ADOTable1.RecNo = 15) or (ADOTable1.RecNo = 15) or
(ADOTable1.RecNo = 17)) then
            begin

```

```

Pen.Color:=clRed;
MoveTo(rect.Left,rect.Top);
LineTo(rect.Right,rect.Top);
MoveTo(rect.Left,rect.Bottom);
LineTo(rect.Right,rect.Bottom);
if Column.Index=0 then
begin
MoveTo(rect.Left,rect.Bottom);
LineTo(rect.Left,rect.Top);
end;
if Column.Index=DBGridEh4.Columns.Count-1 then
begin
MoveTo(rect.Right,rect.Bottom);
LineTo(rect.Right,rect.Top);
end;
end;
end;
procedure TOcenka.FormCreate(Sender: TObject);
var mesto : string;
begin
//mesto:=ExtractFileDir(Application.ExeName) + '\bd.accdb';
mesto:=ExtractFileDir(Application.ExeName) + '\bd.mdb';
ADOTable1.ConnectionString :=
'Provider=Microsoft.ACE.OLEDB.12.0;Data Source='+mesto+';Persist Security
Info=False';
ADOTable1.Active:=False;
ADOTable1.TableName:='sirie';
ADOTable1.Active:=True;
TNumericField(DBGridEh1.Fields[2]).DisplayFormat:='#####0.00';
TNumericField(DBGridEh1.Fields[3]).DisplayFormat:='#####0';
TNumericField(DBGridEh1.Fields[4]).DisplayFormat:='#####0.00';
DBGridEh1.Columns[2].Footer.ValueType := fvtSum;
DBGridEh1.SumList.Active := true;
DBGridEh1.Columns[3].Footer.ValueType := fvtSum;
DBGridEh1.SumList.Active := true;
DBGridEh1.Columns[4].Footer.ValueType := fvtSum;
DBGridEh1.SumList.Active := true;
end;
procedure TOcenka.LabeledEdit1KeyPress(Sender: TObject; var Key:
Char);
begin
if Key in ['0'..'9']+[#8]then
begin
end else Key:=#0;
end;
end;

```

```

end;
procedure TOcenka.PageControl1Change(Sender: TObject);
begin
case PageControl1.TabIndex of
0: begin
    ADOTable1.Active:=False;
    ADOTable1.TableName:='sirie';
//форматированный вывод на экран
    ADOTable1.Active:=True;
    TNumericField(DBGridEh1.Fields[2]).DisplayFormat:='#####0.00';
    TNumericField(DBGridEh1.Fields[3]).DisplayFormat:='#####0';

TNumericField(DBGridEh1.Fields[4]).DisplayFormat:='#####0.00';
    DBGridEh1.Columns[2].Footer.ValueType := fvtSum;
    DBGridEh1.SumList.Active := true;
    DBGridEh1.Columns[3].Footer.ValueType := fvtSum;
    DBGridEh1.SumList.Active := true;
    DBGridEh1.Columns[4].Footer.ValueType := fvtSum;
    DBGridEh1.SumList.Active := true;

    DBGridEh1.Columns[0].Footer.ValueType := fvtStaticText;
    DBGridEh1.Columns[0].Footer.Value:='Bcero';
    DBGridEh1.SumList.Active := true;
end;
1: begin
    ADOTable1.Active:=False;
    ADOTable1.TableName:='SebeSt';
    ADOTable1.Active:=True;
    TNumericField(DBGridEh2.Fields[2]).DisplayFormat:='#####0.00';
    TNumericField(DBGridEh2.Fields[3]).DisplayFormat:='#####0.00000';
    TNumericField(DBGridEh2.Fields[4]).DisplayFormat:='#####0.00';
    DBGridEh2.Columns[4].Footer.ValueType := fvtSum;
    DBGridEh2.SumList.Active := true;
    DBGridEh2.Columns[0].Footer.ValueType := fvtStaticText;
    DBGridEh2.Columns[0].Footer.Value:='Bcero';
    DBGridEh2.SumList.Active := true;
end;
2: begin
    ADOTable1.Active:=False;
    ADOTable1.TableName:='InvestZat';
    ADOTable1.Active:=True;
    / Panel4.Visible := true;
    // BitBtn1.Visible := true;
    // Label2.Visible := true;

```

```

//    DBNavigator1.Visible := true;
//    BitBtn2.Visible := true;
    end;
3: begin
    ADOTable1.Active:=False;
    ADOTable1.TableName:='Budget';
    ADOTable1.Active:=True;
//форматированный вывод на экран
    TNumericField(DBGridEh4.Fields[2]).DisplayFormat:='#####0.00';
    TNumericField(DBGridEh4.Fields[3]).DisplayFormat:='#####0.00';
    TNumericField(DBGridEh4.Fields[4]).DisplayFormat:='#####0.00';
    TNumericField(DBGridEh4.Fields[5]).DisplayFormat:='#####0.00';

    TNumericField(DBGridEh4.Fields[6]).DisplayFormat:='#####0.00';
    TNumericField(DBGridEh4.Fields[7]).DisplayFormat:='#####0.00';
    TNumericField(DBGridEh4.Fields[8]).DisplayFormat:='#####0.00';
    end; end; end;
end.
unit Unit2;                                //построение диаграмм
interface
uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
    System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, VclTee.TeeGDIPlus,
    VCLTee.TeEngine,
    VCLTee.Series, Vcl.ExtCtrls, VCLTee.TeeProcs, VCLTee.Chart;
type
    TDiagSirie = class(TForm)
        Chart1: TChart;
        Series1: TPieSeries;
        procedure FormShow(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
var
    DiagSirie: TDiagSirie;
implementation
    {$R *.dfm}
uses Unit1;
procedure TDiagSirie.FormShow(Sender: TObject);
var i : integer;
begin

```

```

Chart1.SeriesList[0].Clear;
ocenka.ADOTable1.First;
for I := 0 to ocenka.ADOTable1.RecordCount-1 do
begin
    Series1.Add(ocenka.DBGridEh1.Fields[3].Value ,
ocenka.DBGridEh1.Fields[1].Value,Random($FFFFFF));
    ocenka.DBGridEh1.DataSource.DataSet.Next;
end;
end;
end.

```

unit Unit3; **//построение диаграмм**

```

interface
uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, VclTee.TeeGDIPlus,
VCLTee.TeEngine,
    VCLTee.Series, Vcl.ExtCtrls, VCLTee.TeeProcs, VCLTee.Chart;
type
    TdiagSS = class(TForm)
        Chart1: TChart;
        Series1: TPieSeries;
        Chart2: TChart;
        PieSeries1: TPieSeries;
        procedure FormShow(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
var
    diagSS: TdiagSS;
implementation
    {$R *.dfm}
uses Unit1;
procedure TdiagSS.FormShow(Sender: TObject);
var i : integer;
begin
    Chart1.SeriesList[0].Clear;
ocenka.ADOTable1.First;
ocenka.ADOTable1.Next;
for I := 0 to ocenka.ADOTable1.RecordCount-1 do
begin

```

```

        Series1.Add( oценка.DBGridEh2.Fields[4].Value,
оценка.DBGridEh2.Fields[1].Value, Random($FFFFFF));
        оценка.DBGridEh2.DataSource.DataSet.Next;
    end;
    Series1.Add(StrtoFloat (оценка.LabeledEdit5.Text), 'ФЗП',
Random($FFFFFF));
    Series1.Add(StrtoFloat (оценка.LabeledEdit6.Text), 'Потери',
Random($FFFFFF));
    оценка.ADOTable1.First;
    Chart2.SeriesList[0].Clear;
    PieSeries1.Add(StrtoFloat (оценка.LabeledEdit7.Text), 'Себестоимость' ,
Random($FFFFFF));
    PieSeries1.Add(StrtoFloat (оценка.LabeledEdit8.Text) -
StrToFloat(оценка.LabeledEdit7.Text), 'Маржинальный доход производства' ,
Random($FFFFFF));
    PieSeries1.Add(StrtoFloat (оценка.LabeledEdit8.Text), 'Цена реализации' ,
Random($FFFFFF));
    end;
end.
unit Unit4; //построение диаграммы
interface
uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, VclTee.TeeGDIPlus,
VCLTee.TeEngine,
    VCLTee.Series, Vcl.ExtCtrls, VCLTee.TeeProcs, VCLTee.Chart;
type
    Tdiaginvest = class(TForm)
        Chart1: TChart;
        Series1: TPieSeries;
        procedure FormShow(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
var
    diaginvest: Tdiaginvest;
implementation {$R *.dfm}
uses Unit1;
procedure Tdiaginvest.FormShow(Sender: TObject);
var i : integer;
begin for I := 0 to оценка.ADOTable1.RecordCount-1 do

```

```
begin Series1.Add( oценка.DBGridEh3.Fields[2].Value,
оценка.DBGridEh3.Fields[1].Value, Random($FFFFFF));
оценка.DBGridEh3.DataSource.DataSet.Next;
end;
Series1.Add(StrtoFloat (оценка.LabeledEdit9.Text), 'Территория',
Random($FFFFFF));
Series1.Add(StrtoFloat (оценка.LabeledEdit10.Text), 'Оборудование',
Random($FFFFFF));
Series1.Add(StrtoFloat (оценка.LabeledEdit11.Text), 'ВСЕГО
капитальных вложений', Random($FFFFFF));
Оценка.ADOTable1.First;
end;
end
```