

А. В. РУДАКОВ, Г. Н. ФЕДОРОВА

**Бағдарламалық
қамтамасыз ету
ТЕХНОЛОГИЯСЫ**

Практикум

*«Білім беруді
дамытудың федералды институты»
федералдық мемлекеттік мекемесі
бастауыш кәсіптік білім беру бағдарламаларын іске асыратын
білім беру мекемелеріне оқу құралы ретінде*

ҰСЫНЫЛҒАН

Пікірдің тіркеу нөмірі № 325, 16 маусым, 2009 жыл.

5-шы басылым, стереотипті

**Мәскеу
«Академия» баспа орталығы,
2014**

Бұл кітап Қазақстан Республикасының Білім және ғылым министрлігі және «Кәсіпқор» холдингі» КЕАҚ арасында жасалған шартқа сәйкес «ТЖКБ жүйесі үшін шетел әдебиетін сатып алуды және аударуды ұйымдастыру жөніндегі қызметтер» мемлекеттік тапсырмасын орындау аясында қазақ тіліне аударылды. Аталған кітаптың орыс тіліндегі нұсқасы Ресей Федерациясының білім беру үдерісіне қойылатын талаптардың ескерілуімен жасалды. Қазақстан Республикасының техникалық және кәсіптік білім беру жүйесіндегі білім беру ұйымдарының осы жағдайды ескеруі және оқу үдерісінде мазмұнды бөлімді (технология, материалдар және қажетті ақпарат) қолдануы қажет. Аударманы «Delta Consulting Group» ЖШС жүзеге асырды, заңды мекенжайы: Астана қ., Иманов көш., 19, «Алма-Ата» БО, 809С, телефоны: 8 (7172) 78 79 29, эл. поштасы: info@dcg.kz

Пікірсарапшылар:
«Инженерлік және сапа менеджменті» кафедрасының оқытушысы
Г.А.Акимов (Д.Ф. Устинов атындағы ВОЕНМЭХ атындағы БМТУ); Мәскеу автомобильнің компьютерлік даярлау орталығы колледжі зертханасының меңгерушісі А.А.Соломашкин
Рудаков А. В.

P83 Бағдарламалық өнімдерді әзірлеу технологиясы:
Практикалық: оқулық, студияға арналған жәрдемақы, қоршаған ортаны қорғау мекемелері, проф. білім / В.В. Рудаков, Г.Н. Федорова. - 5-ші том., Sr. - М.: «Академия» баспа орталығы, 2014. - 192 б.

ISBN 978-601-333-084-4(каз.) ISBN 978-5-4468-1197-7(рус.)

Оқу құралы жүйелі түрде қажетті теориялық ақпаратты, практикалық тапсырмаларды және оларды іске асыру мысалдарын береді; құрылымдық және объектілі-бағдарлы көзқарас (стандартты UML модельдеу тілі мен қазіргі заманғы CASE-құралдарын пайдалану), тестілеу, анықтамалық жүйе құру және орнату пакеттерін құру бойынша міндеттерді пайдалана отырып бағдарламалық өнімдерді модельдеу бойынша тапсырмаларды ұсынады .

Орта кәсіптік білім беру мекемелерінің студенттері үшін.

ӘОЖ
681.3.06(075.32)
КБЖ
32.973-018я723

Бұл оқу басылымы «Компьютерлік жүйелердегі бағдарламалау» мамандығының оқу-әдістемелік жиынтығының бөлігі болып табылады.

Семинар РМЗ кәсіби модулін «Бағдарламалық модульдерді интеграциялауға қатысу» (МДК 03.01) оқуға арналған.

Жаңа бұынның оқу-әдістемелік жинақтары жалпы және жалпы кәсіби пәндерді және кәсіби модульдерді оқып-үйренуге мүмкіндік беретін дәстүрлі және инновациялық оқу-әдістемелік материалдарды қамтиды. Әрбір жинақта жұмыс берушінің талаптарын ескере отырып, жалпы және кәсіби құзыреттерді меңгеру үшін қажетті оқулықтар мен оқу құралдар, оқу және басқару құралдары бар.

Оқу басылымдары электронды білім беру ресурстарымен толықтырылды. Электрондық ресурстарда интерактивті жаттығулар мен тренерлер, мультимедиялық нысандар, интернетте қосымша материалдар мен ресурстарға сілтемелер бар. Оған қоса теориялық және практикалық модульдер қарастырылған. Оған оқу үрдісінің негізгі параметрлері белгіленетін терминологиялық сөздік және электронды журнал кіреді: жұмыс уақыты, бақылау және практикалық тапсырмалардың нәтижесі. Электронды ресурстар оқу үдерісіне оңай енеді және әртүрлі оқу бағдарламаларына бейімделуі мүмкін.

Семинар «Компьютерлік технологиялар және автоматтандырылған жүйелерге арналған бағдарламалық қамтамасыздандыру» мамандығы бойынша Мемлекеттік білім беру стандарттарына сәйкес дайындалды және оның мазмұнында «Бағдарламалық өнімдерді әзірлеу технологиясы» (2005ж.) Бұрын жарияланған оқулықпен толықтырылды.

Семинардың мақсаты - заманауи әдістер мен құралдарды пайдалана отырып, студенттердің бағдарламалық өнімдерді (ПП) дамытудағы тәжірибелік дағдыларын дамыту. Осы әдістердің басты идеясы ПП-ны құрудағы инженерлік тәсілдерді қолдану болып табылады.

Семинарда «Бағдарламалық өнімдерді әзірлеу технологиясы» тренингінің негізгі аспектілері қарастырылған. Семинарда сол атаудағы оқулықта талқыланған теориялық және практикалық мәселелер толығырақ қарастырылады. Семинарда оларды іске асыру бойынша міндеттер мен нұсқаулардың тізімі келтірілген. Оқулықтың құрылымы оқу жоспарында қабылданған материалды таныстырудың дәйектілігін көрсетеді. Семинар 13 тараудан, сілтемелер тізімі мен қосымшадан тұрады. Әрбір тарауда теориялық бөлік, практикалық тапсырмалар, оларды іске асыру бойынша әдістемелік нұсқаулар және бақылау сұрақтар бар. Материалды таныстыру тапсырманы орындаудың үлгісі - техникалық тапсырманы әзірлеуден бастап дайын бағдарламалық өнімді алу үшін жасалған.

Соңғы тарауда «Бағдарламалық өнімдерді әзірлеу технологиясы» пәні бойынша курстық жобаның құрылымы, құжаттаманың құрамы және курстық жобалардың орындалу мысалдары келтірілген.

Қосымшада тәжірибелік тапсырмалар бойынша тапсырмалар ретінде де, курстық дизайны үшін тақырыптар ретінде қолданыла алатын жеке тапсырмалар тізімі де бар.

ТЕХНИКАЛЫҚ ҚЫЗМЕТТІ ДАМУ

1.1 НЕГІЗГІ АҚПАРАТ

«Техникалық тапсырма» бағдарламалық құжаты ГОСТ 19.201-78 сәйкес әзірленген. Техникалық тапсырма бағдарламалық қамтамасыз етуге қойылатын талаптар жиынтығын қамтиды және әзірленген бағдарламалық қамтамасыз етуді тексеру және қабылдау критерийі ретінде пайдаланылуы мүмкін, сондықтан жеткілікті толық (қосымша бөлімдерді қосу мүмкіндігін ескере отырып) және тапсырыс беруші мен әзірлеуші қабылдайтын техникалық тапсырма іргелі жобаның құжаттары. Бағдарламалық жасақтама өнімін дамыту үшін техникалық тапсырманы дұрыс жасау мүмкіндігі бағдарламашының кәсіби деңгейін белгілейді және тапсырыс берушінің талаптарын қанағаттандырады.

Тұжырымдама - дамудың негізгі мақсаттары, бағдарламалық өнімге қойылатын талаптар тұжырымдалған, дамудың кезеңдері мен кезеңдері айқындалған және қабылдау сынақтарының процесі реттелетін құжат. Тапсырыс беруші мен орындаушының өкілдері техникалық тапсырманы қалыптастыруға қатысады. Бұл құжат тапсырыс берушінің бастапқы талаптарына, алдын ала жобаны зерттеу нәтижелеріне және т.б. негізделеді.

Техникалық тапсырманы әзірлеу келесі кезектілікпен жүзеге асырылады:

- 1) орындалатын функциялардың жиынтығын, сондай-ақ бастапқы деректердің тізбесі мен сипаттамаларын белгілейді;
- 2) нәтижелер тізімін, олардың сипаттамаларын және оларды ұсыну жолдарын айқындайды: бағдарламалық жасақтаманың жұмыс істеуі үшін ортаны белгілейді: техникалық құралдардың нақты жабдықтары мен параметрлері, пайдаланылатын операциялық жүйенің нұсқасы және болашақта бағдарламалық жасақтама өнімі өзара әрекеттесетін басқа орнатылған бағдарламалық жасақтаманың нұсқалары мен параметрлері болуы мүмкін.

Бағдарламалық жасақтама әзірленсе, қандай да бір ақпаратты жинайды және сақтайды немесе кез-келген техникалық процессті басқаруға енгізілген жағдайда, жабдықтың істен шығуы

және электрмен жабдықтау істен шыққан жағдайда бағдарламаның іс- әрекеттерін нақты реттеу керек.

Дамып жатқан бағдарламалық жасақтаманың сипаттамаларын анықтайтын негізгі факторлар:

■ бағдарламаның немесе жүйенің функцияларын анықтайтын бастапқы деректер мен қажетті нәтижелер;

■ әзірленетін бағдарламалық жасақтама жұмыс істейтін қоршаған орта (бағдарламалық жасақтама және аппараттық құрал), анықталуы немесе техникалық тапсырмада көрсетілген параметрлерді қамтамасыз ету үшін таңдалуы мүмкін;

■ Басқа бағдарламалық жасақтамамен және / немесе нақты техникалық құралдармен ықтимал өзара әрекеттесу, сондай-ақ орындалатын функциялар жиынтығы негізінде анықталуы немесе таңдалуы мүмкін.

ГОСТ 19.201-78 сәйкес, «Техникалық тапсырма» бағдарламалық құжатында келесі бөлімдер бар.

Кіріспе.

1. Дамудың негізі.
2. Дамудың мақсаты.
3. Бағдарламаға немесе бағдарламалық өнімге қойылатын талаптар.
4. Бағдарламалық құжаттамаға қойылатын талаптар.
5. Техникалық-экономикалық негіздемесі.
6. Даму кезеңдері мен кезеңдері.
7. Бақылау және қабылдау тәртібі.
8. Өтініштер.

Бағдарламаның немесе бағдарламалық өнімнің ерекшеліктеріне қарай бөлімдердің мазмұнын көрсетуге, жаңа бөлімдерді енгізуге немесе кейбіреулерін біріктіруге болады. Келісімнің бөлімдерінің мазмұнын толығырақ қарастырайық.

Кіріспе бөлімінде бағдарламалық жасақтама өнімі, қолдану саласының қысқаша сипаттамасы және ол қолданылатын объектінің сипаттамасы, яғни, ол қолданылатын нысанның сипаттамасы. тақырыптық аймақтың сипаттамасы.

1. «Дамуға негіздеме» бөлімінде

- әзірлеуге негізделген құжаттар;
- осы құжатты мақұлдаған ұйым және оны бекіту күні;
- даму тақырыбының атауы және / немесе символы.

2. «Дамудың тағайындалуы» бөлімінде белгіленген мақсатқа жету үшін дамыған жүйені шешу керек функционалдық және операциялық міндеттердің анықтамасы бар. Бағдарламаның мақсаты техникалық кешенді басқару, әртүрлі шығын сметаларын,

өндірісті жетілдіру және т.б. болуы мүмкін. Ақпараттық жүйелерді бағдарламалық қамтамасыз етуді ұсыну қажет болса, мақсатты, объективті шешімдерді қабылдау, пайдаланушыны кеңсе жұмысындағы қалыпты жұмыстан алып тастау және мекемені қағазсыз технологияға көшіру және т.б. үшін уақтылы және дәл ақпарат алу мүмкін. Бұл бөлімде проблеманың бастапқы контекстік диаграммасы ұсынылуы керек.

3. Бағдарламаға немесе бағдарламалық өнімге қойылатын талаптар» бөлімінде келесі бөлімдер бар:

функционалдық сипаттамаларға қойылатын талаптар;

■ сенімділік пен қауіпсіздікке қойылатын талаптар;

■ пайдалану шарттары;

■ техникалық құралдардың құрамы мен параметрлеріне қойылатын талаптар;

■ ақпарат пен бағдарламалық қамтамасыз етудің үйлесімділігіне қойылатын талаптар;

■ таңбалауға және орауға арналған талаптар;

■ сақтауға және тасымалдауға қойылатын талаптар;

■ арнайы талаптар.

Функционалдық сипаттамаларға қойылатын талаптар орындалатын функциялардың құрамын, кіріс және шығыс ақпараттарына қойылатын талаптарды, сондай-ақ бағдарламаның қызметтік функцияларын қамтиды. Бағдарламаның функцияларын анықтау үшін болашақ пайдаланушылардың жұмысын мұқият зерделеп, қолмен орындалатын барлық операциялардың тізімін жасаңыз немесе басқа бағдарламаларды пайдаланып, олардың арасында автоматтандыруға жататындарды бөліңіз. Мысалы, ақпараттық жүйенің бағдарламалық қамтамасыз етуінің негізгі функционалдық сипаттамалары мыналарды қамтиды:

■ іздеу жүйесін пайдалана отырып, дерекқордан қажетті ақпаратты іздеу және таңдау мүмкіндігі;

■ Таңдалған деректер негізінде қажетті есеп беру пішіндерін қалыптастыру;

■ деректер базасын пайдалана отырып қажетті есептеулер мен есептеулер;

■ Қолданыстағы деректер базасын басқа қосымшаларға ұсыну мүмкіндігі;

■ пайдаланушының интернетпен жүйеде жұмыс істеу қабілетін және т.с.с.

Бұдан басқа, функционалдық сипаттамаларға сүйене отырып, осы жүйеде қолданылатын деректер файлдарының құрылымы мен мақсаты (электронды анықтамалар, құжаттар журналдары,

электрондық дербес файлдар, мұрағаттар және т.б.) анықталған. Осы кезеңде міндеттерді табысты шешуге қажетті ақпараттық жүйенің (клиент-сервер, файл-сервер) архитектурасын анықтау қажет.

Кіріс деректеріне қойылатын талаптарды сипаттағанда, олардың сипаты, ұйымдастыру және алдын ала дайындық, форматы, сипаттамасы және кодтау әдісі көрсетілуі керек. Бағдарламаның кіріс ақпараты бастапқы құжаттар (шот-фактура, есептер және т.б.), нормативтік-анықтамалық ақпарат (каталогтар, классификаторлар, кодификаторлар және т.б.), электронды құжаттар, кіріс сигналдары және т.б. болуы мүмкін. Бағдарламаның шығыс ақпараты құжаттар (электрондық немесе қағаз), деректер файлдары, шығыс сигналдары және т.б. болуы мүмкін. Шығарылатын деректерге қойылатын талаптарды сипаттағанда, олардың табиғаты, ұйымдастыру, форматы, сипаттамасы және кодтау әдісі көрсетілген.

Техникалық тапсырмадағы негізгі функциялардан басқа, жүйенің параметрлерін (конфигурациясын), деректердің сақтық көшірмесін жасау мүмкіндігін, кіру құпия сөзін өзгертуді, күнтізбелік бағдарламадан, калькуляторды, редакторды және т.б. қалдырмай қоңырау шалу мүмкіндігі сияқты бағдарламаның қызметтік функцияларына қойылатын талаптар сипатталады. Егер әзірленген бағдарламалық қамтамасыз ету техникалық тапсырмада көрсетілген функцияларды орындамаса, ол техникалық сипаттамаға сәйкес келмейді, яғни, сапа критерийлері бойынша дұрыс емес. Болашақ өнімнің әмбебаптығы, әдетте, нақты көрсетілмеген, бірақ ол болжанады.

Сенімділік пен қауіпсіздікті қамтамасыз ету талаптары бағдарламалық өнімдердің сенімді және тұрақты жұмыс істеуін қамтамасыз ету, кіріс және шығыс ақпаратын бақылау, қалпына келтіруден кейінгі қалпына келтіру уақытын және т.б. Сенімділік - бұл белгілі бір уақыт ішінде белгілі бір жағдайларда белгілі бір функцияларды орындауға қабілетті бағдарлама өте жоғары ықтималдықпен. Сенімді бағдарламалық жасақтама өнімі қателіктердің болуын жоққа шығармайды, бірақ белгілі бір жағдайларда тәжірибеде қолданылатын қателер сирек кездеседі. Сенімділіктің дәрежесі белгілі бір уақыт кезеңі ішінде ақауларсыз жұмыс істейтін бағдарламалық өнімнің ықтималдылығымен сипатталады. Жүйенің сенімділігін қамтамасыз етудің көптеген тәсілдері бар (қателерді болдырмау, қателерді түзету, жүйенің сәтсіздіктен кейін өзін-өзі қалпына келтіруі, кіру деректерін

қолайлы мәндер шегінде тексеру және т.б.). Ең оңай жолы - кіруді шектеу. Бағдарламалық жасақтама өніміне және деректер базасына қол жеткізуді бақылау бағдарламаларды іске қосу үшін бағдарламаларды немесе деректерді шектеу үшін негізгі дискіні пайдаланып, пайдаланушыларға қол жетімді өңдеу функцияларын және т.б. іске қосылған кезде құпия сөзбен қорғау бағдарламалары арқылы жасалады.

Құрамына және параметріне, техникалық құралдарға қойылатын талаптар техникалық құралдардың құрамына және олардың негізгі сипаттамаларына нұсқауды қамтиды, атап айтқанда: бағдарламаның жұмыс істеуі үшін қажетті ең аз жүйелік талаптар; (Гц), оның негізінде компьютер жұмыс істеуі керек, оперативті жады (МБ), бос дискілік кеңістіктің қажетті саны, монитордың рұқсаты, CD-ROM дискі бар болуы және т.б., сондай-ақ, бір аппараттық платформасынан басқасына.

Ақпарат пен бағдарламалық қамтамасыз етудің үйлесімділігіне қойылатын талаптар ақпараттық құрылымдарға, бағдарламалау тілдеріне және бағдарламамен пайдаланылатын бағдарламалық құралдарға, атап айтқанда:

- бағдарламалық өнімдерді әзірлеуге болатын операциялық жүйелер мен орталарға қойылатын талаптар;
- бағдарламаны әртүрлі операциялық жүйелерге бейімдеу мүмкіндігі; компьютерлік қосымшаларды әзірлеу құралдарына (бағдарламалық жасақтаманы қайта қарау, жаңғырту немесе пайдалану үшін) бағдарламалық жасақтама пакеттерін орнату қажеттілігі;
- компьютерлік қосымшаларды әзірлеу құралдарына (бағдарламалық жасақтаманы қайта қарау, жаңғырту немесе пайдалану үшін) бағдарламалық жасақтама пакеттерін орнату қажеттілігі;
- түрлі графикалық компоненттерді орнату қажеттілігі және т.б.

4. «Бағдарламалық құжаттамаға қойылатын талаптар». Келешектегі бағдарламалардың дамуын реттейтін негізгі құжаттар бірыңғай бағдарлама құжаттамасы жүйесінің құжаттары болуы тиіс: пайдаланушы нұсқаулығы, әкімшілік нұсқаулық, қолданба сипаттамасы.

5. Жүйенің тиімділігі оны пайдаланудың ыңғайлылығы мен бағдарламалық-аппараттық кешенді енгізуден алынған экономикалық пайдалармен анықталады.

6. «Техникалық-экономикалық негіздеме» бөлімінде әзірленген бағдарламалық өнімнің экономикалық тиімділігі,

кәсіпорынның қолда бар үлгілері немесе аналогтары (немесе қолмен жасалатын операциялармен салыстырғанда) салыстырғанда экономикалық дамудың артықшылықтары.

7. Дамудың сатылары мен сатылары АВ Рудаковтың «Бағдарламалық жасақтама әзірлеу технологиясы» оқулығында сипатталған. «Бақылау және қабылдау тәртібі» жұмыс түрлерін қабылдаудың жалпы талаптары мен түрлерін көрсетуді білдіреді.

«Техникалық тапсырма» бағдарламалық құжатында өтінімдерді енгізу қажет, қажет болған жағдайда:

- кіріс және шығыс құжаттары мен есептерінің үлгілері, деректер файлдарының сипаттамасы және т.б.

- дамуды негіздейтін зерттеулер мен басқа да жұмыстардың тізбесі;

- алгоритмдер схемалары, кестелер, сипаттамалар, негіздемелер, есептер және т.б. ;

1.2 ТЕХНИКАЛЫҚ ТАПСЫРМАНЫ ӘЗІРЛЕУДІҢ МЫСАЛДАРЫ

1.2.1. AIS «Көтерме сауда қоймасы» бағдарламалық жасақтамасын әзірлеуге арналған техникалық тапсырма

Кіріспе

Жұмыс сауда кәсіпорны басқару үшін автоматтандыру жобасының шеңберінде жүзеге асырылады.

Келісімді бекіткен ұйым:

1. Жұмыс сипаттамасы «Көтерме сауданың қоймасы» автоматтандырылған ақпараттық жүйесі

2. Дамудың мақсаты

«Көтерме саудалардың қоймасы» автоматтандырылған ақпараттық жүйесі, көтерме сауда үшін сатып алынған тауарлардың қозғалысы мен қол жетімділігі туралы ақпараттарды қорытуға арналған.

Бағдарламаны пайдаланушылар қойма менеджерлері, бухгалтерия, тапсырыс беру және тіркеу бөлімі болып табылады.

Жеткізушілерден тауарларды сатып алу жеткізу шарттарын анықтайтын сатып алу-сату шарттарымен жүзеге асырылады. Тауарлардың келуіне арналған бастапқы құжаттардың деректері тауардың түбіртегінің журналында, оның түпнұсқасы,

оның күні мен нөмірі, құжаттың қысқаша сипаттамасы, құжаттың тіркелген күні, алынған тауарлар туралы мәлімет бар.

Тауарларды сатуды жүзеге асыру және есепке алу сатып алушы мен сатушы арасында сатып алынған тауарларды есептеу әдісіне байланысты. Тауар ақшалай және қолма-қол ақшасыз төлемдер үшін сатылады. Менеджер тауарларды шығаратын тізілімін жүргізеді, онда: тапсырыс нөмірі, демалыс күні, тауардың атауы, бірліктердің саны және демалыс мөлшері, тауар шығарған тұлғаның аты, аты-жөні және қолы. Бастапқы құжаттардың деректері есепке алу карточкаларында тіркеледі, олар қойманың тізілімі және бухгалтерлік есеп шоттары болып табылады. Бухгалтер - аптасына кемінде бір рет кітаптарда немесе карталарда жазбаларды тексеруді жүзеге асырады. Тауарды алу және жою жөніндегі операцияларды есепке алу кезінде есепке алу үшін негіз болып табылады, оларға қоса берілген құжаттармен тауарлық есептер жатады.

3. Бағдарламаға қойылатын талаптар

3.1 Функционалдық талаптар

«Көтерме сауданың қоймасы» автоматтандырылған ақпараттық жүйесі мынадай функцияларды қамтамасыз етуі керек:

- қоймадағы тауарлардың келуі және сатылуы туралы ақпаратты енгізу, сақтау, іздеу және өңдеу;

- кіріс және шығыс құжаттарының тіркелуін жүргізу;

- қоймадағы тауарлардың болуы туралы ақпаратты уақтылы алу;

- тауарларды алу және сату туралы барлық деректерді қамтитын менеджер мен бухгалтердің талаптарын есепке алу.

«Көтерме саудалардың қоймасы» автоматтандырылған ақпараттық жүйесінің анықтамалық ақпараты контрагенттердің, номенклатураның, өлшеу бірліктерінің анықтамалық материалдарымен ұсынылған.

Көтерме саудада тауарларды тіркеу үшін негізгі құжаттар:

- сатылатын күнін, сатылған тауарлардың тізімін, олардың саны, бағасы мен сатудың жалпы сомасын қамтитын шот-фактуралар, қолма-қол ақшалар мен тауарлық чектер. Шығындық шот-фактуралар тауарларға тапсырыстар негізінде толтырылады;

- жеткізушінің тауарды жеткізуге арналған құжаттары, мынадай

ақпаратты қамтитын: қоймаға тауарларды алу күні, жеткізуші

туралы ақпарат, алынған тауарлардың тізімі, саны, бағасы және жалпы сомасы.

Есептердің келесі түрлері шығарылады:

- жеткізушілер туралы мәліметтерді, алынған тауарлардың тізімін, олардың нөмірін, бағасын, тауарлардың әрбір элементі үшін алынған соманы және алынған жалпы соманы қамтитын белгілі бір мерзімге тауарларды алу туралы есеп;

- сатылған тауарлардың тізімін, олардың бағасы, саны, сатылған тауарлардың әрбір сатысы бойынша сату көлемін және сатудың жалпы сомасын қамтитын белгілі бір мерзімге арналған сату есебін;

- Әрбір өнімнің түсуі мен сатылуы туралы деректерді қамтитын

түгендеу картасы.

Бағдарламада жүйе параметрлерін түзету мүмкіндігін қамтамасыз

ету қажет; резервтік деректерді сақтау; кіру құпия сөзін өзгерту мүмкіндігі; кіріктірілген көмек жүйесінің болуы; қажетті құжаттарды жылдам іздеу және анықтамалық ақпарат және т.б. ;

3.2 Сенімділікке қойылатын талаптар

Бағдарламалық жасақтама болуы керек:

- Ақаулықтан кейін (қуатты өшіру, операциялық жүйеде жұмыс

істемеу және т.б.) өзін-өзі қалпына келтіру мүмкіндігі;

- бағдарламаны іске қосқанда құпия сөзбен қорғау;

- деректерге рұқсатсыз кіруді шектеу;

- дереккордың сақтық көшірмесін жасау мүмкіндігі;

- пайдаланушылардың құқықтарын саралау;

- бағдарламаның рұқсатсыз көшіруін (репликациясын)

болдырмау.

Жүйеде жұмыс істеген кезде енгізілген ақпаратты бақылауды қамтамасыз етеді және пайдаланушының қате әрекеттерін болдырмайды.

3.3.Техникалық құралдардың құрамы мен параметрлеріне қойылатын талаптар

Бағдарламалық өнімді іске қосу үшін жүйелік талаптар келесідей болуы керек: процессордың сағат жылдамдығы ~ 1 200 Гц; ЖЖҚ көлемі - 64 МБ; бос дискілік кеңістіктің көлемі - 50 МБ; 1,024 x 768 қарардың монитормы; CD-ROM жетегінің болуы.

3.4. Ақпарат пен бағдарламалық қамтамасыз етудің

үйлесімділігіне қойылатын талаптар.

Бағдарлама Windows 2000 / XP операциялық жүйелерінде жұмыс істеуі керек. Барлық жасалған есептерді MS Office Excel 2003/2007 электрондық кестесінің редакторына экспорттау мүмкіндігі болуы керек.

3.6 Тасымалдау мен сақтауға қойылатын талаптар

Бағдарлама лазерлік деректер тасымалдаушысына жеткізіледі.

Бағдарламалық қамтамасыздандыру құжаттамасы электрондық және баспа түрінде беріледі. Арнайы талаптар

Бағдарламалық қамтамасыз ету орташа біліктілікті (компьютерлік сауаттылық тұрғысынан) пайдаланушыға арналған ыңғайлы интерфейске ие болуы керек.

Жобаның көлемін есепке ала отырып, міндеттерді кезең-кезеңмен шешу қажет. Сонымен қатар әр түрлі уақытта жасалған программалық модульдер (бағдарламалық қамтамасыз ету) жүйені жоғарылату мүмкіндігін және бір-бірімен үйлесімді болуға тиіс; сондықтан қабылданған операциялық бағдарламалық қамтамасыз етуге арналған құжаттама бағдарламашылармен жұмыс істеу үшін қажетті толық ақпаратты қамтуы тиіс. Бағдарламалау тілі орындалатын файлды таңдаумен анықталады және ол бағдарламалық жасақтаманы MS Office 2003/2007 пакетімен біріктіру мүмкіндігін қамтамасыз етуі керек.

4. Бағдарламалық құжаттамаға қойылатын талаптар.

Бағдарламаны әзірлеу барысында бағдарламаның мәтіні, бағдарламаның сипаттамасы, тестілеу бағдарламасы мен әдістемесі, пайдаланушы нұсқаулығы, техникалық-экономикалық негіздеме жасалуы тиіс.

Кеден қоймасына түсетін түсімдерді және оларды қоймаға жөнелту операцияларын жүргізу кезінде қолмен жұмыс күшін, яғни келу және тауарларды тұтыну кітабы пайдаланылады. Әрине, бағдарламаны пайдалану өнімді тіркеу үшін қажетті уақытты едәуір қысқартады. Бұдан басқа, қоймадағы әрбір зат үшін түгендеу картасы бар; ол сондай-ақ тауарларды қабылдау, шығындар мен теңгерім туралы ақпаратты ұсынады. Белгілі бір өнім үшін осы деректерді алу үшін кем дегенде 8-10 минут қажет. Бағдарламаны пайдалану уақытын 1-ден 2 минутқа дейін қысқартады. Әрбір айдың соңында жауапты қойма қызметкері қоймадағы тауар айналымы туралы есептерді дайындайды және әрбір бөлшекті тауардың қалған бөлігін көрсетеді. Бұл операция 1-ден 2 күнге созылады, яғни. 6-12 сағат компьютерде айналым парағының қалыптасуы 3 - 4 минутты құрайды. Бұдан басқа, кез-келген

уақытқа есептерді алуға болады деп болжануда. Есептерді қолмен жасау кезінде адам қате жібере алады; Қате жасалған әзірленген бағдарламаның алгоритмі дұрыс алгоритмі жоқ.

«Көтерме сауданың қоймасы» автоматтандырылған ақпараттық жүйесін енгізудің экономикалық тиімділігі басшылар жүргізетін операциялардың уақытын қысқарту, есептерді қалыптастырудағы қателіктерді жою, экономикалық қызметті талдау үшін уақытты жоғарылату және т.б. байланысты.

1.2.2. Комбинаторлық мәселелерді шешетін жүйені әзірлеуге арналған техникалық тапсырма

Кіріспе

Осы Техникалық тапсырма осы міндеттерге арналған деректерді енгізу және сақтауға арналған, сондай-ақ оларды шешу, алынған нәтижелерді сақтау және компьютерлік бағдарламалық жасақтаманың және аппараттық құралдарды әзірлеушілердің пайдалануына арналған комбинаторлы оңтайландыру міндеттерін шешудің шешімдер жүйесін әзірлеуге қолданылады.

Техникалық нысандардың, соның ішінде компьютерлердің әртүрлі техникалық тапсырмалары үшін комбинаторлық оңтайландырудың классына жатады, әдетте, қазіргі заманғы компьютерлерде экспоненциалды есептеу қиындыққа ие және оларды іске асыру мүмкін емес нақты әдістер. Қазіргі уақытта осындай есептерді шешу үшін әртүрлі есептеу ресурстарын қажет ететін және шешімнің тең емес дәлдігін беретін жуықтау әдістері мен алгоритмдері кеңінен қолданылады.

Сонымен қатар, осы әдістер мен алгоритмдер жүйелелмеген, олардың есептеу және сыйымдылығы күрделілігіне баға беріліп, алынған шешімдердің ықтимал дәлдігі туралы ақпарат көптеген көздерден толық емес және шашыраңқы болып табылады. Бірыңғай жүйе шеңберінде жобалаудың негізгі комбинаторлық оңтайландыру мәселелерін шешуге арналған алгоритмдердің шектеулі ауқымы үшін бағдарламалық жасақтама жоқ.

Ең көп кездесетін әдістер мен осы проблемаларды шешуге арналған алгоритмдер іске асатын жүйені құру жеке әдістер мен алгоритмдерді бағалау мен зерттеуге және оларды есептеу ресурстарының шығындарымен және алынған шешімдердің дұрыстығымен салыстыруға мүмкіндік береді.

1. Дамуға негіз

Жүйе директордың оқу ісі жөніндегі орынбасарының бұйрығының негізінде әзірленді және ... оқу жоспарына сәйкес 200_____—200__ г.

2. Дамудың мақсаты

Жүйе графиктерде комбинаторлық-оңтайландырудың шағын шеңберін шешуге арналған:

ең аз ұзындық циклін іздестіру (сатушы мәселесі бойынша саяхатшы);

ең қысқа жолды іздеу;

ең аз міндетті ағаштарды іздеңіз.

Пайдаланушылар компьютерлерді жобалауға, тиісті мамандықтардың студенттеріне, сондай-ақ басқа да тақырыптық салалардағы мамандарға ұқсас мәселелер бойынша шешуге тура келетін ғалымдар мен инженерлер бола алады.

3. Бағдарламаға немесе бағдарламалық өнімге қойылатын талаптар

3.1. Функционалдық талаптар

Жүйе мынадай міндеттерге арналған әдістемелік және бағдарламалық шешімдер жиынтығын ұсынуы керек:

■ ең төменгі жабу ағашының құрылысы;

■ ең аз ұзындығы жабу циклін іздестіру (сатушы мәселесі бойынша саяхатшы);

■ Ең қысқа жолды іздеңіз.

Осы міндеттерді орындау үшін төмендегілерді орындау қажет:

■ дәл шешім беретін алгоритм;

■ егер дәл шешім многодинамикалық есептеу қиындықтары жоқ алгоритммен берілсе, онда полиномдық есеп айырысу күрделілігімен шамамен шешімдерді ұсынатын алгоритмді одан әрі дамыту қажет.

Әдістемелік қолдауды жүйенің пайдаланушылық интерфейсінде жүзеге асыру керек, ол мыналарды көздеуі керек:

■ міндеттерді, әдісті және оны шешудің алгоритмін таңдау;

■ деректерді енгізу;

■ жобалық тапсырманы шешу және енгізу деректерін, аралық және соңғы нәтижелерді енгізілген деректер базасына одан әрі талдау үшін сақтау.

3.2 Сенімділікке қойылатын талаптар

Бағдарламалық жасақтама бағдарламалық жасақтаманың бағдарламалық жасақтаманың сенімділік талаптарының қазіргі деңгейіне сәйкес келуі керек:

және енгізілген ақпараттарды бақылауды және жүйемен жұмыс істеген кезде пайдаланушы дұрыс емес әрекеттерді блоктауды қамтамасыз етеді .

3.3. Техникалық құралдардың құрамы мен параметрлеріне қойылатын талаптар

Бағдарламалық өнімді іске қосу үшін жүйелік талаптар келесідей болуы керек: процессордың сағат жылдамдығы ~ 1000 Гц; ЖЖҚ көлемі - 64 МБ; бос дискілік кеңістіктің көлемі 20 Мбайт; 1,024 x 768 қарардың монитору; CD-ROM жетегінің болуы; принтер.

3.4 Ақпарат пен бағдарламалық қамтамасыз етудің үйлесімділігіне қойылатын талаптар

Бағдарлама Windows 2000 / XP операциялық жүйелерінде жұмыс істеуі керек.

4. Бағдарламалық құжаттамаға қойылатын талаптар

Дамыған жүйе жүйенің жұмысы туралы анықтамалық ақпаратты қамтуы және пайдаланушыны шақыруы керек. Ілеспе құжаттама мыналарды қамтуы тиіс: жүйенің сипаттамасын қамтитын есептеу және түсіндірме жазбаны; пайдаланушы нұсқаулығы; жүйелік бағдарламашыны басқару.

5. Даму кезеңдері (1.1-кесте)

Техникалық тапсырманы мақұлдағаннан кейін, әзірлеуші ұйым бағдарламалық жасақтаманы құруды бастайды.

Тапсырма

Қосымшада берілген опциялардың біріне сәйкес бағдарламалық жасақтама өнімі үшін «Техникалық тапсырма» құжатын жасаңыз. ГОСТ 19.106-78 бойынша жұмысқа орналасу.

1.1-кесте. Даму

Кезең номері	Кезеңнің аты	Мерзімі	Есеп беру
1	Негізгі жүйені дамыту	1.1.2008— 31.3.2008	Дерекқордың ішкі пішімдерінің, интерфейстерінің және деректер пішімдерінің сипаттамасы. Интерфейс деңгейінде жүйені енгізу
2	Әдістемелер мен алгоритмдерді әзірлеу және оларды сатушы сатушының	1.4.2008— 30.6.2008	Бағдарламалық қамтамасыз етудің модульдері
3	Кішігірім қосылыс ағашының құрылысы мәселесі бойынша әдістер мен алгоритмдерді жасау және оларды енгізу графиктегі ең қысқа жолды табу мәселесі	1.7.2008— 30.9.2008	Әдістемелер мен алгоритмдердің сипаттамасы. Бағдарламалық қамтамасыз етудің модульдері
4	Бағдарламалық өнімді тестілеу және бағдарламалық құжаттарды жасау	1.10.2008— 31.12.2008	Тесттер. Құжаттама. Бағдарламалық өнім

БАҚЫЛАУ СҰРАҚТАРЫ

1. Бағдарламалық жасақтама әзірлеудің негізгі кезеңдерін көрсетіңіз.
2. Алдын-ала зерттеулерге не жатады?
3. Техникалық тапсырманың негізгі бөлімдерін атаңыз.
4. Бағдарламалық жасақтама өніміне арналған функционалды және операциялық талаптарды көрсетіңіз. Олардың негізгі айырмашылығы қандай?
5. «Техникалық тапсырма» құжатының негізгі бөлімдерін беріңіз.

БАҒДАРЛАМАЛЫҚ ҚАМТАМАСЫЗ ЕТУГЕ ҚОЙЫЛАТЫН ТАЛАПТАРДЫ ТАЛДАУ ЖӘНЕ ҚҰРЫЛЫМДЫҚ ҰСТАНЫМДЫ АНЫҚТАУДА ЖҮЙЕЛІ ӘДІСТІ ҚОЛДАНУ

2.1. НЕГІЗГІ АҚПАРАТ

Шындығында, кез-келген бағдарламалық жасақтаманың дамуы болашақ бағдарламалық жасақтама өнімдеріне қойылатын талаптарды талдау негізінде құрылады. Талдау нәтижесінде әзірленген бағдарламалық қамтамасыз етудің ерекшеліктері алынды: олар шешілетін шешімдердің бөлінуін және мазмұнын қалыптастыруды жүзеге асырады, олардың өзара әрекеттесуін анықтап, операциялық шектеулерді айқындайды. Спецификацияларды анықтау процесінде субъектінің жалпы моделі шынайы әлемнің бөлігі ретінде құрастырылған, оның көмегімен бағдарламалық жасақтама дамып, бір-бірімен өзара әрекеттеседі және оның негізгі функцияларын анықтайды.

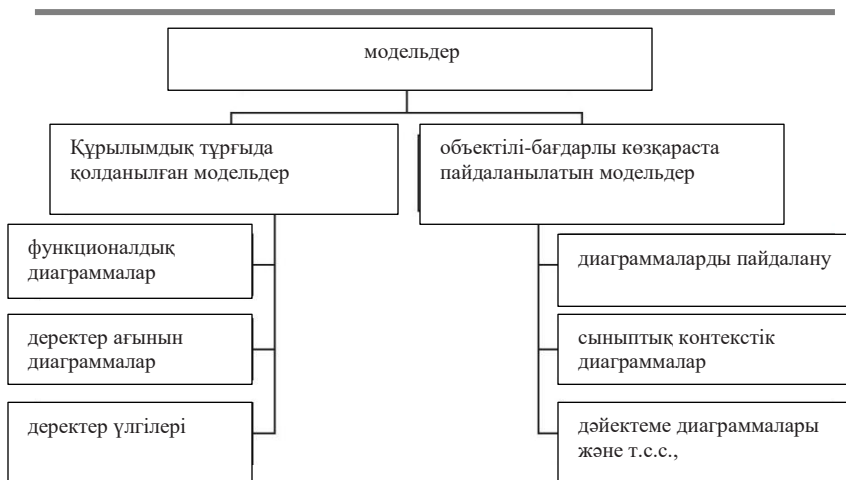
Ерекшеліктер - әзірленген бағдарламалық жасақтаманың функциялары мен шектеулерінің толық және нақты сипаттамасы. Бұл жағдайда функционалдық ерекшеліктер әзірленген бағдарламалық жасақтаманың функцияларын сипаттайды және техникалық сипаттамалар техникалық құралдарға, сенімділікке, қауіпсіздікке қойылатын талаптарды анықтайды. Функционалдық ерекшеліктерге қатысты толықтығы талаптары ең маңызды шешімдерді таңдауға кедергі келтірмеу үшін ешқандай маңызды ештеңе болмағандықтан және ешқандай маңызды емес ақпарат болмауы үшін барлық қажетті ақпаратты қамтуы керек дегенді білдіреді. Дәлдіктің талаптары техникалық сипаттамаларды тапсырыс беруші мен әзірлеуші бірдей қабылдайтындығын білдіреді.

Соңғы талапты орындау өте қиын, өйткені сипаттамаларды сипаттауға арналған табиғи тіл жарамсыз: табиғи тілдегі егжей-тегжейлі спецификациялар қажетті дәлдікті қамтамасыз етпейді. Бағдарламалық жасақтаманың нақты ерекшеліктері осы бағдарламалық жасақтаманың ресми формасын жасау арқылы ғана

анықталуы мүмкін.

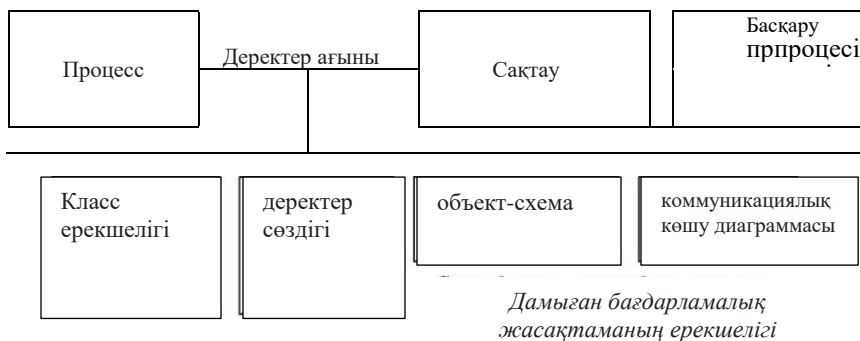
Жасалып жатқан бағдарламалық жасақтаманың барлық функционалдық ерекшеліктері өңделетін деректердің құрамын және құрамының тізімін сипаттайды. Талаптарға талдау жасау және спецификацияларды анықтау процесінде әзірлеуші пайдаланатын басымдықтар (екпін) жүйесінде ғана ерекшеленеді. Мәселен, мемлекеттік көшу диаграммалары уақыт бойынша бағдарламалық қамтамасыз етудің кейбір мінез-құлқын анықтайды, деректер ағынын диаграммалары - деректер ағындарының бағыты мен құрылымы және сыныптың концептуальды диаграммалары - доменнің негізгі түсініктері арасындағы байланыс. Суретте. 2.1-де әзірленген бағдарламалық жасақтама үшін спецификация ретінде қолданылатын үлгілердің классификациясы көрсетілген.

Құрылымдық көзқарас шеңберінде талдау және спецификацияларды анықтау кезеңінде модельдердің үш түрі пайдаланылады: деректерге бағдарланған және деректер ағындарына бағдарланған функцияларға бағдарланған. Модельдің әрбір түрі бағдарламалық жасақтама әзірлеудің нақты сыныбына қолдануға ыңғайлы. Әртүрлі модельдер әртүрлі жағынан болжанған бағдарламалық жасақтаманы сипаттайды. Деректер ағындарын модельдеуге негізделген құрылымдық талдау және жобалау әдіснамасы әдетте төмендегі үлгілер жиынтығы ретінде жасалған бағдарламалық жасақтаманың күрделі көрінісін пайдаланады:



Сурет. 2.1. Спецификация фазасында қолданылатын бағдарламалық жасақтама модельдерінің классификациясы

Деректер ағындарының егжей-тегжейлі диаграммасы



Сурет. 2.2. Деректер ағындарына негізделген құрылымдық талдау және бағдарламалық жасақтама дизайны бойынша әдістемелердің толық спецификациясының компоненттері

- Жүйенің мінез-құлқын уақытында сипаттайтын мемлекеттік көшу диаграммалары (SDT);
- функционалдық диаграммалар (SADT - Құрылымдық талдау және жобалау техникасы);
- Деректер ағынының диаграммасы (DFD - Деректер ағыны схемасы), жүйеде іске асырылуы тиіс процестер арқылы ақпарат көздері мен пайдаланушылары арасындағы өзара әрекеттесуді сипаттайды;
- Дамып жатқан жүйенің деректер базасын сипаттайтын объект-қатынас диаграммалары (ERD).

Құрылымдық талдауға арналған әдістемелік құрамдардың компоненттері 2.2. - сурет

Процесс ерекшелігі әдетте қысқа мәтіндік сипаттама, алгоритм схемалары, псевдокодтар түрінде ұсынылады.

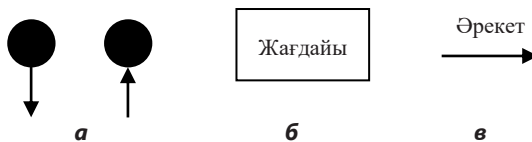
Деректер сөздігі - спецификацияларды жасау кезінде пайдаланылатын негізгі түсініктердің қысқаша сипаттамасы. Сөздік доменнің негізгі түсініктерін анықтауды, деректер элементтерінің құрылымын сипаттауын, олардың түрлері мен форматтарын, сондай-ақ барлық қысқартулар мен белгілерді қамтуы керек. Сөздік тақыр сіну дәрежесін арттыруға арналған.

2.2. ЖАҒДАЙЛАРДЫҢ ӨТПЕЛІ ДИАГРАММАЛАРЫ

Мемлекеттік өтпелі диаграммалардың сипаттамасы (SDT). SDT диаграммалары бақылау әрекеттерін қабылдаған кезде дамитын бағдарламаның мінез-құлқын көрсетеді. Басқару іс-әрекеттері немесе сигналдар астында бұл жағдайда жүйенің сырттан келген бақылау туралы мәліметтері бар; мысалы, басқару пәрмендері компьютердің жүйесімен қосылған датчиктердің пайдаланушы командалары мен сигналдары арқылы қарастырылады. Мұндай бақылауды қабылдағаннан кейін, дамыған жүйе белгілі бір әрекеттерді орындауы керек, содан кейін бірдей күйде қалады немесе жүйеде кейбір өзгерістерді белгілеп, басқа мемлекетке ауысады. Бұл диаграмманың басты мақсаты оның өмірлік циклі кезінде үлгінің элементінің мінез-құлқын сипаттайтын күйлердің және өтулердің ықтимал тізбегін сипаттау болып табылады. Мемлекеттің өтпелі диаграммасы белгілі бір нақты оқиғаларды қабылдау реакциясының ерекшеліктерінің негізінде субъектілердің динамикалық мінез-құлқын білдіреді. Басқа жүйелер немесе пайдаланушылардың сыртқы әсерлеріне жауап беретін жүйелер кейде реактивті деп аталады. Егер осындай әсерлер ерікті кездейсоқ сәттерде басталса, онда модельдің асинхронды мінез-құлқымен танысыңыз. Диаграммалар көбінесе сыныптардың (нысандардың) жекелеген даналарын мінездеумен сипатталады, бірақ олар кей жағдайларда, актерлерде, кіші жүйелерде, операциялар мен әдістерде пайдаланылатын басқа үлгі компоненттерінің функционалдығын анықтау үшін пайдаланылуы мүмкін.

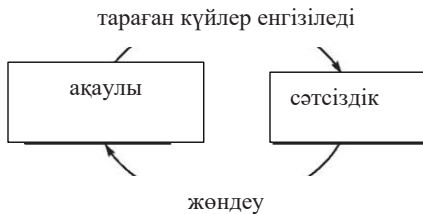
Мемлекеттің өтпелі диаграммасын құру үшін түпкілікті автомат теориясына сәйкес негізгі жағдайларды, басқару әрекеттерін (немесе өту шарттарын), орындалатын іс-әрекеттерді және әзірленетін бағдарламалық жасақтаманың өтуін анықтау қажет. Мемлекеттік өтпелі диаграммаларды құру кезінде қолданылған белгілер сурет. 2.3.

Мемлекеттің өтпелі диаграммасы, негізінен, автоматиканы білдіретін арнайы түрдің кестесі болып табылады.



Сурет. 2.3. Мемлекеттік өтпелі диаграммалардың аңыздары: *a* - бастапқы және соңғы күйлер; *b* - аралық күйі; *в* - көшу

UML контексінде автоматтың тұжырымдамасы автоматика теориясына негізделген нақты семантикаға ие. Осы кестенің шыңдары жай графикалық таңбалармен ұсынылған жай-күйі және автомат элементтерінің басқа түрлері (жалған күйлер) болып табылады. Графиктің шоғырлары мемлекетнен мемлекетке өтуді белгілеу үшін қолданылады. Автоматты түрде формализмге негізделген күйлердің визуалды көріністері мен өтулердің қарапайым мысалы компьютер сияқты техникалық құралдың жағдайына байланысты болуы мүмкін. Бұл жағдайда екі ең кең



Сурет. 2.4. Компьютер сияқты техникалық құрылғыға өтудің қарапайым диаграммасының ең қарапайым мысалы

тараған күйлер енгізіледі: «мінсіз» және «ақаулы» және екі ауысу: «сәтсіздік» және «жөндеу». Графикалық түрде бұл ақпарат компьютерлік диаграмма түрінде ұсынылуы мүмкін (2.4 сурет).

Автоматтың формализміне кіретін негізгі ұғымдар мемлекет пен өтпелі кезең болып табылады. Олардың арасындағы негізгі айырмашылық мынада, бұл жүйені табу уақыты жеке мемлекетте бір мемлекеттен екіншісіне өту уақытынан әлдеқайда көп. Шектеулерде бір мемлекеттен екіншісіне ауысу уақыты нөлге тең болады (егер ештеңе айтылмаса). Басқаша айтқанда, объектіні мемлекеттен мемлекетке көшіру дереу орын алады.

Басқаша айтқанда, объектіні мемлекеттен мемлекетке көшіру дереу орын алады.

Белгілі бір мемлекеттік схеманың семантикасын түсіну үшін модельдеудің мінез-құлқын ғана емес, сонымен қатар

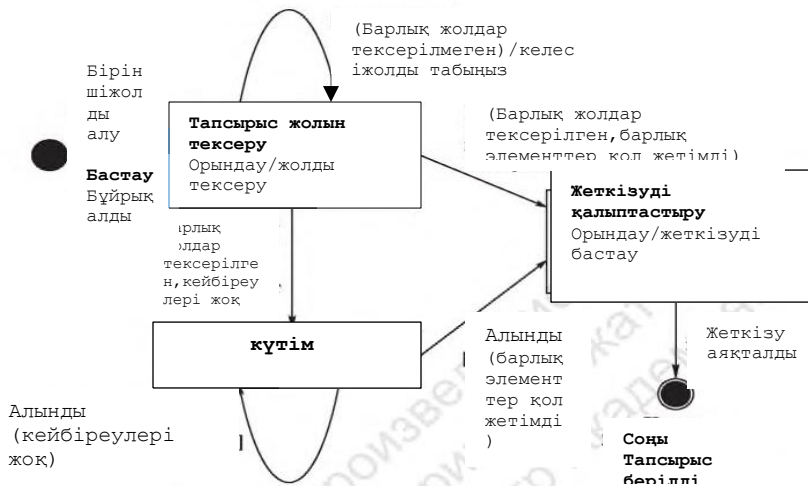
автоматтандыру теориясы туралы жалпы ақпаратты білу керек.

Күту күйі, егер бар болса, триггер оқиғадан кейін әрдайым тік жақшаларда жазылады және кейбір булевалық өрнек ұсынады. Еске салайық, бұл логикалық өрнек екі шын мәнінде айрықша құндылықтардың бірі болуы керек: «шындық» немесе «өтірік». Мемлекеттердің диаграммалары тұрғысынан бұл өрнек семантикасы нақты түрде орындалуы керек. Егер watchdog күйі «true» мәнін алса, онда тиісті өту жұмыс істей алады, нәтижесінде нысан нысаналы күйге өтеді. Егер watchdog күйі «жалған» мәнін алса, өтпелі жұмыс істей алмайды және басқа ауысулар болмаған жағдайда нысан осы өту үшін мақсатты күйге шыға алмайды. Алайда, бағыныстылық жағдайының шынайылығын есептеу онымен байланысты оқиға болғаннан кейін ғана орын алады.

Жетілдірілген пайдаланушы интерфейсі бар интерактивті бағдарламалық жасақтама үшін негізгі басқару әрекеттері нақты уақыт режимінде бағдарламалық қамтамасыз ету үшін - сенсорлардан және (немесе) өндіріс операторынан сигналдар үшін пайдаланушы пәрмендері болып табылады. Бағдарламалық қамтамасыз етудің бұл түрлеріне ортақ жүйе келесі басқару әрекетін алғанша жүйе жұмысын тоқтатқанда күту күйінің болуы болып табылады. Интерактивті бағдарламалық қамтамасыз ету үшін әр түрлі типтегі командаларды қабылдау және нақты уақыттағы бағдарламалық қамтамасыз ету үшін ең тән - сигналдардың бірдей түрін немесе көптеген сенсорлардан немесе ұзақ өңдеуді талап етеді.

Нақты уақыттағы жүйелер үшін интерактивті жүйелерден айырмашылығы, қабылданған сигналдың өңдеу уақытына қатаң шектеулер бар. Мұндай шектеу жүйенің мінез-құлқын уақыт бойынша қосымша зерттеуді талап етеді.

Бағдарламалық қамтамасыз ету, оның дамуы кезінде мінез-құлық сипаттамаларын мемлекеттік күйлердің диаграммасын жасау арқылы талап етеді, желіде жұмыс істеуге бағдарланған бағдарламалық қамтамасыз етуді қамтиды. Бұл жағдайда сервердің және клиенттің мінез-құлқының үлгілері әдетте бөлек салынып, олардың арасындағы басқарушы әрекеттер түрінде жіберілген хабарларды білдіреді.



Сурет. 2.5. «Тапсырыс» объектісінің мемлекеттік өту диаграммасы

Мемлекеттік диаграммалары құсбелгісі ұсынылған диаграмма (сур. 2.5) бірге оның болуы (түбіртек, өңдеу, қалыптастыру жеткізуші) кезінде объект «Тапсырыс» автоматтандырылған ақпараттық жүйесі «көтерме саудасының қоймасы» әрекетін сипаттайтын ықтимал мемлекеттер мен өткелдер, сипаттайды. Ол «Тапсырыс» нысаны арқылы орындалатын функцияларды белгілі бір күйде көрсетеді. детальдарда келесі синтаксис бар: Run / <Қызметі> (мысалы, / тексеру желісін орындау). Өткелдер синтаксистік үш қосымша <Іс-шара> <[шарттары]> </ Іс-шаралар> тұрады (мысалы, [емес барлық жолдар өтті] / келесі жолды алуға) болып табылады.

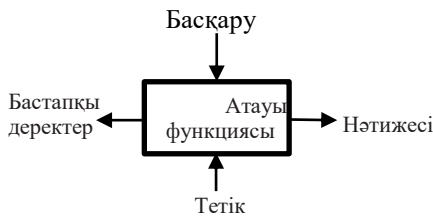
2.3 ФУНКЦИОНАЛДЫҚ ДИАГРАММА

Функционалдық диаграмма сипаттамалары (SADT) - SADT диаграммалары дамыған функциялардың өзара байланысын көрсетеді.

Олар жобалау арналған бағдарламалық қамтамасыз ету жүйесінің негізгі ерекшеліктері мен бөліктерді анықтау мен, егер мүмкін болса, анықтау және айтарлықтай қателерді алып тастауға көмектесу үшін ерте кезеңде жобалау жүйелері құрылады. Функционалдық диагностика-грамм жасау үшін SADT әдістемесі берілген D. Россқа қолдану ұсынылады. IDEFO (ICAM Definition)

АҚШ әуе күштері бастамасымен бағдарламасы ICAM (өндіріс компьютерлендіру) негізгі бөлігі болып табылады, ол салынды күрделі жүйелерді сипаттау әдістемесі SADT әйгілі әдіснамасы негізінде. әдістемесі SADT р құрастыру моделін қолдану диаграммалары Сынықтар мәтіні глоссарий тұратын және бір-біріне аттарын бар. Диаграммалар модельдің негізгі компоненттері болып табылады. Жүйенің функциялары мен интерфейстері блоктар мен доғалар түрінде диаграммаларда ұсынылған. Доғаны блокпен байланыстыру орны интерфейстің түрін анықтайды. Ақпараттық құрылғының сол жағында көрсетілген, және шығыс нәтижелерін оң жағында келтірілген өңделеді, ал бақылау ақпарат жоғарыдан блогында кіреді. Механизмі (адам немесе автоматтандырылған жүйесі) доғаның ұсынылған операция жүргізу, төменгі блогында (сур. 2.6) бөлігі.

SADT әдіснамасының ең маңызды ерекшеліктерінің бірі - модельді білдіретін диаграммалардың біртіндеп кеңейтілген бөлшектерін біртіндеп енгізу. SADT моделінің құрылысы бүкіл жүйенің қарапайым компоненті түрінде - бір блок және жүйеден тыс функциялары бар интерфейстерді білдіретін доғалар түрінде ұсынылуынан басталады. Жалғыз блок толығымен бүкіл жүйені білдіретіндіктен, блокта көрсетілген атау жалпы болып табылады. Бұл сондай-ақ интерфейстік арка үшін де жарайды - олар тұтастай алғанда жүйенің сыртқы интерфейстерінің жиынтығын білдіреді. Содан кейін жүйе бір модуль ретінде берілген блок бірнеше диаграммада көрсетілген

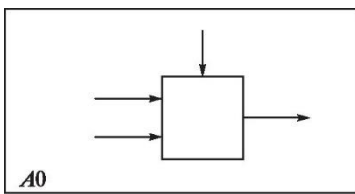


2.6 -сурет. Функционалдық блок және доғаның интерфейстері

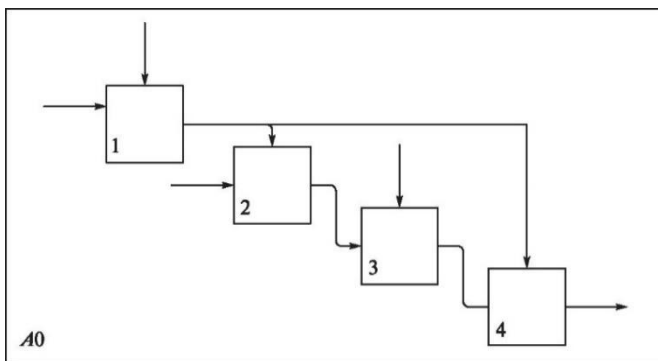
SADT моделі ілеспе құжаттары бар диаграммалардың сериясы болып табылады, ол күрделі объектіні оның құрамдас бөліктеріне блоктар түрінде бөледі. Негізгі блоктардың әрқайсысы басқа диаграммалардағы блоктар ретінде көрсетіледі. Әрбір егжей-тегжейлі диаграмма - жалпы диаграммадан блоктың ыдырауы. Декомпозицияның әрбір сатысында егжей-тегжейлі диаграмма үшін негізгі диаграмма деп аталады.

Жоғарғы деңгейдегі диаграммадағы блокқа кіретін және шығатын доғалар төменгі деңгей диаграммасына кіретін және одан шығатын доғалар сияқты бірдей, себебі блок пен диаграмма жүйенің бір бөлігін білдіреді.

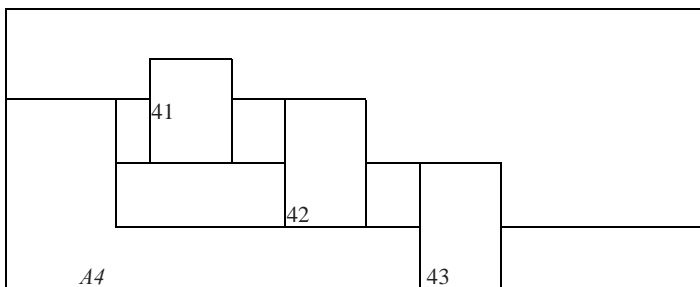
көрсеткілер ата-аналар графиктің түскен немесе оны кетіп қалған , саны, таңбаларды және сандарды пайдаланып. Symbol obo-зпашает қосылым түрі: Мен - Input C - бақылау, M – механизмі , R - нәтижелер. Саны - ата-аналар бөлігіндегі байланыс жағында саны, төменгі және солдан оңға жоғарыдан санағанда. Барлық диаграммалар бір-бірімен иерархиялық нөмірлеу блоктар қосылу: негізгі деңгейін - AҚ, келесі - A1, A2, т.б., келесі - A11, A12, A13, т.б., бірінші нөмірі - ата-аналар бөлігіндегі саны, және соңғы - ата-аналық блоктың нақты субблоктың нөмірі. Алынған функцияларын толық егжей-тегжейлі, сондай-ақ қалай тапсырыс беруші және әзірлеуші түсінген оның мақсаты. Бұл функциялар табиғи тіл немесе псевдокодтар арқылы сипатталады. құрылымын мен сызбаларда көрсетілген деректер элементтерін анықтау, онда ақпарат пен stro-yat деректер сөздігін көрсете бүкіл тіркелген йерархии диаграммаларды құру процесінде. Осылайша, нәтиже - функциялардың иерархиясынан тұратын ерекшелігі.



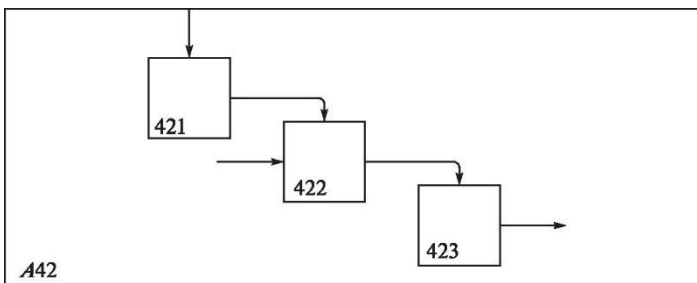
a



6



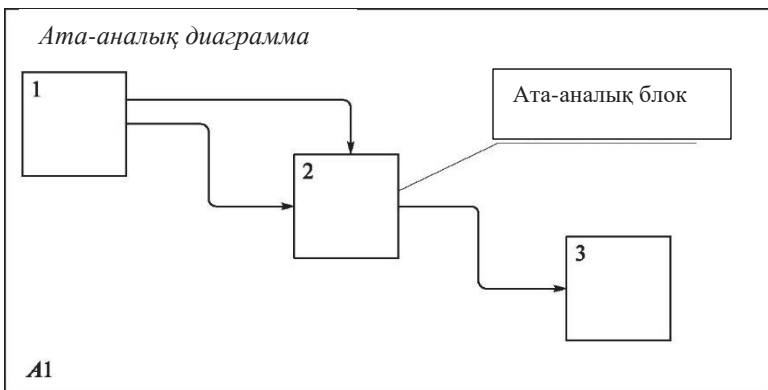
б



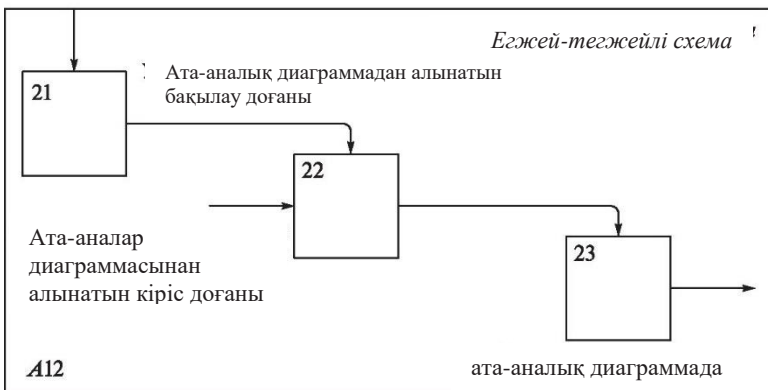
2

2.7 Сурет диаграммалар, төменгі деңгей функцияларының сипаттамалары және бір-біріне сілтеме жасайтын сөздік

Суретте. 2.7 шоу SADT-моделі дәлелдейді бойынша құрылымы ыдырау төрт диаграммалар, жалпы (сур. 2.7, а) және detail-ное өкілдігі А.О. блок (сур. 2.7, б) ыдырау А4 блоктар (сур. 2.7, б) және А42 (Сурет 2.7, d). ARC кірістерін сәйкестеніп, бақылау және шығу ата-аналар бөлігін жалғамаңыз. көзі немесе осы доғаның алушы тек ата аналар диаграммада табылған болуы мүмкін. шеттері бастапқы диаграмманың доғалар сәйкес келуі тиіс жалғамаңыз. Барлық шекара доға ол толық және дәйекті болып табылады, ата-аналық диаграммасын жалғастыру

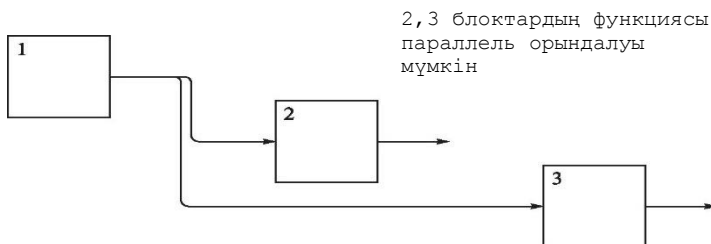


a



б

Сурет. 2.8. Ата-анасының (а)және ата-ана (б) диаграммаларын сәйкестендіру

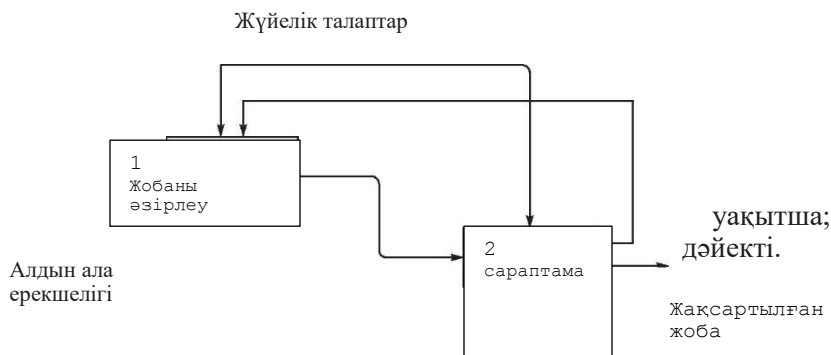


Сурет. 2.9 Ата-анасының (а) ата-ана мен егжей-тегжейлі (б) диаграммаларын сәйкестендіру.

SADT диаграммалары бойынша кезектілік немесе уақыт да анық көрсетілмейді. Жазбалар, итерациялар, жалғасып жатқан

Сурет. 2.9. Функцияларды бір уақытта орындау мысалы. процестер және бір-бірін жабатын (уақытында) функциялары доғаларды пайдалану арқылы да ұсынылуы мүмкін. Кері байланыс түсініктемелер, түсініктемелер, түзетулер және т.б. түрінде болуы мүмкін. (2.10-сурет).

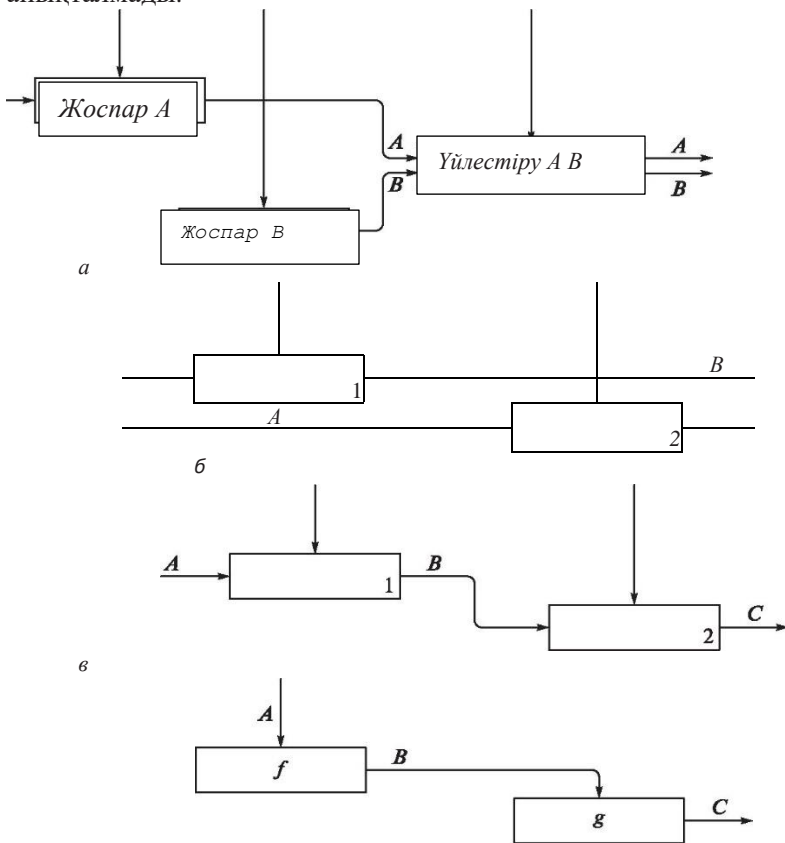
SADT әдіснамасын пайдалана отырып, жүйені модельдеудегі маңызды мәселелердің бірі функция арасындағы қатынастардың түрлерінің дәл сәйкестік болып табылады. Кем дегенде, сілтемелердің келесі түрлері бар: кездейсоқ; логикалық;



Сурет 2.10

Кездейсоқ байланыс функция арасындағы нақты байланыс шағын немесе толық болмаған кезде орын алады. Бұл бір схемада SADT дөңгелектеріндегі деректер атауларының бір-бірімен кішігірім байланысы болған жағдайға нұсқайды

Логикалық байланыс - деректер мен функциялар біріктіріледі, өйткені олар жалпы сыныпқа немесе элементтер жиынына түседі, бірақ олардың арасындағы қажетті функционалдық қатынастар ол анықталмады.



2
Сурет. 2.11 Сілтемелер түрлері:

a - іс жүргізу; б - байланыс; в - дәйекті; г - функционалды

Уақытша сілтеме деректер бір мезгілде қолданылған кезде, немесе функциялар бірізді түрде емес, параллельде ауыстырылған уақытпен байланысты функцияларды білдіреді.

Жоспар, үйлестіру, жоспар

Процедуралық қосылымда функциялар біріктіріледі, өйткені

олар циклдің немесе процесінің сол бөлігінде орындалады (2.11-сурет, а).

Диаграммалар блоктарды бірдей енгізу деректерін пайдаланады және / немесе бірдей шығыс деректерін шығарғандықтан байланыстыру байланысын көрсетеді (2.11-сурет, б).

Сериялық байланыс арқылы бір функцияның шығу функциясы басқа функцияға кіріс ретінде қызмет етеді. Диаграммадағы элементтер арасындағы байланыс жоғарыда талқыланған деңгейлерден гөрі жақсырақ, себебі себеп-салдар байланыстары модельденеді (2.11-сурет, с).

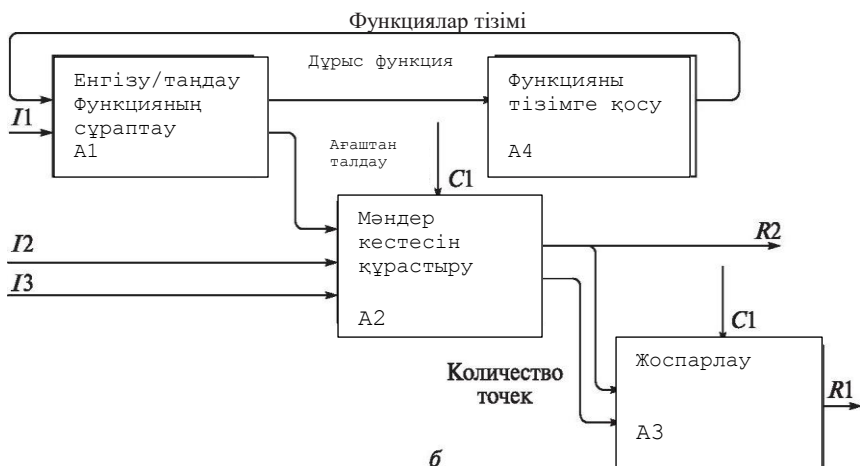
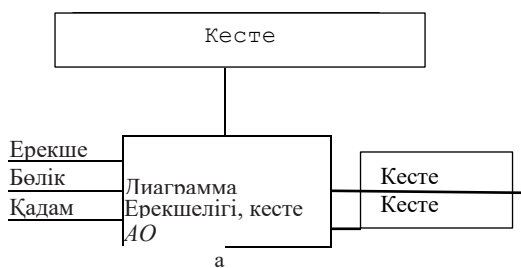
Диаграмма бір функцияның екіншісіне толықтай тәуелді болған кезде толық функционалды байланысын көрсетеді. Таза функциялы схема байланыстың дәйекті немесе әлсіз түріне байланысты шетелдік элементтерді қамтымайды. Функционалдық байланысты диаграммаларды анықтаудың бір жолы - суретте көрсетілгендей басқару доғасының көмегімен қосылған екі блокты қарау. 2.11 сурет.

Математикалық тұрғыда қарапайым функционалдық байланыс түріне қажетті шарт (Figure 2.11, d) мынадай формаға ие: $C = g(B) = g(f(A))$.

SADT әдісі көптеген жүйелерді модельдеу және талаптарды және функцияларды анықтау үшін пайдаланылуы мүмкін. Қолданыстағы жүйелерде SADT әдісі жүйенің атқаратын қызметтерін талдау үшін және оларды іске асыру механизмдерін анықтау үшін пайдаланылуы мүмкін.

Диаграмма бағдарламасының функционалдық диаграммасының даму мысалы. Функционалды диаграммаларды әзірлеу бір айнымалы функцияның кестелерін және кестелерін жасау үшін бағдарламаның спецификациясын нақтылайтын мысалмен көрсетіледі.

2.12 – суретте жоғары деңгейлі диаграмма көрсетіледі, бұл бағдарлама үшін бастапқы деректердің не екенін және қандай нәтижелер күтетінін анық көрсетеді.



Сурет. 2.12. Функционалдық зерттеу жүйесі үшін функционалдық диаграммалар:
a - жоғарғы деңгейлі диаграмма; b - тазарту диаграммасы

Суреттегі диаграмма. 2.12, b бағдарламаның функцияларын анықтайды. Ол төрт блокты көрсетеді: енгізуді таңдау және талдау, тізімге функция қосу, мән кестесін құру және функцияны жасау. Әр блок үшін бастапқы деректер, басқару әрекеттері және нәтижелер анықталады. Диаграммада жалғасы бар кіріс және шығыс белгілерін белгілеу ережелеріне сәйкес келесі диаграммада келесі белгілер қолданылады: I1 - функция; I2 - бірлік; I3 - қадам; C1 - графикалық кесте көрінісі; R1 - сегмент бойынша функцияның кестесі; R2 - сегменттегі функция мәндерінің кестесі.

Бұл жағдайда сөздік жүйеде қолданылған барлық деректердің сипаттамасын қамтуы керек.

Күрделі деректер құрылымдарымен жұмыс істеуді талап етпейтін бағдарламалық жасақтаманың ерекшеліктерін анықтау үшін функционалдық модельді пайдалану ұсынылады, себебі ол функциялардың ыдырауына бағдарланған.

2.4. АҚПАРАТТЫҚ АҒЫН ЖҮЙЕЛЕРІ

Деректер ағынының диаграммасының сипаттамасы (DFD). DFD диаграммасы деректерді өңдеу тораптарынан, сақтау құралдарынан және сыртқы көздерден немесе қолданылған диаграммаға кіретін деректерден тұрады.

Деректер ағынының диаграммасының сипаттамасы (DFD). DFD диаграммасы деректерді өңдеу тораптарынан, сақтау құралдарынан және сыртқы көздерден немесе қолданылған диаграммаға кіретін деректерден тұрады.

Ақпарат көздері (сыртқы ұйымдар) ақпаратты кіші жүйелерге немесе процестерге жіберетін ақпараттық ағымдарды (деректер ағындарын) қалыптастырады; өз кезегінде, ақпаратты басқа үдерістерге немесе кіші жүйелерге, деректерді сақтауға арналған құрылғыларға немесе сыртқы тұтынушыларға - ақпарат тұтынушыларға беретін ақпараттарды түрлендіреді және жаңа ағындарды қалыптастырады.

Деректер ағындарының диаграммаларын көрсету үшін кестеде ұсынылған Йордания және Хайнин-Сарсон дәстүрлі белгілердің екі түрі қолданылады. 2.1.

2.1.кесте.Нотация түрлері		
Ұғым	Йорданның белгіленуі	Гейн-Сарсонның белгіленуі
Сыртқы мәні	Аты	Аты Нөмері
Процесс, жүйе, шағын жүйе	Аты Нөмері	Нөмері Аты Машиналар
Деректер жинаушы	Аты	№ Аты
Деректер ағымы	Аты 	Аты 

Деректерді сақтау, Деректер ағымы, Иорданияның атауы, аты, атауы, нөмірі, аты, Хейн Сарзонның атауы, Санның аты Атауы Механизмі,

Сыртқы ұйым - ақпараттың көзі немесе алушысы болып табылатын материалдық объект немесе жеке тұлға, мысалы, клиенттер, қызметкерлер, жеткізушілер, клиенттер, қойма. Сыртқы объект ретінде кейбір объектілерді немесе жүйелерді анықтау осы объект немесе осы жүйе талданған ақпараттық жүйенің шекарасынан тыс екенін көрсетеді. Талдау үдерісінде кейбір сыртқы эфирлер, қажет болған жағдайда, талданатын ақпараттық жүйенің диаграммасына ауыстырылуы мүмкін, немесе керісінше, ақпараттық жүйенің процестерінің бір бөлігі диаграммасынан тыс және сыртқы субъект ретінде ұсынылуы мүмкін.

Күрделі ақпараттық жүйенің моделін құрастырған кезде, ол жалпы контекстік диаграммада ұсынылуы мүмкін. Бұл диаграмма дамыған жүйенің тұтынушылармен және ақпарат көздерімен өзара әрекеттесетіндігін көрсетеді, яғни сыртқы әлеммен жүйенің интерфейсін сипаттайды. Жүйе бір бүтін ретінде ұсынылуы мүмкін немесе бірнеше кіші жүйеге ыдырауы мүмкін. Шағын жүйенің нөмірі оны анықтау үшін қызмет етеді. Атауы саласында кіші жүйе атауы қосалқы өтініштермен және тиісті анықтамалармен және толықтырулармен ұсыныс түрінде енгізіледі.

Процесс кіріс деректерінің ағындарын белгілі бір алгоритмге сәйкес түрлендіру болып табылады. Физикалық тұрғыдан алғанда бұл процесс әр түрлі жолдармен жүзеге асырылуы мүмкін: ол кіріс құжаттарын өңдеу және есептерді, бағдарламаны, аппараттық қамтамасыз етілген логикалық құрылығыны және басқаларды жүзеге асыратын ұйымның (бөлімнің) бөлімшесі болуы мүмкін. Процесс нөмірі оны анықтау үшін қызмет етеді. Процестің аты атау өрісіне енгізіледі және өңдеуге, жаңартуға немесе редакциялауға арналған этиштерді пайдалану, әдетте, осы процесті терең түсіну және одан әрі талдауды талап етеді. Физикалық іске асыру саласы туралы ақпарат осы процесті қай ұйымдық бөлімше, бағдарлама немесе аппараттық құрал жүзеге асыратынын көрсетеді.

Деректерді сақтау құрылығысы кез-келген уақытта дискке орналастыруға болатын ақпаратты сақтауға арналған дерексіз құрылғы болып табылады және біраз уақыттан кейін оны шығаруға болады; және орналастыру және іздеу әдістері кез келген болуы мүмкін. Деректер аккумуляторы физикалық түрде файл кабинасындағы қораптың түрінде, ЖЖҚ кестесінде, магниттік ортадағы файлда және т.б. болуы мүмкін.

Гейн-Сарсон нұсқада деректерді сақтау құрылығысы D әрпімен

және ерікті нөмірмен анықталады. Дискінің атауы дизайнер үшін барынша ақпараттылыққа байланысты таңдалады.

Деректер жинаушы, әдетте, болашақ дерекқордың прототипі болып табылады және онда сақталған деректердің сипаттамасы ақпараттық модельге байланысты болуы керек.

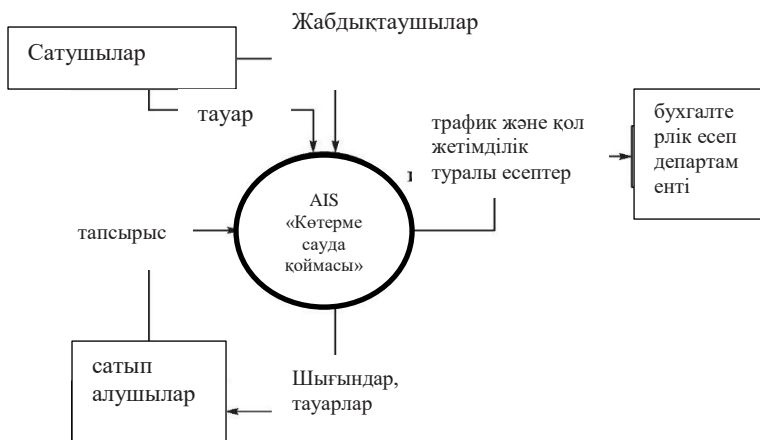
Деректер ағымы көзден қабылдағышқа кейбір қосылым арқылы жіберілген ақпаратты анықтайды. Нақты деректер ағыны хаттармен, магнит таспаларымен немесе бір компьютерден екіншісіне тасымалданған дискеттерге пошта арқылы жіберілген екі құрылғы арасында кабель арқылы берілетін ақпарат болуы мүмкін. Әрбір деректер ағынында оның мазмұнын көрсететін атау бар.

Деректер ағынының диаграммасының иерархиясының құрылысы жүйенің жалпы көрінісін анықтайтын контекстік схемадан басталады. Әдетте бастапқы контекстік диаграммада жұлдыздың түрі бар. Егер жобаланған жүйе көптеген сыртқы субъектілердің (10-нан астам) болса, бөлінген сипаты бар немесе бұрыннан бар ішкі жүйелерді қамтиды, содан кейін контекстік диаграммалардың иерархиясын құрастырады. Бөлшектеу үрдісінде теңдестіру ережесі байқалады, егер шағын жүйе егжей-тегжейлі көрсетілсе, дамыған шағын жүйе ақпараттық қосылым бар (яғни, ол деректер ағындары арқылы қосылған) ішкі жүйенің құрамдас бөліктерін ғана қолдануға болады. Егжей-тегжейлі процестерде осы процестің функцияларын сипаттайтын сипаттамалар бар. Мұндай сипаттама құрылымдық табиғи тілде (жалған кодтар), кестелер мен шешім ағаштарын қолдана отырып, табиғи тілде орындалуы мүмкін, алгоритм схемалары түрінде.

AIS деректер ағынын диаграммасын құрудың мысалы «Көтерме қойма».

Деректер ағындарының диаграммасының иерархиясын құрылымдық сұлбадан бастаймыз, ол жүйенің жалпы көрінісін анықтайды. Осылайша, дамыған жүйенің қабылдағыштар мен ақпарат көздерімен өзара әрекеттесетіндігін анықтаймыз (2.13-сурет).

«Көтерме сауданың қоймасы» автоматтандырылған ақпараттық жүйесіжүк көтеру үшін сатып алынатын тауарлардың қозғалысы мен қолжетімділігі туралы деректер алуға арналған. Тауарлардың келуіне арналған бастапқы құжаттар тауарларды қабылдау журналында тіркеледі. Тауарды сатуды тіркеу және есепке алу сатып алушы мен сатушы арасында сатып алынған тауарларды есептеу әдісіне байланысты. Қойма басқарушысы тауарларды сатып алу және жеткізу тізілімін жүргізеді.



Сурет. 2.13. Иорданиядағы «Көтерме қойма» ААЖ-на арналған бастапқы контекстік диаграмма (нөлдік деңгей диаграммасы)

Жүйенің сыртқы құрылымы жеткізушілер, сатып алушылар, қойма менеджері, бухгалтерлік есеп және бақылау бөлімі, тапсырыстарды қабылдау және тіркеу бөлімі болып табылады. Олар туралы ақпарат тиісті кестелерде (анықтамалықтар) сақталады. Өнім қоймаға тауарларды тасымалдайды, тауарларды жеткізу үшін құжаттар (депозиттік шоттар, шот-фактуралар) деректер базасына енгізіледі. Сатып алушы тауарды сатып алу туралы бұйрықтарды қабылдау және тіркеу бөліміне тапсырыс беріп, тапсырманың әрбір жолын тексереді. Қоймада қандай да бір позиция болмаған жағдайда жеткізушіге талап қойылады, яғни. қажетті тауарларды жеткізу басталды. Өтінім негізінде құжаттың орындалуына арналған құжаттар толтырылады, ол дерекқорда сақталады, басып шығарылады және сатып алушыға жеткізіледі. Әрбір күннің соңында барлық негізгі құжаттар шот басқарушысына (бухгалтер) аударылады. Тауарлардың келуі және сатылуы туралы ақпарат негізінде бухгалтерия бөлімінің қоймалық менеджерлері мен қызметкерлері қоймадағы тауардың айналымы мен қалдықтары туралы есептер шығарады.

Деректер ағындарының диаграммасын құру үшін MS Visio бағдарламасын іске қосу қажет. Shapes / Business Process папкасын таңдау қажет терезе пайда болады. Деректер ағынын диаграммасын құруға арналған пішіндердің ашық тізімінде (Форматы), Деректер ағыны диаграммасының пішіндер элементін таңдаңыз. Жасалған әрекеттердің нәтижесінде экранда сол жақта графикалық символдар жиынтығы көрсетіледі, ал оң жағында - диаграмманы сызу үшін

парақ пайда болады. Қажетті құрамдасты тиісті белгішені басу арқылы таңдаңыз, содан кейін оң жақ батырманы басқан кезде графикалық таңба тышқанмен жұмыс өрісіне ауыстырылады. AIS деректер ағындарының диаграммасы «Көтерме қойма» күріш көрсетілген. 2.14.

Студенттердің бөлінуінің шағын жүйелерінің деректер ағындарының сызбасының құрылысының мысалы. Оқу орнының жоғары курстарына қатысатын студенттер туралы ақпаратты жинауға және сақтауға арналған деректердің автоматтандырылған жүйелерінің ағын схемасын әзірлейтін боламыз. Жүйе бітірушілер туралы Интернет арқылы әлеуетті жұмыс берушілерге ақпарат беру мақсатында оқу орындарында пайдалануға арналған. Гейн-Сарсон белгілеріндегі деректер ағындарының бастапқы контекстік диаграммасы күріш. 2.15. Ондағы сыртқы мәндер жұмыс беруші, әкімші және студент болып табылады.

Жұмыс беруші жүйеде тіркеледі, кандидатты іздеу үшін деректерді енгізеді және жүйеден іздеу нәтижелерін алады. Әкімші дерекқорда сақталған студенттер туралы ақпаратты енгізеді. Студент өзінің байланыс ақпаратын өзгерте алады, өзгертулер деректер базасында да сақталады. Таңдалған студенттерге жұмыс беруші жұмыс туралы ұсыныс жібереді. Студенттердің білім беру жүйесінің ағындарының диаграммасы күріш. 2.16.

Студенттің прогресс жазба жүйесінің деректер ағынының схемасын құруға мысал.

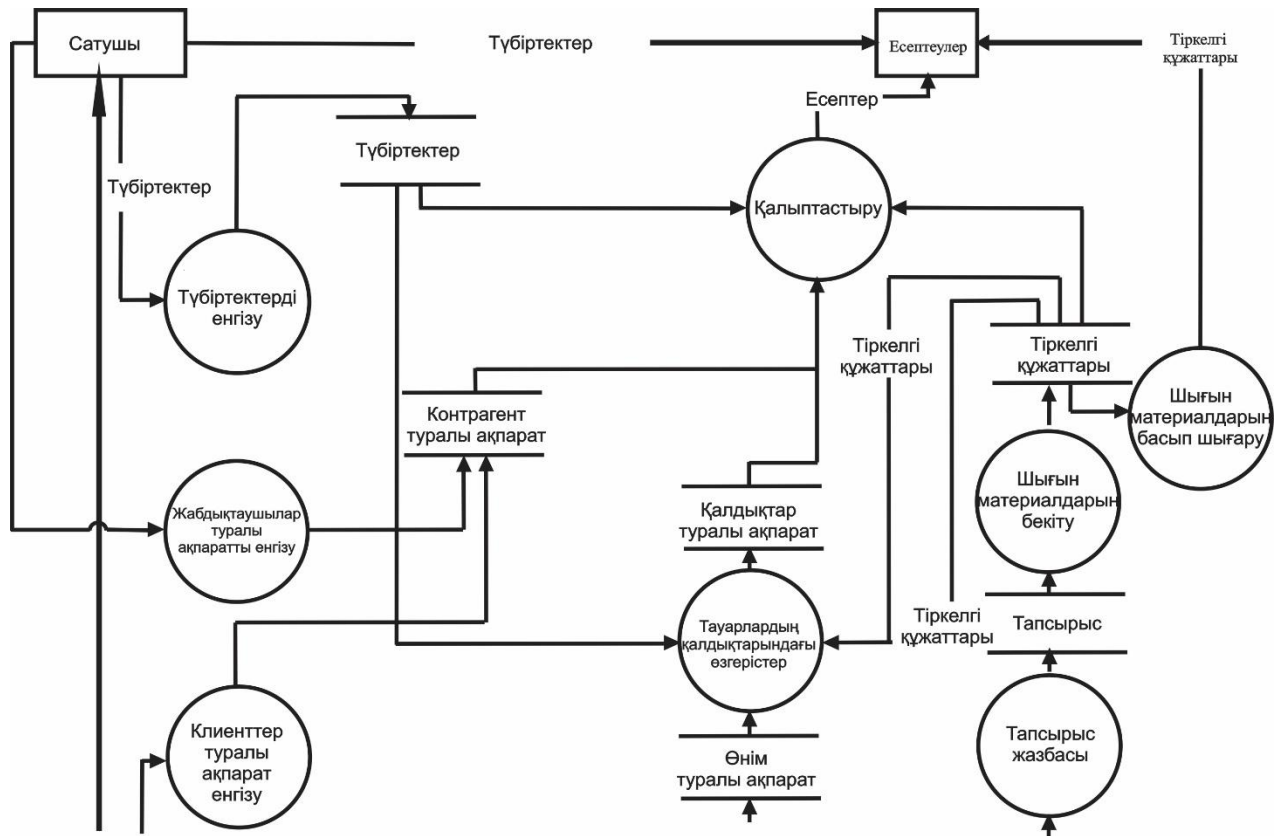
Студенттің ілгерілеу жүйесінің деректер ағындарының диаграммаларын әзірлейміз. Жүйедегі сыртқы субъектілер деканат, декан орынбасары және деканаттың қызметкері. Осы ұйымдар мен жүйе арасындағы деректер ағындарын анықтаңыз.

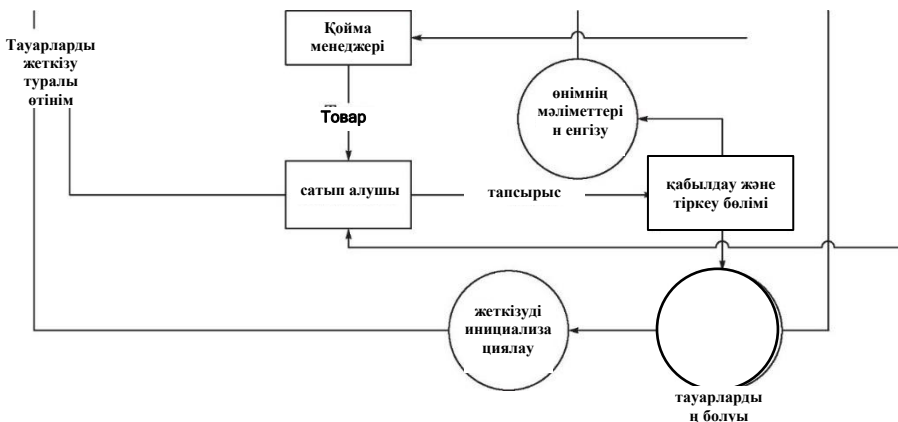
Декан алуға тиісті :

■ факультеттегі прогрестің жиынтығы (топтардың, курстардың және барлық факультеттің прогресі пайызы) ағымдағы немесе белгіленген уақытта;

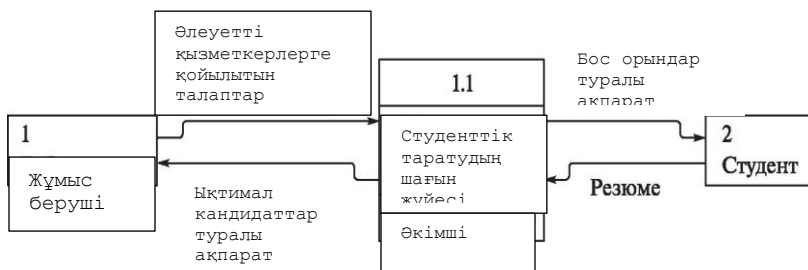
■ студенттің оқуы туралы толығымен ақпарат (барлық аяқталған семестрлердің барлық пәндеріндегі академиялық үлгерімі, қайта қабылдауды ескере отырып).

Деканның орынбасарына тариф бойынша прогресс пайызы;



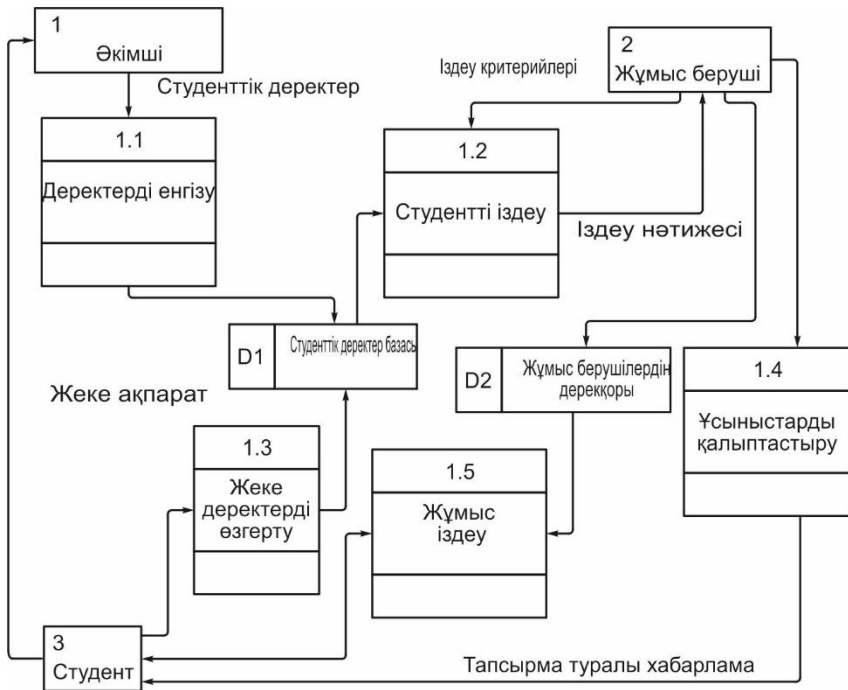


Сурет. 2.14. AIS деректер ағынын диаграммасы «Көтерме қойма»



Сурет 2.15

Тұтынушылар туралы мәліметтерді енгізу, Сатып алушылар туралы ақпаратты енгізу, Есептерді қалыптастыру, Шығындар туралы ақпарат, Тепе-теңдік туралы ақпарат, Тауардың балансын өзгерту, Тауар туралы ақпарат, Тұтынылатын құжаттар, Шығындар, Құжаттар, Шығын кітапшасын басып шығару Тапсырыс беру және Тапсырыс беру бөлімі, Тауарлардың қол жетімділігін анықтау, Жеткізуді инициализациялау



Сурет 2.16

■ студенттің оқуы туралы толық ақпарат (оқудың барлық аяқталған семестрлерінде оқудың барлық пәндерінде академиялық үлгерімі, есепке алуды ескере отырып);

■ берілмеген заттар көрсетіліп, факультетте және топта борышкерлердің тізімі.

Деканаттың қызметкері:

■ бірінші жылы оқитын студенттердің тізімін енгізу;

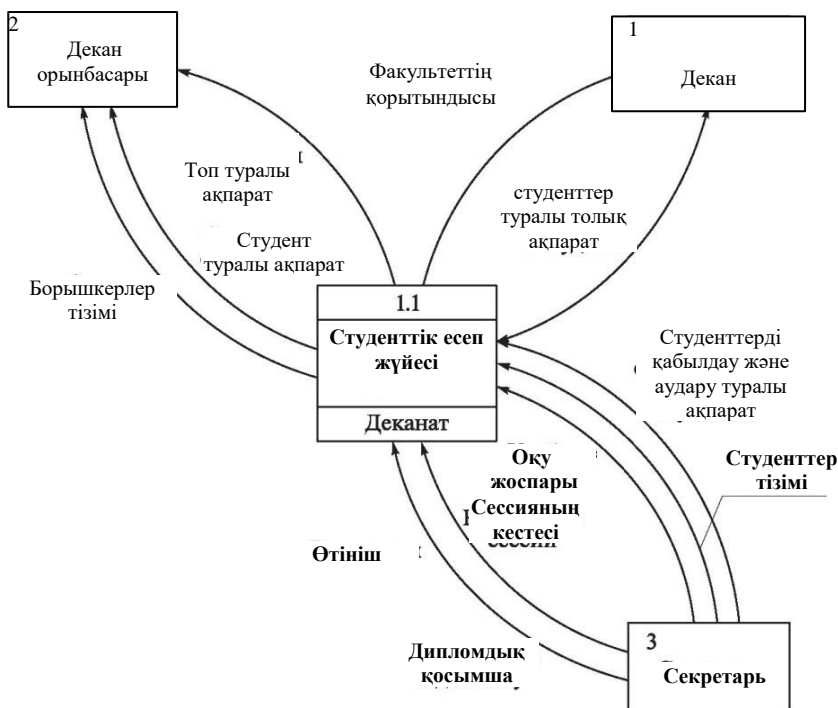
■ Оқуға қабылдау, беру, беру және т.б. тапсырыстарына сәйкес студенттік тізімдерді түзету;

■ департаменттердің оқу жоспарларын енгізу;

■ сессия кестесін енгізу;

■ өтінімдер мен бағыттар негізінде емтихандар мен емтихан тапсыру нәтижелерін енгізу.

Сурет. 2.17. Студенттің есеп беру жүйесінің контекстік диаграммасы



2.18- сурет. Екінші деңгейдегі деректер ағындарының толық диаграммасы

Бұдан басқа, деканаттың қызметкері:

- студенттік пәндер туралы сертификат, сағаттар мен қорытынды баға белгілерін көрсете отырып;
- сағаттар мен қорытынды бағалармен түлектің дипломына

Нәтижесі Гай-на-Сарсон (сур. 2.17) бір контекст диаграмма түсінігіндегі. Әрі қарай егжей-тегжейлі жүйесінде процестер (сур. 2.18). деректер ағыны диаграммада көрсетілгендей екі кіші егжей-тегжейлі: ішкі базалық жүйені толтыру, сондай-ақ құралдарды пайдалана отырып, деректер базасын, сондай-ақ олардың іске асырылуы мүмкін деректер сақтау, есеп. Әрі қарай егжей-тегжейлі процестер айқын, бірақ ол осы даму толық спецификациясы диаграмма «тұлға-қарым-қатынас» түріндегі деректер базасын сипаттамасы керек екені айдан анық.

2.5. МАҢЫЗДЫ БАЙЛАНЫС ДИАГРАММАСЫ

Диаграммалардың сипаттамасы - «мәні - қосылу». Бұл диаграмма (ER-деректер үлгісі) олардың арасындағы деректерді және қатынастарды анықтаудың стандартты әдісін ұсынады. Ол доменнің негізгі бизнес ережелерін көрсететін субъектілер мен қатынастарды қамтиды. Құрылымдық схемалар, функционалдық диаграммалардан айырмашылығы, бағдарламалық жасақтаманың деректер құрылымдарының ерекшеліктерін анықтайды. «Субъект-қатынас» моделінің алғашқы нұсқасын П. Чен ұсынды. Кейінірек, көптеген авторлар ұқсас үлгілердің өздерінің нұсқаларын әзірледі. «Мағынасы-қосылу» диаграммасының барлық нұсқалары бір идеядан келеді - графикалық кескін мәтін сипаттамасынан гөрі айқынырақ.

Барлық осындай диаграммалар домен объектілерінің графикалық көрінісін, олардың қасиеттерін (атрибуттары) және өзара әрекеттестікті пайдаланады субъектілер арасындағы байланыстар. Ең жиі кездесетін Баркердің белгісі, және біз оған сүйенеміз. ER деректер үлгісінің негізгі ұғымдары (ER - Entity-Relationship) - субъект, атрибуция және қарым-қатынас.

Кәсіпорын - бұл тақырыптық домен үшін маңызды болып табылатын ақпарат туралы шынайы немесе дерексіз нысандардың (адамдар, оқиғалар, штаттар, нысандар және т.б.) бір түрі. Деректер құрылымы - жеке деректер бөліктері (элементтері) арасында

болатын қатынастарды көрсететін ережелер мен шектеулер жиынтығы.

Әрбір ұйымда:

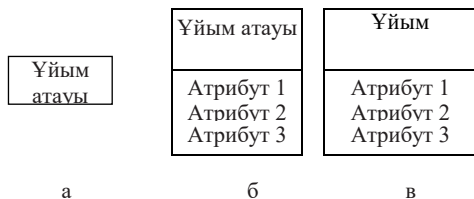
- бірегей атауы;
- Субъектіге тиесілі немесе байланыс арқылы мұраланған бір немесе бірнеше атрибуттар;
- Ұйымның әрбір данасын бірегей түрде анықтайтын бір немесе бірнеше атрибуттар.

Ұйымның данасы - бұл ұйымның нақты өкілі. Ұйым атауы объектінің түрін немесе сыныпын көрсетуге тиіс, оның нақты данасын емес (студент, Иванов емес).

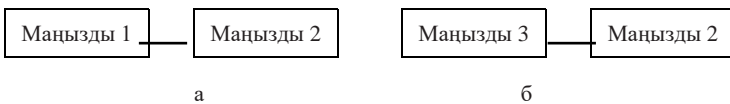
Баркердің белгілеріндегі диаграммада субъект тіктөртбұрыш, кейде дөңгелек бұрыштармен ұсынылған (2.19-сурет, а). Әрбір ұйымның бір немесе бірнеше төлсипаттары бар.

Атрибут — любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности (рис. 2.19, б). Атрибут, таким образом, представляет собой некоторый тип характеристик или свойств, ассоциированных с множеством реальных или абстрактных объектов.

Атрибут данасы - белгілі бір даналықтың ерекше сипаттамасы.



Сурет. 2.19. Баркердің белгілерінде субъектіні сәйкестендіру:
a - атрибутсыз; *b* - атрибуттарды көрсету; *c* - атрибуттарды және олардың сипаттамаларын көрсете отырып түрлері (# - кілт, * - міндетті, о - міндетті емес)



Сурет. 2.20. Байланыс тәсілдері: а - міндетті; b - факультативтік (сызықтың ортасына сызылған)

Атрибуттар кілтке бөлінеді, яғни. кілттің бірегей идентификаторының бөлігі және сипаттама - басқа

Бастапқы кілт - атрибуттар немесе атрибуттар жиынтығы және (немесе) байланыстар, әрбір жеке дананы (объектіні сәйкестендіруге мүмкіндік беретін сипаттамалар жинағын) бірегейлендіруге арналған. Негізгі атрибуттар тізімнің жоғарғы жағында орналасады және «#» белгісімен белгіленеді (2.19-сурет, с).

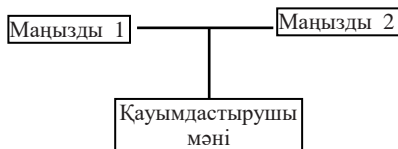
Сипаттамалық атрибуттар міндетті немесе міндетті емес болуы

Байланыс - бұл бір субъектінің қарым-қатынасы. Әр сілтеме сілтемелердің екі әдістерінің біріне ие болуы мүмкін. Бір объектінің кез-келген данасы басқа субъектінің кем дегенде бір данасымен байланысты болса, онда байланыс міндетті болып табылады (2.20-сурет, а). Қосымша қатынастар - субъектілер арасындағы шартты қатынас (сурет 2.20, b).

Байланыстың түрлі мақсаттардан өзгеше болуы мүмкін. Әрбір сілтеме солдан оңға қарай, оңнан солға қарай оқылады. Әрбір ұйымда модельдің басқа субъектілеріне кез-келген байланыстар болуы мүмкін. Байланыстың үш түрі бар (2.21-сурет): «бір-бір»; «Бір-көпшілікке»; «Көптеген-көпшілік».



Сурет. 2.21. Баркердің белгілеріндегі қарым-қатынастардың белгілері:
a - «жеке-жеке»; b - «бірден-көпге»; c - «көптеген-көпшілікке»



Сурет. 2.22. Баркердің белгілеріне байланысты субъектінің тағайындалуы

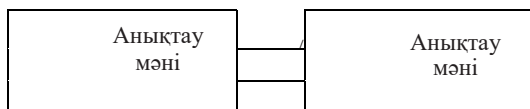
Бір-бірімен қарым-қатынас бірінші инстанцияның бір данасы екінші субъектінің бір данасымен байланыстырылғанын білдіреді. Мұндай байланыс, ең алдымен, бір ұйымның екіге дұрыс емес екенін көрсетеді (кейде деректердің бір бөлігін «жіктеу» қажет болған жағдайда осы байланыс түрі қолданылады). Бір-көп қатынастармен, бірінші субъектінің әрбір данасы екінші субъектінің бірнеше данасымен байланысты. Көптеген-көпшілік қарым-қатынастар бірінші инстанцияның әрбір данасы екінші субъектінің бірнеше даналарына және керісінше байланыстыруға болатындығын көрсетеді. Қосылымның бұл түрі уақытша болып табылады. Бұл модельді дамытудың ерте сатысында қолайлы. Болашақта, мұндай байланыс аралық ұйымды құру арқылы екі «бірден көп» сілтемелермен ауыстырылуы керек.

Тәуелсіз дербес жүйе жүйеде әрқашан болатын тәуелсіз деректерді ұсынады. Олар басқа субъектілермен байланысты болуы мүмкін немесе жоқ.

Тәуелді субъект жүйеде басқа субъектілерге байланысты деректерді ұсынады, сондықтан ол әрқашан басқа ұйымдармен байланыстырылуы керек.

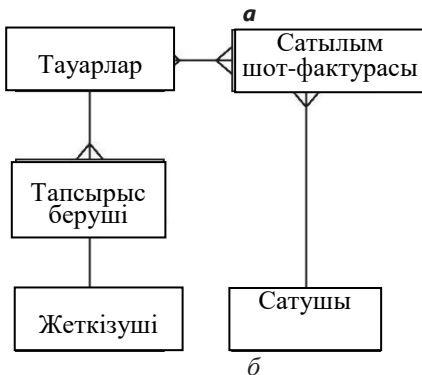
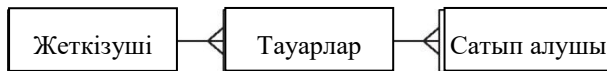
Байланысты ұйым екі немесе одан да көп субъектілер арасындағы байланысқа байланысты деректерді ұсынады. Әдетте бұл «көптегенден көпке» қатынастарды шешу моделінде пайда болады (2.2.2-сурет)

Егер ұйым данасы оның негізгі сипаттамалары арқылы анықталса, онда ұйым толықтай анықталған деп айтылады. Олай болмаған жағдайда, ұйым байланыс желісіндегі сызықпен көрсетілетін байланысты субъектінің атрибуттарын пайдалана



Сурет. 2.23. Баркердің белгілеуінде басқа біреудің сәйкестендіруін анықтау

отырып анықталады (2.23-кесте).



Сурет. 2.24. ER-диаграммасының нұсқалары:
a - бірінші; *b* - аралық; *c* - финал

«Көтерме саудалық қойма» ААЖ «мәні - байланыс» диаграммасының даму үлгісі. Көрсетілген тапсырманы шешудің негізгі мәні болып табылады: өнім беруші, сатып алушы, тауар.

Кәсіпорындардың арасында «сатып алушылар көптеген тауарларды сатып ала алады», «сатып алушылар көптеген сатып алушылармен сатып алынады» дереу байланыс бар. Олардың арасындағы қарым-қатынас «көптегенден-көпке» түріне жатады (2.24-сурет, а). Осы қатынастарды шешу үшін сатып алушы (өнім

беруші) тауарларды сатып алуды (сатуды) көрсететін «Қаптау» байланысты субъектіні енгіземіз (2.24, б).

Субъектілердің атрибуттарын талдайық. Әрбір жеткізуші мен сатып алушы заңды тұлға болып табылады және аты, мекен-жайы, банктік деректемелері бар. Әр тауардың атауы, бағасы бар, өлшем бірлігі. Әрбір жүкқұжатта бірегей нөмір, шығарылған күні, саны мен бағалары бар тауардың тізімі, сондай-ақ шот-фактураның жалпы көлемі бар. Сатып алынатын тауарлардың саны мен бағасы туралы мәліметтер енгізілетін шот-фактураларды бір уақытта алуды сатып алыңыз. Әрбір сатып алушы бірнеше үстеме шығындарды ала алады. Әрбір шот-фактура бір сатып алушыға беріледі. Әрбір жүк жөнелтімінде кемінде бір өнім болуы керек («бос» жүкқұжат бола алмайды). Әрбір өнім, өз бетінше, бірнеше жүк жөнелтілімдерінде бірнеше тұтынушыға сатыла алады. «Өнім» және «Жеткізуші» арасындағы өзара қарым-қатынастарды анықтау үшін ұқсас логикалық тізбекті құруға болады. Сатып алушы бір мезгілде жеткізуші бола алады, сондықтан бұл екі бірлік «Контрагенттің» біріне біріктіріледі. Енді сіз мұның бәрін диаграммада көрсетілгендей етіп жасай аласыз. 2.24,б.

Тазартылған диаграмма техникалық тапсырмада көрсетілген немесе әзірленген жүйенің ағымдық диаграммасында көрсетілген барлық шығыс деректерін (есептерді) алу мүмкіндігімен тексерілуі керек.

Тапсырма

Тәжірибелік жұмыстың техникалық шарттарына сәйкес 1 (Қосымшаны қараңыз), бағдарламалық қамтамасыз етуге арналған функционалды талаптарды құрылымдық талдаңыз және қажетті диаграммаларды жасаңыз. MS Visio көмегімен нәтижелерді аяқтаңыз.

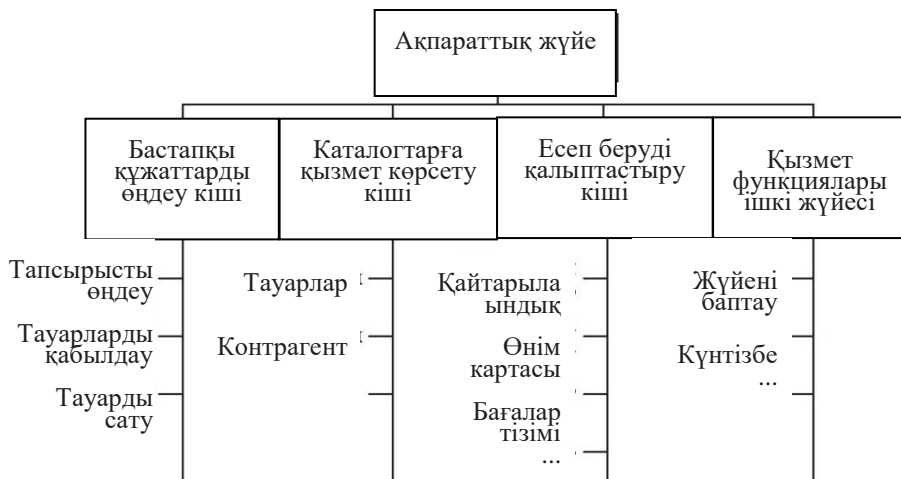
БАҚЫЛАУ СҰРАҚТАРЫ

1. Диаграммаларды құру қағидаларын келтіріңіз. Диаграмманың әр түрінің мақсатын түсіндіріңіз.
2. Құрылымдық тәсілдің негізгі принциптері қандай?
3. SADT функционалды моделі мен деректер ағынының сұлбасы арасындағы айырмашылық неде және қандай айырмашылық бар?
4. Құрылымдық тәсілдің артықшылықтары мен кемшіліктерін атаңыз

БАҒДАРЛАМАЛЫҚ ЖАСАҚТАМА ЖОБАСЫ

3.1. Блок диаграммасы (құрылымдық схема)

Бағдарламалық жасақтама жасау процесі бағдарламалық жасақтаманың құрылымдық компоненттерін және олардың арасындағы қарым-қатынастарды анықтауды қамтиды. Құрылымды нақтылау нәтижесі бағдарламалық жасақтаманың толық түсінігін беретін құрылымдық диаграмма түрінде ұсынылуы мүмкін.



Суретте. 3.1 - «Көтерме сауданың қоймасы» автоматтандырылған ақпараттық жүйесінің бағдарламалық қамтамасыз етуінің блок-схемасы.

AIS бағдарламалық қамтамасыз етуінің құрылымдық диаграммасы «Көтерме қойма»

3.2. Функционалдық схема

Функционалдық диаграмма - ақпараттық ағындардың сипаттамасымен, ағындардағы деректер құрамымен және қолданылған файлдар мен құрылғылардың көрсетілуімен бағдарламалық жасақтама компоненттерінің өзара іс-қимыл схемасы.

Схемалар деректерді өңдеу тапсырмасының өлшеміне және күрделілігіне байланысты деңгейдің санымен бірге әртүрлі деңгейде қолданылуы мүмкін. Егжей-тегжейлі мәліметтер болуы керек, олар әртүрлі бөліктер мен олардың арасындағы қарым-қатынас жалпыға түсінікті болуы керек.

Деректер сұлбалары тапсырмаларды шешу үшін деректер жолын көрсетеді және өңдеу қадамдарын, сондай-ақ пайдаланылатын түрлі деректер тасымалдаушыларды анықтайды. Деректер схемасы келесі таңбалардан тұрады:

- деректер таңбалары (деректер таңбалары деректер тасымалдаушысының түрін көрсете алады);
- деректер таңбалары (деректер таңбалары деректер тасымалдаушысының түрін көрсете алады);
- деректермен жасалған процестің рәміздері (процестің таңбалары компьютермен орындалатын функцияларды көрсете алады);
- процестер және (немесе) деректер тасымалдағыштар арасындағы деректер ағындарын көрсететін сызықтық таңбалар;

Схеманы жазу мен оқуды жеңілдететін арнайы таңбалар.



Бағдарлама диаграммалары бағдарламадағы операциялардың кезектілігін көрсетеді. Бағдарламалық схема келесі таңбалардан тұрады:

- нақты деректерді өңдеу операцияларын көрсететін процестің рәміздері (логикалық жағдайларды ескере отырып, келесі жолды белгілейтін рәміздер);
- Басқару ағынын көрсететін сызықтық таңбалар;
- схеманы жазу мен оқуды жеңілдететін арнайы таңбалар.

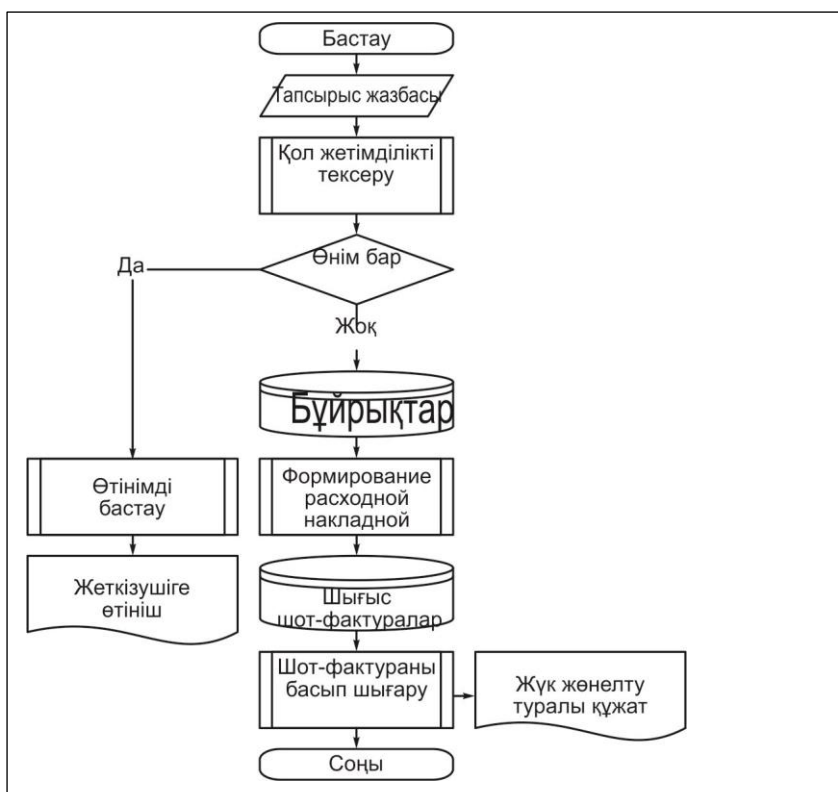
Жүйенің жүйелік диаграммалары жүйедегі деректердің ағымын бақылауды көрсетеді. Жүйенің сұлбасы келесі рәміздерден тұрады:

■ деректердің бар екендігін көрсететін деректер таңбалары (деректер таңбалары деректер тасымалдаушысының түрін көрсете алады);

Кесте 3.1. Алгоритмдердің негізгі блоктарының графикалық рәміздері

Атауы	Графика тағайындау	Кездесу
Терминатор		Таңба сыртқы ортаға және сыртқы ортаға кірісті (бағдарлама схемасының басы немесе соңы, сыртқы пайдалану және деректердің қайнар көзі немесе
Деректер		Таңба мәліметтерді көрсетеді, сақтау ортасы анықталмаған
Тікелей қол жетімді құрылғы		Таңба тікелей қол жеткізу жадында сақталған деректерді (магниттік диск, магниттік барабан, дискета)
Құжат		Таңба адамда оқылатын нысанда (машина бағдарламасы, оптикалық немесе магниттік оқуға арналған құжат, микрофильм, жиынтық мәліметтермен деректерді енгізу формалары бар таспа орамы) деректерді көрсетеді.
Қолмен енгізу		Таңба кез келген түрдегі құрылғыдан (пернетақта, қосқыштар, түймелер, жеңіл қалам, жолақ штрих коды бар) өңдеу кезінде қолмен енгізіледі.
Дисплей		Таңба дисплей құрылғысы түрінде тасымалдаушыда адамның оқитын нысанда ұсынылған деректерін (бейнебақылауға арналған экран, ақпаратты енгізу индикаторлары) көрсетеді,
Процесс		Таңба кез-келген деректерді өңдеу функциясын көрсетеді (белгілі бір опера-

Атауы	Графика тағайындау	Кездесу
		Немесе ақпараттың құны, нысаны немесе орнын өзгертуге немесе бірнеше ағын бағытын ауыстыру керек анықтамаға алып келетін операциялардың тобы)
Алдын ала анықталған процесс		Белгі басқа жерде (кіші, модульде) анықталған бір немесе бірнеше операцияларды немесе бағдарлама қадамдарын қамтитын алдын ала анықталған үрдісті көрсетеді,



Сурет. 3.2. AIS бағдарламалық қамтамасыз етуінің функционалдық схемасының мысалы «Көтерме қойма»

■ деректерде орындалатын операцияларды көрсететін процестің рәміздері, сондай-ақ орындалатын логикалық жолды анықтау;

■ процестер мен (немесе) деректер тасымалдағыштар арасындағы деректер ағындарын көрсететін сызықтық таңбалар, сондай-ақ процестер арасындағы бақылау ағымы;

■ сценарий жазуды және оқуды жеңілдететін арнайы таңбалар.

Функционалды диаграммаларды көрсету үшін ГОСТ 19.701-90 белгілеген арнайы таңбаларды қолданыңыз (3.1 - кесте).

Функционалдық схемалар құрылымдық сипатқа қарағанда, ақпараттылыққа ие. 3.2 - суретте «көтерме қойма» автоматтандырылған ақпараттық жүйенің тауарларын сатуды жүзеге асыратын бағдарламалық қамтамасыз ету жүйесінің функционалдық диаграммасы.

Тапсырма

Техникалық тапсырма негізінде және алдыңғы жұмыстарда бағдарламалық қамтамасыз етуге қойылатын талаптарды талдау негізінде олардың нұсқасы бойынша құрылымдық және функционалдық бағдарламалық қамтамасыз ету схемаларын әзірлейді.

БАҚЫЛАУ СҰРАҚТАРЫ

1. Бағдарламалық жасақтама құрылымының диаграммасына мысал келтіріңіз

2. Қандай функционалдық схемалар қолданылады?

3. Бағдарламалық қамтамасыз етудің функционалдық схемаларының негізгі элементтерін сипаттаңыз.

4. Функционалдық схемаларды пайдалану артықшылықтары мен кемшіліктері қандай?

БАҒДАРЛАМАЛЫҚ ҚАМТАМАСЫЗ ЕТУДІ ТАЛДАУ ЖӘНЕ ЖОБАЛАУҒА ОБЪЕКТІЛІ –БАҒДАРЛАНҒАН ТӘСІЛДІ ҚОЛДАНУ

4.1 Қолданыс түрлеріне қатысты диаграммалар

Қазіргі уақытта бағдарламалық жасақтаманың талаптарын модельдеуге мүмкіндік беретін ондаған техника, техника, көрнекі ұсынымдар бар. Белгілі бір әдістерді қолданудың орындылығын анықтау қажет. Талдаудың талдауы жүйенің не істейтініне сәйкес болуы керек, оны жүзеге асыру мәліметтерінен анықтайды, яғни. ол қалай жасайды.

UML тілі пайдаланушыларға әртүрлі мақсаттарға арналған жүйелердің күрделі модельдерін әзірлеуге және құжаттауға арналған жеңіл түсінікті және айқын визуалды модельдеу тілін береді.

Қолдану жағдайларын анықтаңыз. Бағдарламалық қамтамасыз етудің ерекшеліктерін әзірлеу техникалық тапсырмада көрсетілген функционалдық талаптарға талдау жасаудан басталады. Талдау барысында әзірленген бағдарламалық жасақтаманың сыртқы пайдаланушылары анықталады және белгілі бір пайдаланушылармен өзара әрекеттесу барысында оның мінез-құлқының жеке аспектілерінің тізімі анықталады. Бағдарламалық жасақтаманың мінез-құлқы жағдайларды пайдалану немесе жағдайларды пайдалану деп аталды.

Пайдалану жағдайын болашақ жүйенің нақты әрекеттерімен шатастырмаңыз. Әрбір пайдалану жағдайы тәуелсіз мәнге ие мақсатқа байланысты; мысалы, мәтіндік редактор үшін Мазмұндар кестесін құру пайдалану болып табылады және тақырыптарды арнайы мәнерлермен байланыстыру - мазмұнның кестесін автоматты түрде жасау үшін орындау керек операция.

Процедураның мақсаттарына байланысты төмендегідей қолданулар бөлінеді:

- базалық (негізгі) - әзірленетін бағдарламалық қамтамасыз

етудің

қажетті функционалын қамтамасыз ету;

■ көмекші - қажетті жүйелік параметрлерді және техникалық қызмет

көрсетуді қамтамасыз ету (мысалы, ақпаратты мұрағаттау және т.б.);

■ Қосымша - пайдаланушыға қосымша ыңғайлылықты қамтамасыз

етеді (әдетте, кез-келген ресурстардың ресурстары әзірленуде де,

жұмыс кезінде де талап етілмейді).

UML тілінің графикалық құралдары іс жүзінде бағдарламалық жүйе үшін функционалдық талаптарды көрсету үшін жеткіліксіз болып табылады. UML тілінің талаптарының бірі жобаланған жүйелердің модельдері туралы ақпаратты ұсыну үшін диаграммалардың өзін-өзі қамтамасыз етуі болып табылады. Дегенмен, UML тілінің визуалды құралдары күрделі жүйенің функционалды мінез-құлқының ерекшеліктерін пайдаланудың опцияларын диаграммаларда ескеру жеткіліксіз. Ол үшін осы схемаларды осы сценарийлермен толтыру ұсынылады, немесе оны пайдалану жағдайларын орындау кезінде жүйе орындаған әрекеттердің кезектілігін анықтайды.

UML тілінің контексінде сценарийлер актерлердің өзара әрекеттесуінің және қолдануға болатын нұсқалардың қосымша мысалында пайдаланылады. Осындай сценарийлерді ұсынуға немесе жазуға болатын түрлі тәсілдер ұсынылады. Осы үлгілердің бірі кестеде көрсетілген. 4.1 және концептуалды модельдеудің бастапқы кезеңдерінде қолдану үшін ұсынылуы мүмкін. Абстракцияның деңгейіне қарай пайдалану жағдайын қысқа немесе толық сипаттауға болады. Сипаттаманың қысқа формасы: пайдалану жағдайының атауы, қолданыстағы актерлердің (актерлердің) тізімі және пайдаланудың (негізгі, көмекші немесе қосымша) түрінің қысқаша сипаттамасы.

Пайдалану сценарийлерін жазған кезде, сценарий мәтіні пайдалану диаграммасын толықтыруы немесе нақтылауы керек, бірақ оны толығымен ауыстырмауы керек. Әйтпесе, үлгілерді көрнекілендіру артықшылығы жоғалады.

Негізгі ұғымдар. Пайдалану схемасы. Қолдану жағдайларының диаграммалары бағдарламалық жасақтаманың күтілетін мінез-құлқын бейнелейді. Пайдаланулы диаграммалардың негізгі ұғымдары: актер, пайдалану жағдайы және сілтеме.

Суретте. 4.1 Қолдану схемаларын көрсету үшін пайдаланылатын таңбалар берілген.

Кесте 4.1. Сценарийді бөлек пайдалану жағдайына жазу үшін үлгі

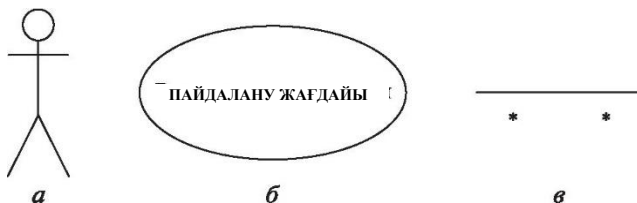
Аты	Іс бойынша табысты іске асыруға әкелетін оқиғалардың әдеттегі бағыты	Ерекшелігі 1	Ескерту: 1
Таңбалар		Ерекшелігі 2	Ескерту: 2
Мақсаты		Ерекшелігі 3	Ескерту: 3
Түрі	
Қысқаша сипаттама			
Басқа пайдаланылатын сілтемелер		Ескерту: <i>n</i>	Ескерту: <i>n</i>

Қолданыстағы адам кез-келген ақпаратты алу немесе қамтамасыз ету мақсатында онымен өзара іс-қимыл жасайтын бағдарламалық жасақтамадан тыс болып табылады. Жоғарыда айтылғандай, актерлер пайдаланушылар, басқа бағдарламалық жасақтама немесе кейбір техникалық құралдар болуы мүмкін.

Пайдалану жағдайы - оның нақты тапсырмасын шешетін адамға белгілі бір рәсім. Қолданудың барлық жағдайлары бір немесе бірнеше тәсілмен дамыған жүйенің функционалдылығына қойылатын талаптарға байланысты және орындалған жұмыстың көлемінде айтарлықтай өзгеруі мүмкін (4.2-сурет).

Байланыс - актерлердің өзара әрекеттесуі және тиісті пайдалану жағдайлары.

Пайдалану параметрлері арасында байланыс болуы мүмкін. Сонымен қатар, пайдалану мен кенейтуді байланыстырады.



Сурет 4.1. Қолдану схемаларының негізгі рәміздері: *a* - жсауапты тұлға; *b* - пайдалану жағдайы; *c* – байланыс

Пайдалану (пайдалану (қамту)) бірнеше бағдарламаларда қайталанатын бағдарламалық жасақтаманың мінез-құлқының бірнеше фрагменті бар екенін білдіреді. Бұл фрагмент бөлек нұсқа ретінде құрастырылған және онымен «пайдалану» түрінің

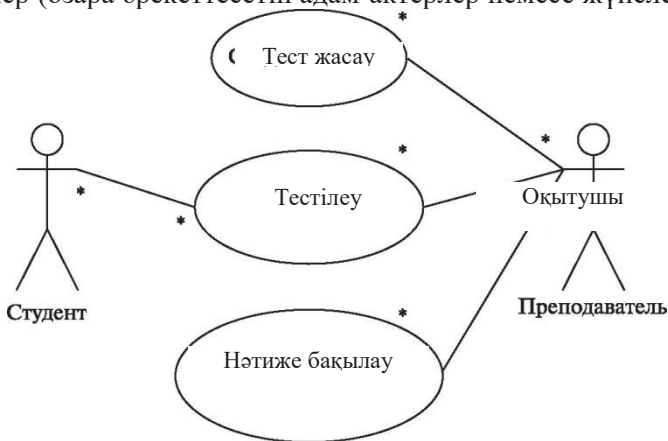
байланысын көрсетеді.

Ұзартқыштар, егер олардың біреуінде бірнеше қосымша әрекеттер болған жағдайда, екі қолданылатын болса, пайдаланылады. Бұл жағдайда қосымша әрекеттер «кеңейту» түріндегі байланыстың негізгі нұсқасымен байланысты жеке пайдалану жағдайында анықталады.

Қолдану диаграммасының негізгі мақсаты жүйеге қойылатын функционалдық талаптарды ресімдеу және дизайнды бастапқы кезеңде тұтынушымен модельдеу мүмкіндігі. Пайдалану нұсқаларының кез-келгені одан әрі түпнұсқалық нысанды қалыптастыратын жекелеген элементтерді пайдаланудың бірнеше қосалқы нұсқаларына айналдырылуы мүмкін.

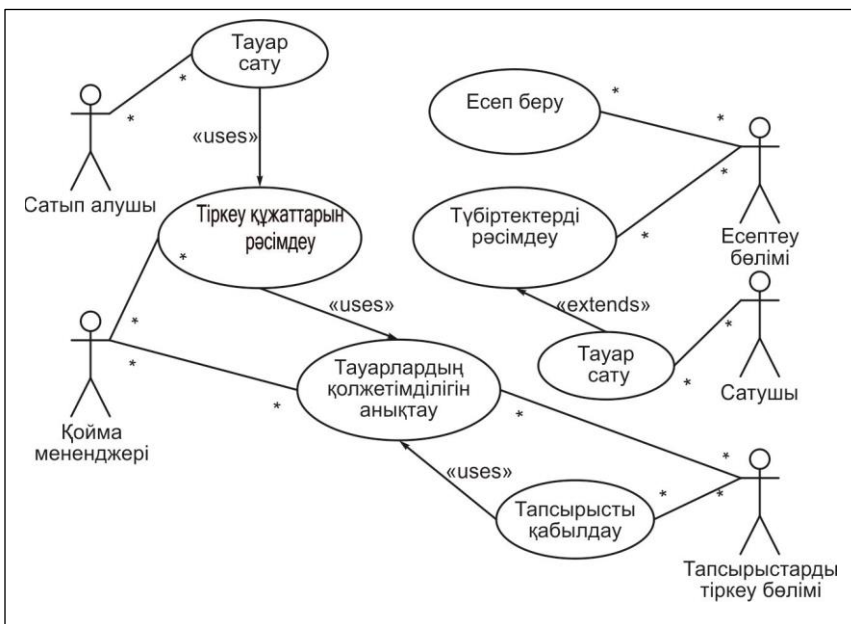
Қолдану жағдайларының диаграммасын жасаудың мысалы.

Пайдалану жағдайларының диаграммасында функционалдық талаптардың ерекшеліктерін көрсету үшін «Көтерме қойма» жүйесінің үлгісін қарастыруға болады (4.3-сурет). Бағдарламалық жасақтаманың құрылымын бастапқы түсіну үшін актерлер (өзара әрекеттесетін адам-актерлер немесе жүйелер)



анықталған.

Сурет 4.2. Сынақ жүйесінде пайдалану жағдайларының диаграммасының мысалы



Сурет 4.3. «Көтерме қойма» АЖЖ-ны жобалауға арналған пайдалану нұсқаларының схемасы

жүйеде 5 актер бар, олардың екеуі қарсы агенттер болып табылады, ал басқалары - барлық операцияларды орындайтын қойма менеджерлері. Осы актерлердің әрқайсысы жүйемен өзара әрекеттеседі, бірақ негізгі актерлар жүйенің функционалдығын бастамағандықтан, сатып алушылар мен сатып алушылар (мердігерлер) болуы мүмкін. Сонымен қатар, жағдайларды пайдалану тұжырымдалады, яғни. актерлердің (актерлердің) байланысын жүзеге асыру үшін жүйе жүргізетін іс-шаралар.

Жүйесіне қатысты нақты мақсаттарды көздеген Әрбір актерлер: сатып алу тауарларды, қойма менеджері сатып алушы үшін тауарларды тапсыруға және өнім бухгалтерлік бөлімі менеджері қабылдауға - жеткізуші кірістер мен сату мөлшерін белгілейді, және тауар қорын талдау. Осы міндеттер негізінде негізгі пайдалану жағдайлары тұжырымдалған және қарым-қатынас Санағыш талдау болады. Шындығында, жағдайларды пайдалану әлдеқайда көп болуы мүмкін. Мысалы, тапсырыс берушінің кредит тексеру өнім туралы мәлімет алуға және қоймада тауарларды қорларын бағалау үшін, және т.б. төлейді Дегенмен, бұл диаграмма жүйенің не істейтінін, оның қалай жұмыс істейтінін түсіндіреді.

Қарастырылып отырған жүйе үшін пайдалану жағдайларын модельдеудің келесі кезеңінде бұл диаграмма бұрын ұсынылған үлгі негізінде жазылған мәтіндік сценариймен

толықтырылуға тиіс. Бұл сценарий жүйені және актерлерді тауарларды енгізу және сату процесінде жүзеге асыратын жеке әрекеттердің мазмұны мен логикалық дәйектілігін айқындайтын диаграмманы толтырады. Бұл жағдайда сценарий кестелер түрінде ыңғайлы түрде ұсынылады, олардың әрқайсысында үлгінің жеке бөлімі сипатталады.

Сценарийдің негізгі бөлімінде сіз қарастырылған қолданыстағы істердің атауын, онымен байланысты актерлердің атауларын, нұсқаны орындаудың мақсаты, шартты типті және басқа мақсаттарға сілтемелерді көрсетіңіз (4.2-кесте).

Сценарийдің келесі бөлімінде қарастырылған іс бойынша табысты іске асыруға әкелетін іс-әрекеттердің дәйектілігі сипатталады. Бұл жағдайда актер әрекеттердің бастамашысы ретінде әрекет етуі керек. Кейінгі сілтемелерге ыңғайлы болу үшін әр актерлердің әрекеттері әрекеттердің реті бойынша жүйелі нөмірмен белгіленеді (4.3-кесте).

Кесте 4.2. Сценарийді қолдану	
Регистрді пайдаланыңыз	Тауарды сату
Актерлер	Сатып алушы, Тапсырыс менеджері, қойма менеджері
Қысқаша сипаттамасы	Сатып алушы тауарды сұрайды. Тапсырыстарды тіркеу бөлімінің бастығы тауарларды сақтайды, тапсырысты ресімдейді, тапсырманы қойма менеджеріне береді. Сатып алушы тауарларды төлейді, тауарларды қоймада алады
Мақсат	Тиісті өнімді алу
Түрі	Негізгі
Басқа пайдаланылатын сілтемелер	тауарлардың қолжетімділігін анықтау;

Кесте 4.3. Актерлердің кезегі

Актерлердің міндетін атқарушы	Жүйенің жауабы
1. Сатып алушы тауарды сұрайды 1. Қоймада сұралған тауарлардың саны талап етілмейді	2. Тапсырысты ресімдеу бөлімінің бастығы қоймада қажетті тауарлардың болуын тексереді 3. Тапсырыс беруші бөлімшесі қажетті тауарларды сақтайды
4. Сатып алушы тауарды төлейді 2. Болдырмау 2. Сатып алушы тауарға төлем жасамды	5. Өтінімдерді тіркеу бөлімінің бастығы тауарларды алуға рұқсат береді 6. Тапсырыстарды өндеу бөлімінің бастығы бұйрықты қоймаға береді 7. қойма жетекшісі сатып алушы үшін тауарлар мен шот-фактураларды шығарады 8. Тапсырысты ресімдеу жөніндегі менеджер сатып алушы тауардың түсуін болдырмайды

Сценарийдің үшінші бөлімінде айрықша жағдайлар орын алғанда немесе ерекше жағдайлар орын алған кезде орындалған әрекеттердің реті сипатталады (4.4-кесте).

Бұл сценарийді «Тапсырыс» және «Тауарлардың қолжетімділігін анықтау» ғана емес, сондай-ақ тұрақты клиенттерге жеңілдіктер жасау сияқты басқа да ерекше жағдайларды қарастыратын сияқты сипаттайды. Бұл жағдайда сценарийлер мен пайдалану жағдайларының толықтығы нақты жобаның шеңберінде тұжырымдалған функционалдық талаптармен анықталады.

Кейбір көлемі сценарий шағын ноталары түрінде диаграммада орналастырылуы мүмкін Ескертпе, дамыту жобасының шеңберінде тікелей байланысты еркін мәтін , моделін енгізу үшін соған арналғанын ескеріңіз. Сияқты ақпарат жасаушыға Пікірлер (мысалы, күні, нұсқасы диаграммасы немесе оның жекелеген компоненттері), шектеулер (облигациялар немесе даналардың мысалы, құндылықтар) болуы және белгіленген мәндерге мүмкін. Қолдану диаграммаларына қатысты жазбада біліктілік болуы мүмкін

Кесте 4.4. Ерекше жағдайлар туындаған кезде актерлердің әрекеттер тізбегі

Актерлердің міндетін атқарушы	Жүйенің жауабы
Ерекшелік 1. Қоймада талап етілетін тауарлардың саны талап етілмейді	
4. Сатып алушы тауарларды төлейді	3. Тапсырыс басқармасының бастығы талап етілетін өнімді жеткізуді бастайды
Ерекшелік 2. Сатып алушы тауарға төлем жасамады	
	8. Тапсырыс беруші сатып алушы тауардың түсуін болдырмайды

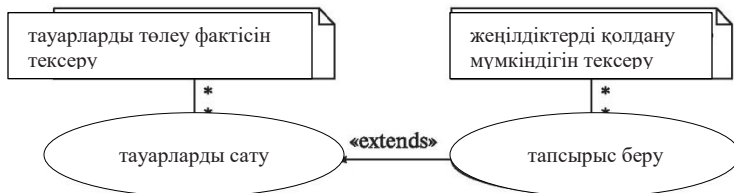
белгілі бір пайдалану жағдайларының контекстіне қатысты ақпарат.

Графикалық түрде диаграммалардың барлық түрлеріне арналған белгілер «тік» жоғары оң жақ бұрышы бар тіктөртбұрышпен белгіленеді (4.4-сурет). Нақты жазба мәтіні осы тіктөртбұрыштың ішіне орналастырылады. Ескерту диаграмманың кез келген элементіне сілтеме жасай алады, бұл жағдайда сызылған сызық оларды байланыстырады. Егер ескерту бірнеше элементтерге жатса, онда одан бірнеше сызық жасалады. Жоғарыда айтылғандай, ескертулер тек пайдалану диаграммасында ғана емес, сонымен қатар басқа канондық диаграммаларда болуы мүмкін.

Пайдаланудың диаграммаларын эзірлеу бойынша ұсынымдар.

Жоғарыда айтылғандай, пайдалану жағдайларының диаграммасының негізгі мақсаттарының бірі жүйеге қойылатын функционалдық талаптарды ресімдеу болып табылады.

Пайдалану нұсқаларының диаграммасы клиентпен келісу үшін негіз бола алады -



Сурет. 4.4. Іс диаграммаларындағы ескертулер мысалы

Қолдану жағдайларының схемасын әзірлеу үшін белгілі бір әрекеттер тізбегін ұстану ұсынылады:

- негізгі, немесе бастапқы және екінші актерлерді анықтау;
- жүйеге қатысты негізгі субъектілердің мақсаттарын анықтау;
- жүйенің функционалды талаптарын анықтайтын негізгі пайдалану жағдайларын тұжырымдау;
- пайдалану жағдайларын оларды іске асыру тәуекелінің төмендеу дәрежесі бойынша реттеу;
- қауіп-қатерді төмендетудің барлық негізгі түрлерін қарастырыңыз;
- таңдаған ісін іске асыру үшін қатысушыларды, мүдделерін, алғышарттарын және кейінгі жағдайларын анықтау;
- таңдалған пайдалану ісін іске асыру үшін сәтті сценарий жазу;
- пайдалану жағдайларының сценарийіндегі айрықшалықты немесе сәтсіздіктерді анықтау;
- барлық ерекшеліктер үшін сценарий жазу;
- ортақ пайдалану жағдайларын айқындауға және өздерінің қарым-қатынастарын стереотипке негізделген пайдалану (пайдалану);
- айрықша жағдайларға назар аударыңыз және стереотиптің кеңеюімен олардың өзара қарым-қатынастарын бейнелеңіз;
- қайталанатын пайдалану жағдайлары мен актерлердің болмауы үшін диаграмманы тексеріңіз.

Қолдану схемасын құрастырудың семантикасы жоғарыда талқыланған модель элементтерінің келесі ерекшеліктерімен анықталады. Мазмұнын пайдалану нұсқасының жеке данасы - актердің данасынан хабардың электронды көшірмесі арқылы инициализацияланатын әрекеттердің кезектілігін орындау. Жауап ретінде немесе актердің хабарламасына жауап -бұл пайдалану жағдайына арналған әрекет. Бұл жағдайда актерлер пайдалану жағдайларын бастау үшін жаңа хабарламалар жасай алады.

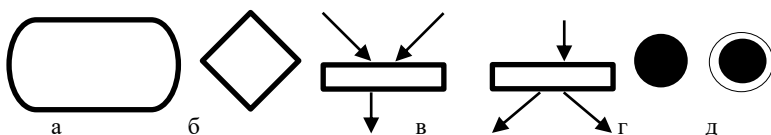
Мұндай өзара іс-қимылдар талап етілетін жұмыс тәртібінің орындалуын пайдалану жағдайының данасы бойынша аяқталғанға дейін жалғасады және модельде көрсетілген актердің қызмет көрсету талап етілетін данасын алмайды. Әрекет етудің соңы - негізгі қолданыс жағдайларында актерлерден хабарларды инициализациялау жоқ.

Пайдалану жағдайлары мәтінмен бірге ескертулермен қосымша көрсетілуі мүмкін, ол болашақта атрибуттармен бірге операциялар мен әдістердің прототипі бола алады. Модельдерді одан әрі дамыту, түпкілікті автомат немесе кез-келген мінез-құлықтың логикалық көрінісі механизмі, соның ішінде алдын-ала шарты мен постпозициясы арқылы әрекет диаграммасы түрінде пайдалану нұсқаларын жүзеге асырумен байланысты. Қолдану жағдайлары мен актерлер арасындағы өзара әрекеттестік осы пайдалану нұсқаларын қамтитын жүйе арасындағы қарым-қатынастар сипатталғанда және осы жүйенің ортасы немесе сыртқы ортасы болған кезде кооперативті диаграммада тазартылуы мүмкін.

4.2. ҚЫЗМЕТ КӨРСЕТУ ДИАГРАММАЛАРЫ

Әрекет диаграммасының сипаттамасы. Егер пайдалану диаграммасы жүйенің функционалдық мүмкіндіктерін «жоғарыдан көруді» ұсынса, екінші жағынан, белсенділік диаграммасы бөлек пайдалану жағдайын және оның сценарийлерін егжей-тегжейлі суреттеуге мүмкіндік береді. Егжей-тегжейлі дәрежесіне байланысты, іс-әрекеттердің диаграммалары дамудың әртүрлі кезеңдерінде қолданылуы мүмкін. Техникалық талаптарды талдауға және спецификацияларды нақтылауға арналған кезеңде, әрекет диаграммалары әзірленген бағдарламалық жасақтаманың негізгі функцияларын анықтауға мүмкіндік береді.

Бұл жағдайда қызметті қолмен немесе автоматтандырудың көмегімен жасау керек тапсырманы (операцияны) түсініңіз. Әрбір өтінімде өз міндеттерінің бірізділігі бар. Теориялық тұрғыдан, қызмет диаграммасы талданған пайдалану жағдайын іске асыратын алгоритмнің жалпыланған көрінісі болып табылады. Диаграммада белсенділік дөңгелек бұрыштары бар тіктөртбұрышпен белгіленеді (4.5-сурет, а). Әрекет диаграммалары баламалы және параллель процестерді сипаттауға мүмкіндік береді.

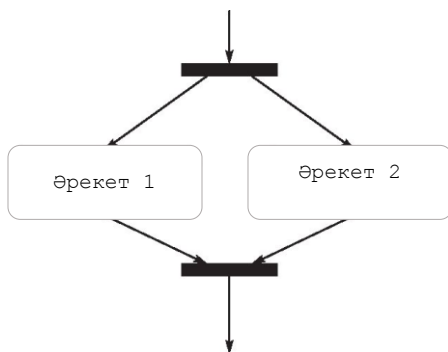


Сурет. 4.5. Іс диаграммасының рәміздері: а - қызмет; б - таңдау; с - үндестіруді басқару; г - басталуы; д – соңы

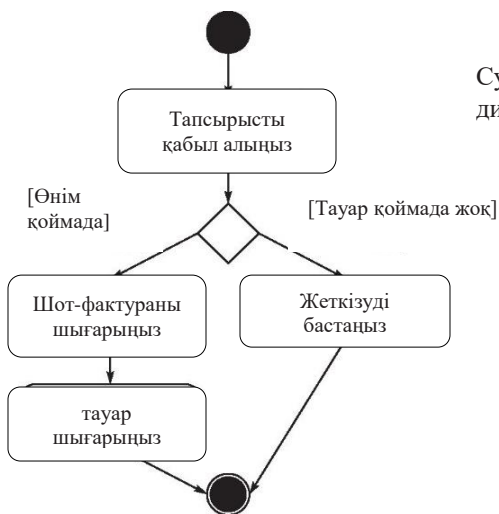
Баламалы Процестерді Көрсету Үшін, Ромб (4,5, В) Қолданылады, Шартты Бірге Көрсетіледі, Ал Балама «Иә», «Жоқ» - Тиісті Нәтижелердің Жанында. Сол Блоктың Көмегімен Циклдік Процесті Жасауға Болады. Әрекет Белсендірудің Көптігі Активтендіру Активтендіру Көрсеткішінің Жанында Орналасқан «*» Символымен Белгіленеді, Ал Қажет Болған Жағдайда «Әр Жолға» Жазумен Көрсетіледі.

Параллельді Процестерді Белгілеу Үшін Үндестіруді Басқарыңыз (Сурет 4.5, С). Функция Диаграммасының Басталуы Мен Аяқталуына Арналған Шартты Белгілер Күріш. 4.5, D, Және D. Процестердің Параллелизмін Көрсететін Әрекет Диаграммасының Мысалы, Күріш. 4.6. Синхрондау Күйін Оны Диаграммада Көрсете Отырып Айқындауға Болады (4.7-Сурет).

Әрекет схемасын құрастырудың мысалы. Алдыңғы мысалда «Көтерме қойма» автоматтандырылған ақпараттық жүйесінде пайдалану жағдайларын сипаттау үшін мәтіндік сценарий жасалды. Бұл сценарий жүйе мен актерлердің жеке әрекеттерінің мазмұнын ашатын диаграмманы толтырады. Алайда, қолдану жағдайларын сипаттаудың орнына немесе олардан басқа, сіз белсенділік диаграммаларын пайдалана аласыз. Әрекеттер кестесіталап етілетін дәрежесі бар пайдалану жағдайын суреттеуге мүмкіндік береді. Пайдаланудың тек қана нұсқаларын түсіндіру мағынасы бар, оның қысқаша сипаттамасы шешілетін мәселелердің мәнін түсіну үшін жеткіліксіз.



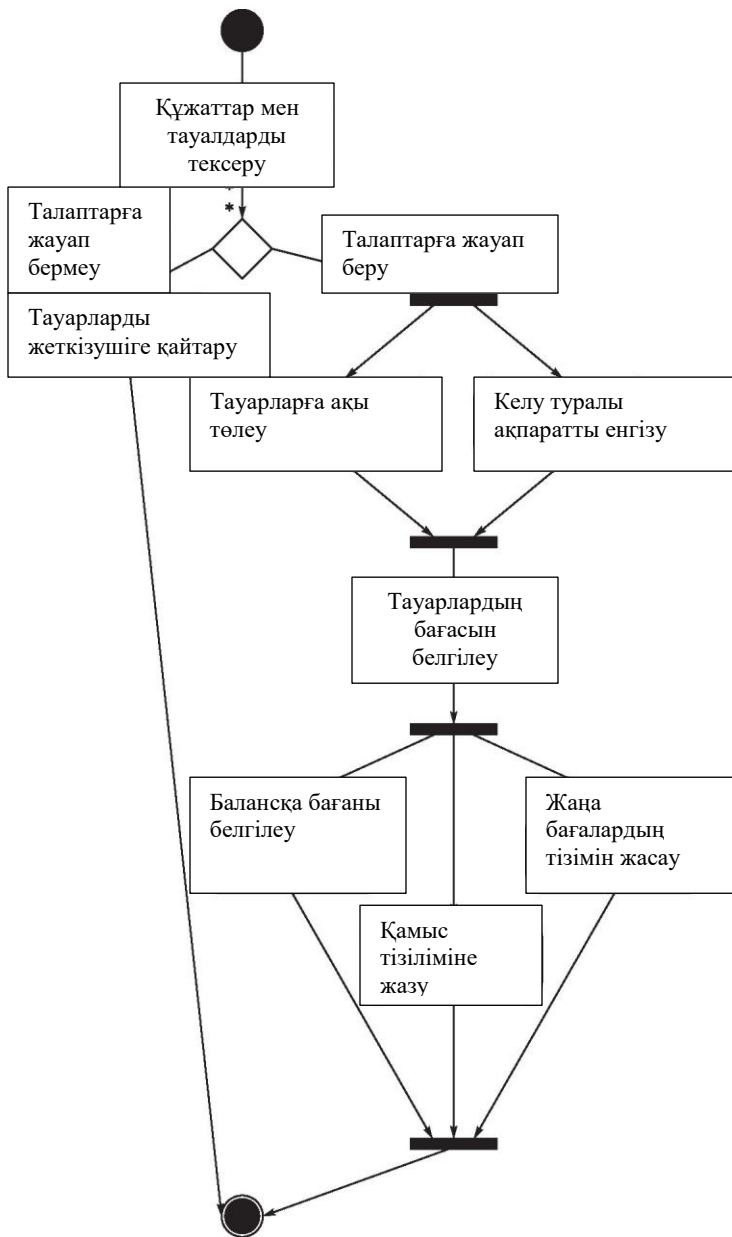
Сурет 4.6



Сурет. 4.7. Әрекет диаграммасының мысалы

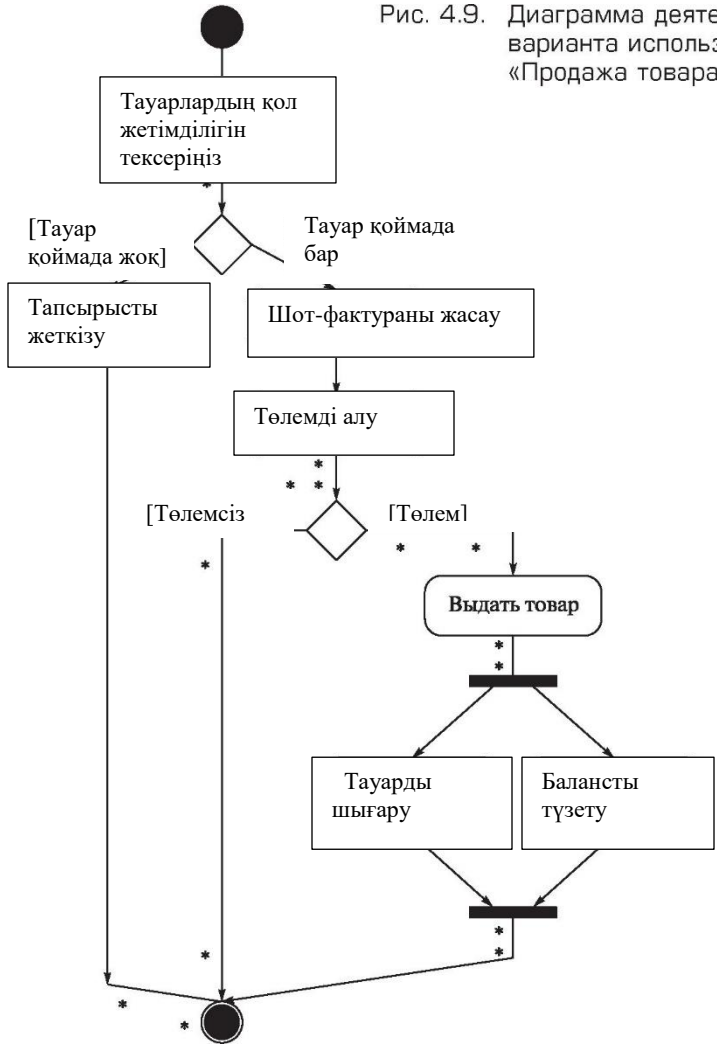
Әзірленетін модель шеңберінде біз «Тауарларды жеткізу» (4.8-сурет) және «Тауарларды сату» (4.9-сурет) іске асырудың белсенді диаграммаларын құрастырамыз. Әрекеттің диаграммасы әртүрлі дәрежелі бөлшектермен пайдалану жағдайын көрсетеді. Жүйенің толық үлгісі бірнеше маңызды диаграммаларды қамтуы мүмкін, олардың әрқайсысы ең маңызды пайдалану жағдайларын (оқиғалар мен барлық ерекшеліктердің типтік бағыты) немесе нетривиальды емес класс операцияларын жүзеге асырудың кезектілігін сипаттайды.

Стандартты сипаттама формасына қосымша, UML «жүзу трассалары» нұсқасын ұсынады. Бұл пішім пайдалану жағдайлары болған кезде жағдайды сипаттауға ыңғайлы бірнеше актер. Бұл жағдайда әрекет диаграммасында әрекет студияның барлық жағдайы бір-бірінен тік сызықтармен бөлінген жекелеген топтарға бөлінеді (4.10-сурет). Тректерді пайдалану бизнестің құрылымдық бөлімшелерінің қызметін көрсетуге мүмкіндік беретін бизнес-процестерді визуалды ұсыну үшін қосымша мүмкіндіктер ашады (4.11-сурет). Типтік жоба болған жағдайда, іс-қимылдарды жүзеге асырудың көп бөлігі бұрыннан белгілі жүйелерді талдау немесе дамып келе жатқан жүйелердің тәжірибесі негізінде белгілі болуы мүмкін. Стандартты шешімдерді пайдалану даму уақытын едәуір қысқартып, жобаны жүзеге асыру барысында мүмкін болатын қателіктерден аулақ болуы мүмкін.






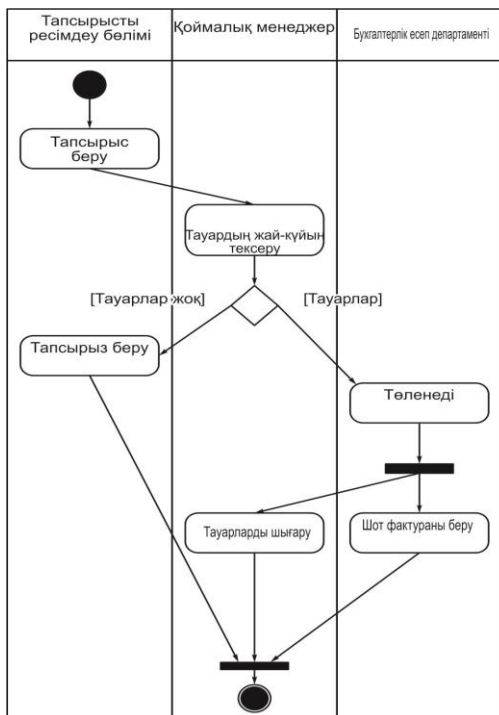
Сурет. 4.8. «Тауарларды жеткізу» жағдайда пайдалану диаграммасы

Рис. 4.9. Диаграмма деятельности для варианта использования «Продажа товара»



Сурет. 4.10. Тректермен жұмыс диаграммасының нұсқасы

Объект 1	Объект 2	Объект 3
		



Сурет 4.11. Әрекет схемасын қолдану үлгісі «Реттеу» жағдайына арналған тректер

Кезектілік диаграммасының сипаттамасы. Іс-әрекеттің диаграммалары бағдарламалық жүйелердің мінез-құлқының динамикасын анықтау үшін пайдаланылады, уақыт оларда ашық емес. Алайда мінез-құлықтың уақытша аспектісі объектілердің өзара әрекеттесуін сипаттайтын синхронды процестерді модельдеуде маңызды рөл атқаруы мүмкін. Бұл үшін UML жүйелі диаграммаларды пайдаланады.

Жүйелік тізбектің диаграммасы графикалық модель болып табылады, ол пайдалану жағдайындағы нақты сценарий үшін объектілердің өзара әрекеттесу динамикасын көрсетеді. Жүйе тізбегі диаграммасын құру үшін сізге қажет:

- әр актерді (объектіні) анықтаңыз және оған өмір сызығын көрсетіңіз;

- пайдалану жағдайларының сипаттамасынан жүйелік оқиғалардың жиынтығын және олардың реттілігін анықтаңыз;

- жүйе оқиғаларын актерлер мен жүйелердің өмірінің жолдары

арасындағы көрсеткі бар жолдар түрінде бейнелейді, сондай-ақ оқиға

аттары мен жіберілген мәндердің тізімдерін көрсетіңіз.

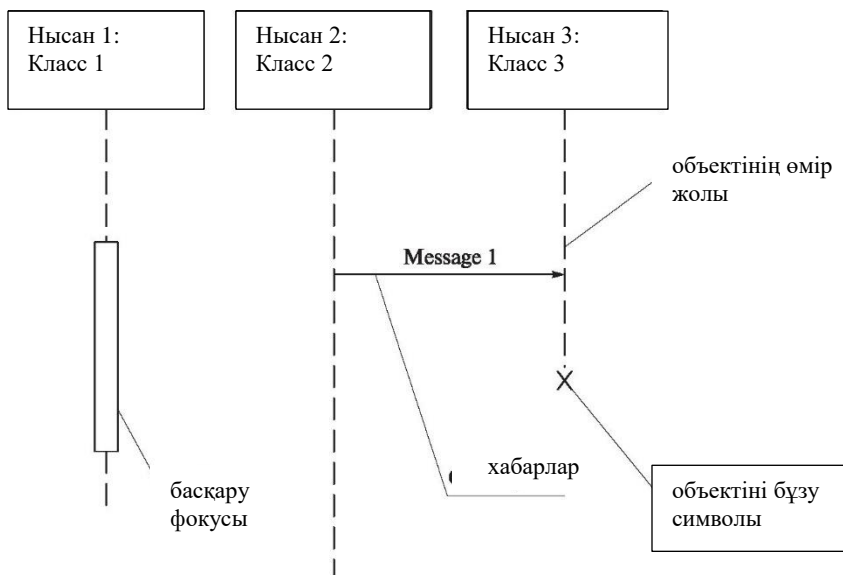
Кезектілік диаграммасы өзара әрекетке тікелей қатысатын және басқа нысандармен ықтимал статикалық қауымдастықтарды көрсетпейтін нысандарды ғана көрсетеді. Кезектілік диаграммасы үшін басты сәт объектілердің өзара әрекеттесуінің динамикасы. Бұл жағдайда тізбектік диаграмма екі өлшемге ие.

Бір өлшемі солдан оңға тік сызықтар түрінде, олардың әрқайсысы өзара әрекеттесуге қатысатын белгілі бір нысанның өмір сызығын білдіреді. Графикалық түрде, әрбір нысан тіктөртбұрышпен бейнеленеді және оның өмір жолының жоғарғы жағында орналасқан. Тіктөртбұрышта нысан атауы мен сынып атауы қос нүктемен бөлінген. Барлық жазба объектінің атрибуты болып табылатын, сыныптың данасы болып табылатын сызықпен сызылады (4.12-сурет).

Нысанның атауы дәйектілік диаграммада болмаған жағдайда жағдай жойылмайды. Бұл жағдайда сынып атауы ғана көрсетіледі және нысанның өзі анонимді болып саналады. Диаграммадағы сол жақ диаграмма объекті болып табылады өзара әрекеттесудің бастамашысы (4.12-суреттегі 1-нысан). Оң жақта - біріншіден тікелей өзара әрекеттесетін басқа зат. Осылайша,

кезектесу диаграммасындағы барлық нысандар бір-бірімен өзара іс-қимыл кезінде осы объектілердің белсенділігі дәрежесімен анықталған тәртіпті құрайды.

Кезектілік диаграммасының екінші өлшемі жоғарыдан төмен бағытталған тік уақытша ось болып табылады. Диаграмманың ең жоғарғы бөлігі бастапқы уақытқа сәйкес келеді. Бұл жағдайда нысанның өзара әрекеті кейбір нысандарға басқаларға жіберетін хабарлар арқылы жүзеге асырылады. Хабарламалар хабардың атауымен көлденең жебелер түрінде көрсетіледі және олардың пайда болу уақытына сәйкес тапсырыс жасайды. Басқаша айтқанда, жоғарыдағы тізбектің диаграммасында орналасқан хабарлар төменде орналасқандардың алдында басталады. Бұл жағдайда, уақыт осінде масштаб көрсетілген емес, өйткені дәйектеме диаграммасы «ертерек» кешіктірілген өзара әрекеттесудің уақытша тәртібін ғана бейнелейді.

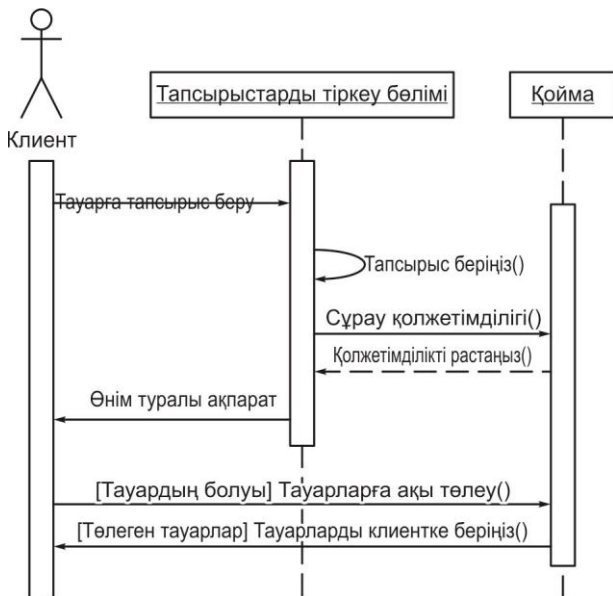


Сурет 4.12. Өртүрлі графикалық жүйелі примитивтер

Нысанның өмір сызығы объектінің жүйеде болған уақыт кезеңін көрсету үшін қызмет етеді және, тиісінше, оның барлық өзара әрекеттеріне қатыса алады. Диаграммада өмір сызығы бір нысанмен байланыстырылған сызықты тік сызықпен бейнеленеді. Егер объект жүйеде тұрақты болса, оның өмір сүру сызығы дәйектілік диаграммасының барлық жазықтық бойында жалғасуы керек.

Құрылыс жүйелі схемасын, симуляцияланған нысандар қатысады. Осыдан кейін, барлық нысандар кейбір тапсырыстың қадағалауымен инициализация хабарлар бар диаграммаға қолданылады. тек қажетті іс-шараларды орындау кезеңінде ол үнемі және уақытша болып табылатын болады қандай нысандар анықтау қажет. Нысандар анықталған кезде, хабарлардың сипаттамасына келеді. Бұл жағдайда жүйеде хабарлар ойнататын рөлдерді есепке алу керек.

Кезектілік диаграммасын жасаудың мысалы.



Сурет 4.13. Тауарды сатуды модельдеу үшін дәйектілік диаграммасының нұсқасы

Жүйенің мінез-құлқының уақытша аспектісін өзара әрекеттесуді сипаттайтын процестерді моделдеу кезінде ескеру үшін жүйелі диаграммаларды құруға болады. Мысал ретінде «Көтерме саудалық қойма» ақпараттық жүйесінде «Тауарларды сату» пайдалану жағдайын іске асыру үшін дәйектілік диаграммаларын саламыз (4.13-сурет).

Бұл диаграммада екі нысан және бір актер бар. Нысандар үнемі белсенді емес, бұл басқарудың тиісті фокусының көмегімен көрсетіледі. Хабарлардың атаулары - тиісті сыныптар үшін көрсетілген әрекеттердің атаулары. Кейбір хабарламалардың сөйлемдері квадрат жақшаларда қарапайым мәтінмен жазылады. Бұл шарттар сату процесін салуға және тиісті пайдалану жағдайының эксклюзивті сценарийін орындау мүмкіндігін көрсетеді, бірақ осы диаграммада басқа мақсаттар көрсетілмейді.

4.4. КЛАССТЫҚ ДИАГРАММА

UML диаграммасы жүйенің ішкі құрылымын түсіндіреді, оның компоненттері (сыныптары, объектілері) арқылы қалыптасатын бағдарламалық жүйені сипаттайтын сынып схемасы, олардың арасындағы қарым-қатынастар мен өзара әрекеттер. Негізгі әдістерде осы әдістердегі сыныптық диаграммалар бетон сыныптарын құрудың ерекшеліктерін көрсету үшін жобалау кезеңінде қолданылады. UML егжей-тегжейлі дәрежеге байланысты үш деңгейлі диаграмма деңгейін қолдануды ұсынады:

- Контекстік сипаттамалы деп аталатын класс диаграммалары тақырыптың негізгі ұғымдарының арасындағы байланысты көрсететін тұжырымдамалық деңгей;
- Сынып схемаларында домендік кластардың интерфейстерін көрсететін сипаттамалар деңгейі, яғни. осы сыныптардың байланыс құралдары;
- Сынып диаграммалары нақты сыныптар мен әдістерді тікелей көрсететін іске асыру деңгейі.

Бұл үш түрлі модель, олардың арасындағы өзара түсініксіз. Сондықтан тұжырымдамалық модель доменнің кейбір тұжырымдамасын сынып ретінде анықтаса, ол бұл тұжырымдаманы іске асыру үшін жеке сынып пайдаланылатындығын білдірмейді. Дегенмен, барлық үш модельдерде біз нысандардың (сыныптардың)

түрлері мен қызығушылықтары бар олардың бірыңғай белгілерін пайдалануға мүмкіндік беретін статикалық қатынастары.

Осы үлгілердің әрқайсысы бағдарламалық жасақтама әзірлеудің нақты кезеңінде қолданылады:

- тұжырымдамалық модель - талдау кезеңінде;
- спецификация деңгейінің сыныптарының диаграммалары - жобалау сатысында;
- іске асыру деңгейіндегі сыныптардың диаграммалары - іске асыру сатысында.

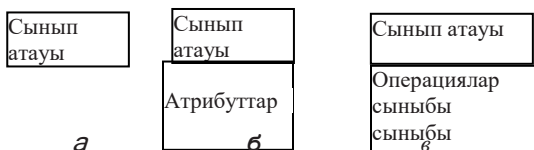
Тұжырымдамалық модельдер, анықтамаға сәйкес, пәндік доменнің концепцияларымен жұмыс істейді, осы тұжырымдаманың атрибуттары мен олардың арасындағы қатынастар. Класс дәстүрлі түрде домендегі нысан нысандарының белгілі бір тобының жалпы атрибуттарының жиынтығы ретінде түсініледі. Осы анықтамаға сәйкес сыныптың диаграммасында әрбір класс жалпы сипаттамалары сыныппен белгіленетін объектілер тобына сәйкес келеді.

Диаграммаларда класс төртбұрыш ретінде ұсынылады, оның ішінде сынып атауы көрсетілген (4.14-суретті қараңыз, а). Қажет болса, сыныптың сипаттамаларын көрсетуге болады, мысалы, символдың арнайы бөлімдерін пайдаланатын атрибуттар (4.14, b суретін қараңыз), сондай-ақ класс операциялары (4.14, c сур.).

Мәселенің шешімі тұрғысынан маңызды болып табылатын кейбір атрибуттар объектілердің сипаттамалары болып табылады, мысалы, сәйкестендіретін мәндер (атауы, нөмірі) атрибуттар ретінде. Объект үшін атрибут әрдайым нақты мәнге ие (4.15-суретті қараңыз).

Сыныптың атауы топтаманың ішіндегі бірегей болуы керек, ол сынып диаграммалары жинағымен сипатталады (мүмкін, бір диаграмма). Төртбұрыштың бірінші жоғарғы бөлігінде көрсетілген. UML элементтерін атаудың жалпы ережесінен басқа, сынып атауы атаудың бөлігінің ортасында жазықтықта жазылады және бас әріппен басталады. Атаулар бос орындарсыз практикалық себептермен жазылған сынып атаулары ретінде қолданылуы ұсынылады. Объектілі-бағытталған талдау және дизайндағы пәндік доменнің сөздігін қалыптастыратын сынып атаулары есіңізде болсын.

Сыныпты анықтаудың бірінші бөлімінде осы сынып алынатын және ол қасиеттер мен әдістерді иеленетін стандартты үлгілерге немесе дерексіз сыныптарға сілтемелер болуы мүмкін. Бұл бөлімде



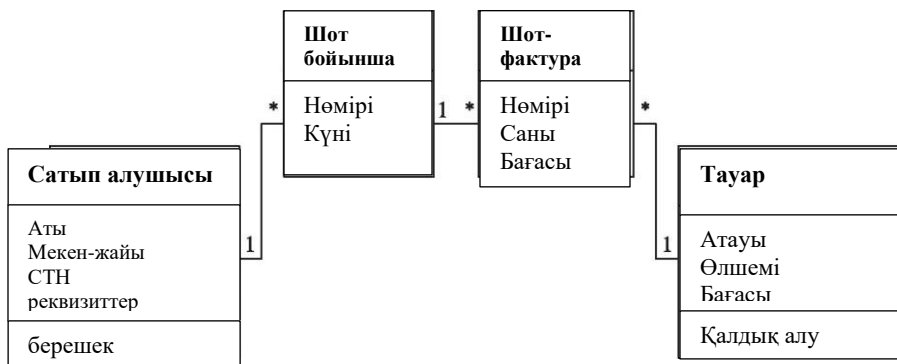
осы сыныпты әзірлеу және мәртебесі туралы ақпарат болуы мүмкін

Сурет 4.14. Класстық диаграммадағы сыныпты белгілеу: *a* - сипаттамаларын көрсетпей; *b* - атрибуттардың сипаттамасымен; *c* - операциялардың нұсқауларымен

сонымен қатар UML тілінің диаграммасының немесе стандартты элементтерінің басқа сыныптарымен байланысты осы сыныптың басқа жалпы қасиеттері жазылуы мүмкін.

Сынып атауларының мысалдары Employee, Company, Manager, Client, Prodavets, Manager, Office және басқалар сияқты тікелей аттары болуы мүмкін алаңы мен функционалдық мақсаты.

Сыныпта даналар немесе нысандар болмауы керек. Бұл жағдайда ол дерексіз класс деп аталады, ал оның атауын белгілеу үшін көлбеу қаріп (курсив) қолданылады. UML-де жалпы конвенциясы дерексіз элементке сілтеме жасайтын кез келген мәтіннің курсивпен жазылғаны болып табылады.



Сурет. 4.15. UML диаграммаларындағы нысандарды графикалық ұсынудың мысалы

Кейбір жағдайларда белгілі бір класқа қандай пакеттің тиесілі екенін нақты көрсетуіңіз керек. Осы мақсат үшін арнайы белгіш таңба пайдаланылады - қос колон «:». Бұл жағдайда сынып атауы жолының синтаксисі:

<пакет атауы>::<сынып атауы>.

Басқаша айтқанда, топтаманың атауы оның тағайындалуы керек сынып атауына дейін айқын көрсетілуі керек. Мысалы, «Банк» деп аталатын пакет анықталса, осы банктегі «Шот» класы «Банк :: Шоты» түрінде жазылуы мүмкін.

Класстың тіктөртбұрышының екінші бөлімінде оның атрибуттары (атрибууттары) немесе сипаттары жазылған. UML тілінде кейбір синтаксистік ережелерге бағынатын сыныптық төлсипат жазбаларының белгілі бір стандартизациясы қабылданды. Әрбір атрибут атрибуттың төлсипатының кванторынан, атрибуттың атауынан, оның көбейткіштерінен, атрибут мәндерінің түрі және, мүмкін, бастапқы мәнінен тұратын жеке мәтін жолына сәйкес келеді:

<көріну кванторы> <төлсипат атауы>
[көпше]:

<attribute type> = <бастапқы мән> {string-property}

Көріну калькуляторы үш ықтимал мәндердің бірін алады және арнайы таңбалар көмегімен көрсетіледі:

■ «+» Көрінетін көрінетін ауқымы бар атрибуты бар ма? Атрибут диаграмма анықталған пакеттің кез-келген басқа кластарынан қол жетімді немесе көрінетін;

■ «#» - қорғалған аумағы бар төлсипат. Атрибут қол жетімді емес немесе осы сыныптың ішкі сыныптарын қоспағанда барлық сыныптар үшін көрінбейді;

■ «-» - көрінетін көрініс көлемімен жеке төлсипат. Бұл ауқымның атрибуты қолжетімді емес немесе барлық сыныптар үшін көрінбейді.

Көрінудің квалификациясы алынып тасталуы мүмкін, бұл жағдайда оның жоқтығы атрибуттың көрінуі – дегенді білдірмейді. Бұл жағдай қарапайым бағдарламалау тілдеріндегі әдепкі конвенциялардан ерекшеленеді, егер көрінетін сандықтаушы болмаса, ол мемлекеттік немесе жеке ретінде қарастырылады. Алайда, шартты графикалық таңбалардың орнына, тиісті кілт сөзді жаза аласыз: public, protected, private.

Төлсипат атауы тиісті атрибуттың идентификаторы ретінде пайдаланылатын мәтін қатарына жатады, сондықтан сыныпта бірегей болуы керек. Төлсипат атауы атрибуттың синтаксисінің жалғыз міндетті элементі болып табылады.

Атрибуттың көптігі бір сыныпты құрайтын берілген түрдегі нақты төлсипаттардың жалпы санын сипаттайды. Тұтастай алғанда, көбінесе тиісті атрибуттың атауынан кейін тік жақшаларда мәтін сызығының түрінде жазылады. Мысал ретінде, атрибуттардың көптігін көрсету үшін келесі параметрлерді қарастырыңыз:

[0..1] атрибуттың көптігі 0 немесе 1 мәнін қабылдай алады дегенді білдіреді. Бұл жағдайда 0 бұл атрибут үшін мән болмайды;

[0..*] атрибуттың көптігі 0-ден жоғары немесе оған тең болатын кез келген оң бүтін мәнді қабылдай алады дегенді білдіреді. Бұл көп түрді жай символы ретінде қысқаруы мүмкін - [*];

[1..*] атрибуттың көптігі 1-ден артық немесе оған тең болатын кез келген оң бүтін мәнді қабылдай алады;

[1..5] атрибуттың сандары сандардан кез келген мәнді қабылдай алады: 1, 2, 3, 4, 5;

[1..3,5,7] атрибуттың көптігі сандардан кез келген мәнді қабылдай алады: 1, 2, 3, 5, 7;

[1..3,7..10] атрибуттардың көптігі сандардан кез келген мәнді қабылдай алады: 1, 2, 3, 7, 8, 9, 10;

[1..3,7..*] атрибуттың көптігі 1, 2, 3 сандарынан және кез келген оң бүтін мәннен 7 мәніне тең немесе одан асатын кез келген мәнді қабылдай алады. Егер атрибуттың көптігі көрсетілмесе, онда оның әдепкі мәні 1..1 болады; дәл 1.

Атрибут түрін - семантикасы тиісті үлгідегі ерекшеліктер тілімен анықталатын өрнек. UML белгісінде атрибут түрін кейде бұл үлгіні іске асыру үшін қолданылатын бағдарламалау тіліне байланысты анықталады. Ең қарапайым жағдайда, төлсипат түрі осы класқа жататын пакетте немесе модельде маңызды мағына беретін мәтін жолымен көрсетіледі.

Атрибуттарды көрсете отырып, екі қосымша синтаксистік

конструкцияны қолдануға болады: атрибут жолының астын сызу және бұйра жақшаларда түсіндірме мәтіні.

Төлсипат жолының асты сызығы тиісті атрибут оның түрімен анықталған төлсипат мәндерінің белгілі бір ауқымынан мәндердің ішкі жиынын алуға мүмкіндік береді. Бұл мәндер бір типті жазбалар жиынтығы немесе сыныптың әрбір нысанын бірлесіп сипаттайтын массив ретінде қарастырылуы мүмкін.

Тіктөртбұрыштың үшінші бөлігінде, операциялар немесе сынып әдістері жазылады. Операция - бұл сыныптың әрбір данасын нақты талаптармен қамтамасыз ететін қызмет. Әрекеттер жиынтығы сынып мінезінің функционалдық аспектісін сипаттайды. UML-те класс операцияларын жазу стандартталған және белгілі бір синтаксистік ережелерге бағынады.

Бұл жағдайда сыныптың әр операциясы әрекеттің көрінетін сәйкестендіргіші, әрекеттің атауы, операциямен қайтарылған мәнің түрін және операцияның сызықтық сипатынан тұратын жеке жолға сәйкес келеді:

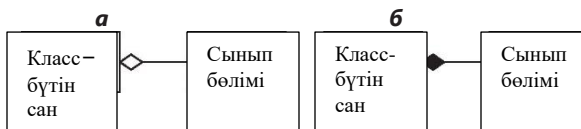
```
<көріну квалифигі> <операция атауы>  
(параметрлер тізімі):  
<өрнек мәнінің өрнегі> {string-property}
```

Көрнекілік квалификациясы, сынып атрибуттары жағдайында, үш ықтимал мәндердің біреуін қабылдап, сәйкесінше арнайы таңба арқылы көрсетіледі. «+» Белгісі ашық түрдегі жалпыға бірдей операцияны білдіреді. «#» Белгісі қорғалған түрдегі ауқымды операцияны білдіреді. Ақырында, «-» белгісі -суа түрін жеке (жеке) түрімен қолдануды белгілеу үшін пайдаланады.

Операция үшін қарау квантизаторы алынып тасталуы мүмкін. Бұл жағдайда оның жоқтығы операцияның көрінуі көрсетілмегенін білдіреді. Шартты графикалық белгілердің орнына сіз тиісті кілт сөзді жаза аласыз: қоғамдық, қорғалған, жеке. Ішкі құрылғы немесе класс құрылымына қоса, бағдарламалық жасақтама жүйесін дамытуда сыныптық диаграммада көрсетілуі мүмкін сыныптар арасындағы әр түрлі қатынастарда маңызды рөл атқарады:

- қауымдастық қарым-қатынасы - сынып арасындағы еркін қарым-қатынастың болуы;

- жалпылама қарым-қатынас - жалпы элемент пен нақты элемент (ата-ана мен бала) арасындағы қатынас,



Сурет. 4.16. Қауымдастықтың ерекше түрлерінің аңызы: а - композиция; б - агрегация

■ агрегация қатынасы (4.16, б) - сыныптардың бірі құрамдас ретінде кейбір басқа субъектілерді қамтитын кейбір субъектіні білдіреді. Класстардың диаграммасын жасау процесі күрделі жүйелер жобаларын жобалау үшін маңызды болып табылады. Оңтайлы класстарды таңдау және олардың арасындағы қарым-қатынастарды орнату қабілетінің арқасында ғана дизайн процестерінің табысы ғана емес, сондай-ақ бағдарламаның орындалуы да байланысты.

Тапсырма

Техникалық тапсырмаларды практикалық жұмыста қолдану арқылы 1 (қосымша қараңыз), бағдарламалық қамтамасыз етудің талаптарын нұсқалық нұсқаға сәйкес талдау, объектілі-бағытталған көзқарас пен UML визуалды модельдеу тілін қолдану. MS Visio-да шығарылатын нәтижелер.

БАҚЫЛАУ СҰРАҚТАРЫ

1. UML тұжырымдамасын сипаттаңыз.
2. UML-ні қолданудың артықшылықтары қандай?
3. Жүйенің мінез-құлқын сипаттайтын қандай құрылымдар?
4. UML-ге субъектілердің түрлері қандай?
5. UML диаграммаларының түрлерін келтіріңіз.
6. Қауымдастық атрибуттарының мысалдарын келтіріңіз.

Қауымдастықтың көптігі неде?

7. Қандай пәндер әдетте сынып диаграммаларын қамтиды?
8. Емтихан өткізу тәртібін модельдеу үшін әрекет диаграммасын құрыңыз.

БАҒДАРЛАМАЛЫҚ ЖАСАҚТАМА ПРОТОТИПІН ЖАСА

5.1.

ПРОТОТИПТЕР ТУРАЛЫ НЕГІЗГІ МӘЛІМЕТТЕР

Біріншіден, қандай бағдарламалық жасақтама үлгісін қолданатынымызды анықтаймыз.

Біріншіден, пайдаланушы жүйеден күтетініне қойылатын талаптарды қалыптастыруды жиі қиынға салды. Бұл жағдайда әңгімелесу нәтижелері бойынша жедел құрастырылған Пайдаланушы интерфейсі (UI) прототипі оны әзірлеуші жүйенің тиісті бөлігін қалай көрсететінін схематикалық жүзеге асыруға мүмкіндік береді.

Прототиптерді пайдаланғанда кез келген нәтиже маңызды. Егер бағдарламашы тапсырыс берушінің талаптарын дұрыс түсінетін болса, бағдарламалық жасақтама жүйесін дамыту дұрыс бағытта жүріп жатқаны анық. Пайдаланушы жүйені енгізудің ұсынылған нұсқасына толығымен қанағаттандырмаса, пайдасы оның бағдарламашы міндетін дұрыс түсінбеуін көрсетуі мүмкін.

Екіншіден, прототиптер баламалы концептуалды шешімдердің бірін таңдауға мүмкіндік береді. Кез келген техникалық проблеманы әр түрлі жолдармен шешуге болады. Бұл бағдарламалық жасақтама жүйесіне қойылатын талаптарды қалыптастыру және оның пайдаланушылық интерфейсін енгізу міндетіне қатысты.

Үшіншіден, функционалдық және функционалды емес талаптардың комбинациясы олардың жүзеге асырылу қаупі бар дегенді білдіреді. Әдетте, мұндай тәуекел оны іске асыру үшін қоршаған ортаға белгілі шектеулермен жүйенің жылдамдығына қойылатын талаптармен байланысты. Бұл жағдайда прототиптер жүйенің тиісті бөлігін енгізіп, оның деректеріне кіретін деректер ағындарын имитациялайтын және осы деректерді өңдейтін

(міндетті түрде пайдаланушы интерфейсіне байланысты емес) жасалады.

5.2

ПРОТОТИП ТҮРЛЕРІ

Прототиптер көлденең және тік, бір реттік және эволюциялық, қағаз және электрондық болуы мүмкін.

Көлденең (мінез-құлық) прототипі пайдаланушылық интерфейсін, өңдеу логикасына және деректер құрылымдарына әсер етпестен модельді. Көлденең прототиптерді көп баламалы іске асыратын айқын емес талаптарды түсіндіру қажет болғанда пайдалану керек. Көлденең прототиптер үшін жүйенің түпкілікті нұсқасы әзірленетін бағдарламалық жасақтаманы пайдалану қажет емес.

Деректер базасы деректер базасын пайдаланса, онда олар бағдарлама кодымен модельленеді; ал экранда көрсетілетін мәтіндер проблемалық аймақтың нақты ерекшеліктерін көрсетуі керек, әйтпесе пайдаланушы қиындықты шоғырландырады. Прототипті жасаған кезде есептер мен сұраулардың нәтижелері сыртқы жүйелерге сұрауларға ұқсас болады. Пайдаланушы жүйенің әрекеттеріне жауап ретінде қалай әрекет ететінін түсінуі үшін, пайдалану параметрлерін орындау кезінде экрандар арасында жылжу үшін жауап беретін осы бөлікті іске асыру жөн.

Горизонталдық прототипін құрастырмас бұрын, қандай негізгі экрандар бар екендігін анықтау қажет, олардың арасында ауысу туралы ереже сақталатын терезелер ашылады. Мұны істеу үшін әртүрлі экрандарға (терезелерге) күйлердің және кейбір интерфейс элементтерінің жабылуына және басқалардың ашылуына себеп болатын белсенді басқару элементтері бар күй диаграммасының үлгісін пайдалану жақсы.

Тік (құрылымдық) прототипі жүйенің тігінен «қисыққа» қатысты қолданушы интерфейсін жобалауға ғана емес, оны іске асырудың барлық деңгейіне әсер етеді. Осындай прототиптерді жасаған кезде, мақсатты жүйені жасау үшін сол тілдерді және іске асыру орталарын пайдалану ұсынылады. Мұндай прототиптер қолданылатындығын талдау үшін, архитектуралық

концепцияларды тексеру үшін пайдаланылады.

Бір уақыттағы (зерттеу) прототипі сіз дамыған бағдарламалық жасақтама жүйесін, оның кейбір аспектілерін және компоненттерін тез арада алу керек болған кезде жасалады. Зерттелетін прототиптерді құрудың мақсаты - жылдам қолдануды дамыту (RAD) технологиясы. Бір уақыттағы прототипі оның дамуымен тез жасалуы керек Біз кодты, сапаны, жылдамдығын, өндірілімділігін және т.б. қайталап пайдалану мәселелеріне назар аудармауымыз керек. Нәтиже - ақаулардың айтарлықтай санын қамтуы мүмкін «шикі» код. Осындай прототиптерді іске асыратын код фрагменттері мақсатты жүйенің бөлігі болмауы үшін шаралар қабылдау қажет. *Эволюциялық прототипі* жүйенің өзі болуға шақырылған жүйенің алғашқы жуықтауы ретінде пайда болады. Эволюциялық прототиптің бағдарлама коды мақсатты бағдарламаның кодын жасау керек. Сондықтан, осы үлгідегі прототиптер біржолғы қолданылатын прототиптерді жасау кезінде оларды тастау керек: мұқият әзірлеу, технологиялық әдістер мен әдістерді қолдану, нәтижелерді тестілеу және т.б.

Қағаздың прототипі - қағаздағы интерфейстердің құрылымы. Әрине, олар даму ортасындағы интерфейсті ауыстырмайды. Дегенмен, барлық кемшіліктер үшін мұндай прототиптердің екі маңызды артықшылығы бар. Тапсырыс беруші түсті шешімдерге назар аудармайды, түймелердің нысаны және функционалдылық талдауларынан шатастырмайды.

UI классының прототиптерінің электронды және қағаз нұсқалары арасындағы аралық шешім электрондық кеңсе құралдарының көмегімен жасалған (мысалы, Microsoft Visio және Microsoft PowerPoint комбинациясы) презентация.

Прецеденттердің суреттелген сценарийлерін ұсынайық. Мәселелік аймақтың ерекшеліктерін жақсы түсіну және оны пайдаланушы интерфейсінде көрсету үшін, пайдалану жағдайындағы сценарий сипаттамасының мәтіні проблемалық аймақтың белгілі бір ерекшеліктерін анықтайтын ақпаратты қамтиды.

Осылайша, пайдаланылған деректердің саны туралы ақпарат жүйенің жұмысына әсер етуі мүмкін деректерді өңдеуде пайдаланушылық интерфейсін оңтайландыруға және жобаның ертерек сатыларын бағалауға мүмкіндік береді. Мысалы, жүйе тілқатысу терезесінде мәндер нұсқаларының аз санынан таңдасаңыз, индикаторлар (күсбелгі) немесе радиобайланыт (лар) жақсы. Таңдау 2-3 пунктпен шектеліп, ашылмалы тізім ыңғайлы. Жүздеген немесе мыңдаған нұсқалардан таңдауға тура келген

жағдайда сүзгілеу және іздеу үшін қосымша терезе қажет болады.

Сондай-ақ, жүйені дамыта отырып, белгілі бір операцияларды жүргізу қарқындылығы туралы ақпарат пайдалы болады. Бұл ақпарат пайдаланушыға ыңғайлы интерфейсті дамытуға мүмкіндік береді, әртүрлі операциялар кезінде әрекеттерді азайту, соның ішінде ақпарат беруді құрылымдау, негізгі интерфейстерден сирек қолданылатын опцияларды жою және т.б. сияқты жүйенің жоғары жылдамдығы.

Біз «Көтерме қойма» автоматтандырылған ақпараттық жүйесінің «Тапсырыс» жағдайын іске асыру үшін көлденең (мінез-құлық) бағдарламалық қамтамасыз ету үлгісін саламыз.

«Тапсырыс» пайдалану жағдайының сипаттамасы.

Прецедентті орындау барысында Тапсырыс беруші менеджер тапсырыс берушінің клиент базасынан (10 000 клиентке дейін) таңдайды немесе жаңа клиентті тіркейді (5% жағдайда), каталогтан тауарлардың элементтерін анықтайды (тауарлар бірнеше түрге бөлінеді, тауарлардың әрбір түрінің элементтерінің саны 100-ден аспайды) және олардың санын көрсетеді (орташа сатып алу - 8 позиция).

Жүйе экранда тауарлардың атаулары, бағасы, саны мен саны қоймада көрсетіледі. Менеджер жеңілдік береді және төлеу тәртібін анықтайды (қазіргі уақытта, ол үш нұсқасы бар төлем сомасы). Жүйе жалпы соманы есептеп береді. Тапсырысты басып шығаруға болады.

«Тапсырыс» пайдалану жағдайының прототиіпін әзірлеу.

Көлденең прототиіптер үшін жүйенің түпкілікті нұсқасы жасалатын бағдарламалық жасақтама енгізу ортасын пайдалану қажет емес. Прототиіптің ыңғайлылығы мен жылдам дамуы үшін «ІС» технологиялық платформасының құралдарын пайдалануға болады. Төменде тапсырыс орындалатын терезе (5.1-сурет). Тұтынушыны (клиентті) таңдау үшін, активтендіру кезінде экранда клиенттік тізімнің нысаны пайда болатын түйме бар. Төлем түрі тиісті түймешік іске қосылған кезде ашылмалы тізімнен таңдалады.

Пішін клиентке төленетін жалпы сома туралы ақпаратты көрсетеді.

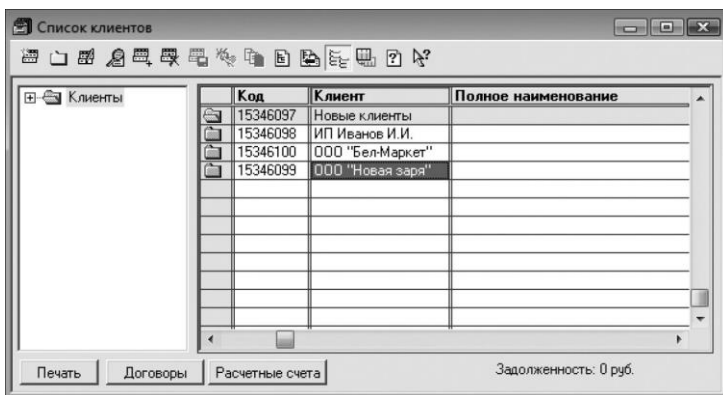
Клиенттердің саны өте үлкен болғандықтан, тізімнен клиенттің таңдауы клиентті тиімді іздеу немесе сүзуді жүзеге асыруға болатын жеке терезеде болуы тиіс (5.2-сурет, а). Терезенің төменгі бөлігінде тапсырыс берушімен өзара есептесу мәртебесі туралы ақпарат көрсетіледі. Сол терезеде клиенттік каталогқа жаңа элемент қосуға болады, яғни жаңа клиентті өңдеу және тіркеу үшін терезені шақырады.

Клиент каталогында көптеген деректемелер бар, каталогтың нысаны каталогтың барлық мәліметтерін көрнекі

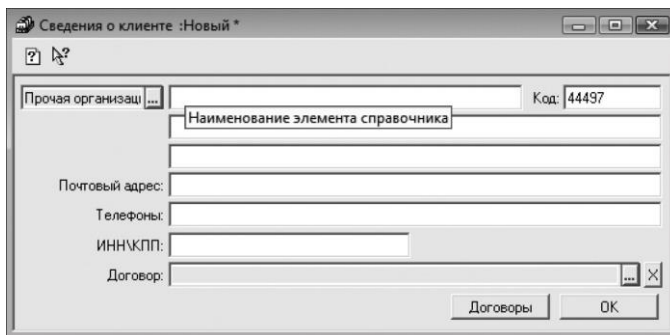
ұсынудың графикасы ретінде көрсетпейтін етіп салынған, әйтпесе кесте тым кең болады; сондықтан клиенттер туралы ақпаратты енгізу және өңдеу бөлек терезеде жүзеге асырылады (5.2-сурет, б).

Тауарларды жинау формасы кестедегі атау өрісіне екі рет басу немесе оған жаңа жол қосу арқылы шақырылады. Иерархиялық тізімнен тауарлар санатын таңдап, тиісті санаттағы тауарлардың тізімін алуға болады. Өнім түріне қарай сүзуге болады. Тауар түрлерінің саны аз болғандықтан, оны төменгі тізімге енгізу арқылы таңдауға болады (5.2-сурет, с). Пішін құжатты жасау кезінде қоймадағы тауардың болуы туралы ақпаратты көрсетеді.

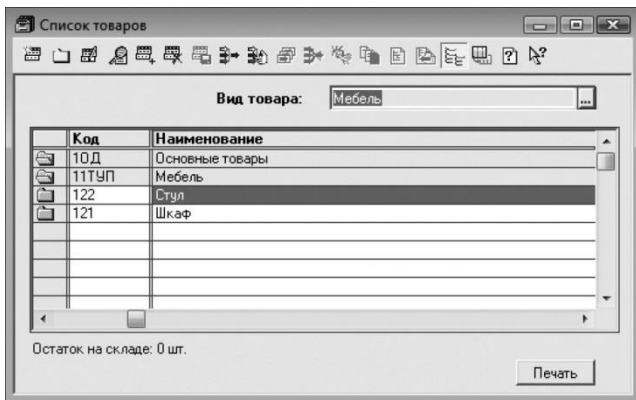
Қажетті клиенттің тізімінен тауып алыңыз немесе тауар аты немесе кодының бірінші таңбаларында болуы мүмкін, яғни, нысандарда «ыстық» іздеуді қолданамыз.



а



б



6

Сурет. 5.2. Экран нысандары:

Прототипте жасалған графикалық интерфейс стандартты графикалық интерфейстің барлық талаптарын қанағаттандырады - түстердің шектеулі жиынтығы пайдаланылады, олардың дұрыс комбинациясы қолданылады, жылдам кнопкалардың құралдар тақтасы бар. таза, басқару элементінің фокустық коммутациялық тізбегі қарастырылады. Негізгі командаларға қол жеткізу үшін қысқа пернелерді пайдаланыңыз. Жылдам түймешіктер мен басқа компоненттерге тінтуір меңзерін жылжытқанда, қалқымалы терезелер бар. Көмек файлын шақыруға болады.

Тапсырма

Тапсырмалардың ұсынылған нұсқасынан пайдалану жағдайларының сипаттамасына сәйкес, бағдарламалық жасақтама үлгілерін құру. Тұтынушыны прототипін көрсету үшін Microsoft PowerPoint бағдарламасында көрсетілім жасаңыз.

БАҚЫЛАУ СҰРАҚТАРЫ

1. Прототиптерді құрудың мақсаты қандай?
2. Прототиптердің түрлерін көрсетіңіз.
3. Қағаздың прототиптерінің артықшылықтары қандай?
4. Қандай жағдайларда зерттеу прототипі құрылады?
5. Неге бейнеленген жағдай сценарийлері қолданылады??

Пайдаланушы интерфейсін жобалау

6.1. Интерфейсті құрудың негізгі ережелері

Сарапшылар әдетте интерфейсін ыңғайлылығын бағалауға және оның ыңғайлылығын арттыратын шешімдерді ұсынуға мүмкіндік беретін принциптер мен ережелер жиынтығын қалыптастырады. Біз осы ережелерді береміз.

Қол жетімділік ережесі. Жүйе бұрын-соңды ешқашан көрмеген, бірақ тақырыптық салада жақсы білетін пайдаланушы ешқандай жаттығусыз оны пайдалана бастауы соншалықты айқын болуы керек. Бұл ереже нақты ұмтылыс ретінде қызмет етеді, ол ұмтылуға тиіс, өйткені іс жүзінде мұндай қолжетімділікке жету мүмкін емес. Дегенмен, мамандар бұл бағытта барлығын жасайды.

Күшіне ену ережесі. Жүйе ұзақ уақыт бойы жұмыс істейтін тәжірибелі пайдаланушылардың тиімді жұмысына кедергі болмауы керек. Бұл ережені бұзудың айқын мысалы жүйенің жаңадан бастағандарға ғана назар аударуы, тәжірибелі пайдаланушыға ыңғайлы құралдарды пайдалану, оны дұрыс емес нәрсе жасау мүмкіндігінде шектеу, бірақ қалай және қайда екенін білетін сарапшы үшін тиімсіз болып табылады. мұны істеу керек.

Үздіксіз даму ережесі.

Жүйе пайдаланушының білімін, дағдыларын және әдеттерін үздіксіз арттыра алады және оның өзгерген тәжірибесіне бейімделуге тиіс. Нашар нәтижелер тек негізгі мүмкіндіктерді қамтамасыз етеді немесе жаңа пайдаланушыны тек сарапшылар сенімді пайдаланатын күрделі интерфейспен қалдырады. Мүмкіндіктердің бір жиынтығынан екіншісіне көшудегі үздіксіздіктің бұзылуы да қолайсыздықты тудырады, өйткені пайдаланушы жаңа контексте қосымша мүмкіндіктермен айналысу керек.

Көптеген пайдаланушыларды үш топқа орналастыруға болады: жаңадан бастаған, тәжірибелі және орташа, жаңадан көп нәрсені білетін және көптеген қателіктер жасамайтын, бірақ көптеген операцияларда автоматизма алған емес және кейде интерфейсінде шатастыруға болады. Жаңа бастағандар оларға жаңа жүйені игеруге және олардың іс-әрекеттеріне, тәжірибелі қолданушыларға - жиі талап етілетін іс-әрекеттердің жоғары тиімділігіне және жиі кездесетін жағдайларда икемді жүйені басқару мүмкіндігіне көмектесу керек, бірақ олардың жеткіліксіз қолдауымен проблемаларды тудыруы мүмкін. Орташа пайдаланушылар туралы жиі ұмытылады, бірақ бағдарламалық жасақтаманы пайдаланушылардың басым көпшілігі осы санатқа жатады. Олар үнемі тиімділік пен икемділікті қажет етеді, сонымен қатар кейде әртүрлі сұрақтар туындайды.

Мәтінмәндік ережелердің сақталуы. Жүйе жұмыс істейтін мәнмәтінге сәйкес болуы керек. Бұл ереже жүйенің «жалпы» емес, ол қолданылатын ортада тиімді болуын талап етеді. Контексте кіріс және шығыс деректерінің ерекшеліктері мен көлемі, жүйе жұмыс істейтін ұйымдардың түрі мен мақсаты, пайдаланушылардың деңгейі, үй-жайдың шуы.

6.2. ПАЙДАЛАНУШЫ ИНТЕРФЕЙСІН ҚҰРУ ПРИНЦИПТЕР

Жоғарыда айтылған ережелер пайдаланушыға ыңғайлы интерфейсін қанағаттандыратын жалпы талаптарды анықтайды. Төмендегі принциптер пайдаланушы интерфейсін ыңғайлылығын арттыратын шешімдерді табуға мүмкіндік береді.

Құрылымдау принципі. Пайдаланушы интерфейсін дұрыс құрылымдалған болуы керек. Мағынасы жағынан жақын, онымен байланысты бөліктер көрінетін түрде және тәуелсіз - бөлінген болуы керек; ұқсас элементтер ұқсас болуы керек, ал айырмашылығы - әртүрлі.

Қарапайымдық принципі. Ең көп тараған операцияларды барынша қарапайым түрде жүргізу керек. Күрделі рәсімдерге көрінетін байланыстар болуы керек.

Көріну принципі. Белгілі бір тапсырманы шешу үшін қажетті барлық функциялар мен деректер пайдаланушы оны шешуге тырысқан кезде көрінуі керек.

Кері байланыс қағидасы. Пайдаланушы жүйенің іс-әрекеттері туралы және оның ішіндегі маңызды оқиғалар туралы хабарлар алу керек. Хабарламалар ақпараттылығы, қысқаша, біржақты болуы және пайдаланушы түсінген тілде жазылуы тиіс. Принцип толерантности.

Интерфейс пайдаланушылық қателіктеріне икемді және төзімді болуы керек. Қателерден келтірілген шығындарды жою және қайталау мүмкіндігімен байланысты және пайдаланушының және оның енгізген деректерінің негізделген әрекеттерінің орынды түсіндірілуіне байланысты азайтылуы керек. Мүмкіндігінше, пайдаланушының еркіндігін шектеуге негізделген мағыналы өзара әрекеттесуден (модальный диалогтардан) аулақ болыңыз.

Қайта қолдану қағидасы. Ішкі және сыртқы компоненттерді бірнеше рет қолдануға тырысыңыз, осылайша интерфейс пен оның ұқсас элементтері арасындағы ұқсастығын қамтамасыз етіңіз. Эргономикалық пайдаланушы интерфейсін жасаудың кейбір мәселелерін қарастырайық.

6.3. ПАЙДАЛАНУШЫ ЖӘНЕ КОМПЬЮТЕР АРАСЫНДАҒЫ БАЙЛАНЫС

Адамның машиналық интерфейсі пайдаланушы мен компьютер арасындағы қарым-қатынасты қамтамасыз етеді, ол алдына қойылған мақсаттарға қол жеткізуге, тапсырманы шеше табуға мүмкіндік береді. Өзара қарым-қатынас - компьютер мен пайдаланушы арасында осы әрекеттерге әрекеттер мен реакцияларды алмастыру. Екі негізгі түрге бөлінетін өзара әрекеттесудің бірнеше стильдері бар.

Пәрмен тілінің интерфейсін пайдалану - мәтіндік құралдарды пайдаланатын пәрмендерді енгізу.

1. Тікелей манипуляция. Осылайша, пайдаланушыға компьютермен байланыс орнатудың бірнеше жолы бар:

■ пәрмен тілдері - пайдаланушы мәтіндік режимде тиісті пәрмендерді енгізу арқылы жүйені басқарады;

■ сұрақ-жауап - компьютер сұрақ тудыратын диалог және пайдаланушы жауап береді (немесе керісінше);

■ нысандар - пайдаланушы тиісті өрістерге деректерді енгізу арқылы диалогтың формаларын немесе өрістерін толтырады;

■ мәзір - пайдаланушыға көптеген опциялар мен басқару элементтері беріледі

■ қажетті элементтерді таңдау арқылы жүйе; тікелей манипуляция - пайдаланушы экрандағы объектілерді тінтуірді түрдегі манипулятор арқылы басқарады.

Бағдарламалық қамтамасыз етудің дамыған жүйесінде интерфейссті құрудың кешенді тәсілі де қолданылады. Ол тікелей манипуляцияны, мәзірлерді, формаларды және диалогтарды пайдаланады.

Эргономикалық интерфейссті құрудың мақсаты адамның қабылдауына мүмкіндігінше тиімді ақпаратты көрсету және экрандағы дисплейді ең маңызды ақпаратқа назар аударту үшін жасайды.

Негізгі мақсат - экрандағы жалпы ақпаратты азайту және тек пайдаланушы үшін не қажет екенін көрсету.

6.4.

КРАНҒА МӘЛІМЕТТЕРДІ ОРНАЛАСТЫРУ

Экранда көрсетілген ақпарат көлемі экранның тығыздығы деп аталады. Зерттеулер көрсеткендей, экранның тығыздығы, көрсетілетін ақпарат пайдаланушы үшін ең қолжетімді және түсінікті, керісінше, егер экранның тығыздығы үлкен болса, бұл ақпаратты игерудегі қиындықтар мен оның анық түсінуі мүмкін. Дегенмен, тәжірибелі пайдаланушылар үлкен экранды тығыздығы бар интерфейсстерді таңдауы мүмкін. Экрандағы ақпарат топтастырылып, елеулі бөліктерде орналасуы мүмкін. Бұған кадрларды (кадрларды), түсті кодтау, кадрларды, теріс кескіндерді немесе назар аударту үшін басқа әдістерді қолдану арқылы қол жеткізуге болады.

Интерфейс элементтерін жарықтық арқылы таңдау. Интерфейсстің кейбір элементтеріне назар аудару үшін сіз осы

элементтердің басқа, қараңғы түстердің фоны бойынша жарықтығын кеңейтуді пайдалана аласыз. Дегенмен, бұл әдісті асықтырудың қажеті жоқ, өйткені

- қызыл дегеніміз - ыстық, көк - суық және т.б.

мәртебесі көрсетілсе, әдетте қызыл дегеніміз қауіпті (тоқтату) білдіреді, жасыл жұмысдың жалғасуын білдіреді, сары ескерту түсін көрсетеді; жарқын элементтердің көп саны пайдаланушыға қолайсыздықты тудыруы мүмкін және қарама-қарсы әсерді - интерфейстің жүктемесін алуға болады. Бұл әдісті қолданыңыз

қажеттілікке ғана байланысты. Жарықтықты бөлектеудің бірнеше жолы бар:

- қозғалысы (жыпылықтайтын немесе өзгертін позиция) - өте тиімді әдіс, өйткені көздің қозғалатын элементтердің арнайы детекторы бар;

- Жарықтық - бұл өте тиімді әдіс емес, өйткені адамдар жарықтықтың бірнеше деңгейлерін ғана ажыратады;

- түсті пайдалану өте тиімді болуы мүмкін;

- пішін (таңба, шрифт, символдың нысаны) деректердің әртүрлі

санаттарын саралау үшін пайдаланылады;

- әртүрлі алфавиттерді (қаріптерді) түрлі формаларда пайдалану;

- мөлшері (мәтін, рәміздер) - әдетте таңдалған нысанды 1,5 есеге

- көбейтеді;

- көлеңкелеу (объектілердің әртүрлі құрылымы) - экранның кез келген бөлігіне назар аударудың тиімді әдісі;

- Қоршаған орта (астын сызу, рамкалар, түрлендірілген кескін). Егер сіз

оны асыра алмасаңыз өте тиімді тәсілі.

Эргономикалық интерфейс ті жобалау кезінде түсті пайдаланыңыз. Түс интерфейс ті жақсартады, бірақ көптеген жүйелер үшін түстің қолданысы пайдаланушының тиімділігіне әсер етпейді. Түстің басты мақсаты - пайдаланушылар үшін қызықты интерфейс терді құру; Алайда түс интерфейс тің дизайнеріне көмектесе алатын жағдайлар бар. Бұл әсіресе ақпараттық ақпаратты немесе қарапайым хабарламаларды (қателер, жағдайлар және т.б.) ажырату кезінде тиімді.

Түс - қуатты көрнекі құрал, сондықтан ол пайдаланушыға қате түс комбинацияларынан ыңғайсыздық туғызбауы үшін өте мұқият қолданылуы керек. Эргономикалық интерфейс ті жобалау кезінде басшылықты қажет ететін түсті пайдаланудың кейбір

қағидаларын келтірейік:

- Экрандағы түстер санын 4-ке дейін және экранның реті үшін 7-ге дейін шектеңіз;

- белсенді емес элементтер үшін бозғылт түстерді пайдалану жақсы;

- Түс кодтау туралы ақпарат болса, пайдаланушының кодты дұрыс түсінетініне көз жеткізіңіз; мысалы, төленбеген вексельдер қызыл түспен белгіленеді;

- түс пайдаланған кезде пайдаланушының көзқарасын ескеру маңызды; мысалы, картограф үшін жасыл - орман, сары - шөл, көк - су. Химикке

- ақ, сары және қызыл түстер назар аудару үшін ең тиімді;

- Егер сіз деректерді ұйымдастырғыңыз келсе, сізге 7 түсті (радуга) спектрін пайдалануға болады;

- Егер деректерді бөлектеу қажет болса, түстер спектрдің әртүрлі бөліктерінен таңдалады: қызыл - жасыл, көк - сары, кез-келген түсті - ақ;

- Деректерді топтау, біріктіру және ұқсастық үшін спектрдің көршілес

түстерін пайдалану қажет: қызғылт-сары, көк-күлгін.

Адамдардың шамамен 9% -ы түстерді (әдетте қызыл-жасыл комбинацияларды) ажыратпайды; Дегенмен, бұл адамдар қара-ақ түстерді айыра алады, сондықтан автоматтандырылған жүйелердің дизайнерлері осы санаттағы пайдаланушылардың қабылдау интерфейстерінде әртүрлі түстерді пайдаланудың бұзылғандығын тексеруі керек.

Сәйкестік және стандарттау. Экрандағы деректер пайдаланушыға қажетті ақпаратты қайдан табуға және күтуге болатындығын біледі:

- дереу байқалуы керек ақпарат пайдаланушының назарын аудару үшін әрдайым көрнекі жерде көрсетілуі керек (мысалы, ескерту хабарламалары мен қате туралы хабарлар);

- сұралатын ақпарат жиі емес (мысалы, анықтама) көрсетілмейді, бірақ қажет болған жағдайда қол жетімді болуы керек. Мысалы, әрбір экранда «анықтама» белгішесі немесе тиісті мәзір опциясы болуы керек.

Мәтіндер мен диалогтар. Мәтін тілқатысу терезелерін және салыстырулар жасау кезінде келесі ұстанымдар қажет:

- төменгі мәтіннің мәтіні толығымен бас әріппен басылған мәтіннен

13% жылдамырақ оқылады;

■ бас әріптер назар аударатын ақпаратты беру үшін ең тиімді болып табылады. Кез келген ақпаратты алуды қаламасаңыз, бас әріптерді пайдаланбаңыз;

■ оң жаққа тураланған мәтін дұрыс емес маржамен тең бөлінген мәтіннен оқуға қиын;

■ жолдар арасындағы оңтайлы интервал кейіпкерлердің биіктігіне қарағанда біршама немесе көп.

Мәзір. Автоматтандырылған жүйенің қажетті элементі - пайдаланушыға қолданбадағы тапсырмаларды орындауға және шешім процесін басқаруға мүмкіндік беретін мәзір. Мәзір - экранда көрсетілетін опциялар жиынтығы, онда пайдаланушылар әрекеттерді таңдай алады және іске асыра алады, осылайша интерфейстің күйін өзгертеді. Мәзірдің артықшылығы, пайдаланушыларға элементтің атын немесе орындағысы келетін әрекетті есте сақтаудың қажеті жоқ, олар тек оны мәзір элементтері арасында тану қажет. Осылайша, тіпті тәжірибесіз пайдаланушы да мәзірді пайдалана алады. Дегенмен, мәзір мәзірін мұқият ойластыру керек - мәзірді тиімді ету үшін, оның элементтерінің атаулары анық болуы керек.

Мәзір көптеген экран кеңістігін қабылдауы мүмкін, бірақ бұл мәселені шешуге белгішені, мәзір жолағын немесе басқа нысанды басу арқылы шақырылатын қалқымалы немесе ашылмалы мәзірді қолдануға болады.

Мәзірлерді құрудың негізгі принциптері. Мәзірдің мәзір жүйесін жобалау барысында мәзірді көрсетудің ең жақсы тәсілі оны анық және оңай пайдалану болып табылады. Әдетте, мәзір пәрмендері иерархиялық түрде сұрыпталады. Басты мәселе әр түрлі мәзір элементтерін әр түрлі деңгейлерде дұрыс таратып, дұрыс топтау керек.

Мәзірді жобалау принциптері:

■ мәзір құрылымы жүйе шешетін мәселенің құрылымына сәйкес келуі

керек;

■ мәзірді ұйымдастыру тапсырманы шешуге әкелетін қадамдардың ең

тиімді ретін көрсетеді;

■ мәзір элементтері қысқа, грамматикалық тұрғыдан дұрыс және тақырыпқа сәйкес болуы керек. Мәзір элементтерінің тәртібі келісімге, жиілікте және пайдалану тәртібіне, сондай-ақ тапсырманың немесе пайдаланушының қажеттілігіне байланысты таңдалады;

- мәзір элементтерін таңдау бірнеше жолмен ұсынылуы керек - пернетақтаны пайдаланып, тінтуірді және пайдаланушы интерфейсінің басқа нысандарын пайдалану. Мәзір элементтеріне тезірек қатынасу үшін есте сақтаудың оңай естілуін жазу маңызды, себебі бұл уақытты үнемдейді.

Формалар. Пішіндер интерфейстің негізгі элементі болып табылады. Кездесу формалары - автоматтандырылған жүйенің деректерін, күйін, хабарламаларын ыңғайлы енгізу және қарау.

Бланкілердің негізгі принциптері:

- пішін тапсырманы ыңғайлы, түсінікті және тез шешуге арналған. Пішін қағаз формадан көшірілсе, көршілес өрістер бойымен қозғалу пайдаланушыға қиындық тудырмауы керек;

- ақпараттық бөлімшелерді форма кеңістігіне орналастыру оның болашақ пайдалану логикасына сәйкес келуі керек: ол ақпараттық бірліктерге қолжетімділік дәйектілігіне, оларды пайдалану жиілігіне, сондай-ақ элементтердің салыстырмалы маңыздылығына байланысты болуы керек;

- нысанның ақпараттық элементтері арасында теңгерімді және симметрияны құру үшін толтырылмаған орынды пайдаланып,

- элементтердің логикалық топтары файлдар, сызықтар, түс немесе басқа көрнекі құралдармен бөлінуі керек;

- тәуелді немесе байланысты элементтер бір нысанда көрсетілуі керек;

- тәуелді немесе байланысты элементтер бірдей түрде көрсетілуі керек.

Пішіндерді құрастырған кезде, жүйе мәзір жолағындағы түймешіктерді белгілі бір терезеде қол жетімді болатынын ойластырып, терезені терезе тақырыбы болуы керек мөлшерін өзгертуге мүмкіндік беру керек. Арнайы қажеттіліксіз, өлшемді өлшемді терезелерді жасаудың қажеті жоқ. Өлшемін өзгерту кезінде арнайы алғышарттар болмаса, терезенің орналасуы үзіледі, бұл пайдаланушы жұмысына теріс әсер етеді. Пайдаланушыда орналасқан дисплей компоненттерінің пайдалы аумағын өзгертуге және ақпаратты өңдеуге: мәтінді, кескінді, тізімдерді және т.б. өзгертуге мүмкіндік беретін болса, өлшемін өзгертуге болатын терезе құру маңызды. Егер айнымалы терезе жобаланған болса, терезенің құрамдас бөліктері өздерінің өлшемін немесе орналасуын өзгертіп, терезе аймағына біркелкі бөлініп, бос орын қалдырмайтынды шаралар қабылдау қажет.

Пішіндерді құрастырған кезде, түстердің шектеулі жиынтығын қолдануға тырысып, олардың дұрыс комбинациясына

назар аударыңыз. Нейтралды түстердің пішінінің фоны үшін (ашық-сұр) таңдалады. Түсті ақпарат берудің негізгі құралдарының сапасында қолдануға болмайды, пайдаланушы өз қалауы бойынша қайта ұйымдастыруға болатын жүйелік түстерді таңдау қажет.

Бақылау элементтері және экранның функционалды түрде қосылған құрамдас бөліктері тақырыптарға қысқаша және нақты мақсатын түсіндіретін топтарға біріктірілуі керек. Әрбір терезеде оның құрамына бағынатын кейбір негізгі тақырып болуы керек. Пайдаланушы осы терезенің не үшін қажет екенін және ең маңыздысы не екенін түсінуі керек. Терезені кіріс және дисплей басқару элементтерінің көп мөлшерімен жүктеуге жол берілмейді. Әр түрлі терезелердегі ұқсас басқару функциялары әртүрлі деп аталады немесе терезелердің әр түрлі орындарына орналастырылады. Жүйенің жұмыс кеңістігі мен оның басқа қосымшалармен өзара іс-қимыл жасайтындығына қалай қолданылатындығын білу маңызды.

Тақырыптар мен өрістерді жобалау. Жеке өрістер үшін тақырыбы солға тураланған болуы керек; тізімдердің өрістері үшін тақырып жоғары және сол жағы сол жағында болуы керек; сандық өрістер оң жақ өріске сәйкес келеді. Ұзын бағаналар немесе жалғыз өрістерге ие ақпараттық бірліктердің ұзын бағандары бос жолмен бөлінген бес элементке топтастырылуы керек - бұл пайдаланушыға таңдалған топтардағы ақпаратты ақылмен өңдеуге көмектеседі.

Ақпараттың үлкен көлемі бар нысандарда олардың иелігіндегі ақпараттың сипатын нақты көрсететін бөлімдердің атауларын пайдалану маңызды. Тақырыптар мен көрсетілімдердің тікелей өрістерін нақты бөлу қажет. Тақырыптар қысқа, таныс және маңызды болуы керек.

Міндетті емес немесе маңызды емес өрістер маңызды және толтырылуы қажет өрістерден көзбен (түс немесе басқа әсерлер) ерекшеленуі керек.

Кіріс форматтары. Әдепкі мәндердің барлық өрістерге енгізілуін қамтамасыз ету керек, бұл мүмкіндік береді және мұндай функция пайдаланушыға тітіркендірмейді. Жиі қайталанатын мәндерді енгізу үшін кілттерді немесе кодтарды тағайындауға болады. Енгізу деректері маңызды және жалпы қабылданған болуы керек. Сандарды және таңбаларды енгізу үшін өрістерді біріктірмеңіз, себебі сандық және әріптік пернелер пернетақтада бір-біріне ыңғайсыз.

Деректерді енгізуді жылдамдату үшін жоғарғы және төменгі тіркеушілер арасында жиі ауысуды болдырмау керек. Сіз

пайдаланушыдан шамалы нөмірді енгізуді сұрамайсыз (мысалы, 00000010-ға қосымша, пайдаланушы тек 10-ға кіруі керек). Сол сияқты, пайдаланушыдан бұрын енгізілген немесе жүйеден автоматты түрде алынатын ақпаратты енгізуді талап ете алмайсыз. Ақпаратты енгізу үдерісін азайту үшін әдепкі мәндерді пайдалану қажет.

Навигация жүйесін ұйымдастыру және жай-күйін көрсету жүйесі Навигация пайдаланушыға автоматтандырылған жүйеде әртүрлі экрандар, ақпараттық бірліктер мен кіші бағдарламалар арасында шарлауға мүмкіндік береді. Толыққанды жүйеде пайдаланушы әрқашан жүйенің күйі, орындалу процесі немесе белсенді кіші бағдарлама туралы ақпаратты ала алады.

Жобалаудың жалпы принциптері.

Пайдаланушыға жүйеге бағдар беруге көмектесетін бірнеше навигациялық құралдар мен әдістер бар. Олар экранның жоғарғы жағындағы ағымдағы файл атауын көрсететін әр экранға, бет нөміріне, жолдар мен бағандарға арналған бет тақырыптарын пайдалануды қамтиды.

Навигация жүйесінің түрі интерфейснің қабылданған стиліне байланысты. Командалық тіл интерфейстері үшін толық навигацияны қамтамасыз етудің бірнеше жолы бар. Мәзірмен интерфейстерде иерархиялық құрылымдалған мәзірді пайдалануға болады. Диалогтық интерфейстер пайдаланушыны қате әрекеттерден қорғайды. Күй туралы ақпарат әдетте экранның төменгі жағында көрсетіледі және сақтау көлеміне қатысты деректерді қамтиды.

Хабарларды жобалау. Мәселені шешу жолында қажетті әрекеттерді орындау кезінде пайдаланушының әрекетін дұрыс бағытта, нұсқаулықта және ескертулерде бағыттау үшін хабарлар қажет. Олар сондай-ақ пайдаланушының әрекеттерін растауды және жүйенің тапсырмалары сәтті аяқталғанын немесе қандай да бір себептермен орындалмағанын растауды қамтиды. Хабарламалар диалог, экран сақтағыштар және т.б. түрінде көрсетілуі мүмкін. Хабарламалар пайдаланушыға ұсынылуы мүмкін:

- ұсынылған баламалардан опцияны немесе опциялар жиынтығын таңдау; мәлімет енгізу;

- ағымдағы контекске байланысты өзгеруі мүмкін параметрлер жиынтығынан опцияны таңдаңыз;

- Енгізуді жалғастырмас бұрын енгізілген ақпараттың фрагментін растаңыз. Хабарламалар кез-келген басқа әрекет басталмас бұрын пайдаланушыны сұраққа жауап беруге

мәжбүрлейтін қалыпты диалогтық терезелерге орналастырылуы мүмкін. Бұл жүйе пайдаланушыны жалғастыру алдында шешім туралы ойлануға мәжбүрлеген кезде пайдалы болуы мүмкін. Модальды емес тілқатысу терезелері интерфейстің басқа элементтерімен жұмыс істеуге мүмкіндік береді, ал терезенің өзі еленбеуі мүмкін.

6.5. ҚАТЕЛЕРДІ БОЛДЫРМАУ, АНЫҚТАУ ЖӘНЕ ТҮЗЕТУ

Кәдімгі мемлекеттегі кәдімгі адам бірнеше түрлі қателіктер жасайды. Компьютерден айырмашылығы адам бейімделгіш аналогтық жүйе болып табылады және оның «жұмыс істеуінің» өзектілігі, ойдың әрекеттері мен тұжырымдарының дәлдігімен емес, тез арада қажетті нәтижеге жақсы жақындатылып, тез арада жеткілікті түрде анықталуымен анықталады деп айтуға болады қажеттілік туындайды.

Пайдаланушы қателері әрекетті немесе процедураны дұрыс түсінбеуге негізделуі немесе теру кезінде теру сияқты кездейсоқ емес кездейсоқ болуы мүмкін.

Екінші түрдегі қателерді алты кіші түрге бөлуге болады:

- опцияны таңдау кезінде қате (мысалы, пайдаланушы кездейсоқ «Шығу» түймесін басып, бағдарлама жабылды);

- деректерді басқарудағы қателер (мысалы, соңғысының дұрыс емес көрсетілуіне байланысты қате файл атауын беру);

- ассоциативті сипаттағы қателер (мысалы, кез келген тұлғаның аты бар файлды сақтау, оны пайдаланушы үнемдеу кезінде ол ойлаған);

- пайдаланушы жұмысты жалғастыру үшін қажетті әрекеттер дәйектемесін ұмытқан кезде қызметті жоғалту;

- пайдаланушы бір күйде екенін ойлағанда, режим немесе күй қателігі, бірақ шын мәнінде басқа; мысалы, сөз процессорындағы мәтіннің жоғарғы жағында басып шығару режимінің орнына кірістіру режимі.

Пайдаланушы бағдарламалық жасақтама жүйесінде тіпті қателіктер жібереді, сондықтан дамыған жүйе әрқашан қателерді қорғауды қамтамасыз етуі керек. Осындай қорғаудың әдістемесі келесі аспектілерді қамтиды:

- қателіктердің пайда болуын болдырмауға немесе кедергі

келтіретін жүйеде міндетті әрекеттер;

- пайдаланушылардың өздерінің қателерін түзетуге мүмкіндік беретін кері әрекеттер қолдануы;

- жүйенің қалыпты диагностикасын қамтамасыз ету, оның барысында

пайдаланушыға қатенің қандай екендігі және оны түзету әдістері көрсетілген.

Кіріс формаларында қателерді өңдеудің негізгі принциптерін қарастырайық:

- енгізу қателіктерін (typos) түзету үшін енгізілген жазбаларды символ-ретті өңдеу мүмкіндігін қамтамасыз ету;

- қате жүйе арқылы анықталса, курсорды қате деректермен өріске қайтару керек, ал қандай да бір жолмен бұл өрісті визуалды түрде таңдаңыз;

- пайдаланушының тілінің мәнерін және сәйкес терминологияны пайдаланатын мағыналы қате туралы хабарларды көрсету;

- оны шешу жолдарын түсіндіретін және ұсынатын қате туралы хабарларды көрсету.

Пайдаланушылардың қателерін болдырмау және жою тиімділігі осы интерфейспен жұмыс істеген кезде пайдаланушылардың жиі қателесетіні және бұрын жасалған қателердің салдарын еңсеру үшін аз уақыт пен күш қажет.

6.6. ГРАФИКАЛЫҚ ИНТЕРФЕЙСКЕ ҚОЙЫЛАТЫН ЖАЛПЫ ТАЛАПТАР

Графикалық интерфейсте пайдаланушы экранда көрсетілім түрін білдіреді, онда пайдаланушы пәрмендерді таңдай алады, тапсырмаларды іске қосып, тізімдегі файлдарды көре алады, экранда көрсетілген мәзір тізімдерін немесе элементтерін көрсете алады (мысалы, Windows жүйесі). Кез келген бағдарламаның графикалық пайдаланушы интерфейсі мыналарды қамтуы керек:

- Басты мәзір;
- Негізгі мәзір бөлімдерін қайталайтын тез түймелердің құралдар тақтасы;

- пайдаланушы белгілі бір құрамдас бөліктегі тінтуірдің оң жақ

батырмасын басқан кезде қалқымалы мәзірдің қалқымалы терезесі;

- Басқару элементтерін фокустық коммутациялаудың ойластырылған тізбегі;

- барлық мәзір бөлімдеріне және барлық басқару элементтеріне тез қол жеткізу кілттері, негізгі командаларға қол жеткізу үшін жылдам пернелер;

- тінтуірдің курсорын жылдам түймелер мен басқа компоненттерге жылжытқаныңызда шығатын қалқымалы кеңестер;

- пайдаланылған кеңестерге арналған және пайдаланушыға әртүрлі ақпаратты шығаратын күй жолағы;

- F1 пернесін басқан кезде немесе пайдаланушы тиісті мәзір бөлігін таңдаған кезде көрсетілген тақырыптар көрсетілетін анықтама файлы;

- қосымшаның белгішесінде тінтуірдің оң жақ батырмасы басылған кезде пайдаланушыға қол жетімді нұсқа туралы ақпарат;

- алдыңғы сеанста орнатылған параметрлер келесі сеанста қалпына келтірілу үшін бағдарламаны конфигурациялау және параметрлерді сақтау мүмкіндігі;

- қосымшаны орнату, оны Windows жүйесінде тіркеу және оны Windows-

Тапсырма

Алдыңғы жұмыста жасалған прототиптер негізінде графикалық пайдаланушы интерфейсінің эргономикасын талдаңыз. Стандартты графикалық пайдаланушы интерфейсінің талаптарына сәйкес әзірленген пайдаланушы интерфейсінің сәйкестік кестесін (сәйкессіздікті) жасау. Анықталған бұзушылықтарды түзету бойынша ұсыныстар дайындау. (Бағдарламаңыздың емес, басқа студенттің интерфейсін талдау қажет.)

БАҚЫЛАУ СҰРАҚТАРЫ

1. Пайдаланушыларға ыңғайлы интерфейс ті құрудың жалпы қағидалары мен ережелерін келтіріңіз.
2. Эргономикалық интерфейс ті дамытуда түс қалай пайдаланылады?
3. Пішіндердің орналасуына қойылатын негізгі талаптарды көрсетіңіз.
4. Мен кіргенде қателерді қалай болдырмауға болады?
5. Мәзірді қалай дұрыс жасау керек?

ОБЪЕКТІЛІ –БАҒДАРЛАНҒАН БАҒДАРЛАМАЛАУ

Объектілі-бағытталған бағдарламалау тұжырымдамасын қолдайтын Delphi тілі сабақтарды анықтауға мүмкіндік береді. Класс - бұл деректерді сипаттаудан басқа, сынып нысанының өкіліне орындалатын рәсімдер мен функциялардың сипаттамасын қамтитын күрделі құрылым.

Мұнда қарапайым класс декларациясының мысалы келтірілген:

```
TPerson = class
  private
    fname: string[15]; faddress: string[35];
  public
    procedure Show;
end;
```

Класс деректері өрістер, рәсімдер және функциялар - әдістер деп аталады. Жоғарыдағы мысалда TPerson - сынып атауы, fname және faddress - өріс атаулары, ал көрсетілім - әдіс атауы. Келісімге сәйкес қабылданған Delphi, өріс атауын басталуы тиіс f (өріс өріс өріс).

Сынып сипаттамасы бағдарлама түрін сипаттау бөлімінде орналастырылған.

Класс өкілдері ретінде бағдарлама var бөлімінде жарияланады, мысалы:

```
var
  student: TPerson; professor: TPerson;
```

Delphi-де объект - динамикалық құрылым. Айнымалы-нысанда деректер жоқ, бірақ нысанның деректеріне сілтеме бар, сондықтан бағдарламашы осы деректерге жад бөлу туралы қамқорлық көрсетуі керек. Жадты бөлу әдетте «Жасау» атымен жасалатын конструктор класының арнайы әдісі арқылы орындалады. Конструктордың ерекше рөлі мен мінез-құлқын атап көрсету үшін, сөздік рәсімнің орнына сынып сипаттамасында конструктор сөзі қолданылады.

Төменде конструктор енгізілген TPerson сыныбының сипаттамасы берілген:

```

TPerson = class private
fname: string [ 15 ];
faddress: string [35];
constructor Create; // конструктор
public
procedure show; // әдіс
end;

```

Нысан деректері үшін жадты бөлу конструктор әдісінің қолданысының нәтиже мәнін нысан түріне (сынып) қосу арқылы жүзеге асырылады. Мысалы, нұсқауларды орындағаннан кейін

```
professor := TPerson.Create;
```

профессор объектісінің деректеріне қажетті жады бөлінеді.

Жадты бөлуден басқа, дизайнер, әдетте, объектінің өрістеріне бастапқы мәндерді тағайындау мәселесін шешеді, яғни, объектіні инициализациялайды. Төменде TPerson нысаны үшін конструктордың мысалға келтірілуі берілген:

```

constructor TPerson.Create;
begin
  fname := ' ';
  faddress := ' ';
end;

```

Егер нысан бағдарламада бұдан былай пайдаланылмаса, осы нысанның өрістерінде орналасқан жадты босатуға болады. Бұл әрекетті орындау үшін Free destructor әдісін қолданыңыз. Мысалы, профессор объектісінің өрістерін босату үшін,

```
professor.Free;
```

Объектілі-бағдарланған бағдарламалау тұжырымдамасы жаңа сыныптарды анықтауға мүмкіндік береді, өрістерді, қасиеттерді және әдістерді бұрыннан бар сыныптарға қосу жолымен қабылдайды. Жаңа сыныптарды алу механизмі ұрпақ деп аталады. Бұл жағдайда, жаңа, балаларға негізделген сынып (бала) ата-ана мен ата-ананың қасиеттері мен әдістерін иеленеді.

Ата-ана сынып баланың сыныптық декларациясында көрсетіледі. Мысалы, TEmployee класы (қызметкер) FDepartment өрісін қосу арқылы талқыланған TPerson сыныбынан алынуы мүмкін. Бұл жағдайда TEmployee сыныптың мәлімдемесі келесідей болуы мүмкін:

```

TEmployee = class (TPerson)
FDepartment: integer; // номер отдела
constructor Create (Name:TName; Dep:integer);

```


end;

Тасымалданған TPMS сынып атауы TEmployee сыныбының TPerson сыныбынан алынғанын көрсетеді. Өз кезегінде TPerson сыныбы TEmployee класы үшін негізгі класс болып табылады.

TEmployee сыныбында ата-аналық сынып пен оның өрістерін инициализациялауды қамтамасыз ететін өз конструкторы болуы керек. TEmployee конструкторын іске асырудың мысалы келтірілген:

constructor

```
TEmployee.Create(Name:TName;Dep:integer);  
begin  
  inherited Create(Name);  
  FDepartment:=Dep;  
end;
```

Жоғарыда келтірілген мысалда мұраланған директива ата-аналық сыныптың конструкторын шақырады. Содан кейін, бала сыныптың өрісінің мәні тағайындалады.

Полиморфизм - әртүрлі сыныптарға түсетін әдістерге бірдей аттарды қолдану мүмкіндігі. Полиморфизм тұжырымдамасы объектіге әдіс қолданған жағдайда объектінің класына сәйкес әдісті қолдануды қарастырады.

Мысалы, үш сынып белгіленді, олардың біреуі екіншісі үшін негізгі:

type

```
// базалық класс TPerson  
= class fname: string; // аты  
  constructor Create(name:string);  
function info: string; virtual; end;  
// алынған от TPerson TStud =  
class(TPerson) fgr:integer; // оқу тобының саны  
  constructor Create(name:string;gr:integer);  
  function info: string; override; end;  
// алынған от TPerson TProf =  
class(TPerson) fdep:string; // кафедраның аталуы  
  constructor Create(name:string; dep: string);  
  function info: string;  
  override;  
end;
```

Осы сыныптардың әрқайсысында ақпараттық әдіс анықталады. Виртуалды директиваны пайдаланатын базалық класста, инфо әдісі виртуалды деп жарияланады. Әдіс виртуалды

деп жариялау сыныптың жоғары деңгейіне виртуалды әдісті өзімен ауыстыруға мүмкіндік береді. Әрбір баланың сыныбында ата-аналық сыныптың тиісті әдісін ауыстыратын өзінің меншікті ақпараттық әдісі бар (ата-аналық сыныптың виртуалды әдісін ауыстыратын баланың сыныбының әдісі переопределение директивасымен белгіленген). Төменде әр сынып үшін ақпараттық әдіс анықтамасы берілген.

```
function TPerson.info:string;  
begin  
result := ' ' ;  
end;  
function TStud.info:string;  
begin  
result := fname + ' rp.' + IntToStr(fgr);  
end;  
function TProf.info:string;  
begin  
result := fname + ' каф.' + fdep;  
end;
```

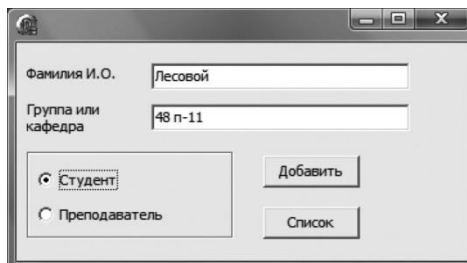
Екі сабақ бірдей негізгі негізінде жасалғандықтан, студенттер мен мұғалімдердің тізімін жариялауға болады (мұнда объект көрсеткіш болып табылатындығын еске түсіру керек):

```
list: array[1..SZL] of TPerson;
```

Тізімді осы жолмен жариялай аласыз, себебі Delphi көрсеткішін ата-аналық сыныпқа баланың сыныбына көрсеткіш мәнін тағайындауға мүмкіндік береді, сондықтан тізбе жиым элементтері TStud нысандары немесе TProf нысандары болуы мүмкін.

Студенттер мен мұғалімдердің тізімін ақпараттық элементтер әдісімен қолдану арқылы көрсете аласыз. Мысалы, бұл:

```
st := '';  
for i:=1 to SZL do // SZL – размер массив  
өлшемі  
if list[i] o NIL  
then st := st + list[i].Info
```



Аты-жөні, топ немесе кафедра меңгерушісі, студент, оқытушы, қосу, тізім

Сурет. 7.1. Бағдарламаның тілқатысу терезесі

```
+ #13; ShowMessage (st);
```

Бағдарлама жұмыс істеп тұрған кезде, массивтің әрбір элементі TStud түрінің нысаны және TProf түрінің нысаны болуы мүмкін. Полиморфизм тұжырымдамасы объектінің түріне сәйкес келетін әдіс атауы нысанына өтініш береді.

Келесі бағдарлама TПерсон, TStud және TProf жоғарыда аталған сыныптық декларацияларын пайдалана отырып, студенттер мен оқытушылардың тізімін жасайды және көрсетеді. Бағдарламаның мәтіні Листинг 7.1-де көрсетілген және диалогтық терезе сурет. 7.1.Листинг 7.1

Полиморфизмді көрсету:

```
unit polimor_;
interface
uses
Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs,
StdCtrls;
type
TForm1 = class(TForm) Edit1: TEdit;
Edit2: TEdit;
GroupBox1: TGroupBox;
RadioButton1: TRadioButton;
RadioButton2: TRadioButton;
Label1: TLabel;
Label2: TLabel;
Button1: TButton;
Button2: TButton;
procedure Button1Click(Sender:
TObject);
procedure Button2Click(Sender:
TObject);
private
```

```

    { Private declarations }
public
    { Public declarations }
end;
type
    // базовый класс
    TPerson = class
    fName: string; // имя
    constructor Create(name:string);
    function info:string; virtual;
    end;
    // класс Студент TStud = class(TPerson)
    fGr:integer; // топ номері
    constructor Create(name :string;gr:integer) ;
    function info:string;
    override;
    end;
    // класс мұғалім
    TProf = class (TPerson)
    fdep:string; // название кафедры
    constructor Create(name :string;dep:string);
    function info:string; override;
    end;
const
    SZL = 10; // тізім өлшемі
var
    Form1: TForm1;
    List: array[1..SZL] of TPerson; // список
    n:integer =0; // адам саны тізім бойынша
implementation
    {$R *.DFM}
    constructor TPerson.Create(name :
string); begin
    fName := name; end;
    constructor TStud.Create(name : string;gr:integer)
begin
    inherited create(name); // базалық кластың
    конструкторын шақырыңыз
    fGr := gr; end;
    constructor TProf.create(name :string; dep:string);
begin
    inherited create(name); // базалық кластың
    конструкторын шақырыңыз
    fDep := dep; end;
function

```

```

TPerson.Info:string;
begin
result := fname; end;
function TStud.Info:string;
begin
result := fname + ' rp.' + IntToStr(fGr);
end;
function TProf.Info:string;
begin
result := fname + ' каф.' + fDep;
end;
// Қосу түймешігін басыңыз

procedure TForm1.Button1Click(Sender:
TObject);
begin
if n < SZL then begin
// тізімді нысанға n: = n + 1;
if Radiobutton1.Checked
then // объект құрасыз TStud
List[n]:=TStud.Create(Edit1.Text,StrToInt(
Edit2.Text))
else // объект құрасыз TProf
List[n] :=TProf.Create(Edit
1.Text,Edit2.Text) ;
/
/
енгізу өрістерін анықтаңыз
Edit1.Text := ' ' ; Edit2.Text := ' ' ;
Edit1.SetFocus; // өріс курсоры Тегі
end
else ShowMessage('Тізім толтырылды!');
end;
// тізім түймешігін басыңыз
procedure TForm1.Button2Click(Sender:
TObject); var
i:integer; // индекс
st:string; // тізім
begin
for i:=1 to SZL do
if list[i] < > NIL then st:=st +
list[i].info + 113;
ShowMessage('Список'+#13+st); end;
end.
Қосу түймешігін басы арқылы басталатын
TForm1.Buttonclick рәсімі TStud немесе TProf класының Тізімін [n]

```

жасайды. Жасалған нысанның класы RadioButton қосқышы күйімен анықталады. Switch to Student (RadioButton 1) параметрін TStud класын және Teacher (RadioButton2) позициясын, TProf класын анықтайды. Әрбір тізім объектісіне (массив элементі) ақпарат әдісін қолдана отырып, Button2 түймешігін басу арқылы басталған TForm1.Button2Click процедурасы бүкіл тізімді білдіретін жолды құрады.

Тапсырма

Жоғарыда келтірілген мысал үшін, бағдарламаны шығару тізімдері алфавиттік түрде сұрыпталатын етіп өзгертіңіз. Есепке тапсырма, бағдарламаның листингі және тест сұрақтарына жауап беру кіреді.

БАҚЫЛАУ СҰРАҚТАРЫ

1. Тұжырымдаманы анықтаңыз: сынып, объект, мұра, инкапсуляция, полиморфизм.
2. Негізгі және сабақтас сыныптарды қалай жариялауға болады?
3. Сыныпты қалыптастыру деп аталатын не?
4. Класс конструкторы қандай міндеттерді шешеді?
5. Бұл әдіс виртуалды деп қалай атала

ТЕСТІЛЕУ СТРАТЕГИЯСЫН ТАҢДАУ ЖӘНЕ ТЕСТІЛЕУ

8. 1. ТЕСТІЛЕУ ДЕҢГЕЙЛЕРІ

Тестілеу - қателерді анықтау мақсатында бағдарламаның орындалуы. Отладтау локализация және қателерді басқару деп аталады.

Модульдік тестілеу бағдарламалардың немесе кіші бағдарламалардың бір бөлігі болып табылатын жеке бағдарлама процедураларын және кіші бағдарламаларды тексеру процесі болып табылады. Модульді тестілеуді тікелей әзірлеуші жүргізеді және әрбір модульдегі барлық ішкі құрылымдар мен деректер ағындарын тексеруге мүмкіндік береді. Тестілеудің бұл түрі даму кезеңінің бөлігі болып табылады. Бірліктерді тестілеу кезінде әзірлеуші анықтайтын сынақтар жиынтығы әр модуль бойынша тестілеуді қамту деңгейі кем дегенде 70 ... 75% құрайды. Модульді тестілеу элементтері:

- синтаксистік тексеру - бағдарлама кода синтаксистік қателерді анықтау үшін кейбір құралды қолдану;
- кодтау стандарттарына сәйкестігін тексеру - компанияның кодтау стандарттарына сәйкестігін тексеру;
- бағдарлама кодын техникалық қарау.

Бірліктерді тестілеу сәтті аяқталғаннан кейін, барлық модификацияланған модульдер мен сынақ люктері жобаның деректер базасында сақталады.

Интеграциялық тестілеу жекелеген модульдердің бірлескен жұмысын тексеру үшін жүргізіледі және бүкіл жүйені бір бүтін ретінде сынақтан өткізеді. Интеграциялық тестілеу барысында модульдер арасындағы байланыстар, олардың үйлесімділігі мен функционалдылығы тексеріледі. Ол тәуелсіздікпен жүзеге асырылады және тестілеу кезеңіне қосылады. Интеграциялық тестілеу элементтері:

- функционалдык тексеру - модульдер жиынтығымен

орындалатын жеке функциялардың, талаптарға сәйкес келтірілген функциялардың сәйкестігін тексеру;

- аралық нәтижелерді тексеру - барлық аралық нәтижелер мен файлдардың қол жетімділігі мен дұрыстығын тексеру;
- интеграцияны тексеру - модульдер арқылы өзара ақпарат берудің дұрыстығын тексеріңіз.

Интеграциялық тестілеу кезінде анықталған қателер қате дерекқорында жазылған. Интеграциялық тестілеудің нәтижелері сынақ циклінің соңында сынақ прогрессінің есебіне қосылады.

Жүйелік тестілеу бағдарламалық жасақтаманың тұтастай жүйесін, оның ұйымдастырылуы мен жұмыс істеуін клиенттің талаптарына сәйкестігін тексеру үшін жасалған. Оны интеграциялық тестілеу сәтті аяқталғаннан кейін тәуелсіз тестерлер жүргізеді.

Жүйелік тестілеу элементтері:

- шекаралық тестілеу - шекаралық шарттарда тестілеу;
- Run-through тестілеу - нақты жүйенің жұмыс істеуінің барлық функционалдық сипаттамаларын тексеру;
- мақсатты тестілеу - мақсатты платформасында тестілеу (мүмкіндігінше);
- құжаттаманы тексеру - дұрыстығына арналған пайдаланушы құжаттамасын тексеру;
- сынаушы анықтаған басқа сынақтар.

Тестілеу барысында анықталған қателер жоба дерекқорында жазылған. Жүйелік тестілеудің нәтижелері сынақ прогрессінің есебіне енгізілген.

Шығарындыларды тестілеу тестілеудің соңғы сатысы болып табылады, ол тапсырыс берушіге жеткізу үшін ПП қолжетімділігін тексереді. Бұл сынақ тәуелсіз тестерлермен жүргізіледі.

Шығару сынақтарының элементтері:

- орнатуды тексеру - орнату нұсқауларының анықтығы мен дұрыстығын тексеру;
- құжаттаманы тексеру - барлық қажетті құжаттаманың толық дайын екендігін және тапсырыс берушіге тапсыруға дайын екендігін тексеру.

Шығуды сынау кезінде анықталған қателер жоба дерекқорында жазылған. Шығару процесін сәтті аяқтағаннан кейін, ПТ тұтынушыға жеткізіледі, сонымен бірге тест нәтижелері туралы

есеп береді.

Қабылдау тестілеуін орнату, бағдарламалық қамтамасыз студі сақтау және соңғы пайдаланушыны оқытуға жауапты ұйым жүзеге асырады.

8.2.

ТЕСТІЛЕУ ТЕХНОЛОГИЯСЫ

Бағдарламалық қамтамасыз студі әзірлеу кезеңінде қолданылатын тестілеу технологиясы «шыны ыдысты» сынау деп аталады. Кейде бұл технология «қара жәшік» (қара жәшік) классикалық тұжырымдамасына қарама-қайшы емес, «ақ қорапты» сынау деп аталады.

«Қара жәшікті» тестілеу кезінде бағдарлама ішкі құрылымы белгісіз объект болып саналады. Сынаушы деректерді енгізеді және нәтижесін талдайды, бірақ ол бағдарламаның қалай жұмыс істейтінін білмейді. Тесттерді таңдаған маман, оның көзқарасынан, стандартты емес нәтижелерге әкелуі мүмкін деректер мен жағдайларды қызықтырады. Ол үшін, ең алдымен, тестілеу бағдарламасының қателіктері ең алдымен орын алатын деректердің әрбір санатының өкілдері қызықтырады.

1. «Шыны ыдысты» сынақтан өткізгенде жағдай мүлдем өзгеше. Сынаушы (бұл жағдайда бағдарламашы өзі) толық қолжетімді болатын бастапқы кодты білуіне негізделген тесттерді әзірлейді. Нәтижесінде, ол келесі жеңілдіктерге ие болады.

2. Тестілеу бағыты. Бағдарламашы бағдарламаны бөліктерде тексере алады, тестіленген модуль деп аталатын арнайы бағдарламаларды жасайды және бағдарламашыға қызығушылығын білдіретін бағдарламаға деректерді жібереді. Жеке модуль дәл «шыны жәшік» ретінде сынақтан әлдеқайда оңайырақ.

3. Толық кодты қамту. Әрбір сынақтағы қай код фрагменттері жұмыс істейтінін бағдарламашы әрқашан анықтай алады. Ол кодтың басқа бөлімдерінің әлі де сыналғанын көреді және олар сыналатын жағдайларды ала алады. Келесі кодтар тестілеуді орындау арқылы бағдарлама кодының ауқымын қадағалауды сипаттайды. Командалардың ағымын бақылау мүмкіндігі.

Бағдарламашы әрдайым келесі бағдарламада қандай функцияны орындау керектігін біледі және оның ағымдағы күйі қандай болуы керек. Бағдарлама ойдағыдай жұмыс істеп тұрғанын білу үшін бағдарламашы бағдарламаның орындалу барысы туралы ақпаратты көрсететін отладтау пәрмендерін қамтуы мүмкін немесе осы мақсатта түзеткіш деп аталатын арнайы бағдарлама құралын қолдануы мүмкін. Отладчик бағдарлама командасының орындалуын қадағалауға және өзгертуге, оның айнымалы мәндерінің мазмұнын және олардың мекен-жайларын жадыда көрсете алады және басқа да маңызды функцияларды орындай алады.

4. Деректердің тұтастығын бақылау мүмкіндігі. Бағдарламашы әрбір бағдарламаның қандай элементінің өзгеруі керектігін біледі. Деректердің күйін қадағалау арқылы (сол сол түзеткішті пайдаланып), ол деректерді дұрыс емес модульдермен, дұрыс емес түсініктемелермен немесе пікірталассыз ұйыммен өзгерту сияқты қателерді анықтай алады. Бағдарламашы бағдарламалық жасақтаманы өз бетінше тексере алады.

5. Ішкі шекаралық нүктелердің көрінісі. сырттан көрінісінен жасырылған бағдарламасы, сол шекаралық нүкте көзі коды көрінеді. Мысалы, бірнеше мүлдем басқа алгоритмдер және коды қарап жоқ, ол таңдаған бағдарламашы нақты әрекетті орындау үшін пайдалануға болады, олардың қандай анықтау мүмкін емес. Тағы бір типтік мысал кіріс деректерін уақытша сақтау үшін пайдаланылатын буферлік толып кету мәселесі болуы мүмкін. бағдарламашы дереу егер деректер толып Аралық кез келген соманы айта аласыз, және сынақтар мыңдаған жұмсайды, ал ол қажет емес.

6. Таңдалған алгоритммен анықталатын тестілеу мүмкіндігі. Деректерді өңдеуді өте күрделі есептеу алгоритмдерін пайдалану үшін арнайы технологиялар қажет болуы мүмкін. Классикалық мысалдар ретінде, матрицаның трансформациясын және деректерді сұрыптауды келтіре аламыз. Бағдарламашыдан айырмашылығы, қандай алгоритмдерді қолдану керектігін білу керек, сондықтан арнайы әдебиетке сілтеме жасау керек.

«Шыны боксты» тестілеу бағдарламалау процесінің бір бөлігі болып табылады. Бағдарламашылар бұл жұмысты үнемі жасайды, олар жазғаннан кейін әрбір модульді тексереді, содан кейін қайтадан оны жүйеге біріктіргеннен кейін. Бірліктерді сынақтан өткізу кезінде сіз технологияны немесе құрылымдық немесе функционалды тестілеуді немесе екеуін де пайдалана аласыз.

Құрылымдық сынақ - шыны ыдысты сынаудың бір түрі. Оның негізгі идеясы тестіленген бағдарлама жолын дұрыс таңдау болып табылады. Керісінше, функционалды тестілеу қара жәшіктің

сынақ категориясына жатады. Әрбір бағдарлама функциясы кіріс деректерін енгізу және шығуды талдау арқылы сыналады. Сонымен қатар, бағдарламаның ішкі құрылымы өте сирек ескеріледі.

Құрылымдық тестілеу әлдеқайда күшті теориялық негізге ие болса да, көптеген тестерлер функционалды тестілеуді қалайды. Құрылымдық тестілеу математикалық моделдеуге ыңғайлы, бірақ бұл ол тиімдірек дегенді білдірмейді. Технологиялардың әрқайсысы басқа пайдаланған кезде жіберіп алған қателерді анықтауға мүмкіндік береді. Осы тұрғыдан олар бірдей тиімді деп атауға болады.

Тесттер белгілі бір тәртіпте жасалады. Сонымен қатар, сыртқы ерекшеліктерге сәйкес сынақтар бөлінеді:

- 1) кіріс деректерінің әрбір класы үшін;
- 2) кіріс деректерінің шекаралық және сандық мәндері үшін.

Барлық шығыс кластары тексеріліп, қажет болған жағдайда тесттер қосылады. Сынақ 1-қадамда сыналмаған функциялар үшін әзірленеді. Бағдарлама мәтініне сәйкес, барлық шартты ауысулар әр бағытта орындалады ма тексеріледі. Қажет болса, жаңа сынақтар қосылады. Осындай жолмен, әрбір цикл үшін жолдар беріле ме, жоқ па - тексеріледі: денені орындамай, қайталанудың жалғыз және ең көп саны. Тесттер дайындалуда, айрықша жағдайларға тексеру жүргізіледі, кіруге рұқсат етілмеген деректер, төтенше жағдайлар.

8.3.

БАҒДАРЛАМА ҚАТЕЛЕРІ

Барлық бағдарлама қателері тиісті санаттарға жатқызылуы мүмкін. Ең жиі қарастырайық.

1. Функционалдық кемшіліктер бағдарламаға тән, егер ол не істемесе, оның функцияларының бірін орындайды нашар немесе толық емес. Бағдарламаның функциялары оның сипаттамасында егжей-тегжейлі сипатталуы керек және ол сынаушы өз жұмысын құрастыратын бекітілген ерекшеліктер негізінде

2. Пайдаланушы интерфейсінің кемшіліктері. Пайдаланушы интерфейсі жұмысының ыңғайлылығы мен дұрыстығын тек онымен жұмыс істеу барысында бағалауға болады. Пайдаланушының осы жұмысқа қатысқаны жөн. Бұған қол жеткізу интерфейсіне қойылатын талаптарды сәйкестендіріп, техникалық сипаттамаға сәйкестендіре отырып, РР прототипінің прототипін жасау арқылы қол жеткізуге болады. Талаптардың сипаттамасын

мақұлдағаннан кейін, кез келген ауытқулар немесе олардың сақталмауы қате болып табылады. Бұл пайдаланушы интерфейсіне толығымен қолданылады.

3. Өндірістің жеткіліксіздігі. Кейбір бағдарламалық өнімдерді әзірлеу кезінде оның маңызды сипаты жұмыс жылдамдығы болуы мүмкін, кейде бұл өлшем клиенттің талаптары бойынша белгіленеді. Пайдаланушы бағдарламаның баяу жүріп жатқанын, әсіресе, бәсекелес бағдарламалардың айтарлықтай тезірек жұмыс жасайтындығын көрсеткенде жаман. бірақ бағдарлама спецификация талаптарына сәйкес келетін техникалық сипаттамаға сай келмесе. Бұл қатені тазалау керек.

4. Қателерді дұрыс емес өңдеу. Қателерді өңдеу процедурасы бағдарламаның өте маңызды бөлігі болып табылады. Қатені дұрыс анықтау, бағдарлама туралы хабар беруі керек. Мұндай хабарламаның болмауы бағдарламада қателік болып табылады.

5. Шекаралық шарттарды дұрыс өңдеу. Көптеген шекаралық жағдайлар бар. Бағдарлама жұмысының кез-келген аспектісі, онда «көп» немесе «аз», «бұрынғы» немесе «кейінгі», «бірінші» немесе «соңғы», «қысқа» немесе «ұзын» ұғымдар ауқымның шекарасында *Z*zone. Бағдарлама ауқымында бағдарлама жақсы жұмыс істей алады, бірақ олардың шекарасында ең күтпеген жағдайлар орын алуы мүмкін, бұл өз кезегінде ҚБ жұмысындағы қателіктерге әкеледі.

6. Есептеу қателері. Бұл есептеу алгоритмін дұрыс таңдамаудан, өңделетін деректерге қолданылмайтын формулалар немесе формулалардан туындаған қателерді қамтиды. Дөңгелектегі қателіктер ең таралған.

7. Ағынды басқару қателіктері. Бағдарламаның логикасына сәйкес, екінші әрекет бірінші әрекетті орындау керек. Егер оның орнына үшінші немесе төртінші әрекеті орындалса, онда ағынды басқаруда қате бар.

8. Жарыс жағдайын тексеру өте қиын. Олар өзара әрекеттесу үрдістері мен жіптер бір мезгілде орындалатын жүйелерге, сондай-ақ көп пайдаланушы нақты уақыт жүйелеріне тән. Осындай жүйелердегі қателерді ойнату қиын, ал әдетте табу өте ұзақ уақытты алады.

Жарыс жағдайын тексеру өте қиын. Олар өзара әрекеттесу үрдістері мен жіптер бір мезгілде орындалатын жүйелерге, сондай-ақ көп пайдаланушы нақты уақыт жүйелеріне тән. Осындай жүйелердегі қателерді ойнату қиын, ал әдетте табу өте ұзақ уақытты алады.

9.Тиеу. Бағдарламадағы ақаулар жадтың болмауы немесе басқа қажетті жүйелік ресурстардың болмауына байланысты болуы

мүмкін. Әрбір бағдарламаның шектері бар, бағдарлама үлкен жүктемелермен күресуге қабілетсіз, мысалы, деректердің тым көп болуы. Сұрақ: бағдарлама шамадан тыс жүктелгенде қалай әрекет етеді.

10.Компьютерлік жабдықпен дұрыс жұмыс істемеу. Бағдарламалар аппараттық құрылғыларға дұрыс емес деректерді жібере алады, қате туралы хабарларды елемейді, бос емес немесе жоқ құрылғыны пайдалануға тырысады.

Кесте 8.1. Бағдарлама қателерінің түрлері және оларды табу жолдары	
Қателер түрлері	Анықтау әдістері
Синтаксистік	Құрастырушылар мен қосылғыштармен байланыстырушыларға статистикалық бақылау және диагностика
Run-time қателері автоматты түрде анықталады: асып кету, жадтан асып кету, өрғау түрі сәйкес келмейді	Динамикалық басқару:
Бағдарлама техникалық талаптарға сәйес емес	Мақсатты тестілеу
Ерекшеліктер талаптарға сәйес келмейді	Сынақтар, бета-тестілеу

Қалаған құрылғы ақаулы болса да, бағдарлама оған қол жеткізуге тырысып жатқанда емес, оны «тоқтатуға» емес, түсінуге тиіс. Бағдарлама қателерінің түрлері және оларды анықтау әдістері Кестеде келтірілген. 8.1.

Мемлекеттер арасындағы өтпелерді тексеру. Әрбір белсенді бағдарламада ауысулар айқын күйден екіншісіне жасалады. Ең қарапайым мысал - бұл бағдарлама: бағдарламаны бастағаннан кейін командалардың бір тізімі бар, олардың біреуін бағдарлама күйінің өзгерісін таңдағаннан кейін және командалар жаңа күйде қол жетімді мәзірде пайда болады.

Бағдарлама ұсынатын әрбір нұсқаны, әрбір мәзір пәрменін тексеру қажет. 10-пәрменді 9 пәрмені немесе 22 пәрмені арқылы ашылған тәртіпте алуға болады. Бұл жағдайда 10 пәрмені екі рет - екі режимде де сыналуы керек. Дегенмен, мәзір пәрмендері, бағдарламаның барлық ықтимал режимдері және осы режимдер жолдары соншалықты көп болуы мүмкін, сондықтан оларды тексеру мүмкін емес. Сондықтан, бағдарламаны орындау жолдарын тексеру үшін тесттерді таңдағанда, келесі қағидаларды ұстану керек:

- пайдаланушы әрекеттерінің барынша ықтимал тізбектерін тексеру;

- пайдаланушының бір режимдегі әрекеттері деректерді көрсету немесе басқа режимде бағдарламамен ұсынылған мүмкіндіктер жиынтығына әсер етуі мүмкін деп болжанса, осы әрекеттерді тексеріңіз;

- жоғарыда келтірілген ең қажетті сынақтарды орындаудан басқа, бағдарламамен еркін режимде жұмыс істеу үшін оны орындаудың кездейсоқ таңдауы.

Мемлекеттер арасындағы ауысу тек мәзір командаларын таңдаудан әлдеқайда күрделі болуы мүмкін. Деректерді енгізудің келесі формасының мазмұны мен құрылымы алдыңғы пішінде енгізілген ақпаратқа байланысты болуы мүмкін, кейбір өрістердің мәндері басқалардың рұқсат етілген мәндерін анықтай алады, белгілі бір ақпараттың кірісі бірқатар қосымша сұрауларды бастауы мүмкін. Мысалы, 1 мен 99 аралығындағы нөмірлерді енгізгенде, бағдарлама пайдаланушыға жіберілетін сұраудың бір түрін көрсетеді және басқа нөмірлерді енгізгенде, екіншісі. Бұл жағдайда эквиваленттік сыныптармен және олардың шекаралық мәндерімен бірге талдау қажет және тесттердің толық жиынтығын жасау үшін бағдарламаны жүзеге асырудың ықтимал жолдары.

Мәзір схемаларын жасау өте пайдалы. Мұндай схема осы жай-күйлер арасындағы өтуді тудыратын барлық бағдарлама күйлері

мен командаларды көрсетеді. Мәзірлер, графикалық құралдар (мысалы, түрлі түймелер) арқылы белсендірілетін пәрмендер және белгілі бір пернелерді басқаннан кейін орындалатын пәрмендер бар. Мысалы, схема «Файл» мәзірінен жолды «Ашу» пәрменіне, одан кейін «Файлды ашу» тілқатысу терезесіне және негізгі бағдарлама күйіне оралуға мүмкіндік береді. Әсіресе ыңғайлы диалогтық терезені бірнеше тәсілмен ашуға және бірнеше түрлі режимдерден шығаруға мүмкіндік беретін мұндай схемалар. Бұл жағдайда, сіз өтудің барлық бағыттарына диаграммада және бағдарламаны тексеруге арналған. Бұл өз мемлекетінің маңызды байланыстарын жоғалтпау үшін тәуекелмен күресудің қандай да бір жоспары жоқ бағдарламамен жұмыс істеуден гөрі сенімді әдіс.

Жарыс уақыты және басқа да уақытша тәуелділік. Бағдарламаға араласуға тырысыңыз, ол екі штаттар арасында ауысуды жүзеге асырса, пернелерді басу керек, әсіресе команда түймелері. Бағдарлама деректерді өңдеу әрекеттерін немесе кіріс-шығысын орындаған кезде кез-келген мәзір элементтерін таңдап көріңіз, бағдарламаға параллельді енгізу немесе шығару үшін бірнеше қосымша ақпарат беріңіз. Мысалы, файлды басып шығарғанда, оны тағы біреуін басып шығаруын сұраңыз.

Егер бағдарлама белгілі бір уақыттағы белгілі бір оқиғаны күтіп, басқа уақытқа шықса, пайдаланушы әрекеттеріне, жүйелік сұрауларға немесе күтілетін оқиғаның уақытша шекараға келуіне жауап беруін тексеріңіз, шығу. Бағдарлама оны күтуді тоқтатпастан бұрын немесе секундтан кейін қайталас бұрын қандай оқиға орын алса, не болатынын қараңыз.

Жүйені жоғары жүктеме кезінде тексеріңіз. Көп мақсатты ортада бірнеше басқа бағдарламаны іске қосыңыз және сіздің жұмысыңыздың қалай жұмыс істейтінін көріңіз, ол оның жұмысымен сәтті күресетін болсын. Принтерге үлкен файл жіберіңіз, сонда процессор техникалық қызмет көрсетуді басып шығару үшін барлық уақытта ауысады. Өртүрлі сыртқы құрылғыларды қосыңыз және мүмкіндігінше жылдам үзілістер жасаңыз. Қысқаша айтқанда, компьютерді мүмкіндігінше азайтыңыз және жүктеңіз. Нәтижесінде, бағдарламаңыз баяу жұмыс жасайды және жылдам деректерді енгізу арқылы оны қабылдау мүмкіндіктерінен асып кетуге болады. Егер бағдарлама қалыпты режимде жұмыс істемесе, ол жүктеменің артуымен байланысты болуы мүмкін.

«Стандартты» тестілеу бағдарламасын өте жоғары жүктемемен орындау, сіз жарыстың толық күтпеген жағдайына тап болуыңыз мүмкін. Егер бағдарлама осы жағынан осал болып шықса, онда осындай жағдайларда толық тестілеу циклін өткізу қажет.

Негізгі міндет - әзірленген бағдарламалық жасақтаманың сенімділігін қамтамасыз ету, ол кез-келген жүйеде және кез-келген қосымша жүктемеде сәтсіздікке қарамастан жұмыс істейді. Кем дегенде, бағдарламаның жұмыс істеуі үшін қандай жүйелік конфигурациялар шектелетінін білу өте қажет.

Тесттерді жүктеңіз. Құжаттамада анықталған бағдарламалық жасақтама өнімінің мүмкіндіктерін шектеуді ұмытпаған жөн. Бағдарлама жұмыс істей алатын файлдардың немесе басқа деректер құрылымдарының максималды санын ашыңыз, оны осындай жағдайда ұзақ уақыт қолдануға тырысыңыз. Егер құжаттама шектеулерді сипаттамаса, бірақ кез келген параметрлердің логикалық қолайлы мәндері болса, онда оларды тексеру керек. Егер бағдарлама пайдаланушының асығыс алатын үлкен сандық мәнмен күреспесе, қате туралы есеп жасалады. Бағдарлама параметрлерді өте аз және өте үлкен мәндерді қабылдауды және өңдеуді жүзеге асырса, оларда ешқандай шектеулер болмауы мүмкін.

Бағдарламаның әртүрлі аппараттық ресурстары таусылғанда қалай жұмыс жасайтынын тексеру керек, мысалы, диск толық немесе принтер қағаздан толып кетеді. Жүйеде аздаған бос жад болған кезде не болатынын біліңіз.

Жүктемені сынау, шын мәнінде, шекаралық шарттарды сынаудың бір түрі болып табылады. Оның мінез-құлқы схемасы мүлдем ұқсас. Біріншіден, бағдарлама жұмыс істейтін шарттарда, содан кейін - ол үшін арналмаған шарттарда басталады. Мүмкіндігінше әртүрлі шарттарды тексеру және тексеру. Мүмкін, әртүрлі жоғары жүктемелермен бөлек жұмыс істей отырып, бағдарлама бәрін біріктірмейді. Жүйені жүктегеннен кейін бір немесе екі сынақты емес, ұзақ және мұқият тестілеуді орындаңыз. Бағдарламаны осындай жағдайларда біраз уақыт басқарыңыз, мүмкін ол дереу сәтсіздікке ұшырайды, бірақ ол орын алады.

Қателерді болжау. Кейде тестерлер сынақтың белгілі бір сыныбы бағдарламаның бұзылуына әкеліп соғатындығын болжайды, бірақ ол логикалық негізде оны ақтау мүмкін емес. Өзіңіздің интуицияңызға сеніңіз және осындай тесттерді жалпы жоспарға енгізгеніңізге сенімді болыңыз. Шекаралық құндылықтар болмаса да, көбінесе бағдарламалық жасақтаманың бұзылуына әкелетін бірқатар жағдайлар мен мағыналар бар. Осындай құндылықтардың типтік үлгісі - 0. Өзіңізге белгілі бір кіріс мәнінің немесе бағдарламаның орналасуының күдікті болып көрінетін себептерді іздейтін уақытты жұмсамаңыз, оны тек сынап көріңіз.

Күрделі жағдайларда интуиция тривиальді логикаға

қарағанда әлдеқайда жақсы тестілеу тактикасын ұсынады. Қауымдық байланыс жұмыс істейді: сіз осындай жағдайларда қате таптыңыз, бірақ оны есіне түсіре алмайсыз. Қандай болмасын, ішкі сезімдеріңізге сеніңіз және оны тындауға үйреніңіз: тәжірибе барынша дамыған және сенімді болады.

Функционалды эквиваленттік тестілеу. Функционалды эквиваленттілікті тестілеу кезінде есептеулердің нәтижелері сол математикалық функцияның әртүрлі бағдарламалары бойынша салыстырылады. «Функционалды эквиваленттілік» термині «эквиваленттік сыныптар» терминімен ешнәрсе байланыстырады. Егер екі бағдарлама сол функцияны есептеу кезінде бірдей нәтиже берсе, онда олар теңдестірілген есептеу әдістерін қолданады.

Математикалық функцияны есептейтін және нәтижені басып шығаратын бағдарлама сыналады делік. Бұл қарапайым тригонометриялық функция немесе матрицаны инвертирует немесе белгілі бір деректер жиынтығын бейнелейтін қысққ сызықты құруға коэффициенттерді қайтаратын әлдеқайда күрделі функция болуы мүмкін. Әдетте мұндай жағдайларда сіз сол әрекеттерді орындайтын басқа бағдарламаны таба аласыз және сонымен бірге жеткілікті сенімді және уақытты тексере аласыз. Екі бағдарлама да кіріс деректерінің бірдей жиындарын өңдейді. Егер нәтижелер сәйкес келсе, сыналған бағдарлама дұрыс жұмыс жасайды.

Тапсырма

Бағдарламаның сынақ нұсқасын өзінің өз нұсқасында жасаңыз. «Шыны ыдыстың» әдісін тестілеу кезінде алдыңғы тәжірибеде әзірленген және жетілдірілген алгоритмдер схемаларын қолдануға болады. Тексеру үшін бірнеше алгоритмді таңдап, осы алгоритмдер бөлімінің әріптерін немесе суретін белгілеңіз. Таңдалған сынақ әдісі бойынша сынақтармен расталуы қажет алгоритм жолдарын жазыңыз. Алгоритм жолдарын өтуге мүмкіндік беретін тесттерді жазып алыңыз.

«Қара жәшік» әдісі үшін сынақтар келесі санаттар бойынша жүргізіледі:

- 1) функционалды мәселелерді шешудің толықтығы;
- 2) кіріс ағынының максималды жүктеме көлемінде тестілеу;
- 3) ресурстарды дұрыс пайдалану;
- 4) жұмысын бағалау;
- 5) деректердің сыбайлас жемқорлықтан және дұрыс емес әрекеттерден қорғаудың тиімділігі;
- 6) түрлі платформаларда орнату және конфигурацияны

тексеру;

7) құжаттаманың дұрыстығы.

Нәтиже дерекқорды (аппараттық және операциялық платформа) және тест нәтижелерін қамтитын есеп түрінде беріледі. Қойынды түріндегі соңғы пішін. 8.2.

Кесте 8.2. Тест нәтижелері		
Сынақ әдісі	Күтілетін нәтиже	Нақты нәтиже

БАҚЫЛАУ СҰРАҚТАРЫ

1. «Ақ жәшік» және «қара жәшік» тестілеу технологиясын анықтаңыз.
2. Тестілеудің негізгі түрлерін көрсетіңіз
3. Нәсілдік жағдай нені білдіреді? Ол оны қалай тексеруге болады?
4. Бағдарламалық жасақтама қателігі дегеніміз не? Сіз қандай бағдарлама қателерін білесіз?
5. Бағдарламалық жасақтама өнімінің жай-күйін тексеру арқылы сіз нені білдіреді?
6. ПП-дың жүктемесін тестілеу дегеніміз не?
7. Қателерді болжау туралы не ойлайсыз?
8. Функционалды эквиваленттің анықтамасын беріңіз.
9. Кездейсоқ кіріс қалай ұйымдастырылуы мүмкін?

СОМ-ТЕХНОЛОГИЯСЫН ПАЙДАЛАНУ КЕЗІНДЕ КОМПОНЕНТТІК ТӘСІЛДІ ҚОЛДАНУ

9.1. СОМ-ТЕХНОЛОГИЯСЫ ЖӘНЕ ОЛЕ АВТОМАТТАНДЫРУ ҰҒЫМДАРЫ

Компоненттік көзқарас стандартталған екілік интерфейстер арқылы бір-бірімен өзара әрекеттесетін бағдарламалық жасақтаманың физикалық тұрғыда қолданыстағы бағдарлама бөліктерінің жекелеген құрамдастарынан бағдарламалық жасақтаманың құрылысын көздейді. Қарапайым нысандардан айырмашылығы, құрамдас бөліктерді динамикалық түрде шақырылған кітапханаларға немесе орындалатын файлдарға жинауға, оларды екілік формада (бастапқы кодсыз) таратуға және оны тиісті технологияны қолдайтын кез келген бағдарламалау тілінде пайдалануға болады. Компоненттік тәсіл COM (Component Object Model - объектілердің компоненттік моделі) негізінде әзірленген технологияларды негіздейді және CORBA (Common Object Request Broker Architecture - Common Object Request Broker Architecture - объектілерді өңдеу сұрауларының делдалымен ортақ архитектура) қосымшаларын жасау технологиялары. Бұл технологиялар осындай қағидаларды қолданады және оларды жүзеге асыру ерекшеліктерімен ерекшеленеді.

Microsoft корпорациясының COM технологиясы - композиттік құжаттарды жасау үшін Windows жүйесінің бұрынғы нұсқаларында пайдаланылған OLE 1 (Объектілерді байланыстыру және ендіру) технологиясының дамуы. OLE бағдарламашыларға комплекстік құжаттармен жұмыс істеу үшін қосымшаларды құруға мүмкіндік береді, олардың кейбір бөліктері әртүрлі бағдарламаларда дамуы мүмкін динамикалық байланысты құрылымдар. Бұл технология Windows 3.1 ішіндегі OLE 1 ретінде пайда болды және пайдаланушы әртүрлі түпнұсқа нысандарды қамтитын күрделі қосылым құжаттарды жасай алатынын білдіреді. Енгізілген нысандарды тиісті деректер элементіндегі тінтуірді басу арқылы

өзгертуге болады.

Мысалы, Word құжатында ендірілген Excel электронды кестесін екі рет басу керек және жеке терезеде редакциялауға дайын жүктелген жұмыс парағы бар Excel бағдарламасын бастадыңыз. Өңдеуді аяқтағаннан кейін, Excel бағдарламасы Word құжатына ендірілген Excel нысанына өзгерістерді сақтауға мүмкіндік берді.

Басқа ерекшелігі - объектілердің байланысы - электрондық кестені Word құжатына байланыстыруға мүмкіндік берді (іс жүзінде, электрондық кестеге сілтеме Word құжатында сақталған). Егер түпнұсқа электрондық кестедегі деректер жаңартылса, келесі жолы Word құжаты жүктелген болса, сілтеме құжатты жаңартты және сол жерде жасалған өзгерістерді көрсетті.

Қосымшаның әрі қарай дамуы және байланыстыру OLE 2.0-де алынды. Бұл жетілдірілген тәсілдің негізі объектінің компоненттік моделі болды (COM). Бұл түрлі тілдердегі әртүрлі компаниялар жазған компоненттердің өзара әрекеттесуінде толық үйлесімділікті қамтамасыз ететін жүйелік бағдарламалық құралдың объектілі үлгісі. Табысқа кілті - бұл компоненттердің модулділігі. Оларды сатып алуға, жаңартуға немесе жалғыз немесе топтарға алмастыруға болады, және бұл тұтастай жұмыс істеуіне әсер етпейді.

OLE 2.0-де енгізілген жаңа функция OLE автоматтандыруы болып табылады, ол сыртқа қолдану объектілеріне қол жеткізуге және басқаруға мүмкіндік береді. Сыртқы пайдалануға арналған осындай нысандар OLE авто нысандары деп аталады. Automaton нысаны ретінде қолдануға болатын нысандардың түрлері құжат, параграф немесе сөйлем болып табылады. Электрондық кесте кестені, диаграмманы, ұяшықты немесе ұяшықтардың тобын көрсетуі мүмкін.

Автоматты нысандар мен қарапайым OLE нысандары арасындағы негізгі айырмашылық мынада, бұл автоматты нысандар бағдарламалық жасақтамада ғана бар, олар бағдарлама коды арқылы жасалады және пайдаланылады, демек, негізінен уақытша болып табылады. Оларды іске асыруға немесе қосуға болмайды. Олар тек бағдарламаларды орындау кезінде ғана өмір сүре алады және тікелей пайдаланушыға көрінбейді.

Автоматты серверлердің екі түрі бар: процестің ішінде және процестен тыс (олар жергілікті серверлер деп те аталады). Процесс ішіндегі сервер автоматты нысандарды экспорттайтын DLL (динамикалық байланысқан кітапхана) болып табылады. Автоматты нысандар DLL-тен басқа қолданбалардан емес, олар клиенттік қосымшаның бөлігі болып табылады. Бұл сервердің әрбір шақыруымен бірге жүретін үлкен шығынды болдырмайды.

Процесстен тыс сервер автоматты нысандарды экспорттайтын жеке орындалатын файл болып табылады.

Автоматтандыру OLE дұрыс жұмыс істеуі үшін үш түрлі ақпарат көздеріне сүйенеді: сыныптар, құжаттар, элементтер.

OLE сыныбы автоматты түрде нысан жасайтын сервер бағдарламасын анықтайды. Мысалы, егер нысан сыныбы .doc құжаты болса, ол Word бағдарламасын пайдалана отырып жасалды.

OLE құжаттары OLE нысанының деректерін қамтитын бастапқы файлға сілтеме жасайды. Кез-келген құжат тиісті объекті болуы керек (яғни ендірілмеген), өйткені тиісті құжаттардың деректері файлда сақталуы тиіс.

OLE элементі құжаттың қай бөлігіне басқа құжатқа байланыстырылатын немесе ендірілетін деректерді анықтайды. Осылайша, бүкіл құжаттың орнына, бірнеше құжаттарды өздерінің деректерінің кішкене бөлігі ғана байланыстыра аласыз. OLE элементтері нысандардың өлшемін азайтады.

Сонымен, COM дегеніміз не? Бұл сұраққа жауап беру үшін бағдарламалық жасақтаманың бір бөлігі басқа бөлік ұсынатын қызметтерге қалай қол жеткізуі керек екенін түсінуі керек. Кітапханамен байланысы бар қолданбалар осы кітапханадан функцияларды шақыратын қызметтерді пайдалана алады. Өтінім басқа да қосымшалардың қызметтерін пайдалана алады, бұл толықтай жеке процесс. Бұл жағдайда екі жергілікті процестер белгілі бір байланыс механизмі арқылы өзара әрекеттеседі, бұл әдетте осы қосымшалар арасындағы хаттаманы анықтауды талап етеді (бір өтінімге сұраулар жіберуге мүмкіндік беретін хабарлар жиынтығы және екіншісіне тиісті түрде жауап беру). Мысалы, бағдарлама амалдық жүйенің қызметтерін немесе басқа компьютерде жұмыс істейтін бағдарламалық жасақтама қызметтерін пайдаланады. Мәселе мынада, бағдарламалық қамтамасыз етудің бір бөлігі басқа бөлік ұсынатын қызметтерге қол жеткізуі керек. Бірақ әр жағдайда, қол жеткізу механизмі әртүрлі.

SOM технологиясында, қолданбаның SOM нысандарын пайдалану арқылы оның қызметтерін пайдалануға болады. Бір бағдарламада кем дегенде бір нысан бар. Әрбір нысанның бір немесе бірнеше интерфейсі бар. Әрбір интерфейс қасиеттерге (деректерге) және операцияларды жасауға мүмкіндік беретін объектілі әдістерді біріктіреді. Әдетте, бір типті операцияларды орындайтын немесе біртекті қасиеттермен жұмыс істейтін барлық әдістер интерфейсте біріктіріледі.

Клиент тек қана интерфейс пен оның әдістерінің көмегімен объектінің қызметтеріне қол жеткізеді. Бұл механизм - кілт.

Объектінің қасиеттері мен әдістерінің құрамы туралы толық ақпарат алу үшін клиенттің бірнеше негізгі интерфейстерін білуі жеткілікті. СОМ спецификациясына сәйкес, бұрыннан жасалған интерфейс ешқандай жағдайда өзгермейді. Бұл кез-келген жаңартуларға қарамастан, қосымшалардың әрқашан СОМ негізінде жұмыс істейтіндігін қамтамасыз етеді.

Объект әрдайым СОМ серверінде жұмыс істейді. Сервер динамикалық кітапхана немесе орындалатын файл болуы мүмкін. Нысанда өз қасиеттері мен әдістерінің болуы немесе сервер деректерін және қызметтерін пайдалану мүмкін. Нысан әдістеріне қол жеткізу үшін, клиент тиісті интерфейске көрсеткішті алу керек. Әр интерфейс үшін таңдамалы көрсеткіш бар. Осыдан кейін клиент объектінің қызметтерін пайдалана алады, жай ғана оның әдістерін шақыра алады. Объектілердің қасиеттеріне қол жеткізу тек оның әдістері арқылы жүзеге асады.

СОМ нысаны электрондық кестеге ендірілсін делік. Клиент пен объект арасындағы өзара әрекеттесуді СОМ негізгі механизмдері қамтамасыз етеді; клиент клиенттің бағдарламалық жасақтамасының тұрғысынан, сол процестің мекенжай кеңістігінде, басқа процесте немесе басқа компьютерде жасырады, сондықтан, клиенттің бағдарламалық жасақтамасын әзірлеуші тұрғысынан, электрондық кестенің функцияларын пайдалану нысанның әдісіне қалыпты сілтеме сияқты көрінеді.

Сұрақ пайда болады: СОМ нысанын жасау және инициализациялау клиенттің бірінші данасында қалай пайда болады? Операциялық жүйе олардың біреуі қажет болған жағдайда тіркелген барлық сыныптардың даналарын өз бетімен жасауға тиіс емес. Бірақ объектінің жұмысы үшін сізге серверлер қажет. Windows жүктеген сайын Word, Excel, Internet Explorer және т.б. іске қосылғанын елестетіп көріңіз.

Кез келген СОМ нысаны оның қасиеттері мен әдістерін сипаттайтын кейбір сыныптың тұрақты данасы болып табылады. Осы операциялық жүйеде тіркелген және қол жетімді барлық СОМ сыныптары туралы ақпарат арнайы сыныптағы СОМ кітапханасында жиналады, ол объект классының данасын бастау үшін пайдаланылады.

Біріншіден, клиент қажетті кітапхананың атауын және бірінші қажетті интерфейссті тапсыру арқылы СОМ кітапханасына кіреді. Кітапхана қажетті классты табады және алдымен серверді бастайды, содан кейін сыныптың даналық объектісін жасайды. Осыдан кейін кітапхана объектіге және интерфейске клиенттік көрсеткішке қайтарады. Кейінгі жұмыста клиент тікелей объектіге

және оның интерфейстеріне жүгіне алады.

Біріншіден, клиент кітапханаға қажетті кітапхананы және оған қажетті СОМ интерфейсмен кіреді. Кітапхана қажетті класты табады және алдымен серверді бастайды, содан кейін сыныптың даналық объектісін жасайды. Содан кейін кітапхана объекіге және Интерфейс тұтынушы көрсеткішке қайтарды. Жобаның объектісі болып табылатын «Каяйинг жимиста» клиенті алға интерфейсінде.

Сұрақ туындайды: клиент объект туралы ақпаратты қалай ала алады? Мысалы, клиент бағдарламалық жасақтамасының әзірлеушісі электрондық кестенің СОМ сипаттамасына сәйкес жасалғанын біледі, бірақ клиенттерге оның қызметтерімен қамтамасыз ететін СОМ нысандары туралы білмейді.

Осындай жағдайларды шешу үшін, СОМ нысанының әзірлеушісі объектімен бірге түр туралы ақпарат таратуы мүмкін. Онда интерфейстер, олардың қасиеттері мен әдістері, әдістің параметрлері туралы ақпарат бар. Бұл ақпарат арнаулы интерфейссті анықтау тілі (IDL) арқылы жасалған түр кітапханасында болады.

Енді Delphi-да СОМ нысандарын жасау механизмін қарастырыңыз. Жоғарыда айтылғандай, СОМ нысаны интерфейске сілтегіш арқылы қол жетімді әдістердің белгілі бір тіркесімі ретінде түсінікті интерфейстердің ерікті санын жасау мүмкіндігін қамтамасыз етуі керек.

Осындай талаптарды объектілі-бағдарлы бағдарламалаудың стандартты тәсілдері шеңберінде тікелей орындау өте қиын. Delphi-де бұл іске асыру төменде келтірілген. СОМ нысанының өзі әдеттегі TComObject сыныбы арқылы сипатталады, ол тікелей TObject-ден жасалады.

Нысанның мақсатын жүзеге асыратын барлық қасиеттер мен әдістер өзінің декларациясында жарияланады және сипатталады, сондықтан жаңа СОМ нысанының сыныбы негізінен өзгеше болмайды. TComObject сыныбы СОМ объектісін жасайтын негізгі нысан функцияларын қамтамасыз етеді. Төменде Excel кестесін OLE автоматтандыру механизмі арқылы қол жеткізудің мысалы келтірілген. Excel және Word бағдарламаларына қатысты OLE автоматтандыруын пайдалану VBA анықтамасы барлық бағдарламалардың, сипаттар мен әдістердің сипаттамасын қамтитын осы бағдарламаға ендірілгеніне байланысты жеңілдетілген. Осылайша, әзірлеуші Delphi қосымшасынан жұмыс істеуге мүмкіндік алады.

9.2. DELPHI-дегі OLE АВТОМАТТАНДЫРУДЫ EXCEL кестесі АРҚЫЛЫ КӨРСЕТУ

Код редакторындағы MS Excel серверімен байланыс орнату үшін, пайдалану бөліміндегі ComObj модулін пайдаланыңыз. Бұл модуль COM нысандарымен жұмыс істеу үшін қажетті барлық функцияларды сипаттайды. Төменде MS Excel тізіміндегі бағдарлама кодының мысалы болып табылады, ол BookTable кестесінде сақталады:

```
procedure TMainForm.ExcelButtonClick(Sender
: TObject);
var
  XLApp, Sheet, Colum:Variant;
  index, i:Integer;
begin
  XLApp:= CreateOleObj
ect('Excel.Application'); XLApp.Visible:=true;
  XLApp.Workbooks.Add(-4167);
  XLApp.Workbooks[1].Worksheets[1].Name:='Есеп
п';
  Colum:=XLApp.Workbooks[1].Worksheets['Есеп'
].Columns;
  Colum.Columns[1].ColumnWidth:=20;
  Colum.Columns[2].ColumnWidth:=20;
  Colum.Columns[3].ColumnWidth:=20;
  Colum.Columns[4].ColumnWidth:=20;
  Colum.Columns[5].ColumnWidth:=20;
  Colum:=XLApp.Workbooks[1].Worksheets['Есеп'
].Rows; Colum.Rows[2].Font.Bold:=true;
Colum.Rows[1].Font.Bold:=true;
Colum.Rows[1].Font.Color:=clBlue;
Colum.Rows[1].Font.Size:=14;
  Sheet:=XLApp.Workbooks[1].Worksheets["Есеп"
];
  Sheet.Cells[1,2]:='Таяп тізімі';
```

```

Sheet.Cells[2,1]:='аталуы м'; Sheet.Cells[2,2]
    ='Өлшем бірлігі '
    ; Sheet.Cells[2,3]:='Бағасы';
    Sheet.Cells[2,4]:='Қал
дығы'; index:=3;
    DataModule1.BookTable.First;
    for i:=0 to
DataModule1.BookTable.RecordCount-1 do
    begin
        Sheet.Cells[index,1]:=DataModule1.BookTable
.Fields.Fields[1].
        AsString;
        Sheet.Cells[index,2]:=DataModule1.BookTable
.Fields.Fields[2].
        AsString;
        Sheet.Cells[index,3]:=DataModule1.BookTable
.Fields.Fields[3].
        AsInteger;
        Sheet.Cells[index,4]:=DataModule1.BookTable
.Fields.Fields[5].A
        sInteger;
        Inc(index);
        DataModule1.BookTable.Next;
    end;
end;

```

Кодтың бірінші жолы Excel нысанын жасайды және оны XLApp айнымалысына жазады:

```

XLApp:= CreateOleObj
ect('Excel.Application');

```

CreateOleObject функциясы COM технологиясы бойынша басқа бағдарламамен байланыс орнатуға мүмкіндік береді. Осыған байланысты деректерді басқа қолданбаларға тасымалдауға болады. Ол үшін қосылым жасалатын бағдарлама деректерді сыртқы (мысалы, Word немесе Excel) қабылдау үшін тиісті мүмкіндіктері болуы керек, сондықтан сіз жұмыс істей алатын функциялар туралы білуіңіз керек.

Екінші код коды Excel бағдарламасын көрсетуді күшейтеді. Содан кейін жаңа жұмыс кітабы қосылады:

```
XLApp.Workbooks.Add(-4167);.
```

Жақшадағы нөмір тұрақты болып табылады, яғни кітапты жасауды білдіреді және оны өзгерте алмайсыз. Барлық тұрақты мәндер туралы қосымша ақпаратты әзірлеуші нұсқаулығынан табуға болады. Бұдан әрі осы кітаптың атауы қалыптасады:

```
XLApp.Workbooks[1].Worksheets[1].Name:='Отчет';
```

Бұл әрекет міндетті емес, бірақ бұл қажет, өйткені әдепкі атау «Sheet 1».

Енді бізде Excel жұмыс істейді және жаңа кітап жасайды, сіз деректерді жіберуді жүзеге асыра аласыз. Бірақ бұған дейін бағандар мен жолдарды пішімдеу қажет; бұл үшін жұмыс кітабының бағандарында индексті алу керек:

```
Column:=XLApp.Workbooks[1].Worksheets['Отчет'].Columns;
```

Нәтиже Variant түрінің Varium айнымалысына жазылады. Енді баған енін өзгертеміз (ColumnWidth):

```
Column.Columns[1].ColumnWidth:=20;
```

Содан кейін бір айнымалы мәнде жұмыс кітабының жолдарына көрсеткішті жазыңыз:

```
Column:=XLApp.Workbooks[1].Worksheets['0T4eT'].Rows;
```

Есептің бірінші екі жолында шрифт түрін орнатыңыз:

```
Column.Rows[2].Font.Bold:=true;
```

```
Column.Rows[1].Font.Bold:=true;
```

Төрт жақшаларда енді сызық нөмірі бар. Содан кейін, бірінші жолдың түстерін көк түске орнатыңыз және қаріп өлшемі 14:

```
Column.Rows[1].Font.Color:=clBlue;
```

```
Column.Rows[1].Font.Size:=14;
```

Пішімдеу аяқталды, енді сіз деректерді жібере аласыз. Мұны істеу үшін параққа көрсеткішті аламыз:

```
Sheet:=XLApp.Workbooks[1].Worksheets['Отчет'];
```

Деректерді көрсету үшін, сізге мәнді тағайындау керек

```
Sheet.Cells[строка, колонка]
```

Кестенің шығу кодын қарастырыңыз:

```

        index:=3;
        DataModule1.BookTable.First;
        for          i:=0                               to
DataModule1.BookTable.RecordCount-1 do
    begin
        Sheet.Cells[index,1]:=DataModule1.BookTable
.Fields.Fields[1].
        AsString;
        Sheet.Cells[index,2]:=DataModule1.BookTable
.Fields.Fields[2].
        AsString;
        Sheet.Cells[index,3]:=DataModule1.BookTable
.Fields.Fields[3].
        AsInteger;
        Sheet.Cells[index,4]:=DataModule1.BookTable
.Fields.Fields[5].
        AsInteger;
        Inc(index);
        DataModule1.BookTable.Next;
    end;

```

Айнымалы индекс енді Excel кестесінің қандай жолын көрсететінін көрсетеді. Алғашқы екі жол баяндама тақырыбы бойынша жұмыс істеп тұрғандықтан, деректер үшінші жолдан шығуы керек (index = 3). Excel бағдарламасындағы жолдар мен кестелер қалған кестелерде және массивтердегідей нөлден емес, бірден басталады.

Осыдан кейін, ADOTable құрамдас бөлігінің бірінші әдісін қолданып, дерекқордағы кестенің бірінші жолына өтеміз. Бұл пайдаланушы есеп кестесін басудан бұрын кестенің ортасындағы кез келген жолды таңдауы мүмкін, себебі бұл есеп осы таңдалған жолдан шығады. Енді барлық циклдар айналдырылатын циклды бастау шарттары дайындалып, олардан ақпарат Excel бағдарламасына қойылады. Цикл нөлдік жолдан басталады және кестедегі жолдар санының мәніне жеткенге дейін жалғасады:

```

        For          i:=0                               to
DataModule1.BookTable.RecordCount-1

```

Осылайша, біз барлық жазбаларды өткізіп, әрбір жазбаны Excel-ге, дәйекті барлық бағандарға жаздық (1-5).

Бұл OLE автоматтандыруымен орындалатын типтік операцияларды шолуды аяқтайды. Жалпы, OLE автоматтандыруды

пайдалану VB-ге өте жақын және жоғарыда қарастырылмайтын Excel-мен жұмыс істеудің барлық мүмкіндіктерін Excel бағдарламасының кіріктірілген бағдарламасымен танысу арқылы іске асыруға болады.

Біз Excel-мен жұмыс істеуді қарастырдық. Word бағдарламасымен жұмыс істеу үшін, OLE автоматикасы ұқсас түрде пайдаланылады, тек нысандар, олардың қасиеттері мен әдістер Excel-тен ерекшеленеді. Word бағдарламасының жаңа данасын келесідей:

```
uses ComObj;  
var  
WordApp:Variant;  
WordApp:=CreateOleObject('Word.Application'  
);  
WordApp.Visible:=true;
```

Көріп отырғаныңыздай, Excel бағдарламасымен байланысудың жалғыз айырмашылығы - басқа класқа арналған атау, ал басқалары Excel бағдарламасымен бірге жұмыс істейтін барлық қарастырылған әдістер Word функциясына, мүмкіндіктері ескерілгенде қолданылады.

Тапсырма

Delphi бағдарламасында MS Excel / Word серверіне қосылыңыз. Тапсырманың жаңа нұсқасы үшін MS Excel / Word бағдарламасында есеп шығару үшін бағдарлама модулін жасаңыз.

БАҚЫЛАУ СҰРАҚТАРЫ

- 1.COM технологиясын сипаттаңыз.
- 2.OLE технологиясын дамыту кезеңдерін сипаттаңыз.
3. Delphi-да Excel және Word бағдарламаларына OLE автоматтандыру қалай қолданады?

КОМПОНЕНТТІ БАҒДАРЛАМАЛАУ НЕГІЗІНДЕ ДИНАМИКАЛЫҚ КІТАПХАНАНЫ ҚҰРУ

10.1. ДИНАМИКАЛЫҚ КІТАПХАНА ЖӨНІНДЕ НЕГІЗГІ МӘЛІМЕТ

COM нысаны әрқашан серверде іске асырылады. Сервер бағдарламаның жұмыс істеуі кезінде жүктелген динамикалық түрде қосылған кітапхана немесе жеке оқшау процесс болуы мүмкін.

Динамикалық түрде қосылған кітапхана (DLL) немесе динамикалық кітапхана - функцияларды және ресурстарды орындалатын файлдан бөлек сақтау үшін қолданылатын орындалатын файлдың ерекше түрі. Әдетте, бағдарлама әзірленгенде және жұмыс жасағанда, ресурстар жасалады, олар орындалатын файлға жиналады. DLL қолданылған кезде, оны шақыртылған функцияларды орындау кезінде модуль пайдаланылады. DLL файлы пайдалы және жиі пайдаланылатын функцияларды көптеген бағдарламалар үшін бір уақытта қол жетімді етеді, бірақ жұмыс тек бір көшірмеде және жадта орындалады. Әдетте, DLL қажет болғанша жадқа жүктелмейді, бірақ жүктелгеннен кейін, оның функциялары мен ресурстарын кез келген бағдарламаға қол жетімді етеді. DLL-ті құру бағдарламалардың икемділігін арттырады. Мысалы, қолданба пайдаланатын барлық мәтіндерді, мысалы, жапсырмалар, кеңестер және тағы басқа бірнеше кітапханаларды жасауға болады. Осы кітапханалардың әрқайсысында бір немесе басқа тілде мәтіндер болуы мүмкін: орыс, ағылшын, неміс. Содан кейін, пайдаланушы осы кітапханалардың қайсысын қолданатынына қарай бағдарламалар тиісті тілде онымен байланысады.

DLL көмегімен сіз қосымшаларды түрлендіруге көмектесе аласыз. Кәдімгі түрде орындалатын және өтінімге енгізілген кейбір функциялар бар. Бұл бағдарлама пайдаланушыларға таратылады, содан кейін осы функцияларды кеңейте алатын жаңа функциялар бар, мысалы, функцияларды теңшеуге болатын жаңа файл пішімдері.

Бұл жаңа мүмкіндіктерді іске асыру үшін алдыңғы функцияларды

қайта жазу, жаңа қосымшаны жасау және оны барлық пайдаланушыларға жіберу қажет болады. Егер функциялар DLL түрінде жүзеге асырылса, онда бағдарламаларды қайта құрастырудың қажеті жоқ, жаңа атаумен бірдей жаңа атаумен және кішкентай DLL файлын жасау жеткілікті, және пайдаланушыларға жаңа DLL ғана таратады.

DLL-дің тағы бір артықшылығы - олар басқа алгоритмдік тілдерде жазылған қосымшалармен қолданылуы мүмкін. Мысалы, сіз C ++, Visual Basic, Access Basic және т.б. кітапханаларын пайдалана аласыз. Delphi-де жасалған кітапханалар DLL-ке аталатын кез келген жүйені, олар жазылған алгоритмдік тілдерге қарамастан, пайдалана алады.

ДИНАМИКАЛЫҚ КІТАПХАНАНЫ ҚҰРУ МЫСАЛДАРЫ

10.2.

DLL файлын құрудың мысалын қарастырайық, онда біз өз атына пайдаланушыдан тілқатысу терезесінің нысанын орналастырамыз.

Файлды / Жаңа / Басқа пәрменін бастау арқылы DLL файлын жасаңыз және New Items терезесінде New DLL Wizard белгішесіндегі оны таңдап алыңыз. Нәтижесінде редактор терезесіндегі мәтін пайда болады:

```
library Project1;  
{ Important note about DLL memory  
management: ShareMem must be the  
first unit in your library's USES clause AND  
your project's (select  
Project-View Source) USES clause if your DLL  
exports any procedures or  
functions that pass strings as parameters or  
function results. This  
applies to all strings passed to and from  
your DLL- even those that  
are nested in records and classes. ShareMem  
is the interface unit to  
the BORLNDMM.DLL shared memory manager,
```

```

which must be deployed along
with your DLL. To avoid using BORLNDMM.DLL,
pass string information
using PChar or ShortString parameters. }
uses
SysUtils,
Classes;
{$R *.res}
begin
end.

```

Бұл DLL-нің дайындық модульның дайындығына өте ұқсас. Негізгі айырмашылық - бірінші жолда бөлімнің орнына Кітапхана кілтсөзі. Save your special katata for your project that has been invoked by MyDLL. Жоғарыда келтірілген кітапханада MyDLL-мен автоматикадағы кітапханада айтылады. Жобаның кітапханасының кілтсөзімен жинақталғанда, dll кеңейтімі бар кітапхана файлдары жасалады. Түсініктеме DLL функциясындағы аударудың ерекшеліктеріне және олардың жол мәндерін алуға қатысты. Түсініктеме DLL жадының ұйымдастырылуында маңызды бақылау болып табылады. ShareMem модулі алдымен кітапханадағы және жобада Пайдалану бөліміне енгізілуі керек (жоба / көзін көру). Егер DLL кейбір рәсімдерді немесе функцияларды экспорттайды, не олардан жолдар параметрлері немесе функционалдық мәндер түрінде алынады, бұл DLL-ден берілген немесе алынған барлық жолдарға, тіпті жазбада енгізілгендерге де қолданылады және сабақтар. ShareMem - BORLNDMM.DLL ортақ жады менеджерінің интерфейс модулі. BORLNDMM.DLL пайдалануды болдырмау үшін, жол туралы ақпаратты PChar немесе ShortString сияқты параметрлер арқылы беріңіз.

Пікірдің мәтіні, әрине, кодтан алынуы мүмкін, Пайдаланушы операторы SysUtils және Classes модульдерін кітапханаға қосады. Көптеген жағдайларда бұл модульдер талап етілмейді, сондықтан олардың қосылымы кодтан алынуы мүмкін. Алайда SysUtils модулін қосу кітапхана функциялары мен рәсімдерін орындау кезінде ерекше жағдай туғызу мүмкін болған жағдайда өте пайдалы болуы мүмкін. Бұл модуль болмаған жағдайда, кітапханада ұсталмаған ерекшелік бағдарламаны апатқа ұшыратады, тіпті шақырылатын бағдарламаның жабу функциялары тіпті шақырылмайды, жай ғана тапсырма жадтан босатылады. Егер SysUtils модулі кітапханаға қосылса, онда ол туралы ескертілмейтін оқиға қоңырау шалу бағдарламасына берілетін болады және оператордан басқа әдеттегі әрекет арқылы өңделуі мүмкін.

Дегенмен, DLL-ті әртүрлі тілдерде жасалынған бағдарламалардан шақыруға болатындығын ескерсек, онда барлық ерекшеліктерді өңдеп, шақырылған бағдарламаға қате көрсететін кейбір арнайы жалауды аудару керек.

```
{ $R *.res }
```

Degenman, DLL-ті әртүрлі тілдерде жасалынған бағдарламалардың шақырылуы мүмкін екенін ескерсек, ондағы барлық ерекшеліктерді өңдеңіз, шақырылған бағдарлама кейте ксерететін кілтсөз арнайы жалауда аудару керек. Басталу туралы мәлімдеме DLL жүктелген кезде орындалатын есептерді бастайды. Олар кейбір қажетті түзетулер жасай алады. Егер бұл параметрлер алдын ала қабылданбаған болса, онда бастапқы мәлімдеме кодтан алынуы мүмкін. Осылайша, кодтағы ең қарапайым жағдайларда, сіз тек кітапхананың жетекшісін және соңғы ақырғы мәлімдемені қалдыра аласыз.

Біз осы кілтпен шифрланған немесе шифрланған жолды қайтаратын жолға және кілтке жіберілетін DLL функция кодын орналастырамыз. Кейінірек, осы DLL-де пайдаланушыға оның атын сұрайтын диалогтың басқа нысаны орналастырылады. Төменде осы DLL кодының үзінділері келтірілген:

```
Library MyDLL;  
Function Code (S: PChar; Key: integer) :  
    PChar;  
  
stdcall;  
Var i:integer;  
Ss: string;  
Begin  
    ss:=S;  
    for i:=1 to length(S) do  
        ss[i]:=char(ord(ss[i]) xor key);  
    code:=PChar(ss);  
end;  
exports  
Code;  
end.
```

Функция кодын түсіндірейік. S жолын және кілттің барлық мәнін алады. Содан кейін, циклде, бастапқы жолдың әрбір таңбасы арнайы НЕМЕСЕ оператор хороны арқылы берілген таңбамен ауыстырылады және осы таңбаның индексі мен Кілт пернесі болады.

Нәтижесінде жол шифрланған. Егер сол функцияны бірдей кілтпен бірнеше рет шифрланған жолға шақырсаңыз, онда шифрлеу

орындалады және функция түпнұсқа жолын қайтарады. Бұл принцип бойынша әдетте мәтінді шифрлау жүргізіледі, содан кейін қосымша шифрлау ережелері қолданылуы мүмкін. Әрбір келесі кейіпкердің кілтін кейбір ережелермен өзгертуге болады. Мысалы, егер шифр операторын оператормен ауыстырсаңыз:

```
ss[i]:=char(Ord(ss[i]) xor (key + I mod 2));
```

онда кілт әр екі таңба үшін циклды орындайды.

Кодекстің функциясын S параметрінің және қайтару мағынасының «Жол» ретінде анықталатын болса, іске асыру қарапайым еді. Бұл жағдайда сізге String түріндегі жергілікті айнымалы мәнді енгізіп, жолдарды түрлендірудің қажеті жоқ. Бірақ мұнда Delphi кітапхананың кодын жасаған және бұрын берілген мәтінді орналастырған түсініктемелерді есте сақтау пайдалы. Егер сіз DLL-ке жіберілгіңіз келсе немесе одан жолды алсаңыз, ShareMem-ды пайдалану тарауында көрсетуіңіз керек.

Бұл кітапхана BORLNDMM.DLL интерфейс модулі, мұндай жады ұйымдастыруды қамтамасыз етеді, онда DLL-мен ұзын жолдарды айырбастауға болады. Сонымен қатар, DLL-ді қолданатын қосымшаның негізгі файлында ShareMem модулі бағдарламада бірінші кезекте көрсетіледі. Сондықтан, DLL пайдаланушыларына осындай сұрауларды ұсынудан және PChar сияқты параметрлер арқылы жолдардың берілуін қолданған дұрыс.

DLL процедураларынан және функцияларынан экспортталған тақырыптардан кейін анықталған stdcall спецификациясы параметрлерді беру кезінде белгілі бір конвенцияларды анықтайды; атап айтқанда параметрлерін солдан оңға қарай жүйелеу. Осындай аудару Windows API-де және DLL деп аталатын қосымшаларды жасау үшін қолданылатын бағдарламалау тілдерінде қабылданады. Демек, stdcall іріктеуішін әрдайым пайдалану ұсынылады, әйтпесе тіркелушінің спецификациясы әдепкі бойынша қабылданады. Бұл параметрлердің жылдам алмасуын қамтамасыз етеді, бірақ мұндай DLL тек Pascal-да жазылған қосымшаларда қолданыла алады.

DLL-кодындағы барлық функциялар мен рәсімдер мәтіндерінен кейін экспортты жеткізу орналасқан. Ол DLL арқылы экспортталатын процедуралар мен функцияларды, яғни сыртқы бағдарламаға қоңырау шала алатындарды тізеді. Экспортталған функциялардан басқа, DLL-де ішкі қолдануға арналған утилиталарда сипатталған кейбір қосалқы функцияларды сипаттауға болады. Әрине, мұндай коммуникациялар экспорттық ұсынысқа қосылмаған.

Егер экспорттау процедураның немесе функцияның атауын ғана көрсетсе, онда сыртқы бағдарлама оны шақыра алатын атаумен анықталады. Бірақ кілт сөздің атауымен сыртқы қолдануға арналған басқа атауды белгілей аласыз. Мысалы, жоғарыда келтірілген экспорттау туралы мәлімдемені келесідей ауыстырсаңыз:

```
exports
DoMy name 'My',
Code;
```

онда сыртқы бағдарламаның DoMy функциясы «Менің» атын қолданып қоңырау шалуы керек.

Сыртқы бағдарлама DLL функцияларына және рәсімдеріне аттар емес, индекстер арқылы қоңырау шалуы мүмкін. Көрсеткіштер экспортта тізімделген тегі бойынша автоматты түрде тағайындалады. Индекстер кілт сөз индексін пайдалану арқылы мәжбүр болады. Мысалы:

```
exports
DoMy index 1,
Code index 2;
```

Егер индекстермен қоңырау шалу мүмкін болса, индекстерді DLL-ге жаңа функциялар мен рәсімдерді қосқан кезде өзгертуі үшін индекстерді ұқсас жолмен орнату керек. Басқа тілдерге және жүйелерге түсініксіз функциялар атауында кейбір таңбалар пайдаланылса, индексті шақыру қажет болуы мүмкін. Шын мәнінде, кілт сөз индексі бар индекстерді көрсету Windows үшін ғана рұқсат етіледі, басқа платформаларда индекстер автоматты түрде жасалады.

Экспорттау туралы ереже DLL-ге бірнеше рет енгізілуі мүмкін және кодтың кез келген жеріне орналастырылады, бірақ оның функциялары мен рәсімдерін жариялағаннан кейін ғана. Егер DLL шамадан тыс жүктелген функциялар болса, олар параметрлер тізімімен қатар экспорттау мәлімдемесіне қосылып, оларды әртүрлі экспортталған аттармен беру керек. Мысалы, кітапханада екі функция сипатталсын:

```
function Average (X, Y:      integer):
                        real; overload;
stdcall;
begin
Average := (X + Y) / 2
end;
function Average (X, Y: real): real;
```

```

overload; stdcall;
begin
Average := (X + Y) / 2
end;

```

Одан кейін, экспорт мәлімдемесінде олардың декларациясы ұқсас болуы керек

```

exports
Average (X, Y: integer) name 'AverageInt',
Average (X, Y: real) name 'AverageReal';

```

Әртүрлі атаулар қоңырау шалушы бағдарламада шамадан тыс жүктелген ретінде осы функциялардың қолданылуына кедергі келтірмейді. Енді DLL-дегі құпия сөзді енгізу пішінін енгізіңіз. Әрбір қосымшаға мұндай форманы енгізу, оны DLL-де үнемді үнемдеу және оны қажет болған барлық қосымшалардан шақырту ұсынылмайды. Сонымен қатар, егер сіз бұл нысанды уақыт ішінде ауыстырғыңыз келсе, тек DLL-ді ғана өзгертуге болады және барлық қосымшалар өзгеріссіз қалуы мүмкін.

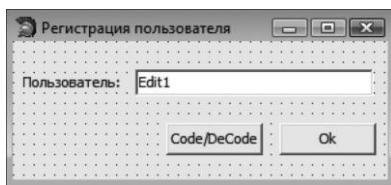
DLL-ге пайдаланушы атын сұрайтын диалогтың қарапайым нысанын енгізейік. DLL файлындағы жаңа пішінді қамтитын Файл / Жаңа / Пішін пәрменін орындаңыз. Белгіні, Edit1 және Edit батырмасын басыңыз, пішінге ModalResult сипатында, mrOk орнатыңыз (10.1-суретті қараңыз).

Пішін модулін UMyDialog атымен сақтаңыз. Пішін модулінде бұл жағдайда ешқандай кодты жаза алмайсыз және DLL коды келесідей өзгертілуі керек:

```

Library MyDLL;
uses
Forms {для переменной Application},
UMyDialog in 'UMyDialog.pas' {form1};
function MyDialog (User: PChar) : PChar;
stdcall;
Var form: TForm1;

```



Пайдаланушы

Сурет. 10.1. Диалогтың нысаны

begin

```
Form := TForm1.Create (Application);  
Form.Edit1.Text := User;  
Form.ShowModal;  
Result := PChar (Form.Edit1.Text);  
Form.Free;
```

end ;

exports

```
Code,  
MyDialog;
```

end .

Жоғарыда келтірілген кодта, MyDialog функциясы диалогқа қоңырау шалып, пайдаланушы 1-дегі терезеде көрсетілген жолды қайтарады. Параметр ретінде User string функциясына - пайдаланушы атының бастапқы мәні беріледі. Функцияда жергілікті айнымалы пішін. Алғаш орындалатын оператор тілқатысу пішінінің данасын жасайды, Пайдаланушы жолы осы пішіннің өңдеу терезесіне енгізіледі. Содан кейін, ShowModal әдісін қолданып, пішін пайдаланушыға модаль ретінде көрсетіледі. Пайдаланушы осы пішінмен жұмыс жасағаннан кейін редакциялау терезесінің мәтіні функциямен қайтарылған мәнде сақталады және пішін жадтан Тегін әдіс арқылы жойылады.

Осылайша, диалогқа шақыру басқа функцияның шақыруынан ерекшеленбейді, тек пайдалану туралы мәлімдеме пішін модуліне және Forms модуліне сілтеме қосады, онсыз компилятор қолданбаның айнымалы мәнін түсінбейді.

Тапсырма

Қолданбаны қотару кезінде пайдаланушыны анықтау үшін кіру және құпия сөзді DLL жасаңыз.

БАҚЫЛАУ СҰРАҚТАРЫ

1. DLL-ні пайдаланудың артықшылықтарын сипаттаңыз.
2. stdcall іріктеуішті пайдалану дегеніміз не?
3. Operator операторында пайдаланылатын ShareMem дегеніміз не?
4. SysUtils модулін қосу дегеніміз не?

ПАЙДАЛАНУШЫ ҮШІН ҚҰЖАТТАМА ЖАСАУ. БАҒДАРЛАМАЛЫҚ ӨНІМДЕРДІҢ АНЫҚТАМАЛЫҚ ЖҮЙЕСІН ҚҰРУ

11.1. «ПАЙДАЛАНУШЫНЫҢ НҰСҚАУЛЫҒЫ»

Пайдаланушыларға арналған құжаттарды ресімдеу, өз кезегінде, пайдаланушы, әдетте, бағдарламалық қамтамасыз етуді дамыту саласында кәсіби емес екеніне байланысты өз ерекшеліктеріне ие. «Пайдаланушы нұсқаулығын» әзірлеу кезінде, «Бағдарламалық нұсқаулықтың бірыңғай жүйесі» ГОСТ 19.504-79 және «Оператордың нұсқаулығы» ГОСТ 19.505-79 бірыңғай бағдарламалық құжаттама жүйесінің жұмыс құжаттары негіз бола алады. «Пайдаланушы нұсқаулығы», бәлкім, келесі бөлімдерді қамтиды:

- «Бағдарламаның мақсаты»;
- «Бағдарламаны орындау шарттары»;
- «Бағдарламаны орындау»;
- «Пайдаланушыға хабарламалар».

Бағдарламаның ерекшеліктеріне қарай бөлек бөлімдерді біріктіруге немесе жаңа енгізуге болады.

«Бағдарламаны тағайындау» бөлімінде сіз бағдарламаның мақсаты туралы ақпаратты, оның функцияларының қысқаша сипаттамасын, енгізілген әдістерді және ықтимал қосымшаларды көрсетуіңіз керек.

«Бағдарламаны орындау шарттары» бағдарламаны орындау үшін қажетті (ең төменгі және (немесе) максималды аппараттық және бағдарламалық құрамы және т.б.) болуы тиіс.

«Бағдарламаны іске қосу» бөлімінде бағдарламаны жүктеуді, орындауды және аяқтауды, функциялардың сипаттамасын және пайдаланушы интерфейсін қамтамасыз ететін пайдаланушы әрекеттерінің дәйектілігін көрсету керек. Мысал ретінде, біз пайдаланушы нұсқаулығының кішкене фрагменті береміз.

- «Пайдаланушыға арналған хабарламалар» бөлімінде

бағдарламаны орындау барысында берілген хабарламалар, олардың мазмұнын сипаттайтын және тиісті пайдаланушы әрекеттерін сипаттайтын хабарлар берілуі керек (бағдарламаның сәтсіздігі, бағдарламаны қайта іске қосу және т.б.).

Бөлімдердің мазмұны түсіндірме мысалдармен, кестелермен, диаграммалармен, графиктермен суреттелуі

мүмкін. Пайдаланушы нұсқаулығындағы қосымшаларда нұсқаулықтың бөлімдеріне орналастыруға болмайтын түрлі материалдар енгізілген.

Пайдаланушы құжаттамасын жазу бойынша ұсыныстар:

- Пайдаланушылардың мүдделерін ескере отырып, нұсқаулық пайдаланушы талап ететін барлық нұсқауларды қамтуы тиіс;

- нақты анықтап, қысқа сөйлемдерді қолданыңыз;

- Кейбір терминдерді пайдалану қажет болса, техникалық жарнамадан және тар мамандандырылған терминологиядан аулақ болыңыз;

- дәл және ұтымды, ұзақ және күрделі бағыттаушылардың әдетте ешкім оқымауы мүмкін, мысалы, ұзақ уақыт бойы сипаттаудан гөрі, пішінді бейнелеу жақсы.

- Нақты және ұтымды, ұзын және күрделі нұсқаулықтарды әдетте

ешкім оқымайды, мысалы, ұзындықты суреттеуден гөрі, нысанын

пішінін жақсарту жақсы.

Пайдаланушыны бағыттау үшін әзірленген материалдың негізінде бағдарламалық жасақтама өнімінің анықтамалық жүйесі құрылады немесе жай ғана көмек файлы. Бағдарламаны пайдаланушыны бағдарламамен жұмыс істеу және осы бағдарламалық жасақтама өнімінің қажеттілігі туралы толық және толық ақпаратпен қамтамасыз ету үшін әзірленген.

Эталондық жүйе келесі талаптарға сай болуы керек:

- бағдарламаның толық сипаттамасын беруге;

- графикалық материалдарды;

- бағдарламаның кез келген түрінен қоңырау шалу үшін қол жетімді болуы;

- контекстік сипаттама және ыңғайлы ақпарат іздеу жүйесі бар

- Мүмкіндігінше аз мөлшерде болыңыз.

АЖС «КӨТЕРМЕ САУДА ҚОЙМАСЫ»ҮШІН «ПАЙДАЛАНУШЫ НҮСҚАУЛЫҒЫ» ҚҰЖАТЫНЫҢ ҮЛГІЛЕРІН ӘЗІРЛЕУ ҮЛГІСІ

Мысал ретінде, «Бағдарламаны іске қосу» бөлімінен «Тауарларға тапсырыс беру» құжатымен жұмыс істеу тәртібін қарастырыңыз.

«Тауар орамы» құжаты сатып алушының тауарлық-материалдық құндылықтарды сатып алу ниеті туралы алдын-ала келісімді ресімдеуге арналған. Осы құжаттың орындалуы міндетті емес, оның негізгі мақсаты баспа түрін және оның негізінде жүк жөнелтімінің үзінді көшірмесін жасау болып табылады. Құжаттың экрандық нысаны тақырыптан және кесте бөлігінен тұрады (5.1 суретті қараңыз). Тақырып: «Клиент» - «Клиенттер» каталогының элементі;

«Шарт» - тапсырыс жасалатын шарт;

«Төлем түрі» - қолма-қол ақша немесе банк аударымы. Құжаттың кесте бөлігін толтыру құжаттың жаңа жолын әдеттегі енгізу арқылы жүзеге асырылады. Бұл «Тауарлар» каталогын ашады, онда сіз құжаттың элементтерінің ерікті санын таңдай және енгізе аласыз (таңдалған өнімдегі тінтуірдің сол жақ батырмасын екі рет басу немесе Enter пернесін басу). Барлық таңдалған тауарлар құжаттың кестесінде жазылған.

Егер қажетті «Баға» «Тауарлар» каталогында толтырылған болса, онда реквизиттердің құны кесте бөліміне көшіріледі. Бұл сома автоматты түрде есептеледі. Құжатты толтыру нысаны қосымша басқару элементі - «Басып шығару» түймесіне ие. Бұл батырма басылғанда басып шығарылған тапсырыс формасы жасалады, оны принтерге шығаруға болады. Тапсырыс құжаты бухгалтерлік жазбаларды құрмайды. Экран нысанын толтырғаннан кейін ОК түймешігін басыңыз.

АНЫҚТАМАЛЫҚ ЖҮЙЕНІ ДАМУ

Анықтама жүйесі - Windows жүйесінің ажырамас бөлігі болып табылатын Winhelp бағдарламасы hlp-файлында көмек туралы ақпаратты көрсетеді. Көмек жүйесін құру процесі (hlp-файл) келесі екі қадамның реті ретінде ұсынылуы мүмкін:

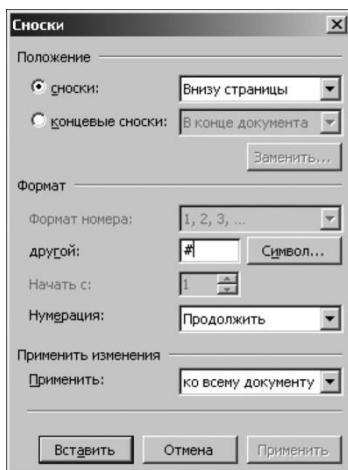
1) анықтамалық ақпаратты дайындау (анықтамалық ақпараттың құжаттық файлын құру);

2) анықтамалық файлды анықтама файлына түрлендіру.

Анықтамалық құжаттардың файлдары белгілі бір құрылымның rtf файлы болып табылады. Біз осы файлдардың тақырыбына (тақырып файлдары) қоңырау шалдық. Анықтамалық файлдардан (hlp) тақырыптық файлдардан алу үшін олар Көмекші семинар бағдарламасы (hwc.exe) арқылы өңделуі (жинақталуы) керек. Түпнұсқалық мәтіндер мен суреттерден басқа, тақырыптық файлдар Winhelp әртүрлі мүмкіндіктерін іске асыру үшін өздерінің тақырыптарымен мазмұн кестесін байланыстыратын сілтемелер арқылы сілтемелер жасау үшін қажетті ақпаратты қамтитын арнайы түзетулерді қамтуы мүмкін.Әдетте, көмек бірнеше тақырыпты қамтиды және осы тақырыптарға баруға болатын мазмұндар кестесі бар. Ең қарапайым нұсқасы: тақырып біреуі және мазмұны жоқ. Бұл жағдайда сізге не қажет екенін жазыңыз және оны .rtf кеңейтімі бар файлға сақтаңыз. Бірнеше тақырып жасау үшін, процесс біршама күрделене түседі.

Әрбір тақырып жаңа бетке қатты көшу арқылы аяқталады. Мұны істеу үшін, тақырып аяқталғаннан кейін, мәзірде (MS Word) «Кірістіру» ішіндегі «Break / Start a new page» тармағын таңдау керек.

Мазмұнның кестесінен көмекке дейін тақырыпты алу үшін, оның идентификаторын көрсетуіңіз керек. Мұны орындау үшін, мәтінді курсорлардан басталатын (тақырыптың басталуы немесе басқа орын) басталатын мәтінді орналастыру керек, арнайы жазбаны, атап айтқанда соңғы ескертуді енгізіңіз. Ескертудің таңбасы «#» белгісін таңдайды (11.1-сурет). Тақырып мәтіні - бұл түсіндірме мәтіні. Мысалы, «Қолдау» тақырыбын жасаңыз, оны басқа тақырыптардан бөліңіз және оны қолдау үшін орнатыңыз. Ол үшін тақырыптың тақырыбының жанындағы енгізу таспасын орналастырыңыз да, мәзірден «Кірістіру / Ескерту ...» дегенді таңдаңыз. Тілқатысу терезесіндегі «Соңы» сілтемесін таңдап, нөмірлеу «басқа», терезеде «#» (тырнақшасыз) жазу үшін терезеде. ОК түймешігін басамыз, сілтеме қосылады және тасымалдау автоматты түрде сілтеме мәтініне аударылады. Қолдау жазамыз. Аяқталды. Барлық анықтамалық тақырыптар үшін бірдей қайталаңыз. Файлды сақтаңыз.



11.1. Сурет

Анықтамалық семинар бағдарламасын іске қосыңыз (файл Hcw.exe). Файлды / жаңа / анықтама мәзірінің мәзірін қолданып жаңа жоба жасаңыз. Панельдің оң жағында бірқатар файлдар бар, олардан файлдар таңдалады. Тілқатысу терезесінде тақырыптық файлды қосып, осы терезені жабыңыз. Жобаны сақтаңыз - бұл кеңейтім hrj (Help Project) бар файл болады.

Бірінші іске қосылғаннан кейін, Help Workshop hrj файлдарына, сондай-ақ көмек файлдарына (cnt) сілтеме жасайды, осылайша оларды екі рет басу арқылы ашуға болады. Анықтамалық файлды жасау үшін Сақтау және Құрастыру түймешігін басыңызға болады. Жаңа терезе жинақтың нәтижесі туралы хабармен ашылады. Бәрін тәртіпке келтірейік делік, біз осы терезені жабамыз. Енді біздің жобамыз (. Hрj) орналасқан каталогта анықтама файлы пайда болуы керек. Алайда, егер сіз оны екі рет бассаңыз, бірінші тақырыпты ғана көре аламыз; Барлық тақырыптарды көру және олардың арасында шарлау үшін, мазмұн файлының кестесін қосу керек.

Анықтама файлы қарапайым мәтін пішіміне ие, бірақ оны көмекші семинарда жасау ыңғайлы. Бұл әрекетті орындау үшін, Help File File (Файл) / New (Жаңа) / Help Contents (Анықтама мазмұны) мәзірінің Help мәзірін таңдаңыз. Терезенің жоғарғы бөлігінде негізгі анықтама файлының атауын (әдетте бірнеше файл болуы мүмкін) және анықтама кестесінің атауын енгізу керек. Сонымен қатар, Өңдеу түймесін басқан кезде ашылатын диалогтық терезеде де болады.

Енді нақты мазмұн кестесін жасаңыз. Ол келесі екі түрдегі элементтерден тұрады: Көмек тақырыптары, олар бірнеше

тақырыпты қамтиды және көмек кітапшасының мазмұнын кестесінде және тақырыптардың өзі ұсынылады - анықтама белгісінің мазмұнын кестеде көрсетілген мәтіндер мен суреттер оған сұрақ белгісімен (сіз кез келген анықтаманың мазмұны кестесінде қараңыз).

Сондай-ақ, мазмұндар кестесінде макростарды кірістіріп, файлдарды (қоса) қосуға болады. Панельдің оң жағында мазмұн кестесінің элементтерін қосу және басқару үшін түймелер жиынтығы бар (төменге қосу - төменде қосу, жоғарыдан қосу - жоғарыдан қосу, Оңға жылжыту - Оңға жылжыту, Солға жылжыту - Солға жылжыту, Өңдеу, Жою). Олардың көмегімен біз мазмұн кестесінің қажетті құрылымын құрдық.

Тілқатысу терезесінде бөлімді қосқанда, оның атауын ғана көрсету керек; Тақырыпты қосқанда, атау, ID (соңғы кезі енгізілген кезде rtf файлында көрсетілген біз), анықтама файлының атауы және терезенің аты көрсетіледі. Егер тақырып негізгі файл ретінде көрсетілген анықтамалық файлда болса, анықтама файлының атауы көрсетілмеуі керек. Терезенің атын да көрсету мүмкін емес - тақырып әдепкі терезеде ашылады. ТОС файлы (ол .cnt кеңейтімі) анықтамалық файлмен бірдей каталогта сақтау керек, ол жақсырақ анықтама файлы сияқты.

Енді қайтадан жоба файлы .hрj ашып, Параметрлер түймешігін басу қажет, Файлдар қойындысындағы ашық диалог терезесінде біздің мазмұн файлы көрсетіңіз. Тілқатысу терезесін жабыңыз, Сақтау және құрастыру түймешігін қайта басыңыз. Енді көмек файлының белгішесін екі рет бассаныз, барлық тақырыптарға қол жеткізуге болатын мазмұнның кестесін аша аласыз.

Әдетте, көмек тақырыптары басқа бөлімдерге сілтемелерді қамтиды. ұғымдарды (сөз) анықтамалық жүйесінде, басқа Анықтама бөлімінде үшін шапшаң әкеледі, ол таңдау, негізгі мәтін анықтамалық түсті және негізгі сәттері бастап айырмашылығын атап өтті. мәтінді анықтамалық сөз-буын дайындау барысында, келесі Анықтама бөлімінде көшуді қамтамасыз ету болып табылады, ол таңдау, ол өтпелі жасалуы тиіс ID анықтамалық бөлімін, қоюға жоқ кеңістікте бар қос сызықпен, және бірден осы сөздің артында атап өткен жөн. Кірістірілген идентификатор жасырын мәтін ретінде ресімделуі керек.

Мәселен, суретке сурет салу керек. Оны MS Word бағдарламасына әдеттегі жолмен rtf құжатына қосу арқылы жасауға болады. Бір сурет бірнеше жерде пайдаланылса, сурет бір көшірмеде сақталатын етіп суретке кірістіруге арналған арнайы таңбаларды қолдануға болады.

Көмек алу үшін Delphi ортасында жазылған бағдарламада қол жетімді болу үшін бағдарламаны .hlp файлына бағыттау қажет. Анықтамалық файлды орналастырудың ең оңай жолы - оны орындалатын файл сияқты бір қалтада орналастыру.

Қолданбаның анықтамалық файлын анықтаңыз: Delphi-дегі жобаны (қосымшаларды) ашыңыз және Project / Options / Application (Бағдарлама / Параметрлер / Қолданба) мәзірін таңдап, Анықтама файлы өрісінде анықтама файлының атауын енгізіңіз. Сонымен қатар файлдың атын жолсыз көрсетуіңіз керек. WinHelp анықтамалық файлды табуға тырысқан кезде, көрсетілген каталогтардың бірі бағдарламаның орындалатын файлы орналасқан жері болып табылады. Басқа әдіс: келесі бағдарламаны негізгі бағдарлама пішінінің OnCreate оқиға өңдегішіне кірістіріңіз:

```
Application.HelpFile:=ExtractFilePath(Application.ExeName)+ + «MyHelp.hlp»;
```

Мұнда MyHelp.hlp — анықтама файлының атауы.

Бағдарламадағы мәзірден мазмұн кестесін келтіру үшін функцияны пайдалануыңыз қажет

```
Application.HelpCommand(HELP_FINDER, 0) ;
```

Біз анықтаған анықтама бөлімдерінің біріне өту үшін функцияны шақыру қажет

```
Application.HelpJump('MyTopic'),
```

мұнда MyTopic - тақырып идентификаторы.

Көмек алудың бір жолы - F1 пернесін басу. F1 пернесін басу арқылы контекстке сезімтал көмек шақыруын реттеуге болады, бір немесе басқа басқару элементі белсенді болғанда. Мұны орындау үшін, көмек туралы тиісті тақырып нөмірге тағайындалады және содан кейін бұл санды басқарманың HelpContext сипатына тағайындайды. Көмек тақырыбына арналған нөмірлерді анықтау үшін, HelpWorkshop бағдарламасындағы көмек жобасын ашып, терезенің оң жағындағы Карта түймешігін басыңыз. Тілқатысу терезесіндегі Қосу түймешігін нұқып, тақырып кодын және кез келген нөмірді енгізіңіз, барлық қажетті тақырыптар үшін (әрқайсысы өз нөмірімен) қайталаңыз, терезені жабыңыз және Сақтау және құрастыруды қайтадан басыңыз. Содан кейін, нысанның инспекторы терезесіндегі Delphi-де қажетті элементтерге керекті сандарды белгілеңіз.

Тапсырма

Тапсырманың нұсқасы үшін «Пайдаланушы нұсқаулығы» бағдарламалық құжатын жасаңыз. Жасалған құжаттың негізінде көмек жүйесін құрыңыз.

БАҚЫЛАУ СҰРАҚТАРЫ

- 1.Бағдарламалық құжат туралы қандай ақпарат бар?
- 2.Пайдаланушы нұсқаулығының негізгі бөлімдерін атаңыз.
- 3.Көмек көрсету жүйесін дамытудың негізгі кезеңдері қандай?
- 4.Анықтамалық файл үшін мазмұн кестесін қалай құрастыруға болады?
- 5.Бір бөлімде басқа бөлімге сілтеме қалай жасау керек?

12 тарау

БАҒДАРЛАМАЛЫҚ ЖАСАҚТАМАНЫ ОРНАТУ

12.1. ОРНАТУДЫҢ МАҢЫЗДЫЛЫҒЫ

Көптеген тұтынушыларға арналған бағдарламаны орнату үдерісі өте қиын. Ол, әдетте, тек қана каталог жасайды және орындалатын файлдарды және деректер файлдарын аралық ортасынан ғана емес, сонымен қатар жүйені орнатады. Бағдарламалық жасақтаманы орнату бағдарламалық жасақтама өнімді пайдаланушылардың компьютерлеріне орнату процесін автоматтандыруға мүмкіндік береді, оларды әртүрлі орнату сценарийлерін таңдауға және одан әрі пайдаланудың дұрыстығын қамтамасыз етуге мүмкіндік береді.

Сонымен қатар, бағдарламалық жасақтаманың әзірлеушілері рұқсат етілмеген қондырғыларға тыйым салудың жақсы мүмкіндігі бар. Бағдарламалық жасақтаманы орнату процесі «Пайдаланушы нұсқаулығы» бөлімінде сипатталуы керек. Орнату кезінде орындалатын тапсырмалар әдеттегі болып табылады, сондықтан орнату бағдарламасының, дәлірек айтқанда, кодтың бір жолын жазбастан орнату дискісін жасау үшін тез пайдалануға болатын құралдар бар. Бағдарламалық жасақтама өнімді орнату үшін көптеген құралдар бар, олардың бірі - InstallShield Express бағдарламасы.

12.2. БАҒДАРЛАМАЛЫҚ ЖАСАҚТАМАНЫ ОРНАТУ ПРОЦЕСІ

Орнату бағдарламасын тікелей InstallShield Express бағдарламасында жасауды бастамас бұрын, төмендегілерді орындауыңыз керек:

12.1-кесте. Орнатуды қажет ететін файлдардың тізімі

Файл	Кездесу	Қайда орнату керек

1) пайдаланушының компьютеріне орнату қажет файлдардың тізімін жасайды;

2) мәтін редакторын пайдаланып, EULA (Түпкі пайдаланушы лицензиясы келісімі) RTF файлдарын және қысқаша анықтаманы (Readme файлы) дайындаңыз.

Пайдаланушы компьютеріне көшірілуі қажет бағдарлама файлдарының тізімі келесі кестеде келтірілген. 12.1.

Файлдар тізімінен кейін, InstallShield Express бағдарламасын іске қосу керек, Файл мәзірінен Жаңа параметрін таңдап, жоба файлының атауын Жобаның атауы мен Орналасу өрісіне енгізіңіз.

ОК түймешігін басқаннан кейін орнату бағдарламасын құруға арналған жоба терезесі ашылады. Терезенің сол жағында құру үрдісі және құрылатын бағдарламаның параметрлерін анықтау үшін қолданылатын командалар табылады. Конфигурация пәрмендері біріктіріліп, атау және тізбегі орнату бағдарламасын жасау үдерісінің мәнін көрсетеді. Топтық айдарлар нөмірленеді, орнату бағдарламасын орнату командаларды дәйекті таңдау арқылы орындалады. Негізгі терезенің оң жағындағы пәрменді таңдау нәтижесінде параметрлер тізімі пайда болады, орындалатын пәрмендер күсбелгілермен белгіленеді.

12.2-кесте. Орнатуды қажет ететін параметрлер тізімі

Параметр	Нені анықтайды	Мағынасы
Product Name	Орнатылатын бағдарламаның атауы	Sklad 2008
Product Version	Орнатылатын бағдарламаның нұсқасы	1.01.0001
INSTALLDIR	Бағдарлама орнатылған пайдаланушы компьютерінің каталогы	[ProgramFilesFolder] Sklad 2008

Орнатуды ұйымдастыру тобыңыздың командалары орнату бағдарламасының құрылымын көрсетуге мүмкіндік береді. Орнатылған бағдарламаны және оның әзірлеушісін анықтайтындарды қоспағанда, көптеген параметрлердің мәндері өзгеріссіз қалуы мүмкін. Мәндерін өзгерту қажет параметрлері кестеде келтірілген. 12.2.

INSTALLDIR параметріне назар аудару керек. Әдепкі бойынша, бағдарлама бағдарламаларға арналған каталогқа орнатылады деп болжанады.

Орнату бағдарламасын жасағаннан кейін, бағдарлама каталогы пайдаланушының компьютерінде және оның қайда орналасқандығы туралы білуге болмайды, содан кейін оның нақты аты-жөні орнына - [ProgramFilesFolder] - пайдаланылады. Бағдарламаны пайдаланушының компьютеріне орнату барысында, орнату бағдарламасы Windows тізілімінен бағдарлама каталогының атауын алады және бүркеншік атын осы атпен ауыстырады.

InstallShield Express бағдарламасында пайдаланылатын басқа бүркеншік аттар:

[WindowsVolume] - Windows орналасқан дискідегі түбірлік каталог;

[Windows Folder] - Windows каталогы, мысалы C: \ Windows;

[SystemFolder] - Windows жүйелік каталогы, мысалы C: \ Windows \ System32;

C: \ Program Files; [PersonalFolder] - жұмыс үстеліндегі «Менің құжаттарым» қалтасы (қалта орналасқан жер амалдық жүйенің нұсқасына және кіргеніңізге байланысты). Белгіленген бағдарламаның мүмкіндіктері орнатылған компоненттердің құрамымен анықталады. Мысалы, егер көмек жүйелік файлдары орнатылған болса, бағдарламамен жұмыс істеу барысында пайдаланушыға фондық ақпарат беріледі.

Ерекшеліктер пәрмені бағдарламаның мүмкіндіктерін анықтайтын және бөлек орнатуға болатын құрамдастар тобын жасауға (анықтауға) мүмкіндік береді. Компоненттерді топтарға бөлу бағдарламаны орнатуды қоса алғанда, пайдаланушы анықтайтын көп өзгерісті ұйымдастыруға мүмкіндік береді. Ең қарапайым жағдайда, Мүмкіндіктер тобы бір уақытта әрқашан орнату элементінен тұрады. Мүмкіндіктер тобына элементті қосу үшін Мүмкіндіктерді тінтуірдің оң жағымен нұқыңыз, пайда болатын мәтінмәндік мәзірден Жаңа мүмкіндіктер мүмкіндігін таңдаңыз және жаңа топтың атын енгізіңіз, мысалы, анықтама файлдары мен үлгілер. Осыдан кейін, Сипаттама өрісінде элементтің қысқаша сипаттамасын енгізіңіз және Түсініктемелер өрісіне түсініктеме жазыңыз.

Орнату түрлері пәрмені бағдарламаны орнату кезінде орнату нұсқасы кезінде пайдаланушының таңдауға болатындығын (орнату түрі диалогында) анықтауға мүмкіндік береді. Орнату әдеттегі, ең аз немесе теңшелетін болуы мүмкін. Орнатылған бағдарлама күрделі болса, бірнеше тәуелсіз компоненттерден тұрады, әдетте бұл мүмкіндік беріледі. Егер бағдарлама бір ғана орнату опциясын қабылдайтын болса - типтік болса, Минималды және Өзгертпелі жалаушалар қайта орнатылуы керек.

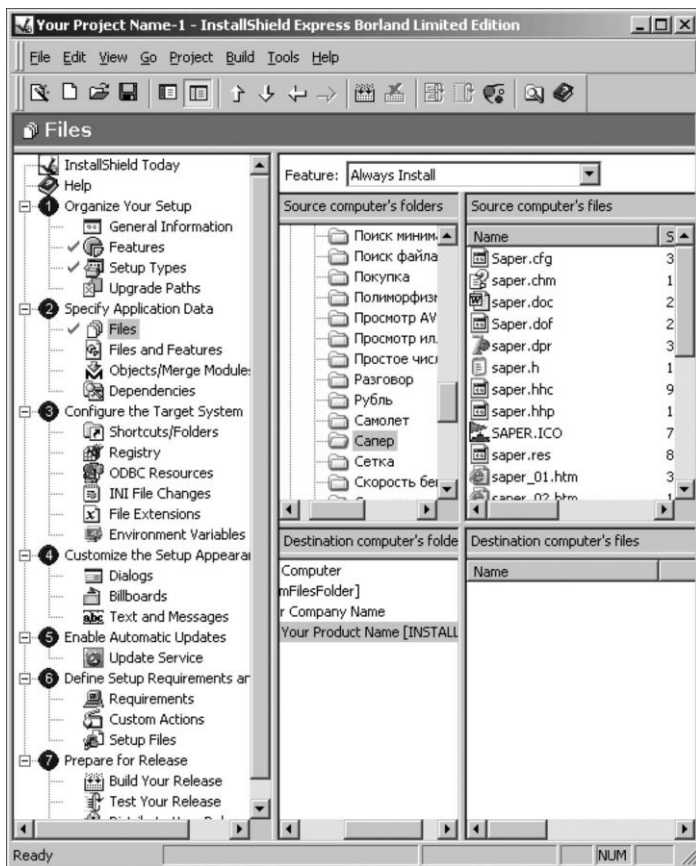
Бағдарламаны деректерді көрсету тобын көрсету (12.1-сурет) бағдарламаның компоненттерін пайдаланушы компьютеріне орнатуға мүмкіндік береді. Егер жобада бірнеше құрамдас топтар анықталса («Ерекшеліктер» пәрменін қараңыз), әр топ үшін құрамдас бөліктерді анықтау керек.

Файлдар пәрменін таңдау нәтижесінде терезенің оң жақ бөліктері аймақтарға бөлінеді (12.1-суретті қараңыз). Бастапқы компьютердің файлдар аймағында пайдаланушының компьютеріне тасымалданатын файлдарды таңдауға болады. Тағайындалған компьютердің қалталар аймағында осы файлдар орналастырылатын қалтаны таңдауыңыз керек. Пайдаланушының компьютеріне қай файлдарды орнату керектігін анықтау үшін, қажетті файлдарды Көз компьютерінің файлдар аймағынан Мақсатты компьютердің файлдар аймағына апарып тастаңыз. Мүмкіндіктер тобында бірнеше элементтер болса, әр элементтің файлдарын анықтауыңыз қажет.

Object / Merge Modules пәрмені пайдаланушының компьютерінде және, демек, орнату дискетасында қандай нысандарды, мысалы, динамикалық кітапханалардың немесе құрамдас бумалардың орналасуын көрсетуге мүмкіндік береді. Орнату дискетасында орналастыру қажет нысандар InstallShield

Objects / Merge Modules тізімінде таңдалады.

Нысаналы жүйе тобын конфигурациялау тобының командалары жүйені орнатылған бағдарламамен жұмыс істеу үшін конфигурациялау үшін пайдаланушы жүйесінде қандай өзгерістер жасау керектігін анықтауға мүмкіндік береді.



Сурет. 12.1. Пайдаланушының компьютеріне тасымалдайтын файлдарды таңдау

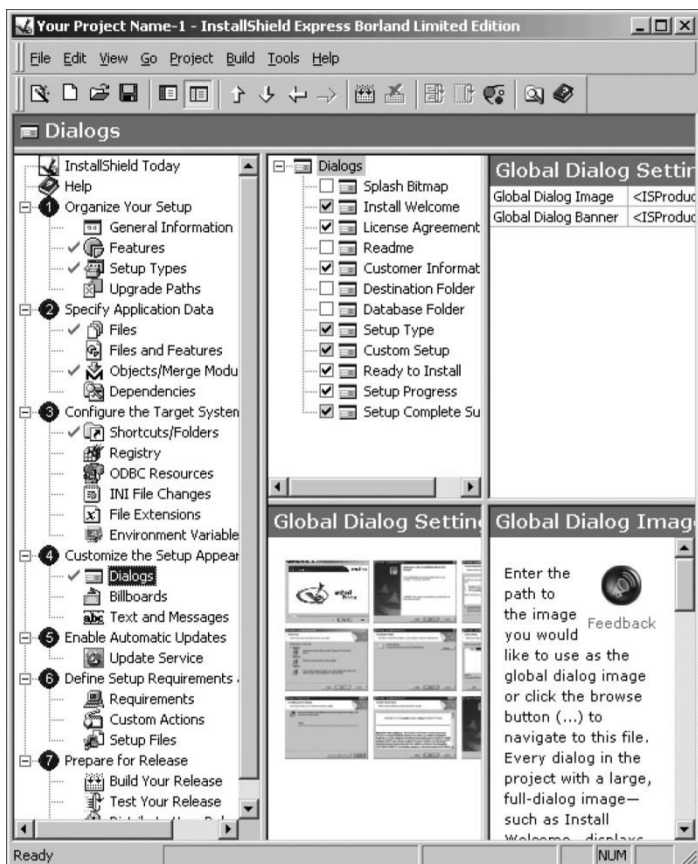
Таңбашалар / Қалталар пәрмені орнатылған бағдарламаның іске қосылуын қамтамасыз ететін төте жолды қайда орналастыру керектігін көрсетуге мүмкіндік береді. Осы пәрменді таңдау нәтижесінде, иерархиялық тізім терезенің оң жағында көрсетіледі, ол бағдарламаның таңбашасын қоюға болатын мәзірлер мен қалталарды тізімдейді. Бұл тізімде таңбашаны орналастыру керек мәзірді таңдап, тінтуірдің оң жақ батырмасымен нұқыңыз және пайда болған тізімнен Жаңа тіркесімін таңдаңыз.

Содан кейін, Төте жолға шолу үшін тілқатысу терезесінде бағдарлама файлы таңдап, Ашу түймешігін басып, жапсырма атын енгізіңіз. Осыдан кейін, сіз қысқа жолдың соңғы конфигурациясын орындауға болады - мысалы, дәлелдер өрісінде пәрмен жолы

параметрлерін және Жұмыс каталогында жұмыс каталогын енгізіңіз.

Пайдаланушымен өзара әрекеттесу үшін, Орнату стандартты диалогтық терезелерді пайдаланады. Орнату бағдарламасын жасаған кезде бағдарламашы бағдарламаны орнату кезінде пайдаланушы қандай диалогтарды көретінін көрсете алады.

Орнату бағдарламасы жұмыс істеп тұрған кезде монитор экранында пайда болатын диалогтық терезелерді орнату үшін, орнату терезесін теңшеу бөліміндегі және ашық диалогтар тізіміндегі Диалогтар пәрменін таңдаңыз, орнату бағдарламасына қосқыңыз келетін диалогтық терезелерді таңдаңыз (12.2-суретті қараңыз). Бұл тілқатысу терезелері бағдарламаның пайдаланушы компьютерінде орнатылуы кезінде пайда болады.



Сурет. 12.2. Диалогтар тізімі

Сипаттар кестесі (тілқатысу терезесінің оң жағында) таңдалған диалогтың сипаттарын көрсетеді. Бағдарламашы осы қасиеттердің мәнін өзгерте алады және осылайша диалогты теңшей алады. Мысалы, Readme тілқатысу терезесінде, орнатылған бағдарлама туралы қысқаша анықтаманы қамтитын файл атауын (Readme File сипаты) көрсету керек.

Көптеген диалогтар үшін баннерді (Banner Bitmap сипаты) анықтай аласыз - диалогтық терезенің жоғарғы жағында көрсетілетін иллюстрация.

Төменде орнату бағдарламасы кезінде пайда болатын диалогтық терезелер:

Splash Bitmap - белгіленген бағдарлама туралы ақпарат болатын иллюстрациялар шығару;

Қош келдіңіз - ақпарат туралы хабарды суреттің фонында орнатыңыз;

Лицензиялық келісім - RFT файлында лицензия хабарламасын шығару. Пайдаланушы ұсынылған шарттармен келіспеген жағдайда бағдарламаны орнату процесін тоқтатуға мүмкіндік береді;

Readme - белгіленген бағдарлама туралы қысқаша ақпарат шығару;

Тұтынушы туралы ақпарат - пайдаланушы туралы ақпаратты (аты, ұйым атауы) және, қажет болса, орнатылған көшірменің сериялық нөмірін сұрайды;

Destination Folder - пайдаланушыға бағдарлама орнатылған алдын ала анықталған каталогты өзгертуге мүмкіндік береді;

Дерекқор қалтасы - пайдаланушыға дерекқорларға арналған алдын ала анықталған каталогты өзгертуге мүмкіндік береді;

Орнату түрі - пайдаланушыға бағдарламаны орнату түрін таңдауға мүмкіндік береді (әдеттегі - қалыпты орнату, минималды - минималды орнату, Custom - selective installation);

Custom Setup (Пайдаланушылық баптау) - пайдаланушыға Custom орнату арқылы орнатылатын компоненттерді таңдауға мүмкіндік береді;

Setup Complete Success - орнату процесі аяқталғанын пайдаланушыға хабарлайды. Орнату аяқталғаннан кейін іске қосылатын бағдарламаны көрсетуге мүмкіндік береді (бұл әдетте орнатылған бағдарлама), сондай-ақ Readme файлының мазмұнын шығаруға мүмкіндік береді;

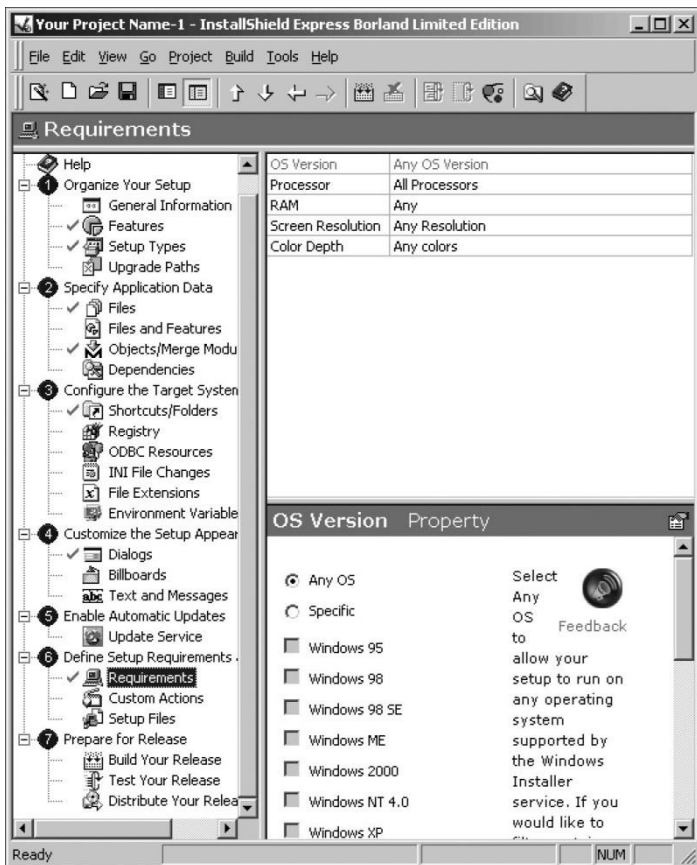
Setup Progress - бағдарламаны орнату барысында орындалған жұмыстардың пайызын көрсетеді;

Орнына дайын - бағдарламаны тікелей орнатпастан бұрын оны тексеру үшін алдыңғы қадамдарда пайдаланушы енгізген ақпаратты шығару.

Тілқатысу терезесі орнату бағдарламасында пайда болу үшін тілқатысу терезесінің атауының сол жағындағы күшбелгіні қоюыңыз керек. Терезелер үшін лицензиялық келісім және Readme, сізге тиісті ақпарат орналасқан RFT-файлдардың атауларын көрсету керек.

Ең қарапайым жағдайларда орнату бағдарламасы келесі диалогтық терезелерді көрсету үшін шектелуі мүмкін:

Readme;
 Destination Folder;
 Ready to Install;
 Setup Progress;
 Setup Complete Success.



Сурет. 12.3. Жүйені сипаттайтын параметрлер

Орнатылған бағдарламада жүйелік ресурстарға арналған арнайы талаптар болса, онда орнату талаптары мен әрекеттерін анықтау командасының пәрмендерін қолдансаңыз, бұл талаптар көрсетілуі мүмкін.

Нұсқаулықтарды таңдау нәтижесінде, кесте экранда пайда болады (12.3-сурет), онда жүйені сипаттайтын параметрлердің мәндерін енгізу керек: OS нұсқасы, Процессор, ЖЖҚ), экран ажыратымдылығы (Экранның ажыратымдылығы) және түс бояғыштары (Түстер тереңдігі). Типтік мәндер параметр мәні мән өрісінде басу нәтижесі ретінде пайда болған ашылмалы тізімнен таңдау арқылы орнатылады.

Егер бағдарлама жүйе конфигурациясына ешқандай ерекше талаптар қоймаса, Орнату талаптары мен әрекеттерін анықтау командасының өткізіп жіберілуі мүмкін.

Шығаруға дайындалу пәрмендері сізге орнату дискісінің (CD-ROM) кескінін жасауға және орнату бағдарламасының жұмысын тексеруге мүмкіндік береді. Орнату дискісінің (CD-ROM) кескінін жасау процесін белсендіру үшін, «Шығаруды қалыптастыру» пәрменін таңдап, орнату бағдарламасын орнатуға болатын медиа белгішесін тінтуірдің оң жағымен басып, алынған мәтінмәндік мәзірден пәрменді таңдаңыз Құрастыру. Осы әрекеттердің нәтижесінде компьютерлік дискіде жоба қалтасында орнату дискісінің суреті жасалады.

Тапсырма

Бағдарламалық жасақтаманың өнімділігін көлденең міндеттің нұсқасына сәйкес орнату.

БАҚЫЛАУ СҰРАҚТАРЫ

1. Бағдарламалық жасақтама өнімін орнатудың мақсаты қандай?
2. Орнату барысында қандай проблемалар туындауы мүмкін және оларды қалай шешуге болады?
3. Орнату түрлері пәрмені InstallShield Express бағдарламасында қандай мақсаттарға ие?
4. Орнату бағдарламасында пайда болатын диалогтық терезелерге ат беріңіз.

КУРСТЫҚ ЖОБАЛАУ

13.1. ТҮСІНДІРМЕ ЖАЗБАНЫҢ ҚҰРЫЛЫМЫ МЕН МАЗМҰНЫ

Курстың дизайны - «Бағдарламалық өнімдерді әзірлеу технологиясы» пәнін оқитын студенттердің соңғы сатысы. Курстың дизайны барысында студенттер өздігінен жұмыс істеу дағдыларын дамытады және ақпарат, технология және бағдарламалық қамтамасыз ету үшін дизайн шешімдерін әзірлейді. Оқушылар:

1) тақырыптық аймақтың ақпараттық ортасын талдау және ақпараттық кеңістіктің басқа компоненттерімен құрылымдық өкілділік пен қарым-қатынас орнату;

2) ақпараттық ағындарды талдау, жобаның логикалық құрылымын қалыптастыру;

3) деректер базасын ұйымдастыру;

4) тақырыптық аумақты алгоритмдеу және бағдарламалық қамтамасыз етуді ұйымдастыру;

5) заманауи алгоритмдік бағдарламалау тілдерін және ДҚБЖ-ны қолдануға.

Курстық жоба барысында студент бағдарламамен тікелей жұмыс істеуден гөрі қиындықтар мен проблемаларға әкеп соғатын бағдарламалық құжаттарды әзірлеуі керек. Төменде түсініктеме жазбада көрсетілген курстық жобаның құрылымы және оның бөлімдерінің мазмұны берілген.

Сонымен, курстық жобаның шеңберінде «Түсініктеме жазба» бағдарламалық құжаты келесі құрылымға ие болуы мүмкін.

Кіріспе

Жүйелік жобаны әзірлеу

1.1. Даму мақсаты

1.2. Талаптар сенімділік пен қауіпсіздікке қойылатын функционалдық талаптар

1.3. Техникалық құралдардың құрамы мен параметрлеріне

қойылатын талаптар

1.4. Ақпарат пен бағдарламалық қамтамасыз етудің үйлесімділігіне қойылатын талаптар.

2. Техникалық жобаны дайындау

2.1. Талаптарды талдау және бағдарламалық қамтамасыз етудің ерекшеліктерін анықтау Проектирование модели данных

2.2 Бағдарламалық жасақтаманың толық дизайны (прототиптің құрылысы)

3. Іске асыру

3.1 Даму құралдарын таңдау негіздемесі

3.2 Негізгі бағдарламалық модульдердің сипаттамасы

Бағдарламалық өнімді сынау.

Қорытынды

Қолданбалар

Әдебиеттер тізімі

Бөлімдердің мазмұнын елестетіп көрейік.

Кіріспе

Жобаның мақсаты, әзірленген бағдарламалық жасақтама өнімінің қолдану саласының қысқаша сипаттамасы және ол қолданылатын объектінің сипаттамасы (тақырыптық аймақтың сипаттамасы).

1. Жүйелік жобаны әзірлеу

1.1. Бұл бөлім шын мәнінде курстық жобалау бойынша техникалық тапсырма болып табылады.

1.2. Дамудың мақсаты дамыған жүйенің қойылған мақсатқа қол жеткізу үшін шешуі керек функционалдық және операциялық тапсырмалардың анықтамасын қамтиды.

1.3. Функционалдық сипаттамаларға қойылатын талаптар орындалатын функциялардың құрамын, кіріс және шығыс ақпараттарына қойылатын талаптарды, сондай-ақ бағдарламаның сервистік функцияларын қамтиды.

1.4. Сенімділік пен қауіпсіздікке қойылатын талаптар бағдарламалық қамтамасыз етудің сенімді және тұрақты жұмыс істеуін қамтамасыз ету, кіріс және шығыс ақпаратын басқару үшін,

істен шыққан сәттен бастап қалпына келтіру және т.б.

1.5. Техникалық құралдардың құрамы мен параметрлеріне қойылатын талаптарға техникалық құралдардың қажетті құрамын және олардың негізгі сипаттамаларын, атап айтқанда, бағдарламаға қажетті ең төмен жүйелік талаптарды көрсетуге болады.

Ақпарат пен бағдарламалық қамтамасыз етудің үйлесімділігіне қойылатын талаптар ақпарат құрылымдарына, бағдарламалау тілдеріне және бағдарламалық жасақтамаға қойылатын талаптарды қамтиды.

2. Техникалық жобаны әзірлеу

Бұл кезеңде жүйелік жобаның негізінде жүйенің нақты дизайны, оның ішінде дизайнды жобалау және үлгілерді жобалау жүзеге асырылады.

2.1. Талаптарды талдау және бағдарламалық қамтамасыз етудің ерекшеліктерін анықтау

2.1.1. Дизайн технологиясын таңдау

Мұнда бағдарламалық жасақтамаға - құрылымдық немесе объектілі-бағдарланған көзқарасты таңдайсыз. Осы тәсілдерді салыстырайық. Олардың әрқайсысында артықшылықтар мен кемшіліктер бар. Құрылымдық тұрғыда процестер мен деректер бір-бірінен бөлек (ұйымның қызметін модель ретінде де, бағдарламалық жасақтама үлгісінде де) және жобалар деректерден процестерге дейін жүргізіледі. Бұл құрылымдық тәсілдің негізгі кемшілігі. Объектілі-бағдарлы көзқараста, объектілі модельдің негізгі санаты - бұл сынып. Ол қарапайым деңгейде деректерді де, оларда орындалатын әрекеттерді де (әдістерді) біріктіреді. Процестер мен деректерді бөлісу үстінде. Процестермен салыстырғандағы деректер жүйенің бір бөлігін сирек өзгеріп отырады, сондықтан нысанға бағдарланған жүйелер ашық әрі оңай өзгерістер енгізеді.

Сонымен қатар, нысанға бағдарланған модель өзара әрекеттесетін объектілердің жиынтығы болып табылатын нақты әлемді барабар түрде көрсетеді, объектілер арасындағы хабар алмасу арқылы өзара әрекеттеседі.

Алайда іс жүзінде объектілі көзқарастың ерекшеліктерін көрсететін диаграммалар (класс диаграммалары және т.б.) әлдеқайда айқын және нашар түсініксіз, сондықтан әзірлеуші туралы түсінік негізінен құрылымдық әдістермен қамтамасыз етіледі.

Үлгілік ғимарат.

2.1.2 Бұл кезеңде жобаланған жүйенің үлгілері құрылады, олар қажетті деңгейге дейін анықталады және таңдалады (таңдалған технологияға байланысты: деректер ағынын диаграммалары, функционалдық диаграммалар, пайдалану жағдайлары, белсенділік диаграммалары және т.б.). Осы бөлімде мәтіннің сипаттамасы бар әзірленген модельдер ұсынылуы керек.

2.2. Деректер үлгісін жобалау

Құрылымдық әдіс қолданылған жағдайда, жүйенің функционалды талдауын аяқтағаннан кейін, деректер ағындарының құрамы анықталады және тұжырымдамалық деректер схемасы бір модель немесе бірнеше жергілікті модель түрінде құрылады. Деректерді модельдеудің ең көп тараған құралы - «субъект-қатынас» диаграммасы (ERD).

2.3. Толық бағдарламалық жасақтаманы әзірлеу

Бағдарламалық қамтамасыз етудің құрылымдық схемасы жасалды, жүйенің функционалды схемалары, экрандық нысандардың дизайны, есептер, диалогтар (бағдарламалық жасақтама прототипін жасау) арқылы жүзеге асырылады. Экран формаларының тізбектік диаграммаларын пайдалану арқылы экрандық пішіндердің иерархиясы модельденеді. Осындай диаграммалардың жиынтығы - бұл жүйенің пайдаланушылық интерфейсінің абстрактілі үлгісі, ол экрандағы экрандық формалардың пайда болуын көрсетеді.

3. Реализация

Бағдарламаны іске асыру сатысында бағдарламалық жасақтама тікелей дамыған. Алынған үлгілерге, сондай-ақ функционалды емес талаптарға (сенімділікті, орындалуды талап ету және т.б.) негізделіп, бағдарлама коды жасалады, бірлік тестілеу жүргізіледі.

3.1 Даму құралдарын таңдау негіздемесі.

Мұнда бағдарламалау тілін таңдаудың негіздемесі келтірілген, даму ортасын таңдауға әсер ететін негізгі факторлар (тапсырма үшін программалау тілінің салыстырмалы жарамдылығы, таңдалған әдіс және т.б.) келтірілген.

3.2 Негізгі бағдарламалық модульдердің сипаттамасы

4. Бағдарламалық жасақтаманы тестілеу

Жоспарлау кезеңінің басталуымен және спецификация талаптарының жасалуымен бірге тестілеу стратегиясы әзірленді. Талаптардың сипаттамалары мақұлданғаннан кейін тестілеу жоспары әзірленді және егжей-тегжейлі және интеграция және жүйелік тестілеу үшін сынақ люктер жасалды.

Тестілеу сынақтың барлық нәтижелерін ұсынатын сынақ есебін жасау арқылы аяқталады. Баяндамада келесі бөлімдер болуы керек:

1. Сынақ объектісі;
2. Сынақ мақсаты;
3. Құжаттаманың құрылымы
4. Техникалық талаптар;
5. Тестілеу тәртібі;
6. Тестілеу әдістері.

Алғашқы үш бөлімде: тестілеу кезінде бағдарламалық жасақтаманың атауы, көлемі, тағайындалуы; сынақтардың мақсаты; тестілеуден бұрын талап етілетін құжаттама тізімі.

«Техникалық талаптар» секциясы екі бөлімнен тұруы мүмкін: 1) бағдарламалық құжаттамаға қойылатын талаптар; 2) техникалық сипаттамаларға қойылатын талаптар.

Бірінші бөлімде құжаттаманың толықтығы, мазмұны және сапасына қойылатын талаптар көрсетілуі тиіс; Екінші бөлікте бағдарламаның жұмыс сипаттамаларына және ақпарат пен бағдарламалық қамтамасыз етуге үйлесімділікке қойылатын талаптар сипаттамаларына қойылатын талаптар сипатталады.

«Тестілеу процедурасы» нұсқауларын талап етеді: тестілеу реті бойынша, әр сынақ экспериментінің нақты тәртібі, тестілеу жүргізілетін техникалық құралдардың құрамы мен құрылымы, тестілеу үшін қажетті қосымша бағдарламалық жасақтама мен аппаратуралардың тізімі.

«Сынақ әдістері» бөлімінде пайдаланылған сынақ әдістерінің сипаттамалары келтірілген. Әдістері «Техникалық талаптар» бөлімінде көрсетілген техникалық сипаттамалардың кезектілігіне сәйкес келетін тізбеге енгізілуі керек.

Сонымен қатар, тестілеудің нәтижелерін көрсете отырып, сынақтардың сипаттамалары көрсетілуі керек, олар мыналарды қамтуы мүмкін: сынақ жағдайларының тізбесі, мысалдардың өздері мен нәтижелерін бақылау, кестелер, графиктер және т.б. Сынақ жағдайлары (басып шығару, кестелер, диаграммалар және т.б.) Қосымшада берілген.

Қ о р ы т ы н д ы

Жұмыстың орындалуына талдау, жобаның маңыздылығы туралы тұжырымдар, жобаны пайдалану бойынша ұсынымдар, жобаны әрі қарай қайта қарау немесе модернизациялау мүмкіндігі туралы ұсыныстар және т.б. қамтылған.

Курстық жобаның практикалық нәтижесі - «Түсініктеме жазба» және «Пайдаланушы нұсқаулығы» бағдарламалық

құжаттарын қамтитын бағдарлама мен құжаттама пакеті. Бұдан басқа, бағдарламаның мәтіні (бастапқы код) қажетті түсініктемелермен магниттік ортада ұсынылуы керек.

Курстық жобаны қорғау үшін бағдарламалық жасақтамаға қойылатын талаптарды, оны дамытудың негізгі кезеңдерін, диаграммаларды, деректер үлгілерін, жұмыстың қорытындыларын көрсететін 10-12 слайдтың презентациясы жасалады.

13.2. КУРСТЫҢ ЖОБАЛАРДЫ РӘСІМДЕУ

«Абитуриент» автоматтандырылған ақпараттық жүйесінің бағдарламалық жасақтамасы.

Төменде бағдарламалық жасақтама өнімі дизайнының үзіндісі келтірілген. «Абитуриент» автоматтандырылған ақпараттық жүйесінің бағдарламалық жасақтамасы» тақырыбындағы курстық жобаның түсіндірме жазбасы.

Кіріспе

«Абитуриент» автоматтандырылған ақпараттық жүйесі орта кәсіби білім беру мекемесі автоматтандырылған басқару жүйесінің бір бөлігі болып табылады. Талапкерлерді қабылдауды ұйымдастыру, емтихан тапсыру үшін ақпараттарды дайындау және қабылдау қорытындысы бойынша қажетті есептерді қалыптастыру үшін қабылдау комиссиясының жұмысын автоматтандыру.

1. Жүйелік жобаны әзірлеу

1.1 Қабылдау комиссиясына үміткерлерді қабылдаудың шағын жүйесі келесі мәселелерді шешуге арналған: ведение базы данных абитуриентов;

- кіріс сынақтарын бақылау және өткізу;
- түсу емтихандарының нәтижелері бойынша студенттерді қабылдау;
- өтініштер қабылдау және түсу емтихандарын өткізу туралы қажетті
- есептерді қалыптастыру (түрлі тізімдерді жасау, тапсырыстарды алыптастыру, статистикалық есептер және т.б.).

1.2 Функционалдық талаптар

1.2.1. Орындалған функциялардың құрамы

«Абитуриент» автоматтандырылған ақпараттық жүйесі:

- оқу орнына түсушілер туралы ақпарат жинау;
- Пайдаланушы белгілеген әртүрлі критерийлер бойынша талапкерлер туралы қажетті ақпаратты тез іздестіру және басып

шығару;

- емтихан парақтарын басып шығару;
- өтініш және қорытынды емтихандар нәтижелері бойынша статистикалық есептерді қалыптастыру.

Жүйедегі бастапқы ақпарат үміткерлердің өтініштерінен, білім туралы құжаттардан, сертификаттардан және т.б. Енгізу деректері мыналар:

- өтінімнің күні мен нөмірі;
- S тегі, аты, әкесінің аты;пол;
- туған күні;
- Азаматтығы;
- СТН;
- сақтандыру нөмірі;
- үй мекен-жайы;
- төлқұжат деректері;
- аяқталған білім туралы ақпарат
- жеңілдіктерге құқық беретін құжаттар;
- Қосымша ақпарат (мүгедектер, жетім балалар, әскери іс-шараларға қатысушы және т.б.);
- ата-аналар туралы мәліметтер (аты-жөні, жұмыс орны);
- сертификаттың орташа балы.

Үміткерлер қажетті мамандықты таңдап алады, оқыту түрі. Талапкерлер туралы барлық мәліметтер дерекқорға енгізіледі.

Жүйе бірнеше пайдаланушы режимінде жұмыс істеуі керек:

- деректер қорының жазбаларын, соның ішінде әртүрлі сүзгілердің көмегімен қарау;
- жаңа жазбаларды қосып, қолданыстағы өзгертулерді енгізіңіз;

■ жазбаларды жойыңыз.

Кіріс емтихандарының нәтижелері де дерекқорда сақталады.

Шығу көздері:

- өтініш берушілер ұсынған өтініштер туралы есептер;
- емтихан нәтижелері туралы есептер;
- оқу орнына түскендер тізімі;
- қабылдау және қабылдау туралы статистикалық есептер;
- студенттерді қабылдау үшін жоғарғы деңгейдегі ұйымдарға реттелетін есептер.

Ескерту: Кіріс құжаттары мен шығыс есептерінің үлгілері

курстық жобаға қосымшада ұсынылуы керек.

1.3. Сенімділік пен қауіпсіздікке қойылатын талаптар

Жүйе көп пайдаланушы режимінде жұмыс істеуге тиіс, сондықтан әрбір пайдаланушыда жүйеге жеке кіру паролі болуы керек. Бұдан басқа, жүйе деректердің сақтық көшірмесін жасау мен қалпына келтіру мүмкіндігін қамтамасыз етуі керек, ал бағдарламалық жасақтама амалдық жүйедегі немесе электр қуатының үзілістеріндегі ақаудан кейін өздігінен жөндеуге қабілетті болуы керек.

1.4. Техникалық құралдардың құрамына және параметрлеріне қойылатын талаптар Жүйелік талаптар:

- процессордың сағат жылдамдығы > 2 000 Гц;
- 512 Мб RAM көлемі;
- бос дискілік кеңістіктің көлемі - 50 МБ;
- Монитордың ажыратымдылығы - 1 024 x 768.

1.5. Ақпарат пен бағдарламалық қамтамасыз етудің үйлесімділігіне қойылатын талаптар

Компьютерлерде амалдық жүйе Windows 2000 / XP болуы керек. Барлық жасалған есептерді MS Office Excel 2003/2007 электрондық кестесінің редакторына экспорттау мүмкіндігі болуы керек. Компьютерде «1С: Кәсіпорын» қорғау жүйесімен қайшы бағдарламалық және аппараттық пакеттер орнатылмауы керек.

2. Техникалық жобаны әзірлеу

2.1 Талаптарды талдау және бағдарламалық қамтамасыз етудің техникалық сипаттамаларын анықтау (IDEF0 белгісінде функционалдық модельді құру)

Біз бағдарламалық жасақтаманың болашақтағы бағдарламалық өнімдеріне қойылатын талаптарды талдауды бастаймыз. Доменнің жалпы моделін нақты әлемнің бөлігі ретінде құрайық, оның көмегімен дамыған бағдарламалық жасақтама өзара әрекеттесе алады, содан кейін оның негізгі функцияларын көрсетеміз. Техникалық сипаттамалар тапсырыс беруші мен әзірлеуші тарапынан анық көрінуі тиіс. Бұл талапты тек осы бағдарламалық жасақтаманың ресми үлгісін жасау арқылы ғана қамтамасыз етуге болады. Талдау және спецификацияларды анықтау сатысында құрылымдық тәсіл қолданылуы мүмкін.

Біз бағдарламалық жасақтама жүйесінің негізгі функциялары мен құрамдас бөліктерін анықтау үшін және мүмкіндігінше елеулі

қателерді анықтау және жою үшін функционалдық диаграмма құрдык. Функционалды модельдеу әдіснамасының ең маңызды ерекшеліктерінің бірі - модельді білдіретін диаграммалардың барлығына біртіндеп егжей-тегжейлердің біртіндеп енгізілуі.

Суретте көрсетілгендей диаграмма. 13.1, жоғары деңгейлі диаграмма. Бағдарлама үшін бастапқы деректердің не екенін және қандай нәтижелер күтетінін анық көрсетеді. Талапкерлер туралы ақпарат жүйеге кіруге жіберіледі, Қабылдау комиссиясының жұмысы туралы есептер шығарылады. Ақпаратты басқару - қабылдау ережелері, білім беру мекемесін аккредитациялау туралы куәлік, лицензиялар және т.б.

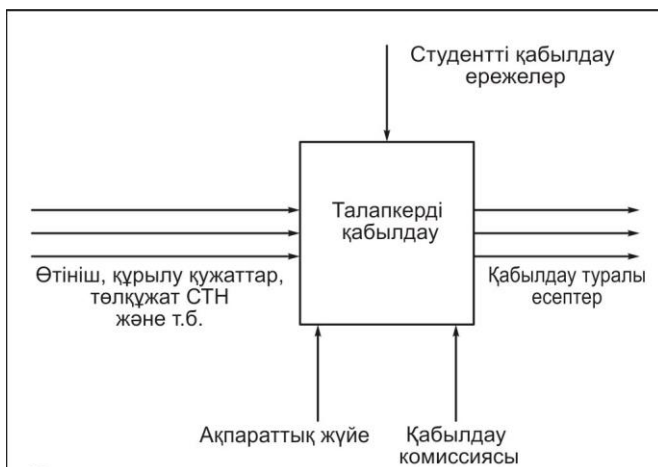
Суреттегі диаграмма. 13.2 бағдарламаның функцияларын анықтайды. Диаграммада төрт блок бар: 1) құжаттарды қабылдау және орындау; 2) түсу емтихандарын өткізу; 3) емтихандар қорытындысы бойынша оқуға қабылдау; 4) есептерді қалыптастыру.

Орта кәсіптік оқу орындарына қабылдау ережелері өтінімдерді қабылдау және оқуға қабылдау тәртібін реттейді. Аккредиттеу туралы куәліктің негізінде мамандықтардың тізімі жасалады.

A1 блокының ыдырауы (құжаттарды қабылдау және өңдеу функциясы) сурет 13.3.

Оқу орнына түсу үшін құжаттарды қабылдау және рәсімдеу тәртібі, ең алдымен, мамандықты таңдау және оқу орнында бар мекемелерден оқу нысанын қамтиды, содан кейін өтініш нысанын толтырады. Мекенжайланған студенттер туралы барлық мәліметтер абитуриенттердің тізімін қалыптастыру, емтихан ведомостарын басып шығару, қабылдау емтихандарының нәтижелерін енгізу және қабылдау нәтижелерін кейінгі талдау үшін деректер базасында сақталуы тиіс.

Өтініш өтініш берушінің жеке құжаттарының негізінде - паспорт, сақтандыру сертификаты, білім туралы куәлік, медициналық анықтама және т.б. негізінде толтырылады. Өтінішті тіркеуден кейін өтініш берушіге емтихан ведомосі беріледі. Қабылдау емтихандарына арналған пәндер тізімі таңдалған мамандық және қабылдау ережелері бойынша анықталады.

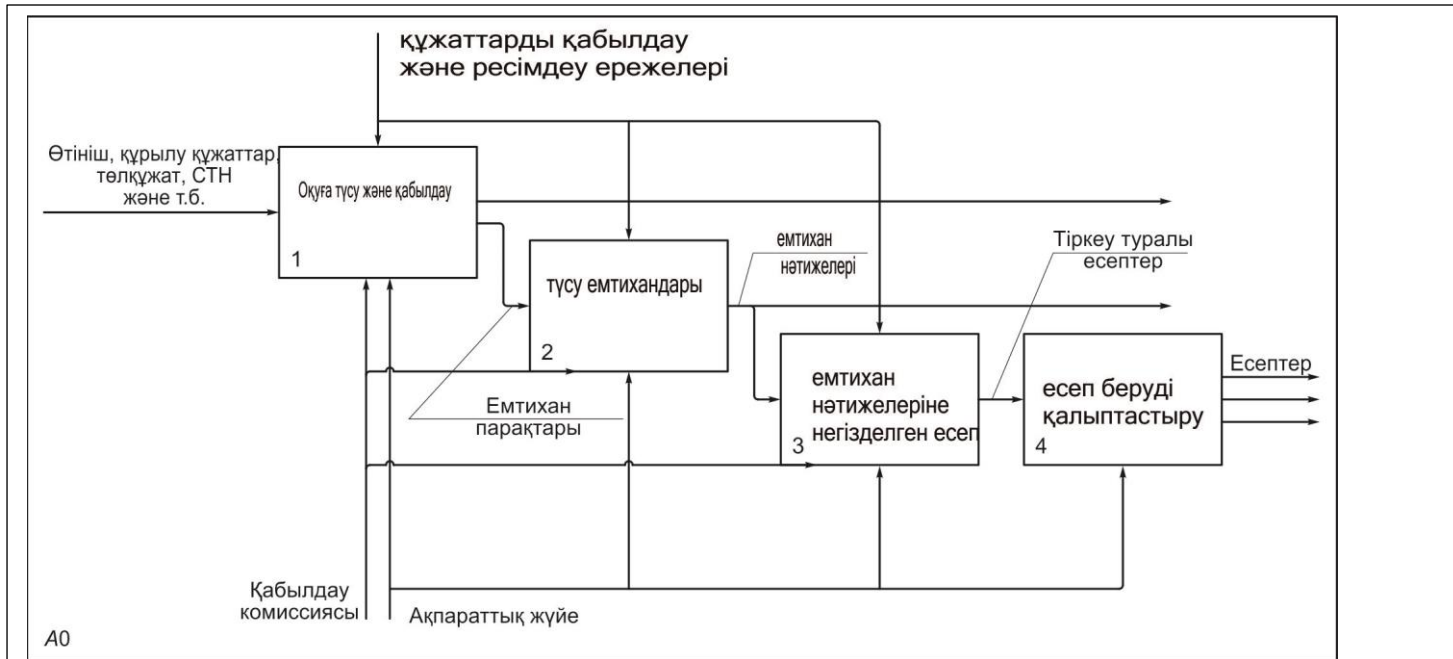


Сурет. 13.1. Кіру деңгейінің функционалдык диаграммасы

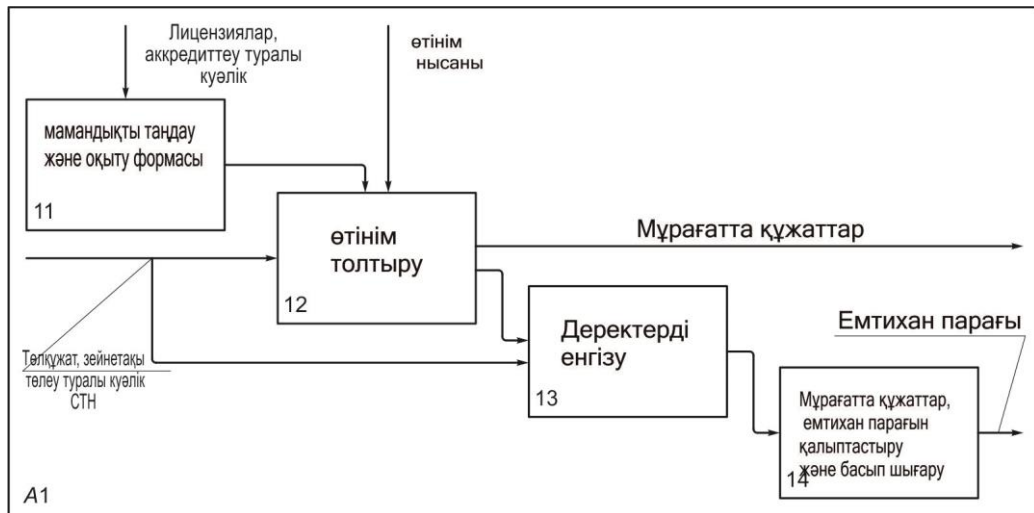
2.2. Деректер үлгісін жобалау

Енді олар арасындағы деректерді және өзара қарым-қатынастарды анықтауға мүмкіндік беретін «субъект-қатынас» диаграммасын (ER-деректер үлгісі) дамытамыз. Деректер үлгісі субъектілер мен олардың арасындағы қатынастарды қамтиды. Құрылымдық схемалар, функционалдык диаграммалардан айырмашылығы, бағдарламалық жасақтаманың деректер құрылымдарының ерекшеліктерін анықтайды. Дамыған автоматтандырылған ақпараттық жүйенің деректер үлгісінде негізгі субъект, әрине, талапкер болып табылады.

Бұл ұйымның атрибуттары: жынысы, туған күні, туған жері, азаматтығы, СТН, сақтандыру нөмірі, үй мекен-жайы, төлқұжат деректері, аяқталған оқу орны туралы ақпарат, артықшылықтар беретін құжаттар, қосымша ақпарат (мүгедек, жетім бала, әскери қызметкер) әрекеттер және т.б.), ата-аналар туралы мәліметтер (аты, жұмыс орны), орташа балл



Сурет 13.2.



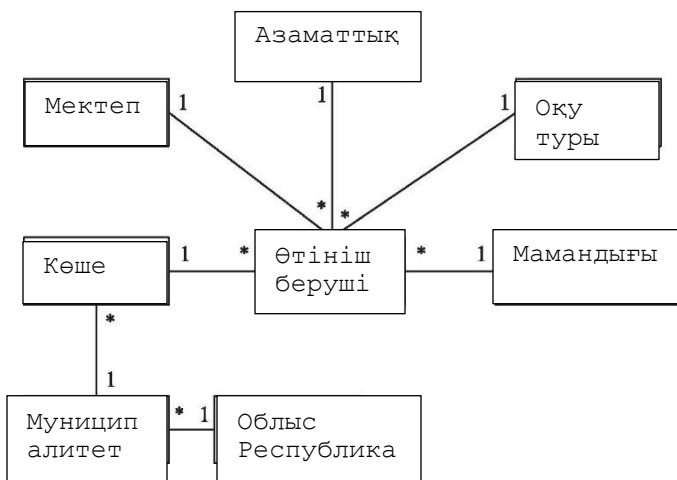
Сурет. 13.3. Құжаттарды қабылдау және орындау

куәлік және т.б. «Мектеп», «Азаматтық», «Оқу түрі», «Мамандық», «Көше», «Қала», «Аймақ» атрибуттары жеке тұлғалардың атаулары болып табылады; «Есептеу» төлсипатының мәні «Көше» ұйымының құнын толығымен анықтайды және «Аймақ» атрибутының мәні «Есеп айырысу» мәні бойынша анықталады.

Бұл субъектілер арасындағы негізгі қатынастар күріш. 13.4. Келесі кадамда біз әрбір субъектінің атрибуттарын анықтап, олардың түрлерін нақтылаудамыз (субъектіні басқа субъектімен сәйкестендіру үшін қолданылатын атрибуттар, олар сәйкес ұйымда сипатталғандықтан). Енді сіз мұның бәрін диаграммаға қосасыз (сурет 13.5).

Бағдарламалық жасақтаманың толық дизайны (прототиптің құрылысы)

Бағдарламалық жасақтаманың толық дизайны жасақталған бағдарламалық жасақтама туралы толық түсінік беретін құрылымдық схеманың дамуын қамтиды. «Abiturient» бағдарламалық жасақтама жүйесі бірнеше функционалды кіші жүйелерге бөлінуі мүмкін: абитуриенттер базасының деректер базасы, анықтамалық ақпаратты қамтамасыз ету, шығыс ақпаратын қалыптастыру, қызмет көрсету функциялары.

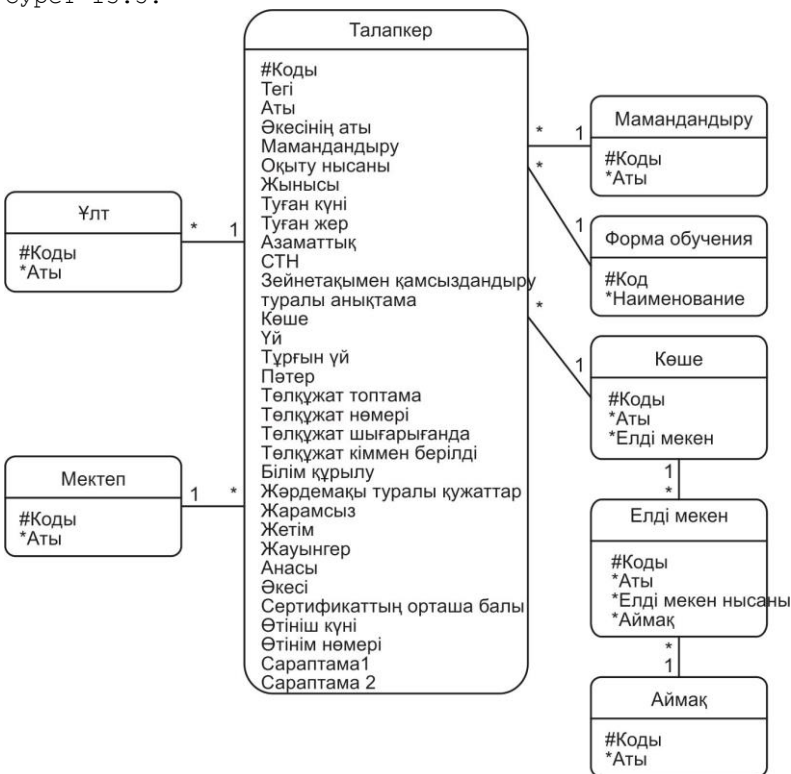


Сурет. 13.4. «Талапкер» ақпараттық жүйесінің дерекқорын сипаттауға арналған «мәні-сілтеме» диаграммасы (бастапқы нұсқасы)

Осыған орай, бағдарламаның құрылымдық сұлбасы салынып, оның негізінде Бағдарламаның Негізгі мәзірінің құрылымы жобаланып отырады. «Өтініш беруші» ақпараттық жүйенің бағдарламалық жасақтамасының құрылымдық схемасы сурет 13.6.

Бағдарламаның құрылымдық схемасына сүйене отырып, экранның формаларының пайда болуын көрсететін жүйенің пайдаланушылық интерфейсінің дерексіз моделін құруға болады. Негізгі нысаны, анық, бағдарламаның құрылымдық схемасына сәйкес келетін Мәзірді қамтуы керек. Өз нұсқалары бойынша абоненттерді енгізу және редакциялау, журналистерді басқару, есеп беру, бағдарлама параметрлерін және т.б. белсендіруге болады.

Сурет 13.5.



Сурет 13.6.



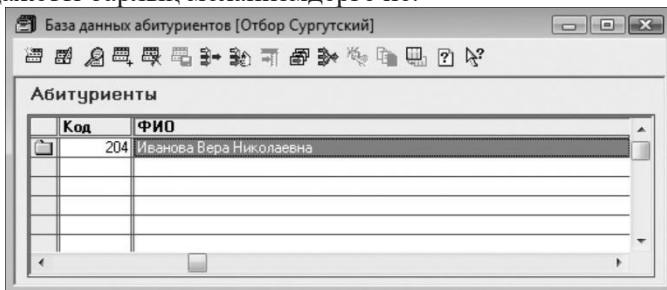
жазбаларды жойып, енгізуді және редакциялау терезесін шақыруға, қажетті ақпаратты іздеуге және таңдауға болады (13.7-сурет). Талапкерлер туралы ақпаратты енгізу және редакциялау бөлек терезеде жүзеге асырылады. Пайдаланушының сұранысы бойынша икемді есептерді жасау үшін тілқатысу формалары есеп формасын қалыптастыру критерийлерін анықтауға арналған (13.8, а, в, с, d).

Осылайша, эволюциялық прототип жасалды, ол кейіннен аяқталған бағдарламалық жасақтама деңгейіне дейін аяқталады.

3. Іске асыру

3.1. Даму құралдарын таңдау негіздемесі

Ең қолайлы, қолда бар тапсырмаға сәйкестігі тұрғысынан алғанда, жеке шешімдерді дамыту үшін жаппай қолданылатын «1С: Кәсіпорын» платформасы. Біріншіден, бұл жүйе бақылау және есепке алу тапсырмаларын автоматтандыруға арналған және тез арада қажетті барлық механизмдерге ие.



Талапкерлердің дерекқоры [Селекциялық Сургут], Өтініш берушілер, Кодекс, Аты

Сурет. 13.7. Талапкерлердің деректер базасын жүргізу нысаны

қолданбалы шешімдерді әзірлеу. Екіншіден, оны пайдалану жобаны дамытуды едәуір жеңілдетуге және жеделдетуге және оның құнын едәуір азайтуға мүмкіндік береді.

Негізгі бағдарламалық модульдердің сипаттамасы

Бағдарламалық қамтамасыз ету жүйесінің жекелеген модульдерінің жұмысын сипаттау үшін есептік генерациялайтын модульдерді таңдауға болады (13-суретті қараңыз, b, c және d). Бастапқы пішінді үзу рәсімдерінің алгоритм схемасы сурет. 13.9. Суреттегі 1 белгісі есепті қалыптастыру критерийлерін енгізуге қатысты әрекетті көрсетеді.

Қажетті деректер тиісті мәтін өрістеріне енгізіледі. 2 символы «Жасау» батырмасын басудан тұратын әрекетті білдіреді. Осы батырманың көмегімен рәсімде, бағдарлама модуліне қол жеткізіледі, ол есеп жасау үшін алгоритмді тікелей іске асырады. «Жабу» батырмасын басу (3 белгісі) пішінді жабады.

Одан кейін, ГОСТ 19.701-90 стандартына сәйкес есепті құрастырудың алгоритмін сипаттаңыз, бұл бағдарламаның мәтініне сілтеме жасай отырып, оқу жоспарына қосымшада келтірілген.

Автоматтандырылған тестілік жүйенің бағдарламалық жасақтамасы.

Төменде бағдарламалық жасақтама өнімі дизайнының үзіндісі келтірілген. «Автоматтандырылған сынақ жүйесінің бағдарламалық жасақтамасы» тақырыбындағы курстық жобаның түсіндірме жазбасы.

Білімді компьютерлік тестілеу жүйесі студенттердің білімін терең және жүйелі тестілеуді қамтамасыз етуге арналған бағдарламалық қамтамасыз ету жүйесі болып табылады. Жүйені пайдалануға болады

Абитуриенты:Новый *

Основные сведения | Дополнительные сведения | Вступительные экзамены

Фамилия: Лесовой
Имя: Захар
Отчество: Борисович
 Пол: Мужской Женский
 Дата рождения: 23.01.89
 Место рождения: >>
Национальность: русский
 ИНН: Страховой №:

Удостоверение личности:
 Документ: №: Серия: №:
 Дата выдачи: Кем выдан: >>

Страховой полис:
 №: Когда выдан: Кем выдан: >>

Пополнение в группу

Форма обучения:
 Специальность, на которую подано заявление:
 Вторая специальность, указанная в заявлении:
Специальность, на которую зачислен(а):

Финансирование: с полным возмещением затрат (козрасчетное)

Поступил(а)

а

Абитуриенты:Новый *

Основные сведения | Дополнительные сведения | Вступительные экзамены

Законченное образование:
 Год окончания образоват. учреждения: 0
 Законченное образовательное учреждение: СОШ №15
 Документ об образовании: Серия: №: На базе 11 классов

Льготы:
 Инвалид
 Сирота
 Военный запаса
 Участник военных действий
 Политтехнический класс Целевое направление

Изучаемый иностранный язык: французский

Льготы:
 Медаль(аттестат, диплом с "отличием")
 Победитель всероссийский олимпиад (член сборной)
 Льготы при поступлении:
 Документ, предоставляющий право на льготы:
 Ждагайство:

Трудовой стаж:
 0 лет, 0 месяцев

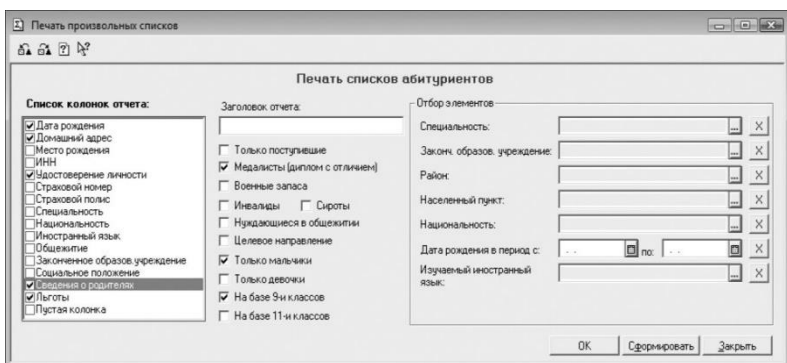
Средний балл аттестата: 0.00
 Дата приема заявления:

Сведения о родителях:
 Мать: Отец: Социальное положение:

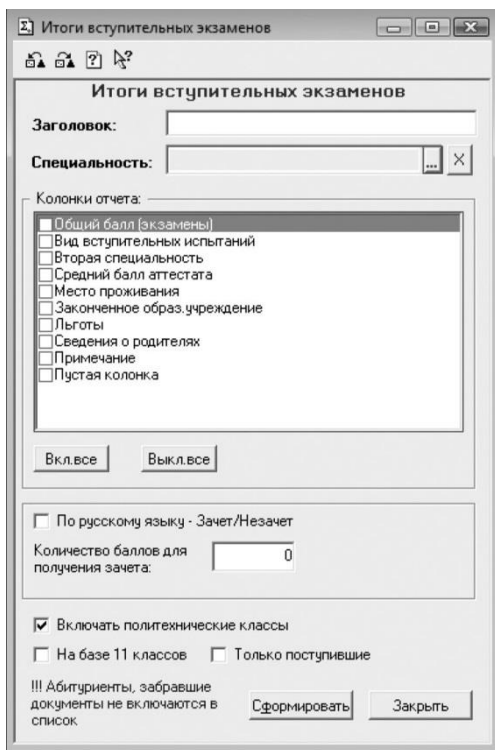
Забрал(а) документы

Поступил(а)

б



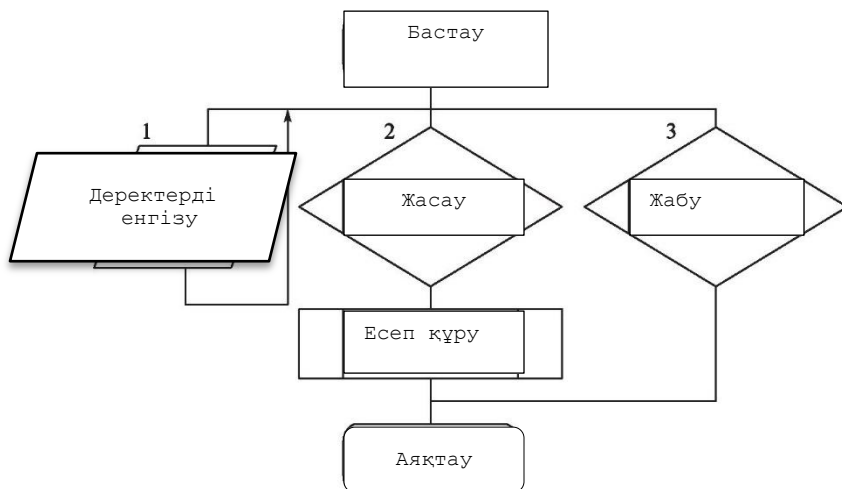
6



2

Сурет. 13.8. Экран нысандары:

a - өтініш беруші туралы ақпарат енгізу; b) өтініш беруші туралы ақпаратты редакциялау; c - абитуриенттердің тізімін жасау; г - түсу емтихандарының нәтижелері бойынша есеп беру



Сурет. 13.9. Бастапқы пішінді үзуге арналған рәсімдер алгоритмі схемасы

білімді нақты бақылау үшін және өзін-өзі сынау үшін (сынақ тестілеуі). Мақсаты - компьютерлік техниканы оқытудың соңғы нұсқаларын пайдалану, сондай-ақ оқу орындарында оқу үрдісінде сынақтар мен электронды оқулықтарды пайдалану кезінде психологиялық-педагогикалық мәселелерді зерттеу.

1. Жүйелік жобаны әзірлеу

1.1. Даму мақсаты

Білімді тексеру жүйесі студенттердің білімін сынақтан өткізу процесін кез-келген тақырыптар мен пәндер бойынша тестілеуден өткен кез-келген топпен автоматтандыруға арналған. Жүйе кез-келген тест сұрақтары бойынша сұрақтар қоюға, сұрақтарға әртүрлі жауаптарды таңдауға, тест нәтижесін бағалауға, білімді бағалауға, тест нәтижелерін жинақтауға және өңдеуге және сынақ элементтерінің тізімін сақтауға мүмкіндік береді.

Жүйе тестілеу қатысушыларына уақыт параметріне дифференциалды көзқарасты қолдануы, тестерлердің тапсырмаларды қарауға және редакциялауға қол жеткізуін шектейді, тестіленген жауаптардың дұрыс жауаптарымен салыстырады. Сынақ нәтижелері сыртқы файлға сақталып, басып шығарылуы керек. Жүйе материалдарды Интернетте таратуды ұйымдастыруы керек.

1.2. Функционалдық талаптар

1.2.1. Орындалған функциялардың құрамы

Дайындаған автоматтандырылған тестілеу жүйесінде тест сұрақтарының атауы, сұрақтарға иллюстрациялар, жауап нұсқалары сақталуы керек. Әрбір сұраққа жауапты дайындауға белгілі бір уақыт бөлу керек және дұрыс жауап үшін ұпайлар санын беріңіз.

Сондай-ақ, дерекқорда тестіленген оқушылар (атауы, топ), тестілеу және тестілеу нәтижелері туралы ақпарат (сұрақ нөмірі)

таңдалған жауаптың нөмірі, дұрыс немесе қате, жауап берілді), студенттің қойған баллдары болуы тиіс. Жүйе пәндердің нақты бөлімдеріне арналған оқу сынақтарын, барлық тақырып бойынша аралас тесттерді, сондай-ақ әртүрлі курстардағы барлық тақырыптар бойынша арнаулы қорытынды тесттерді қамтуы керек.

1.2.2. Кіріс және шығыс деректерін ұйымдастыру

Сынақ жүйесінің кіріс деректері:

- тестілер - тест атауы, тестілеуді құрушы, пән; сұрақтарға жауаптар, жауаптар, дұрыс жауап нөмірі, дұрыс жауап үшін баллдар саны (сұрақтың «салмағы»), жауап уақыты, сұрақ қою, сұраққа түсінік беру;

- тестіленген (аты, топ), тестілеудің және тестілеудің нәтижелері туралы сұрақ (сұрақ нөмірі, таңдалған жауаптың нөмірі, дұрыс немесе дұрыс емес, жауап берілді), студенттің қойған баллдары.

Деректер :

- тестілеу күнін көрсететін тестілеу хаттамалары, сынақ атауы, кесте түрінде сынақ пәні мен нәтижелер (мәселе, таңдалған жауаптың нөмірі ұрыс немесе дұрыс емес жауап), ұпай саны
- сынақ нәтижелерімен академиялық көрсеткіштерді талдау (топтар бойынша, пәндер бойынша, жеке оқушылар).

1.3. Сенімділік пен қауіпсіздікке қойылатын талаптар

Жүйе көп пайдаланушы режимінде жұмыс істеуге тиіс, сондықтан әрбір пайдаланушыда жүйеге жеке кіру паролі болуы керек. Пайдаланушы құқықтарын бөліп көрсету қажет, яғни. Студенттер сынақтарды өңдеп, сынақ нәтижелерін өзгертуге рұқсат бермеңіз, каталогтарды өңдеңіз. Сонымен қатар жүйеде бірнеше ондаған пайдаланушылар болуы мүмкін (мысалы, бірнеше топты сынағанда). Жүйе клиент-сервердің архитектурасын қолдануды талап ететін деректер базасында бірдей деректерге бір мезгілде кіруді қамтамасыз етуі керек.

Бұдан басқа, жүйе деректерді резервтік көшіру мен қалпына келтіруді, сондай-ақ амалдық жүйедегі немесе электр қуатының үзілуінен кейінгі жүйенің өзін-өзі қалпына келтіруін қамтамасыз етуі керек.

1.4. Техникалық құралдардың құрамы мен параметрлеріне қойылатын талаптар

Серверлік компьютерге арналған жүйелік талаптар:

- Процессордың сағат жылдамдығы > 2 000 Гц;
- Бос дискілік кеңістіктің көлемі - 50 МБ;
- Рұқсат берілген

- Монитор 024 x 768 деңгейінде.

Жұмыс станциясы үшін жүйелік талаптар:

- процессордың сағат жылдамдығы ~ 1000 Гц;
- ЖЖҚ көлемі - 64 МБ;
- Бос дискілік кеңістіктің көлемі 20 МБ;
- Монитордың ажыратымдылығы - 1 024 x 768.

1.5 Ақпарат пен бағдарламалық қамтамасыз студия үйлесімділігіне қойылатын талаптар

Серверде Windows Server 2003 және SQL Server 2003 орнатылған болуы керек Клиент компьютерлерінде Windows 2000 / XP амалдық жүйесі орнатылған болуы керек. Барлық жасалған есептерді MS Office Excel 2003/2007 электрондық кестесінің редакторына экспорттау мүмкіндігі болуы керек.

3. Техникалық жобаны әзірлеу

2.1. Талаптарды талдау және бағдарламалық қамтамасыз студия ерекшеліктерін анықтау

Преценденттердің анықтамасын қарастырыңыз (пайдалану жағдайлары). Тестілеу жүйесі ең алдымен келесі мүдделі тараптар үшін қажет:

- (студент);
- тестілеу құрастырушысы (мұғалім);
- емтихан тапсыратын мұғалім;
- прогресті бақылайтын оқу бөлімінің бастығы;
- Мектептің желілік әкімшісі.

Жүйені құрудың бастапқы кезеңінде сіз актерлердің тек екі маңызды рөлін ғана шектей аласыз: студент (тестілік адам) және әкімші (ол мұғалім және тесттердің композиторы).

Тиісінше, жүйе үшін негізгі преценденттер (пайдалану жағдайлары) келесідей.

- Студентке арналған прецендент:
- П1 - тесттен өту.

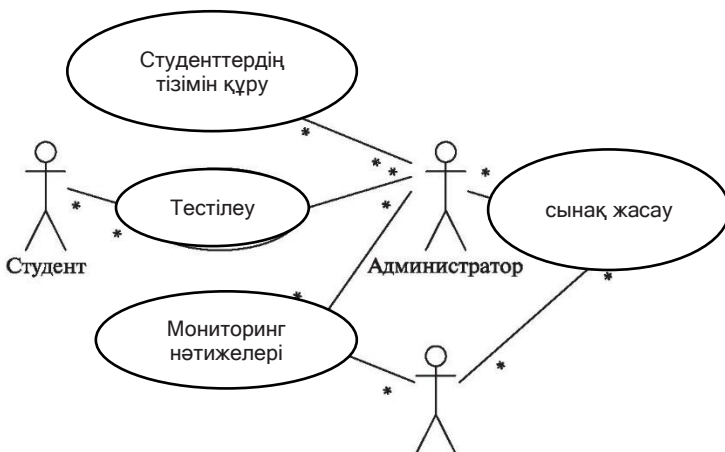
Әкімшінің преценденттері:

- P2 - тестті жасау (өзгерту);
- P3 - тест нәтижелерін көру;
- P4 - пайдаланушыларды қосу (өзгерту) және т.б.

Пайдалану жағдайын қысқаша немесе егжей-тегжейлі сипаттауға болады. Сипаттаманың қысқа формасында пайдалану жағдайының атауы, оның мақсаты, актерлер, пайдалану жағдайының түрі (негізгі, көмекші, қосымша) және оның қысқаша

сипаттамасы келтірілген (13.1-кесте). «Тестілеуді» қолдану нұсқасының толық сипаттамасы Кестеде келтірілген. 13.2.

Неғұрлым анық болу үшін пайдалану жағдайлары схемасын



құр. Студент пайдалану жағдайлары (13.10-сурет).

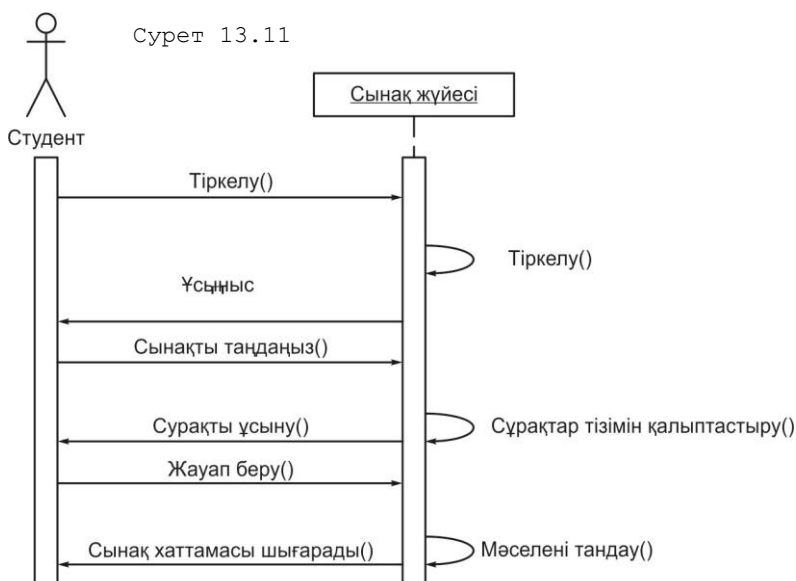
Жүйенің мінез-құлқының уақытша аспектісін ескеру үшін жүйелі диаграмманы құруға болады. Мысал ретінде, біз іске асыру үшін жүйелі диаграмма жасаймыз

Сурет. 13.10. Сынақ жүйесін пайдалану нұсқаларының диаграммасы

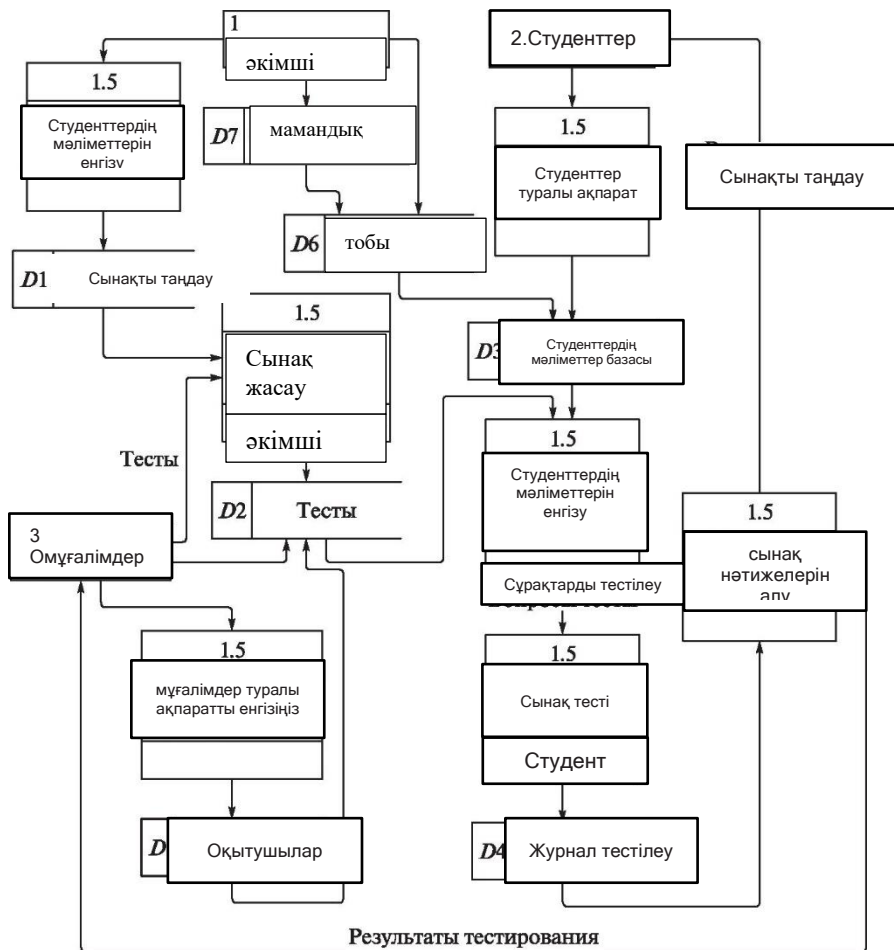
Кесте 13.1. Пайдалану жағдайының сипаттамасы	
Нұсқа атауы	Тестілеу
Максаты	Сынақтан өту, бағалауды алу
Актерлер	Студент
Қысқаша сипаттамасы	Студенттің тіркелуі, тестті бастау, сұрақтар мен жауаптар беру, бірнеше сөйлемнен жауап таңдау немесе жауап беру, тестілеуді аяқтау, бағалауды алу
Нұсқа түрі	Негізгі

Кесте 13.2. «Тестілеу» пайдалану жағдайының сипаттамасы

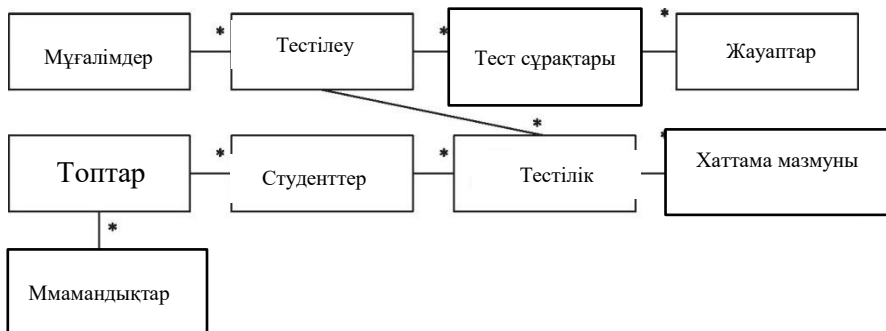
Орындаушы әрекеттері	Жүйенің жауабы
1. Студент өзінің деректеріне (аты, топ) кіріп, жүйеде тіркеледі	2. Жүйе деректерді тексереді, тестті таңдауды ұсынады
3. Студент сынақты таңдайды	4. Жүйе сынақты бастайды
5. Студент үнемі сұрақтарға жауап береді	6. Жүйе дұрыс және дұрыс емес жауаптарды тіркейді
7. Студент тестілеуді аяқтайды	8. Жүйе дұрыс жауаптардың пайызын есептейді
9. Студент нәтижені күтеді	10. Жүйе нәтижелерді көрсетеді және нәтижелерді тестілеу хаттамасын басып шығаруды ұсынады
11. Студент нәтижені басып шығару туралы шешеді	12. Егер басып шығаруды тандасаңыз, жүйе сынақ протоколын басып шығарады
13. Студент нәтижені сақтап қалады ма, жоқ па деп шешеді	14. Егер сақтау тандалса, жүйе нәтижені сақтайды
15. Студент жұмысын аяқтайды	16. Жүйе жабылады



ақпараттық тестілеу жүйесіндегі «Тестілеуді» қолдану нұсқасы (13.11-сурет). Талаптарға талдау жасау үшін құрылымдық әдісті қолдануға болады. Суретте. 13.12, 13.13 жүйеде орын алған негізгі процестерді сипаттайтын Гейн-Сарсон белгілеріндегі деректер ағындарының диаграммасы ұсынылған. Біріншіден, біз кіріс деңгейінің контекстік диаграммасын жасаймыз. Out-



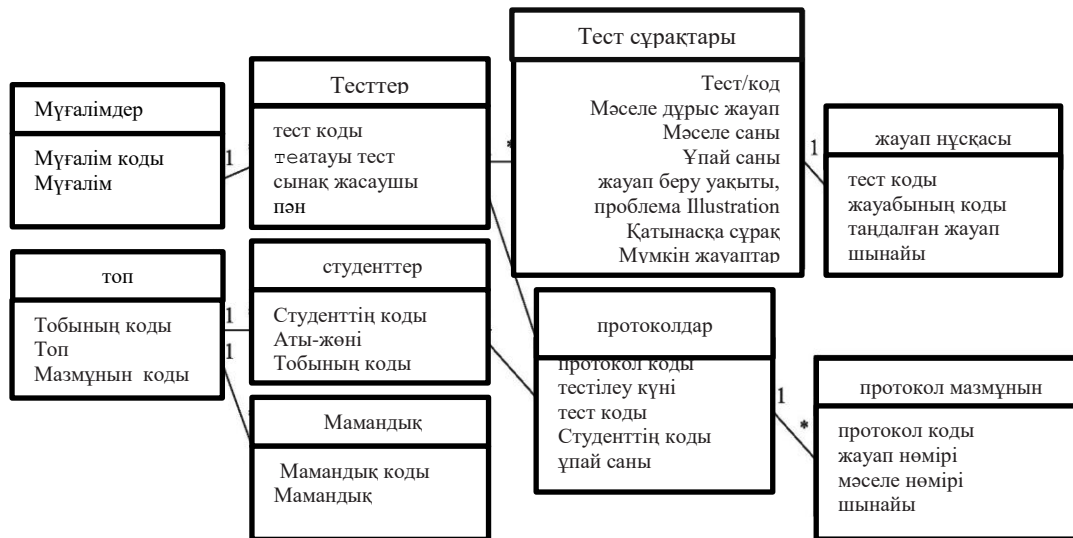
Сур. 13.13. Бірінші деңгейдегі жүйелік процестердің диаграммасы



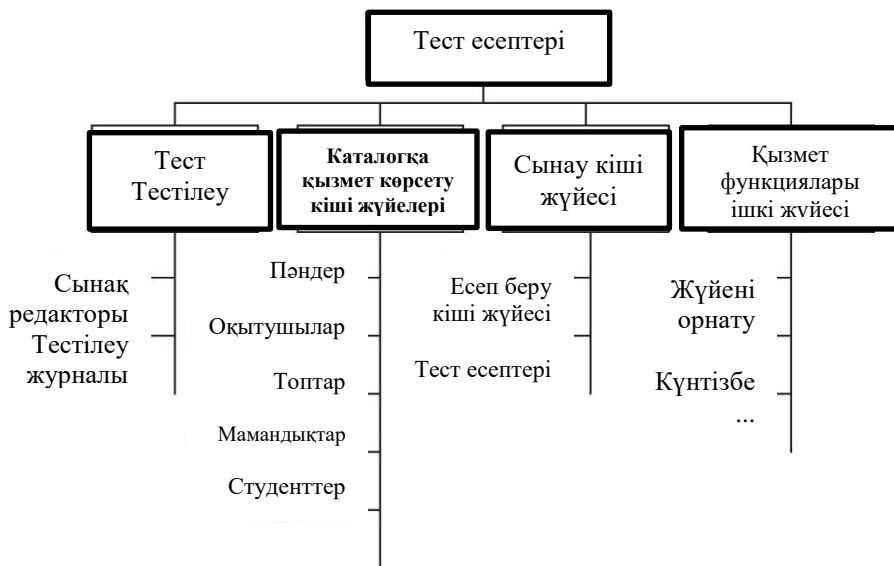
13.14. Сурет. Тұжырымдамалық деректер моделінің алғашқы нұсқасы



13.12. Сурет.



Сурет. 13.15. Автоматтандырылған сынақ жүйесіндегі «мән-сілтеме» диаграммасы



Сурет. 13.16. Автоматтандырылған сынақ жүйесінің сызбалары:
 а - құрылымдық; b – функционалды

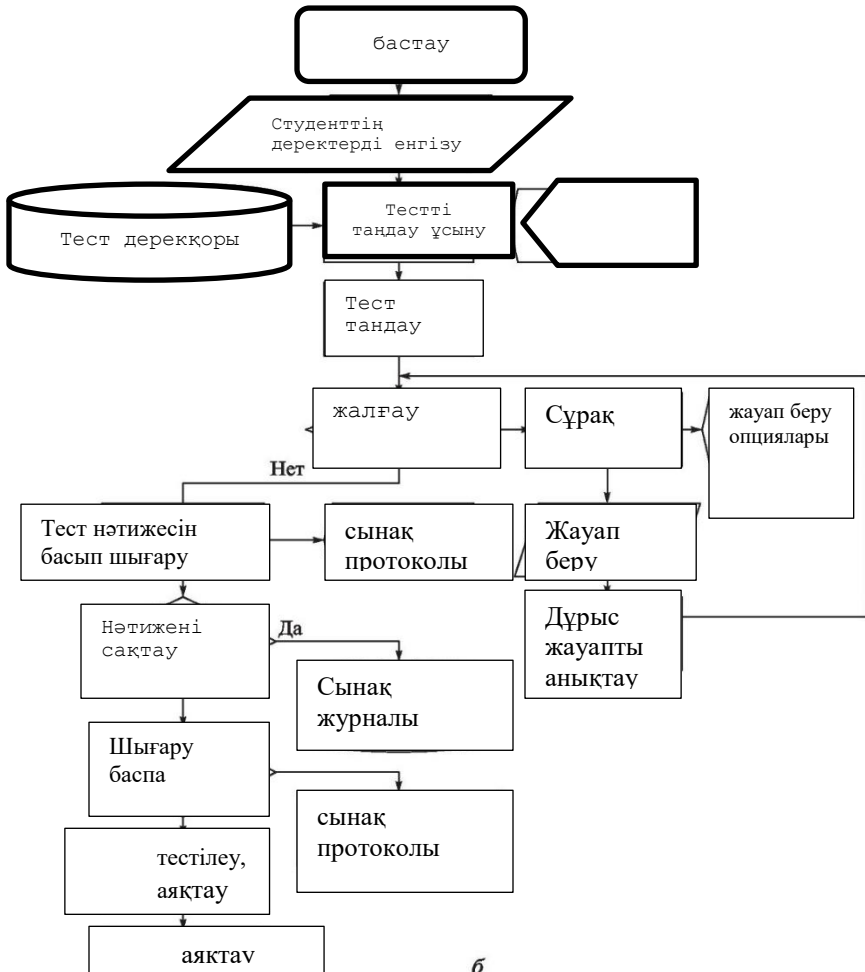
Студенттер мен мұғалімдер - бұл жүйеде жалғыз ғана субъектілер. Мұғалімдер сынақ жасап, оларды жүйеге енгізеді, студенттер тестілеуді таңдап, ұсынылған сұрақтарға жауап береді (13.12-суретті қараңыз). Толық мәліметтердің келесі деңгейінде барлық үрдістер мен деректер ағындары жүйеде бөлінеді, деректерді сақтау құрылғылары бөлінеді, олар деректер құрылымдары арқылы сипатталады (13.13-суретті қараңыз). Деректер ағындарының құрамы анықталғаннан кейін бастапқы ақпарат негізінде тұжырымдамалық деректер моделін құруға болады.

2.2. Деректер үлгісін жобалау

Алдыңғы бөлімде орнатылған үлгілерге негізделген, әрбір нысанға арналған объектілерді автоматтандырылған сынақ жүйесінде таңдап алыңыз. Олардың арасында байланыс бар екенін белгілейік:

1. мұғалімдер сынақтар өткізеді;

2. Тесттер Сұрақтарды қамтиды;
- 3 . жауаптар сұрақтарға жауап береді;



б

5. Мамандандыруға жатады;
6. Студенттер тестілеуден өтеді (Хаттаманы алады);
7. Хаттамада хаттаманың мазмұны бар. Хабарламалар

субъектілер арасындағы өзара әрекеттестікті, және осы өзара әрекеттесу туралы ақпарат жүйеде сақталуы тиіс. «Мағынасы - қосылу» диаграммасын сызыңыз (сурет 13.14) және осы өзара әрекеттесу туралы ақпарат жүйеде сақталуы тиіс. «Мағынасы - қосылу» диаграммасын сызыңыз (сурет 13.14). Әрі қарай, біз әрбір субъектінің атрибуттарын анықтаймыз. «Субъект-қатынас» диаграммасы сурет. 13.15.

2.3. Бағдарламалық жасақтаманың толық дизайны (прототиптің құрылысы) Бағдарламалық жасақтаманың егжей-тегжейлі дизайны үшін біз бағдарламаның құрылымдық және функционалдық схемаларын құрастырамыз (Сурет 13.16, а, б). Бұл схемаларды пайдалану экрандық нысандардың иерархиясын жасау оңай (прототипті жасаңыз).

Бағдарламалық жасақтаманың егжей-тегжейлі дизайны үшін біз бағдарламаның құрылымдық және функционалдық схемаларын құрастырамыз (Сурет 13.16, а, б). Бұл схемаларды пайдалану экрандық нысандардың иерархиясын жасау оңай (прототипті жасаңыз).

БАҚЫЛАУ СҰРАҚТАРЫ

1. «Курстың дизайнын түсіндіру туралы ескерту» қандай негізгі бөлімдерден тұрады?
2. Курстың дизайны қандай тәртіпте жасалады?

Тәжірибелік сабақтарда орындалатын тапсырмалардың нұсқалары

1. «Кітап дүкені» автоматтандырылған ақпараттық жүйесіне арналған бағдарламалық қамтамасыз етуді әзірлеу. Дерекқорда түбіртектерді алу туралы, соның ішінде құжаттың нөмірі және жеткізуші туралы мәліметтер қамтылуы тиіс; кітаптар туралы ақпарат - жанр, аты, автор (лар), жарияланған жылы, баспагер, баспа орны, беттер саны, баға; кітаптарды сату туралы ақпарат (сату күні, көшірмелер саны, сома). Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

2. Параметрлік теңдеу арқылы берілген функцияның графигін құру бойынша бағдарламаны әзірлеу. Бұл жобаны іске қосқан кезде, ауқымды өзгертуге, өндік түстер мен жолдарды өзгертуге мүмкіндік беретін дамыған интерфейс күтесіз. Оң жақ батырманы басқан кезде тінтуірдің курсорының координаттарын және t параметрін көрсету мүмкіндігін қамтамасыз етіңіз.

3. «Студенттік кітапхана» автоматтандырылған ақпараттық жүйесіне арналған бағдарламалық қамтамасыз етуді әзірлеу. Дерекқорда кітаптар туралы мәліметтер - жанр, аты, инвентарлық нөмірі, автор (лар), жарияланған жылы, баспагер, баспа орны, беттер саны, бағасы; студенттерге кітаптарды беру және қайтару туралы ақпарат (нысан); кітаптарды есептен шығару туралы деректер. Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

4.«Қалалық кітапхана» автоматтандырылған ақпараттық жүйесінің бағдарламалық қамтамасыз етуін әзірлеу. Дерекқорда кітаптар - жанр, аты, инвентарлық нөмірі, авторы, жарияланған жылы, баспагері, басылған орны, беттер саны, баға туралы мәліметтер болуы тиіс; оқырмандарға кітаптарды беру және қайтару туралы ақпарат (нысан); оқырмандар туралы мәліметтер (аты, мекен-жайы, паспорт деректері). Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

5.Paint Editor-дің функционалды редакторы жасаңыз.

6.«Мектеп кітапханасы» автоматтандырылған ақпараттық жүйесінің бағдарламалық жасақтамасын құрыңыз. Дерекқорға кітаптар алу туралы ақпарат, оның ішінде жеткізу күні мен жеткізушілер, құжаттың нөмірі; кітаптар туралы ақпарат - жанр, аты, инвентарлық нөмірі, автор (лар), жарияланған жылы, баспагер, басылған орны, беттер саны, баға; студенттерге кітаптарды беру және қайтару туралы ақпарат (нысан); кітаптарды есептен шығару туралы деректер. Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран

пішіндерін жасаңыз.

7. «Кәсіпорынның кадрлық бөлімі» автоматтандырылған ақпараттық жүйесі үшін бағдарламалық қамтамасыз етуді әзірлеу. Дерекқорда кәсіпорынның қызметкерлері туралы ақпарат, оның ішінде аты-жөні, жынысы, туған күні, білімі, лауазымы, мамандығы, бөлімі, жұмыс күні, жалақысы, төлқұжат деректері, мекен-жайы көрсетілуі тиіс. Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

8.«Студенттік кадрлар бөлімі» автоматтандырылған ақпараттық жүйесінің бағдарламалық қамтамасыз етуін әзірлеу. Деректер базасында техникалық мектеп оқушылары туралы толық ақпарат, оның ішінде толық аты-жөні, жынысы, туған күні, тұрғылықты жері, телефон нөмірі, ата-аналар туралы мәліметтер, ата-аналардың телефондары, курсы, топтары, мамандығы, саласы, қаржыландыру түрі, түсу жылы, бітірген жылы туралы ақпарат болуы керек. , студенттік билеттің нөмірі. Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

9.«Музыкалық аспаптар дүкені» автоматтандырылған ақпараттық жүйесінің бағдарламалық қамтамасыз етуін әзірлеу. Деректер базасында дүкендерде музыкалық аспаптарды алу туралы ақпарат (оның түсу күні, құжаттың нөмірі, жеткізушінің ақпараты, саны, сомасы), құралдар туралы ақпарат (атауы, құралы, бағасы) болуы тиіс; музыкалық аспаптарды клиенттерге сату туралы ақпарат (сату күні, саны, сомасы). Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

10. «Музыкалық аспаптар дүкені» автоматтандырылған ақпараттық жүйесінің бағдарламалық қамтамасыз етуін әзірлеу. Деректер базасында дүкендерде музыкалық аспаптарды алу туралы ақпарат (оның түсу күні, құжаттың нөмірі, жеткізушінің ақпараты, саны, сомасы), құралдар туралы ақпарат (атауы, құралы, бағасы) болуы тиіс; музыкалық аспаптарды клиенттерге сату туралы ақпарат (сату күні, саны, сомасы). Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

11.«Көтерме саудалардың қоймасы» автоматтандырылған ақпараттық жүйесінің бағдарламалық қамтамасыз етуін әзірлеу. Деректер базасында тауар туралы қоймада (тауардың түсу күні, құжаттың нөмірі, жеткізуші туралы ақпарат, тауарлар саны, сомасы), тауарлар туралы мәлімет (атауы, тауардың түрі, бағасы) туралы ақпарат болуы тиіс; сату туралы мәлімет (сату күні, сатылған тауардың саны, сату көлемі, сатып алушы туралы ақпарат). Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

12. «Көтерме саудалардың қоймасы» автоматтандырылған ақпараттық жүйесінің бағдарламалық қамтамасыз етуін әзірлеу. Деректер базасында тауар туралы қоймада (тауардың түсу күні, құжаттың нөмірі, жеткізуші туралы ақпарат, тауарлар саны, сомасы), тауарлар туралы мәлімет (атауы, тауардың түрі, бағасы) туралы ақпарат болуы тиіс; сату туралы мәлімет (сату күні, сатылған тауардың саны, сату көлемі, сатып алушы туралы ақпарат). Деректерді кестелерде және барлық қажетті

шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

13. Телефондар мен аксессуарлардың келуі туралы ақпарат (жеткізу күні, құжаттың нөмірі, жеткізуші туралы ақпарат, жеткізілетін тауарлар саны, сома), сондай-ақ сату туралы ақпарат (сату күні, сатылған тауардың сомасы, сату көлемі) туралы ақпарат енгізу қажет. Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

Қаланың жоспарын жобалау бағдарламасын әзірлеу. Рәміздер мен элементтердің кітапханасын сақтау керек. Жобада элементтерді масштабтау, сызу және жою функциялары жүзеге асырылуы тиіс.

14. «Материалдық қойма» автоматтандырылған ақпараттық жүйесінің бағдарламалық қамтамасыз етуін әзірлеу. Дерекқорда мыналар болуы керек:

- қоймадағы материалдарды алу туралы ақпарат (оның алған күні, құжаттың

- нөмірі, жеткізуші туралы мәліметтер, материалдың көлемі, сома);

- материалдар туралы ақпарат (атауы, материалдың түрі, бағасы); материалдарды

- Материалды өндіріске беру туралы ақпарат (аудару күні, құжаттың нөмірі, сома, сома).

- есептен шығару туралы ақпарат (есептен шығару күні, құжаттың нөмірі, сома);

15. Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

- қойма үшін комбинезондар алу туралы ақпарат (түбіртек алған күні, құжат нөмірі, жеткізуші туралы мәліметтер, жинақталған комбинезон саны);

- комбинезондар туралы ақпарат (атауы, комбинезон түрі (аяқ киім, аяқ киім және т.б.), баға);

- қызметкерлерге киім-кешек беру туралы ақпарат - шығарылған күн, киім (атын, кәсібін, лауазымын) алған қызметкер туралы, киім-кешектерді пайдалану мерзімі.

16. Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз. «Білім беру бөлімі» автоматтандырылған ақпараттық жүйенің бағдарламалық жасақтамасы. Дерекқорда студенттердің студенттік кітабына енгізілген барлық ақпарат (топ нөмірі, мамандық, кафедра, студенттің толық аты, семестр, пәндер, емтихан немесе тест тапсыру күні, мұғалімдер, бағалау) енгізілуі тиіс. Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

17. «Абитуриент» автоматтандырылған ақпараттық жүйесінің бағдарламалық қамтамасыз етуін әзірлеу. Дерекқорда жеке деректер болуы керек, ол өтінімді тапсыру уақытын көрсетеді. Аты-жөні, туған күні, азаматтығы, жынысы, мекен-жайы, таңдаған мамандығы, телефоны, аяқталған оқу орны және оның аяқталу жылы, ата-аналар туралы мәліметтер, қосымша ақпарат (мүгедектер, жетім балалар, жатақхана қажет), оқылатын шет тілі, сертификаттың орташа балы. Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін

экран пішіндерін жасаңыз.

18. Студенттерді оқытуға ақы алуды бақылауға арналған автоматтандырылған ақпараттық жүйенің бағдарламалық жасақтамасын әзірлеу («Төлем алуы»). Деректер базасында студенттер туралы ақпарат (атаулар, топтар, курстар, мамандықтар), ата-аналар туралы ақпараттар, касса бөлімшесіне ақшаны алу туралы ақпарат (түбіртек және сома). Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

19. Жергілікті желіні басқару бағдарламаларын әзірлеу. Бағдарлама клиенттік компьютерлерде және компьютерлік серверде орнатылады, оның ішінде компьютерлік құрылғылар сұралады. Сервердегі пайдаланушы кез-келген уақытта клиенттік компьютерді басқара алады. Тесттердің атауын сақтайтын автоматтандырылған тестілік жүйені бағдарламалық қамтамасыз етуді әзірлеу, сауалдарға арналған суреттер, жауап нұсқалары, әрекет ету мерзімінің лимиттері, дұрыс жауап нөмірі және дұрыс жауап үшін балл саны.

20. Сонымен қатар тестілеуден өткен (аты, топ), тестілеудің және тестілеудің нәтижелері туралы сұрақ (сұрақ нөмірі, таңдалған жауаптың нөмірі, дұрыс немесе қате, жауап берілді), оқушы ұпайлары сақталуы керек. Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

21. Электрондық сынып журналын жүргізу үшін бағдарламалық қамтамасыз етуді әзірлеу. Дерекқор әрбір топтағы топтардың нөмірін, студенттердің тізімін сақтайды. Әрбір пән үшін журналдың, оқытушының, пәннің пәні, сыныпқа кіру туралы белгілер мен таңбаларды толтыру күні жазылады. Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз. Жергілікті желіні басқару бағдарламаларын әзірлеу. Бағдарлама клиенттік компьютерлерде және клиенттік компьютерлік құрылғылар сұралатын серверлік компьютерде орнатылуы керек. Пайдаланушы кез келген уақытта жергілікті желідегі компьютерлерде қол жетімді құрылғылардың және жүйелік бағдарламалық құралдардың тізімін ала алады.

22. «Оқу тобы» автоматтандырылған ақпараттық жүйесінің бағдарламалық қамтамасыз етуін әзірлеу. Дерекқорда техникалық мектептің бір студенттік тобы, оның ішінде аты, жынысы, туған күні, тұрғылықты мекен-жайы, телефон нөмірі, ата-аналар туралы мәліметтер, ата-аналардың жұмыс телефоны, студенттік карта нөмірі, сондай-ақ студенттердің жетістіктері туралы мәліметтер (сынақ деректері кітаптар). Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз. «CD-дискілердің электронды каталогы» автоматтандырылған ақпараттық жүйесінің бағдарламалық жасақтамасын әзірлеу.

23. «Компьютерлік салон» автоматтандырылған ақпараттық жүйесінің бағдарламалық жасақтамасын әзірлеу. Деректер базасында компьютерлер мен олардың компоненттерін алу және сату туралы ақпарат: алу күні, құжаттың нөмірі, жеткізуші, компьютердің түрі, оның үлгісі және өндірушісі болуы тиіс. Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін

жасаңыз.

24 Жергілікті есептегіш желілерді (LAN) жобалау схемаларын жасау бойынша бағдарламаны әзірлеу. Жоба ғимаратты құру, LAN элементтерін құру және редакциялау үшін бөлек қабатты жоспар құру мүмкіндігін, ғимараттың жоспарына сәйкес осы элементтерді құрастыру құралын қолдауға тиіс.

25. Оқу орындарында компьютерлер мен компоненттерді есепке алудың автоматтандырылған ақпараттық жүйесі үшін бағдарламалық жасақтама әзірлеу. Дерекқорда білім беру мекемесі шегінде компьютерлер мен олардың компоненттерінің бар болуы мен қозғалысы туралы ақпарат болуы тиіс: компьютер нөмірі, орналасқан жері (шкаф нөмірі және материалдық жауапты тұлға), компьютерлік құрамы - компоненттердің тізімі (компонент түрі, моделі және өндірушісі). Кестелер және барлық шығыс есептерде енгізу және редакциялау деректер үшін экрандық нысандары жасаңыз.

26. «Қалалық телефон анықтамалығы» автоматтандырылған ақпараттық жүйесінің бағдарламалық жасақтамасын әзірлеу. Дерекқорда абоненттің тегі, үй мекенжайы және телефон нөмірі болуы керек. Қалалық елді мекендер мен ауылдарды есепке алу қажет. Сонымен қатар, бизнес-телефондар (фирманың атауы, мекен-жайы, бөлімі, телефон нөмірі) деректер базасында есепке алынады.

27. «Қалалық деректер банкі» автоматтандырылған ақпараттық жүйесі үшін бағдарламалық жасақтама әзірлеу. Деректер базасында жеке деректер болуы тиіс: аты-жөні, туған күні, азаматтығы, жынысы, мекен-жайы, туған жері, СТН, сақтандыру нөмірі, телефон нөмірі, отбасылық жағдайы, қосымша ақпарат (мүгедек, жетім бала) сайлау учаскесінің. Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

28. Сұрыптау әдістерінің жұмысын анық көрсететін бағдарламаны әзірлеу: іріктеу және енгізу.

Сорттарды салыстыру санымен, алмасу санымен салыстыру. Мұны істеу үшін біз осы сандардың тәуелділіктерін массивтің элементтерінің санына салуымыз керек.

29. Сұрыптау әдістерін суреттейтін бағдарламаны жасаңыз: қарапайым кірістіру және екілік енгізу. Сорттарды салыстыру санымен, алмасу санымен салыстыру. Мұны істеу үшін біз осы сандардың тәуелділіктерін массивтің элементтерінің санына салуымыз керек.

Сұрыптау әдістерінің жұмысын анық көрсететін бағдарламаны әзірлеу: іріктеу, қарапайым кірістіру және екілік енгізу.

30. Сорттарды салыстыру санымен, алмасу санымен салыстыру. Мұны істеу үшін біз осы сандардың тәуелділіктерін массивтің элементтерінің санына салуымыз керек.

31. «Видеалон» автоматтандырылған ақпараттық жүйесі үшін бағдарламалық жасақтама әзірлеу. Дерекқорда сатылатын кассеталар мен CD-ROM туралы барлық ақпарат болуы тиіс.

32. Жұмыстарды (жанрды, атын, орындаушысын, шығарылған жылы мен елін), видеотаспалар мен дискілерді алу туралы ақпараттарды қамтуы тиіс (оның ішінде алынған күні, құжаттың нөмірі, жеткізуші

туралы ақпарат, жеткізілетін дискілердің саны (кассеталар), алынған сома), сондай-ақ бейне дискілерін сатуға қатысты ақпарат (сату күні, сатылған дискілердің саны, сатылған сома). Деректерді кестелерде және барлық қажетті шығыс есептеріне енгізу және өңдеу үшін экран пішіндерін жасаңыз.

Әдебиеттер тізімі

1. ГОСТ 19.701—90 (ИСО 5807-85). Единая система программной документации (ЕСПД).
2. *Архангельский А. Я.* Delphi 7 / А.Я.Архангельский. — М.: ЗАО «Издательство БИНОМ», 2004.
3. *Архангельский А. Я.* Приемы программирования в Delphi / А.Я.Архангельский. — М.: ЗАО «Издательство БИНОМ», 2004.
4. Базы данных / под ред. А.Д.Хомоненко. — СПб.: КОРОНА принт, 2002.
5. *Бобровский С.* Delphi 7 / С. Бобровский. — СПб.: Питер, 2005.
6. *Бутаков Е.А.* Методы создания качественного программного обеспечения ЭВМ / Е.А. Бутаков. — М.: Энергоатомиздат, 1984.
7. *Ван-Тассел Д.* Стилль, разработка, эффективность, отладка и испытание программ / Д.Ван-Тассел. — М.: Мир, 1981.
8. *Вендров А. М.* Практикум по проектированию программного обеспечения экономических информационных систем / А. М. Вендров. — М.: Финансы и статистика, 2002.
9. *Гагарина Л. Г.* Основы технологии разработки программных продуктов / Л. Г. Гагарина, Б. Д. Виснадул, А. В. Игошин.— М.: ФОРУМ — ИНФРА-М, 2006.
10. *Голицына О.Л.* Базы данных / О.Л.Голицына, Н.В.Максимов, И. И. Попов. — М.: ФОРУМ — ИНФРА-М, 2003.
11. *Горев А.* Эффективная работа с СУБД / А. Горев, Р.Ахаян, С. Макашарипов. — СПб.: Питер, 1997.
12. *Дарахвелидзе П. Г.* Программирование в Delphi 7 / П.Г.Дарахвелидзе, Е.П. Марков, О. А. Котенок. — СПб.: БХВ — Санкт-Петербург, 2005.
13. *Емельянова Н. З.* Основы построения автоматизированных информационных систем / Н. З. Емельянова, Т.Л.Партыка, И. И. Попов. — М.: ФОРУМ: ИНФРА-М, 2005.
14. *Жилинский А. А.* Самоучитель Microsoft SQL SERVER 2005 / А.А. Жилинский. — СПб.: БХВ-Петербург, 2007.
15. *Жоголев Е.А.* Введение в технологию программирования: конспект лекций / Е.А.Жоголев. — М.: ДИАЛОГ-МГУ, 1994.
16. Информатика / под ред. Н. В. Макаровой. — 3-е изд., перераб.; — М.: Финансы и статистика, 2004.
17. *Кандзюба С. П.* Delphi. Базы данных и приложения / С. П. Кандзюба, В.Н.Громов. — СПб.: ООО «ДиаСофтЮп», 2005.
18. *Ковязин А.* Мир InterBase / А. Ковязин, С. Востриков. — М.: КУДИЦ-ОБРАЗ, 2006.
19. *Кузьменко В. Г.* Базы данных в Visual Basic VBA. Самоучитель / В.Г.Кузьменко. — М.: ООО «Бином-Пресс», 2007.
20. *Культин Н.* Программирование на Object Pascal / Н. Культин. —

СПб.: БХВ-Петербург, 1998.

21. **Орлов В.В.** Технологии разработки программных продуктов / В.В. Орлов. — СПб.: Питер, 2003.
22. **Пономарев В. А.** COM и ActiveX в Delphi / В. А. Пономарев. — СПб.: БХВ — Петербург, 2001.
23. **Рудаков А. В.** Технология разработки программных продуктов / А. В. Рудаков. — М.: Издательский центр «Академия», 2005.
24. **Сорокин А. В.** Delphi. Разработка баз данных / А.В.Сорокин. — СПб.: Питер, 2005.
25. **Тейлор Аллен** Дж. SQL для «чайников» / Аллен Дж.Тейлор; пер. с англ. — М.: Вильямс, 2005.
26. **Фаронов В.В.** Программирование баз данных в Delphi 7 / В. В. Фаронов. — СПб.: Питер, 2005.
27. **Фаронов В. В., П.В. Шумаков.** Delphi. Руководство разработчика баз данных / В. В. Фаронов, П.В. Шумаков. — М.: Нолидж, 1999.
28. **Фленов М. Е.** Delphi 2005. Секреты программирования / М. Е. Фленов. — СПб.: Питер, 2006.
29. **Фленов М. Е.** Библия Delphi / М. Е. Фленов. — СПб.: БХВ-Петербург, 2005.
30. **Шкрыль А. А.** Разработка клиент-серверных приложений в Delphi / А. А.Шкрыль. — СПб.: БХВ-Петербург, 2006.
31. **Шураков В. В.** Надежность программного обеспечения / В. В. Шураков. — М.: Финансы и статистика, 1987.

Алғысөз.....	4
1 Тарау. Техникалық қызметті дамыту	5
1.1. Негізгі ақпарат	5
1.2.Техникалық тапсырманы әзірлеудің мысалдары.....	10
1.2.1. AIS «Көтерме сауда қоймасы» бағдарламалық жасақтамасын әзірлеуге арналған техникалық тапсырма.....	14
1.2.2. Комбинаторлық мәселелерді шешетін жүйені әзірлеуге арналған техникалық тапсырма.....	14
2 тарау. Бағдарламалық қамтамасыз етуге қойылатын талаптарды талдау және құрылымдық ұстанымды анықтауда жүйелі әдісті қолдану.....	18
2.1. Негізгі ақпарат.....	18
2.2. Жағдайлардың өтпелі диаграммалары.....	21
2.3. Функционалдық диаграммалар.....	24
2.4. Ақпараттық ағын жүйелері.....	34
2.5. «Маңызды байланыс» диаграммасы.....	44
3 тарау. Бағдарламалық жасақтама жобасы	
3.1. Блок диаграммасы (құрылымдық схема).....	50
3.2. Функционалдық схема.....	51

4-тарау. Бағдарламалық қамтамасыз етуді талдау және жобалауға объектілі-бағдарланған тәсілді қолдану.....	56
4.1. Қолданыс түрлеріне қатысты диаграммалар.....	56
4.2. Қызмет көрсету диаграммалары.....	65
4.3. Кезектілік диаграммалары.....	72
4.4. Класстық диаграмма.....	75
5-тарау Бағдарламалық жасақтама прототипін жасау	
5.1. Прототиптер туралы негізгі мәліметтер.....	83
5.2. Прототип түрлері.....	84
5.3. Прототип құру мысалдары.....	87
6-тарау. Пайдаланушы интерфейсін жобалау.....	90
6.1. Интерфейсті құрудың негізгі ережелері.....	90
6.2. Пайдаланушы интерфейсін құру принциптері.....	91
6.3. Пайдаланушы мен компьютер арасындағы өзара байланыс.....	92
6.4. Экранға мәліметтерді орналастыру.....	93
6.5. Қателерді болдырмау, анықтау және түзету.....	100
6.6. Графикалық интерфейске қойылатын жалпы талаптар.....	101
7 тарау. Объектілі-бағдарланған бағдарламалау.....	103
8 тарау. Тестілеу стратегиясын таңдау және тестілеу.....	112
8.1. Тестілеу деңгейлері.....	112
8.2. Тестілеу технологиясы.....	114
8.3. Бағдарлама қателері	116
8.4. Тестілеу түрлері.....	119
9 тарау. COM технологиясын пайдалану кезінде компоненттік тәсілді қолдану COM-технологиясы және OLE автоматтандыру ұғымдары.....	124
9.1. Delphi -дегі OLE автоматтандыруды.....	124
9.2. Excell кестесі арқылы көрсету.....	129

10 тарау. Компонентті бағдарламалау негізінде динамикалық кітапхананы құру.....134

10.1 Динамикалық кітапхана жөнінде негізгі мәлімет..... 134

10.2. Динамикалық кітапхана құру мысалдары..... 135

11-тарау. Пайдаланушы үшін құжаттама жасау. Бағдарламалық өнімдердің анықтамалық жүйесін құру.....144

11.1. «Пайдаланушы нұсқаулығы».....144

11.2. AIS «Көтерме сауда қоймасы» үшін «пайдаланушы нұсқаулығы» құжатының үзінділерін әзірлеу үлгісі
.....146

11.3. Анықтамалық жүйені дамыту..... 146

12 тарау. Бағдарламалық жасақтаманы орнату..... 153

12.1 Орнатудың маңыздылығы.....152

12.2. Бағдарламалық жасақтаманы орнату процесі..... 152

13 тарау.Курстық жобалау..... 163

13.1. Түсіндірме жазбаның құрылымы мен мазмұны.....163

13.2. Курстық жобаларды рәсімдеу.....168

Әдебиеттер тізімі.....198

Оқу басылымы
Рудаков Александр Викторович,
Федорова Галина Николаевна

Бағдарламалық қамтамасыз етуді әзірлеу технологиясы.

Семинар

Оқу құралы
Редактор Ж.Ж. Сыдықова
Техникалық редактор Н.И. Горбачев

Компьютер орналасуы: Е.Назарова
Түзеткіштер А.С. Сазов, Г.А.Форенцова
Ed. №105114016. Баспасөзге қол қойылды 04.04.2014ж. Формат 60 x 90/16.
Гарнитура Baltika. Қағаздар. №1. Офсетті басып шығару. Us1. принтер. л. 12.0.

Тираж 1 200 дана. Тапсырыс №
«Академия» баспа орталығы »ЖШС. www.academia-moscow.ru 129085, Мәскеу қаласы,
Мира даңғылы, 101Б, 1-б.Тел./факс: (495) 648-0507, 616-00-29.
Санитарлық-эпидемиологиялық есеп № РОСС RU. АЕ51. Н 16476 04.05.2013
Баспа үйі берген Саратов полиграфиялық комбинатының электронды баспадан басылған.
www.sarpk.ru 410004, Саратов, ул. Чернышевский, 5