

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

«Компьютерлік технологиялар» кафедрасы

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

(КОЛЫ)

« _____ » _____ 2014 ж.

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: ««Алматы Энергетика және Байланыс Университетіне
арналған мобильді қосымша құру»
5B070400 – «Есептеу техникасы және бағдарламалық камтамасыз ету»
мамандығы бойынша

Орындаған Сағындықов Ербол Тұрарбекұлы тобы: ВТк-10-2

Жетекші доцент, ф.-м.ғ.к. Аманбаев А.А.

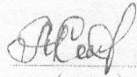
Кеңесшілер :

Экономикалық бөлім бойынша :

 доцент Боқанова Г.Ш.
« 30 » 05 2014 ж.
(колы)


Өмір тіршілігі қауіпсіздігі бойынша:

т.ғ.к., аға оқытушы Муташева Г.С.

 « 15 » 05 2014 ж.
(колы)

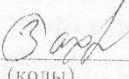
Есептеу техникасын қолдану бойынша :

ф.-м.ғ.к., доцент Аманбаев А.А.

 « _____ » _____ 2014 ж.
(колы)

Мөлшер бақылаушы:

аға оқытушы Рахимжанова З.М.

 « 04 » 06 2014 ж.
(колы)

Пікір жазушы :

т.ғ.д., ҚазЭУ ҚИ кафедрасының проф. Абдылдаев Э.К.

 « _____ » _____ 2014 ж.
(колы)

Алматы 2014

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

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

жобаны орындауға берілген

ТАПСЫРМА

Студент Сағындықов Ербол Тұрарбекұлына

Жоба тақырыбы Алматы Энергетика және Байланыс Университетіне арналған
мобильді қосымша құру
ректордың «24» қараша № 115 бұйрығы бойынша бекітілген.

Аяқталған жұмысты тапсыру мерзімі: «03» маусым 2014 ж.

Жобаға бастапқы деректер (талап етілетін жоба нәтижелерінің параметрлері және нысанның бастапқы деректері): Android жүйесіне мобильді АЭЖБУ-не арналған қосымша құру үшін келесі деректер қолданылды:

- АЭЖБУ-нің электронды ақпараттары;
- АЭЖБУ-нің электронды кітапханасы;
- Eclipse бағдарламалау ортасы мен MySQL деректер қоры.

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

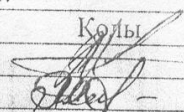
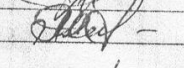
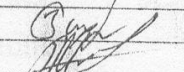

- Берілген жобаның ER моделін жобалау, жобаның жұмысын жақсарту үшін қалыптау әдістерін қолдану;
- Жобаның алгоритмін құру, пайдаланушы және админдік интерфейстің сызбасын жасау;
- Ақпараттық жүйелер, дерекқорлар (ДҚ), дерекқорларды басқарудың жүйелері (ДҚБЖ) түсініктері;
- Жүйенің дерекқорын жобалау және жүзеге асыру, жобалау құралдарын таңдау, дерекқорды құру;
- Пайдаланушы интерфейсін жобалау, дерекқормен байланысын орнату
- Техникалық-экономикалық негізделуі;
- Өндірістегі еңбек қорғау мен қауіпсіздік шараларын ұйымдастыру.
- Құрылған жобаны тестілеу.

- Құрылған жобаны тестілеу.
- Сызбаматериалдарының (міндетті түрде дайындалатын сызуларды көрсету) тізімі:
- Дерекқордың ER диаграммасы.
 - Пайдаланушы және админдік интерфейстің сызбасы.
 - Жасалған бағдарламалық қосымшаның жұмыс істеу қабілеттілігін мен функционалдығын көрсететін графикалық материал.

Негізгі ұсынылатын әдебиеттер:

1. Герберт Шилдт. Java: полное руководство = 8-изд. The Complete Reference. - М.: «Вильямс», 2012. — С. 1104.
2. Android 2: программирование приложений для планшетных компьютеров и смартфонов: [пер. с англ.] Рето Майер. - М. Эксмо 2011 - 672 с.
3. Харди Б., Филлипс Б. Программирование под Android: Для профессионалов - СПб.: Питер, 2014. - 592 с.
4. Эккель Б. Философия Java. Библиотека программиста. 4-изд. - СПб.: Питер, 2009. - 640 с.
5. <http://developer.android.com/design/index.html>
6. Рик Роджерс, Медниекс Зигурд и др. «Android. Разработка приложений» 2010г. . — С. 685.
7. Shane Conder, Lauren Darcey «Android Wireless Application Development» Sept. 7, 2009. — С. 847.
8. Dave MacLean и др. «Pro Android 2» (Paperback - March 15, 2010) . — С. 675.

Жоба тараулары бойынша кеңес берушілер және оның мерзімі:

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

ДИПЛОМ ЖОБАСЫН ДАЙЫНДАУ

КЕСТЕСІ

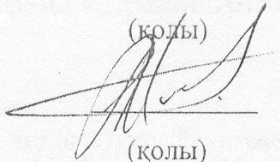
№ р/с	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
1	Кезігі бойы	09.09.2013	
1.2	Мобильді құрылғыларға қосымша мәлімет құрудың кезіңдері	16.09.2013	
1.3	Еclipse IDE платформасы туралы	23.09.2013	
1.4	Android SDK туралы туралы	30.09.2013	
1.5	GetHub жобалар қосымши веб сервисі туралы	1.10.2013	
2	Арнайы бойы	25.10.2013	
2.1	АЭЖБҮ мобильді қосымши сайынғы жаңы сипаттамасы	28.10.2013	
2.2	Қажетті құралдарды дайындау	10.12.2013	
2.3	Қосымшаның Java Eclipse IDE платформасында жүзеге асырылуы	15.01.2014	
3	Тұрақты құдісіздігі	08.04.2014	
3.1	Оператордың жұмыс тәртібінің талдауы	15.04.2014	
3.2	Таблиця тәртібінің есептеу	22.04.2014	
4	Техникалық экономикалық кезің	27.04.2014	
4.1	Өзіндік жұмыс нәтижесінің кестесі	29.04.2014	

Тапсырманың берілген уақыты « _____ » _____ 2014 ж.

Кафедра меңгерушісі
З.К.

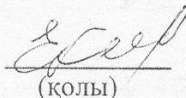
ф.-м.ғ.д., проф. Құралбаев

Жоба жетекшісі

(КОЛЫ)


ф.-м.ғ.к., доцент Аманбаев А.А.

Орындалаты тапсырманы қабылдаған студент

(КОЛЫ)


Сағындықов Ербол Тұрарбекұлы

АҢДАТПА

«Алматы энергетика және байланыс университетіне арналған мобильді қосымша құру» дипломдық жұмысында көптеген зерттеулер жасалды.

Дипломдық жобаның басты мақсаты – студенттер мен талапкерлерге қолжетімді және қолайлы түрде ақпаратпен қамтамасыз ете алатын анықтамалық қосымша құру. Дипломдық жобаны орындау кезінде PHP және Java тілі және мобильді құрылғылардың Android SDK құралы қолданылды.

Өміртіршілік қауіпсіздігі бөлімінде жұмыс бөлмесіндегі жасанды және табиғи жарықтанудың есептері келтірілген.

Экономикалық бөлімде енгізілген жүйеден алынған экономикалық пайданы есептеу жүргізілген.

АННОТАЦИЯ

В дипломной работе «Разработка мобильного приложения для Алматинского университета энергетики и связи» проводились множество исследований.

Целью дипломной работы является разработка доступного и удобного информационного мобильного приложения для студентов и абитуриентов .

В ходе выполнения дипломной работы были использованы PHP, объектно-ориентированный язык программирования Java и Android SDK.

В части по безопасности жизнедеятельности представлены расчеты искусственного и естественного освещения в рабочем помещении.

В экономической части произведен расчет полученной экономической выгоды от внедряемой системы.

ABSTRACT

In degree work "Developing mobile application for Almaty university of power engineering and telecommunications" conducted great number research.

The main purpose of the degree project is the creation of application for students and entrants to give information more comfortable and available .

In the course of the degree project used PHP, object-oriented programming language Java, a tool for mobile application development Android SDK.

As part of Health and Safety presents calculations of artificial and natural lighting in the operating room.

In the economic part of a calculation of economic benefits derived from the system being implemented.

Мазмұны

Кіріспе	8
1 Мобильді құрылғыларға қосымшалар құрудың негіздері	Ошибка! Закладка не определена.
1.1 Мобильді құрылғылар платформалары мен құру жабдықтарына анализ	Ошибка! Закладка не определена.
1.2 Eclipse IDE платформасы туралы	11
1.3 Java Eclipse IDE платформасында мобильді құрылғыларға қосымшалар құру мүмкіндіктері	11
1.4 Android платформасы туралы	12
1.5 Android SDK құралы туралы.....	Ошибка! Закладка не определена.
1.6 Java бағдарламалау тілі туралы	12
1.7 Git файл нұсқаларын басқарудың таратылған жүйесі туралы	12
1.8 GitHub жобалар хостингы веб сервисі туралы	Ошибка! Закладка не определена.
2 Арнайы бөлім.....	Ошибка! Закладка не определена.
2.1 АЭЖБУ мобильді қосымшасының жалпы сипаттамасы	Ошибка! Закладка не определена.
2.2 Қажетті құралдарды дайындау	19
2.3 Қосымшаның Java Eclipse IDE платформасында жүзеге асырылу	21
3 Тіршілік қауіпсіздігі.....	32
3.1 Оператордың жұмыс жағдайын талдау	32
3.2 Жұмыс орнының микроклимат жағдайы, өрт қауіпсіздігі	32
3.3 Жұмыс орнының табиғи және жасанды жарықталуы, электр қауіпсіздігі	32
3.4 Бөлме жоспары	32
3.5 Жасанды жарықтандыруды есептеу.....	33
3.6 Табиғи жарықты есептеу	34
4 Техникалық экономикалық негізделуі.....	56
4.1 Жоба сипаттамасы.....	56
4.2 Бағдарламалық қамтаманың өңделуінің еңбек өнімділігінің есептелуі....	60
4.3 Бағдарламалық қамтаманы өңдеуге кеткен шығынның есептелуі	61
4.4 Қосымша шығындар статьясы	66
4.5 Өзіндік құн нәтижесінің кестесі	66
4.6 Бағдарламаның бағасын есептеу	66
4.7 Минималды бағаны, кірісті және пайданы есептеу	72
4.8 Бағдарламалық өнімді иемденудегі кәсіпорындардың бірмезгілдегі шығындарын есептеу	73
4.9 Қолдану аясындағы жылдық бірмезгілдегі шығындарды есептеу	74
4.10 Ақпараттық өнімді енгізуден алынған табысты есептеу	75
4.11 Пайда мен шығындарды есептеу	76
4.12 Ақшалай құралдардың қозғалысы.....	74
4.13 Экономикалық тиімділікті есептеу	75
4.13.1 Таза ағымдағы құндылықты есептеу	76
4.13.2 Пайда индексін есептеу	76
4.13.3 Табыстың ішкі нормасын есептеу	76
4.13.4 Өтімділік периодын есептеу	76
Қорытынды	77
Пайдаланылған әдебиеттер тізімі	78
А қосымшасы.....	79

Кіріспе

«Алматы энергетика және байланыс университетіне мобильді қосымша құру». Осы дипломдық жұмысты орындау кезінде жасаушы құралдардың жаңашыл тәсілі Android платформасы, Java объектіге бағытталған программалау тілі және мобильді құрылғылардың Android SDK құралы қолданылды.

Бүгінде мобильді құрылғылар нарығында үш компанияның басымдылығын байқауға болады. Оларды қолданушылар арасындағы танымалдылығы бойынша бөлетін болсақ, бірінші орында Apple, iOS платформасы (iTunesAppStore), екінші Google, Android платформасы (AndroidMarket), үшінші Microsoft, Windows Phone платформасы (Windows marketplace). Олармен қатар Symbian, BlackBerry OS и HP webOS платформалары да бар.

Шет ел мамандары мобильді құрылғыларға арналған мобильді қосымшалардың қарқынды дамып кетуін алдынан болжаған болатын. 2009 жылдың өзінде IDC маркетингтік–аналитикалық агенттігі 2013 жылға қарай әлем бойынша 1,19 миллиардтан астам маман (жалпы жұмыс күшінің 34,9%–ы) мобильді технологияларды қолданатынын айтқан болатын.

Қазақстанда мобильді технологиялардың дамуы жоғарыда айтылған әлемдік даму қарқынынан төмен. Қазақстанда мобильді қосымшаларды жасау және

ендірумен айналысатын компаниялар көп емес. Көптеген мобильді қосымша жасаушылар өздері жұмыс жасауда. Бұған себеп – сұраныстың болмауы. Мобильді қосымшалар бағасының жоғары болу себебінен аз сұранысқа ие болып отыр.

Қазірге Қазақстанда мобильді қосымшаларға тек бірнеше үлкен банктер, интернет алаңдар және БАҚ ие.

Қазнет мониторингі нәтижесінде танымал мобильді қосымшалар тізімі жасалды.

Мысалға, MobileCreators командасы iOS және Android үшін «People o'Clock», «DialerOne», «Кинозавр» және «GQ BarLab» қосымшаларын жасады.

«Глатис» студиясы өзінің iDrive.kz сайтына Android платформасында «iDrive: ПДД и штрафы РК» қосымшасын жасап шағарды. Және де Homebank.kz iPhone қосымшасының және Альянс–Банктың iOS қосымшасының дизайнын құрды.

Kaznetmedia командасымен iOS және Android үшін «NewsBox» және Apple компаниясының құрылғылары үшін «Vnews.kz», «7 канал (Қазақстан)», «Фокус», «Вести» және «PM.kz» қосымшалары жасалды.

Android қосымшаларының арасынан қазақ тілінде жазылған шығармаларды табуға болады. Бірақ бұл қосымшалар бір немесе бірнеше автордың шығармалар жинағы болып табылады және басқа шығармаларды оқуға мүмкіндік бермейді.

1. Мобильді құрылғыларға қосымшалар құрудың негіздері

1.1 Мобильді құрылғылар платформалары мен құру жабдықтарына анализ

Мобильді құрылғыларға қосымша түсінігі.

Мобильді қосымша – мобильді телефон, коммуникатор, смартфон және т.б. мобильді құрылғыларға арналған бағдарламалық жабдық.

Мобильді қосымшалар бастамада жалпы құрылғының тиімділігін арттыру және ақпарат беру мақсатында жасалған болатын. Мысалға: электронды хат жолдау, күнтізбе, контактілер, фондтық биржа және ауа райы қосымшалары. Дегенмен көпшілік сұраныс пен қосымша жасау құралдарының дамуы мобильді ойындар, GPS және оның негізіндегі сервистер, банкинг, ұсыныс қабылдау, билет сату секілді қосымшалардың пайда болуына әкелді.

Android операциялық жүйелердің арасында жаңалық болып табылады. Бұл операциялық жүйе мобильді құрылғылардың кең шеңберіне арналған. Android операциялық жүйесі коммуникаторларға, планшетті компьютерлерге, смартфондарға және нәтбуктерге орнатылады.

Бұл жүйені жасап шығарған – "Android Inc." компаниясы. Бұл компанияны кейін "Гугл" (Google) компаниясы сатып алған болатын. Қазіргі уақытта Android негізіндегі жүйелерді даму және өндіруді "OpenHandsetAlliance" компаниясы қолға алған. "OpenHandsetAlliance" Гугл ғана емес, сонымен

қатар Моторола, НТС, Интэл, Самсунг және көптеген басқа да техникалық өндіріс алыптарын біріктіреді.

Android операциялық жүйесі Linux операциялық жүйесі негізінде жасалған. Бірақ Linux-тің барлық ерекшеліктерін қамтымайды. Бұл "Дэлвик" виртуалды машинасын пайдаланумен байланысты. Барлық программалық жабдықтың жұмысы осы виртуалды машинада атқарылады.

Өндірушілер бір орында тұрмайды. Android жүйесі үнемі жетілдіріліп және техниканың жаңа түрлеріне ендірілуде.

Android операциялық жүйесі 2008 жылдың қыркүйегінде пайда болды. Бастапқыда ол T-Mобайл Г1 (T-MobileG1) коммуникаторы үшін жасалды. Жарты жылдан кейін жаңа, жетілдірілген әрі өте ыңғайлы 1.0 нұсқасы пайда болды.

2009 жылы операциялық жүйенің төрт жаңартуы таныстырылған болатын. Осылай, ақпанда әр түрлі қателерді түзетумен 1.1 нұсқасы шықты. Сәуірде және қыркүйекте тағы екі жаңарту жарыққа шықты – 1.5 «Cupcake» және 1.6 «Donut». «Cupcake» жаңартуы маңызды өзгерістерді алып келді: виртуалды пернетақта, видео жазу және ойнату, браузер және т.б. «Donut» жаңартуында алғаш рет әр түрлі экранның тығыздығы мен кеңейтілімдерін және CDMA желілерін қолдау пайда болды. Сол жылдың қазанында Google-дың бірнеше аккаунттарын қолдануға мүмкіндік беретін, HTML5 тілін және басқа да жаңалықтарды қолдайтын браузері бар Android 2.0 «Eclair» операциялық жүйесі жасап шығарылды.

2010 жылдың ортасында Google «Froyo» атауымен Android 2.2 нұсқасын, ал 2010 жылдың соңында – Android 2.3 «Gingerbread» нұсқасын таныстырды. «Froyo» жаңартуынан кейін смартфонды қолжетімділік нүктесі ретінде пайдаланып, смартфонда дәстүрлі цифрлық немесе әріптік – цифрлы кілттерді қолдануға мүмкіндік пайда болды, ал «Gingerbread» жаңартуы «көшіру және қою» функцияларына толық бақылау жүргізуге, қоректі басқару және қосымшаларды бақылауды жақсартуға, бір құрылғыда бірнеше камераларды пайдалануға және т.б. мүмкіндік берді.

2011 жылдың 22 ақпанында планшеттарға бағытталған Android 3.0 «Honeycomb» нұсқасы ресми таныстырылған болатын.

2011 жылдың 19 қазанында шыққан Android 4.0 «IceCreamSandwich» – планшеттерде және смартфондарда қолдануға арналаған алғаш әмбебап нұсқа.

2012 жыл операциялық жүйенің «JellyBean» нұсқасына дейін жаңартуын әкелді. Ол маусым айында 4.1-ші нұсқа ретінде таныстырылды, ал қазанның соңында шағын жаңартудан кейін өз нөмір 4.2-ге ауыстырды.

2008 жылдың 22 қазанында GoogleAndroid операциялық жүйесі үшін онлайн қосымшалар дүкені – AndroidMarket-тің ашылуы туралы жариялады. Келісімге бойынша өндірушілер – 70 %, мобильді байланыс операторлары 30 % пайдаға ие болады. 2009 жылдың ақпанда АҚШ және Ұлыбритания өндірушілері AndroidMarket-ті өз қосымшалары үшін төлем алу мүмкіндігін алды. SonyEricsson – AndroidMarket қосымшалар дүкенінде өз онлайн арнасын іске қосқан бірінші компания. Арнада компания қолдануға кеңес

беретін ойындар және қосымшалар таныстырылған. 2011 жылдың желтоқсанына дейін AndroidMarket–тен 10 миллиард қосымша жүктеп алынған болатын.

Google компаниясы 2012 жылдың наурызында «Кітап», «Android Market», «Музыка» және т.б. мультимедиа сервистерін Google Play біртұтас сервисіне топтастыды. Google Play интернет дүкені 190 елде жұмыс істейді, 700 мыңнан астам қосымшаны қамтиды, ал сервистің жұмыс уақытында 25 миллиард шамасында жүктеу жасалған. Windows Phone – 2010 жылдың 11 қазанында жарық көрген Microsoft компаниясымен құрастырылған мобильді операциялық жүйе. 21 қазаннан бастап жаңа платформаға құралған алғашқы құрылғылар сатылуға түсе бастады. Windows Phone сәйкес болмаса да Windows Mobile операциялық жүйесінің толығымен жаңартылған интерфейсі бар және алғашқы рет Microsoft–тың Xbox Live және Zune медиаплеері қызметі интеграцияланған ізбасары саналады. Жүйенің тұсаукесері Барселонада өткен Mobile World Congress 2010 шарасына байланыстырылды.

1.2 Eclipse IDE платформасы туралы

Eclipse IDE. Eclipse IDE – Java бағдарламалау тілінде қосымшаларды өңдеуге арналған біріктірілген өңдеу ортасы және есептеу техникасы платформасы фреймворкі. Өзге бағдарламалау тілдері JavaScript, PHP, Python (Eclipse IDE 7–нұсқасында қосылмаған), Groovy, C, C++, Scala, Clojure және өзгелері. Eclipse IDE IDE 7.0 нұсқасында Ruby және Ruby on Rails бағдарламалау тілдері алынып тасталды, дегенімен қосымша plug–in ретінде орнатуға болады.

Eclipse IDE жобасының демеушісі Oracle компаниясы болып табылады, дегенмен Eclipse IDE ортасын жасауды бағдарламашылардың тәуелсіз бірлестігі (Eclipse IDE Community) және Eclipse IDE Org компаниясы жүргізеді.

Сапа және мүмкіндіктері бойыша Eclipse IDE соңғы нұсқалары Java тілінде бағдарламалауға арналған IntelliJ IDEA секілді коммерциялық біріктірілген өңдеу орталарынан кем түспейді. Eclipse IDE рефакторинг, ортаның жұмыс мінездемелерін жинақтау, синтаксистік құрылымдарды түспен ерекшелеу, енгізіліп жатқан құрылымдарды автоматты түрде толықтыру, алдын ала анықталған код үлгілері және т.б. мүмкіндіктерді қолдайды.

Eclipse IDE ортасында бағдарлама жазу үшін, ортаны орнатуы үшін және оның жұмысына алдын ала орнатылған Sun JDK немесе J2EE SDK–тың қажетті нұсқасы болуы керек. Eclipse IDE ортасында J2SE және J2EE платформаларына арнап бағдарламалау қамтамасыз етілген. Eclipse IDE 6.0 нұсқасынан бастап жүйеде қосымша құрамдас бөліктерді орнатпай J2ME, C++ мобильді платформаларына және PHP платформасына арнап бағдарламалау мүмкіндігі бар.

Eclipse IDE Java тілінде жазылған, ол Windows, OS X, Linux, Solaris OS және өзге JVM бар платформаларда істеледі.

Eclipse IDE мүмкіндіктері. Eclipse IDE 6.1 нұсқасы UML, SOA, мобильді телефондарға қосымшалар жасау үшін арналғын J2ME құралдарын, Ruby бағдарламалау тілін (Ruby on Rails – ты қоса) қолдайды. 6.5 нұсқасында PHP тілін қолдау қосылған. Және де тестілеуді қолдау үшін Python модулі қосылған.

Eclipse IDE бағдарламашыларға ортаның мүмкіндіктерін кеңейтуге мүмкіндік беретін плагиндерді қолдайды. Танымал плагиндердің бірі – iReport (JasperReports кітапханасына негізделген) қуатты есептер дизайнері.

Eclipse IDE Microsoft Windows, Linux, FreeBSD, Mac OS X, OpenSolaris және Solaris (SPARC үшін де, сол сияқты x86 – Intel және AMD үшін де) платформалары үшін дайын дистрибутивтер түрінде таратылады. Барлық қалған платформал үшін Eclipse IDE–ты түпнұсқалардан компиляция жасау арқылы алу мүмкіндігі бар.

Eclipse IDE 6.7 нұсқасында Project Kenai, Groovy тілін және Grails веб–фреймворкін қолдау қосылған. 6.8 нұсқасында Symfony PHP–фреймворкін қолдау, ал 6.9 нұсқасында Zend Framework–ты қолдау мүмкіндігі бар.

Eclipse IDE 7 нұсқасында өндіруші Ruby және Ruby on Rails тілдерін қолдаудан бас тарытты (Ruby and Rails құрамдас бөліктерін дамытуды бірлестік өз қолына алды). Қорыта келгенде, соңғы Eclipse IDE нұсқалары «қораптан» Java (Java FX, Java ME, Java EE–ды қоса), C/C++, Groovy, PHP, HTML, JavaScript, CSS–ты ғана қолдайды және де Eclipse IDE 7 нұсқасында UML–моделдеу құралдары қарастырылмаған.

Eclipse IDE тарихы. Eclipse IDE ортасын жасау 1996 жылы Xelfi атауымен Прага Карлов университеттің математика және физика факультінің басшылығымен студенттердің Java IDE жасау жобасы ретінде басталды. 1997 жылы Станек Роман жоба негізінде компания құрып, 1999 жылы Eclipse IDE IDE құқықтарын Sun Microsystems–ке беруге дейін ортаның коммерциялық нұсқаларын шығарумен айналысты. Sun келесі жылдың маусымында Eclipse IDE ортасының бастапқы кодтарын жариялады.

Соңғы нұсқалар. Алдыңғы 5.5.1 нұсқа негізде жасалған Eclipse IDE 6.0 нұсқасы IDE модульдарын және Eclipse IDE платформасына негізделген интернет қосымшаларды жасауды, интерфейстердің жаңа дизайнерін («Matisse жобасы» атауымен белгілі), CVS нұсқаларды басқару жүйесін қолдаудың жаңа нұсқасын, Weblogic 9 және JBoss 4 серверлерін қолдау және жетілдірілген редакторын пайдалануға мүмкіндік береді. Eclipse IDE 6.0 Ubuntu 8.04 және Debian дистрибутивтерінің құрамына қосылған.

2008 жылды қараша айында шығарылған Eclipse IDE 6.5 (Java Persistence, EJB 3 және JAX–WS–ты қолдауды қоса алғанда) Java EE мүмкіндіктерін кеңейтеді. Бұған қосымша ретінде, Eclipse IDE Enterprise Pack SOA визуалды құралдарын, XML схемамен жұмыс құралдарын, веб–сервиспен жұмыс құралдыран (BPEL үшін), UML тілінде моделдеу мүмкіндіктерін қосады. Eclipse IDE IDE Bundle for C/C++ жиыны C/C++ тілдеріндегі жобаларды қолдайды.

Eclipse IDE 7.2. нұсқасында қолданылатын тілдер тізімінен Ruby алып тасталынды.

1.3 Java Eclipse IDE платформасында мобильді құрылғыларға қосымшалар құру мүмкіндіктері

Java ME, Java Card мобильді және кірістірілген қосымшаларын жасауды қолдау. Eclipse IDE платформасында келесі технологияларға толық қолдау қарастырылған:

- MIDP 1.0, 2.0 және 2.1 профилі;
- CLDC 1.0 және 1.1 конфигурациясы;
- CDC конфигурациясы;
- Java ME SDK 3.0.5 пакетімен бірге жеткізіледі;
- Java-карталары;
- Apache Ant 1.8 сценарийі;
- мәліметтерді оқшаулауды және жалғастыруды қолдайтын GUI Visual Mobile Designer көзбен шолу жобалаушысы;
- мобильді қосымшаларға арналған экран жобалаушысы;
- API MIDP 2.0 Game интерфейсіне арналған Mobile Game Builder;
- SVG (JSR 226) графикалық элементтерін қолдаушы пайдаланушы интерфейсінің SVG құрамдастары бар SVG Composer құралы, SVG сипат редакторы;
- құрамдастар палитрасы, пайдаланушылардың құрамдастарын жасаушы;
- JUnit 1.1.0 тестілеу;
- мидлет сертификаттары мен қолтаңбаларын басқару;
- Over-the-Air (OTA) біріктірілген эмуляциясы;
- Push-registry эмуляциясы;
- SMS және CBS хабарламалары үшін WMA эмуляциясының функциялары;
- хабарламаларды сымсыз жіберуге арналған API интерфейстері және API мультимедиялық интерфейстері;
- ProGuard 4.2 көмегімен кодты шатастыру;
- бірнеше жобалары бар ортаны күйге келтіру;
- құрылғыда тестілеу және ретке келтіру;
- Java ME (JSR 172) мобильді веб-қызметі;
- ОС Mac арналған SDK MpowerPlayer платформасын қолдау (орнату).

Мобильді қосымшалардың көбісінде Java Micro Edition (Java ME) платформасы қолданылады. Бұл платформа мобильді телефон сияқты шағын құрылғыларға арналған, бірақ алуан түрлі құрылғыларда қолданылып келеді. Java ME платформасында Java SE құрамдастарының, виртуалды машиналардың және API интерфейстерінің қысқартылған ішкі көпмүшелері қолданылады. Оған қоса платформада мақсатты түрде мобильді және кіріктірілме құрылғыларға арналған API интерфейстері анықталған. Eclipse

IDE Java ME–мен бірге Java ME платформасының екі базалық күйін және екі қосымша мобильді мен кіріктіріме платформаларын қолдайды:

- CLDC (Connected Limited Device Configuration) нұсқасы CDC негізіндегі құрылғыларға қарағанда жадының көлемі кіші және есептеу қуаты аз құрылғыларға арналған. CLDC нұсқасында MIDP (Mobile Information Device Profile) бейіні негізделген; қазіргі таңда әлемде MIDP құрылғыларының саны екі миллиардтан астамға жетеді;

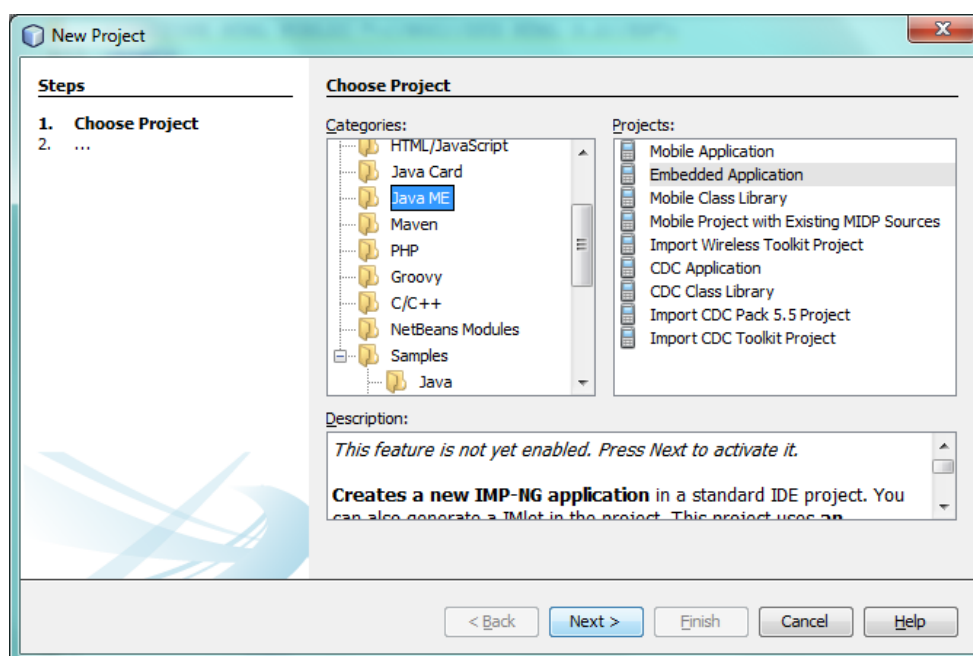
- CDC (Connected Device Configuration) нұсқасы жадының көлемі үлкен, есептеу қуаты айтарлықтай көп және желі ішінде байланысты қолдайтын құрылғыларға арналған. Мұндай құрылғыларға смартфондарды, телевизиялық компьютерлі қосымшаларды, кіріктіріме серверлер немесе құрылғыларды жатқызуға болады;

- Java Card платформасы Eclipse IDE ортасын қолдайды және смарт–карталар мен басқа шағын құрылғыларға арналған қосымшаларды жасап шығаруға мүмкіндік береді;

- Mobile Information Device Profile (MIDP) 1.0, 2.0, 2.1 (MSA), Connected Limited Device Configuration (CLDC) 1.0 және 1.1, оған қоса Connected Device Configuration (CDC) арналған қосымшаларды жасаңыз, тестілеңіз және ретке келтіріңіз;

- IDE CLDC және CDC әзірленімдерді қолдайды Java ME SDK жиынтығымен бірге әкелінеді. «Қызмет көрсету» мәзіріндегі басқа да мобильді және кіріктірімелі платформаларды тіркеугі болады.

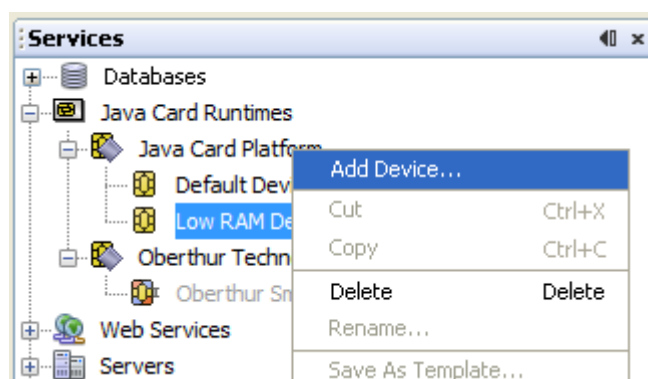
1.1 суретте Java ME жобасын құру көрсетілген.



Сурет 1.1 – Java ME жобасын құру

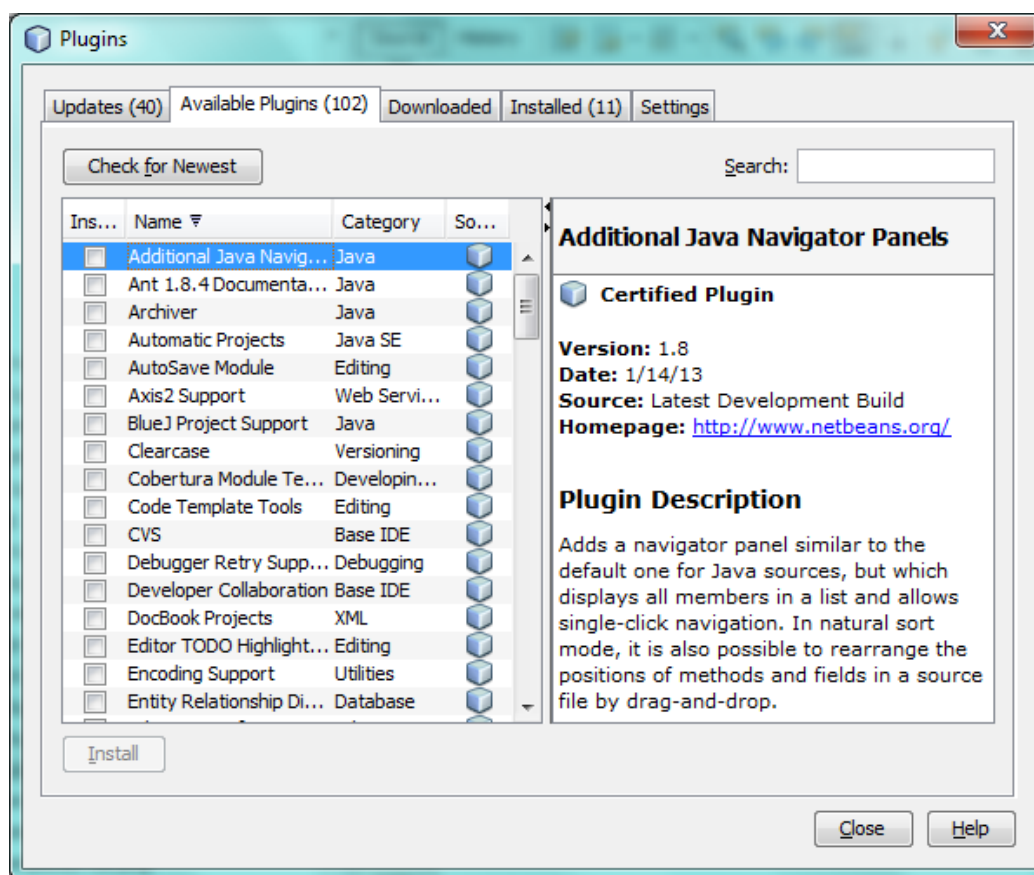
Java Card. Java Card Platform – смарт–карталарда қосылатын Java виртуалды машинасы. Платформа таныс API Java интерфейстарын, мысалы,

сервлеттерді қолдайды да, алып жүруге ыңғайлы ықшам құрылғыларда орындалатын веб-қосымшаларды жасауға мүмкіндік береді. IDE Java Card жобаларын, Java Card апплеттерінің ашылуы мен Java Card құрылғылары 1.2 суретте консольдарының күшеюін қолдайды.



Сурет 1.2 – Java Card құрылғылары консолі

Visual Mobile Designer Visual Mobile Designer көмегімен қолданушылардың сызбалық интерфейстерін жылдам жасау. Күту экраны, жүйеге ену экраны, файлдарды шолушы, SMS-хабарламаларды құрастыру құралы және экран-сурет сияқты орын ауыстырылатын құрамдастар қолдау табады. Талдау құралы идентификация және қолданбайтын құрамдастарды жою есебінен файлдың өлшемін азайтуға мүмкіндік береді, оған қоса MIDP 1.0 платформасының сәйкестігін тексеруге мүмкіндік береді. VMD қолданушылардың сызбалық интерфейстерін шоғырлауды жеңілдетеді. Анығырақ 1.3 суретте қолжетімді плагиндер көрсетілген.

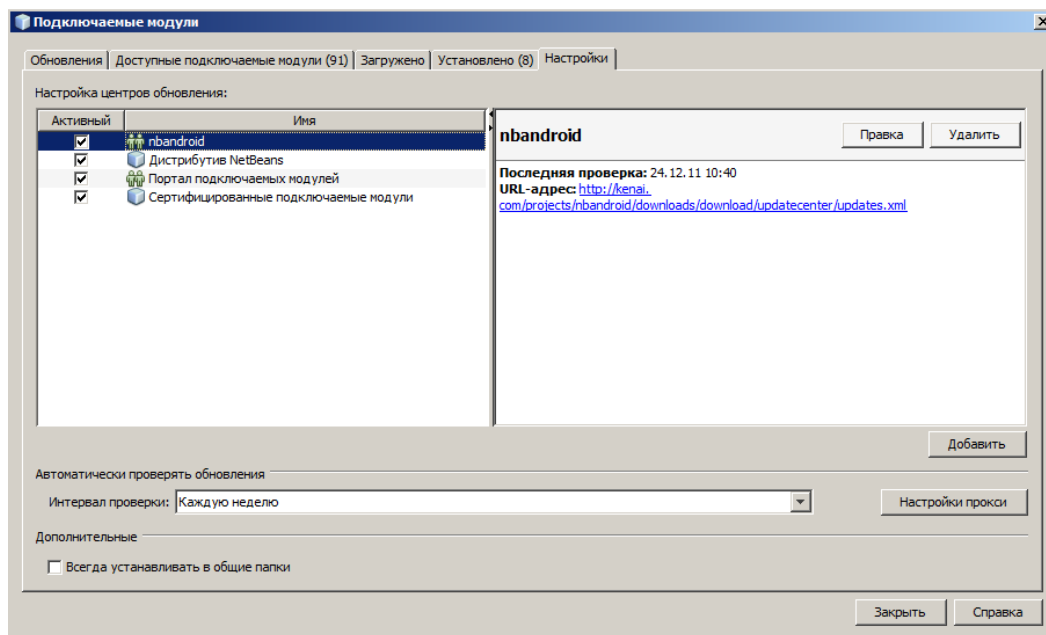


Сурет 1.3 – Қолжетімді плагиндер

Android платформасы үшін мобильді қосымшаларды жасауды қолдау. Eclipse IDE IDE Android плагинін орнату үшін Eclipse IDE’те «Қосылатын модульдер» терезесін («Қызмет көрсету» → «Қосылатын модульдер») ашып, «Күйге келтіру» қосымша бетіне өтеміз. «Қосу» тетігін басамыз. Жаңадан шыққан «Жаңару орталығын күйге келтіру» терезесінде «nbandroid» атын (плагин жобасы осылай аталады) және URL–мекен–жайын «<http://kenai.com/projects/nbandroid/downloads/download/updatecenter/updates.xml>» жазамыз. «ОК» тетігін басып, «Қол жетімді қосылатын модульдер» қосымша бетіне өтеміз, керек болған жағдайда «Каталогты қайта қосу» тетігін басамыз да, одан кейін іздеу жолында «Android» сөзін енгіземіз. Қол жетімді модульдер аумағында төмендегілер шығуы керек:

- Android;
- Android Test Runner for Eclipse IDE 6.9.1;
- Android Test Runner for Eclipse IDE 6.9;
- Android Test Runner for Eclipse IDE 7.0+.

Төменде 1.4 суретте Android платформасының қолжетімді модульдері көрсетілген.



Сурет 1.4 – Қолжетімді модульдер

Өзіңіздің Eclipse IDE нұсқасына сәйкес келетін «Android Test Runner» модулімен бірге «Android» модулін белгілеп «Орнату» тетігін басу керек, одан кейін Eclipse IDE'ті қайта қосу керек.

Осы амалдарды орындағаннан кейін «Android Project» және «Android Test Project» екі жобасы қол жетімді болады.

1.4 Android платформасы туралы

Android жүйесі архитектурасының ерекшелігі. Android қосымша жасаудың алдында, жүйенің архитектурасымен және оның негізгі ерекшеліктерімен танысу қажет.

Android жүйесі – операциялық жүйеден, аралық қабаттың бағдарламалық қамтамасыз етуінен (middleware), сонымен бірге негізгі қолданбалы қосымшалардан құралған мобильді құрылғыларға арналған программалық стек.

Android архитектурасын төрт деңгейге бөлу қабылданған:

- ядро деңгейі;
- орындалу ортасының кітапханасы деңгейі;
- қосымшалар қаркасының деңгейі;
- қосымшалар деңгейі.

Суретте операциялық жүйенің негізгі құрамдас бөліктері және олардың өзара әрекеттесуі көрсетілген.

Android ядросы Linux операциялық жүйенің 2.6–шы нұсқасына негізделген, бірақ Android жүйесі таза күйіндегі Linux–жүйе емес, ол бірқатар ерекшеліктер мен өз жады бөлу механизмдері, процестер арасындағы қатынас және т.б. Android жүйесіне тән ядроның қосымша кеңейтілмелеріне ие.

Ядро жабдық пен программалық стектің қалған бөлігі аралысындағы абстракция қабаты болып табылады. Бұл деңгейде процестерді басқару, жадты үлестіру және файл жүйесін басқару тәрізді қызметтер орналасады.

Ядро деңгейінің негізгі құрамдас бөліктері:

- процессораралық өзара әрекеттесу драйвері (IPC Driver);
- қоректенуді басқару драйвері (Android Power Management);
- мобильді құрылғыны құрамына кіретін жабдықты басқару драйверлеі.

Ядро деңгейінен «жоғары», аралық қабат бағдарламалық қамтамасыз етуі ретінде қосымшалар үшін ең маңызды негізгі функционалды қамтамасыз етуі үшін кітапханалар жиынтығы (Libraries) орналасқан. Яғни осы деңгей жоғары жатқан деңгейлерге жүзеге асырылған алгоритмдерді беру, файлдық форматтарды қолдау, ақпаратты кодтау және кері кодтауды жүзеге асыру (мысалы, мультимедиа кодектері), сурет салу және т.б. үшін жауап береді. Кітапхана C/C++ тілінде жазылған және нақты аппараттық қамтамасыз етуге арналып компиляция жасаған, сондықтан алдын ала орнатылған күйде жеткізіледі.

Кейбірін қарастырып өтейік:

- Surface Manager – Android жүйесіндегі терезелердің композитті менеджері Linux-тың Compoz менеджеріне ұқсас, бірақ ықшамдалған. Бұл жүйеге терезелердің мөлдірлігі және бір қалыпты өту әсерлерін жасауға мүмкіндік береді;
- Media Framework – PacketVideo OpenCORE негізінде жүзеге асырылған кітапхана. Олардың көмегімен жүйе аудио және видео контентті жазу және ойнатуды, сонымен қатар статикалық бейнелерді көрсетуді жүзеге асыра алады; Көптеген кең таралған форматтарды қолдайды: MPEG4, H.264, MP3, AAC, AMR, JPG және PNG;
- SQLite – Android жүйесінде деректер қорымен жұмыс жасауға арналған жеңіл және өнімді реляциялық ДҚБЖ;
- FreeType – бит карталарымен жұмыс жасауға, сонымен бірге шрифттардың растеризациясы және олармен операцияларды жүзеге асыруға арналған кітапхана. Шрифттарды және мәтінді бейнелеу үшін арналған жоғары сапалы қозғағыш;
- LibWebCore – Google Chrome және Apple Safari браузерлері пайдаланылатын белгілі WebKit браузер қозғағышының кітапханалары;
- SSL – аттас криптографиялық хаттаманы қолдау үшін арналған кітапхана. Libc – C тілінің стандартты кітапханасы, атап айтқанда оның Linux негізіндегі құрылғыларда жұмыс жасауға бағыттылған BSD нұсқасы. Bionic атауына ие. Бұл деңгейде Android Runtime орындалу ортасы орналасқан. Оны маңызды құраушылары: ядро кітапханалар жиынтығы және Dalvik виртуалды машинасы.

Әр Android қосымшасы өз Dalvik виртуалды машинасында іске қосылады. Осының арқасында барлық жұмыс үстіндегі процестер операциялық жүйеден және бір-бірінен оқшауланған. Жалпы Android Runtime архитектурасы

программаға виртуалдық машина ортасы шеңберінде ғана жұмыс жасауға мүмкіндік береді. Осылай операциялық жүйе ядросын оның басқа бөліктерінен келуі мүмкін зияннан қорғаныс жүзеге асырылады. Сондықтан қате код немесе зиянды программа Android–ты және оның базасындағы құрылғыларды бүлдіре алмайды. Мұндай қорғаныс Android Runtime–ның маңызды функциясы болып табылады.

Бұл деңгейден жоғары кейде қосымшалар қаркасы деп аталатын Application Framework деңгейі орналасады. Қосымшалар қаркасы деңгейі арқылы өндірушілер төмен деңгейлерде жататын жүйенің құрамдас бөліктері беретін API–ға рұқсат алады. Одан басқа, фреймворк архитектурасының арқасында кез–келген қосымша басқа қосымшалардың рұқсат етілген жүзеге асырылған мүмкіндіктеріне қол жеткізе алады. Әр қосымшаның негізі және фреймворк бөлігі болып табылатын сервистер мен жүйелердің негізгі жиынына кіреді:

- қосымшаларының визуалды компоненттерін(мысалы, тізім, мәтіндік алаң, кестелер, батырма немесе тіпті кіріктіріме web–браузер) жасау үшін қолданылатын көріністердің (Views) бай және кеңейтілетін тізімі;
- бір қосымшалар басқа қосымшалар өз жұмысына пайдалану үшін ашқан мәліметтерді басқарушы контент – провайдерлер (Content Providers);
- функционалсыз ресурстарға (жол мәліметтерге, кесте, файл және т.б.) қол жетімділікті қамтамасыз ететін ресурс менеджер (Resource Manager);
- әрбір қосымшаға өз мәлімдемелерін қалып–күй жолында бейнелеуге мүмкіндік беретін хабарламалар менеджері (Notification Manager);
- қосымшалардың өмірлік циклын басқаратын, әрекеттермен жұмысы туралы мәліметтерді сақтайтын, сонымен бірге әрекеттер арасында навигация жүйесін қамтамасыз ететін әрекеттер менеджері (Activity Manager);
- қосымшаларға құрылғының ағымдағы географиялық орын туралы жаңартылған мәліметтерді мерзімді алуға мүмкіндік беретін орналасу менеджері (Location Manager).

Қорыта келгенде, Application Framework арқасында Android қосымшасы қосалқы функционалды қолдану мүмкіндігіне ие. Яғни қосымшалар және операциялық жүйе құрамдас бөліктерін бірнеше рет пайдалану қағидасы жүзеге асырылған.

Android программалық стегінің жоғарғы қабатында қосымшалар деңгейі (Applications) орналасқан. мұнда Android ОЖ–не алдын ала орнатылған қосымшалардың негізгі жиыны жатады. Мысалы, оған SMS жіберу қосымшасы, карталар, күнтізбе, браузер, пошталық клиент және т.б. бағдарламалар кіреді. Интегралданған қосымшалардың тізімі құрылғы түріне және Android нұсқасына байланысты өзгеруі мүмкін. Және бұл қосымшалар деңгейіне негізгі жиыннан басқа Android қосымшаларының бәрі жатады.

1.5 Android SDK құралы туралы

Мобильді құрылғыларға арналған Android операциялық жүйесінің алғашқы шығарылымынан біраз уақыттан кейін, артынша танымалдылығының айтарлықтай артуынан кейін жасаушыларға арналған құрал-жабдықтар шығарылды.

Android SDK – Android операциялық жүйесі үшін қосымшаларды жасау ортасы. Бағдарлама мобильді құрылғы камерасын, акселерометр, компас, GPS мәліметтерді, Bluetooth, Wi-Fi, EDGE және 3G бойынша рұқсатты қолданатын Android-қосымшаларын жасау мен тестілеуге мүмкіндік береді. Оған қоса мультимедиялық контентпен (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG және GIF форматтарындағы аудио, бейнелер, суреттер), SQLite мәліметтер базасымен, WebKit қозғалтқышындағы біріктірілген браузермен, Dalvik виртуалды машинасымен, GSM телефониямен және тағы басқаларымен жұмыс жасауды қолдайды. Оған қоса Android SDK қолданушылары қосымша орнатылған эмулятор көмегімен өздері жасап шығарған қосымшаларды тестілей алады.

Linux'ке негізделген Android SDK ортасы қосымшаларды іске қосу үшін виртуалды құрылғыны қолданады және 3G, WiFi, GPS, сенсорлы экран сияқты және заманауи мобильді телефондарда бар басқа да стандартты функцияларды қолдайды.

1.6 Java бағдарламалау тілі туралы

Java тілі. Java тілі бұл Internet желісінде жұмыс істейтін объектілі-бағытталған, платформалы – тәуелсіз, желі ішінде жұмыс істейтін тармақталған қосымшалардың өңдеуіне қолданылатын программалау тілі. Java жобасы 1995 жылы Sun Microsystems бірлестігінде басталды. Java бағдарламалау жүйесі кішкене интерактивті қолданбалы программалар – апплеттерді таратуға интернетті қолдануға мүмкіндік береді. Олар Internet серверлерінде орналасып, таратылады, клиентке желі арқылы тасымалданады, автоматты түрде орнатылады және www құжатының бөлігі ретінде орындалады. Апплетке клиент компьютері қорларына қолжетімділігіне шек қойылған, сондықтан да ол жанама мультимедиялық интерфейсін бере алады және дискіде сақталған мәліметтерге ешқандай ақау келтірместен күрделі есептеулерді жүргізе алады. Бағдарламалардың басқа түрі ретінде Java қосымшаларын қарастыруға болады. Олар кез келген компьютерде, тіпті оның архитектурасына қарамастан орындалатын, тасымалдана алатын бағдарламалар болып табылады. Генерацияланатын байт код виртуалды Java – машинадағы (JVM – Java Virtual Machine) виртуалды код интерпретаторында орындалатын нұсқаулардың жиынтығы болып табылады. Клиенттерге сервердегі қосымшаларға және деректер базасына қолжетімділікке мүмкіндік беретін JSP (Java Server Pages) және сервлеттер өте кең ауқымда таралуға мүмкіндік алды.

Java тілі C++ тілінің синтаксисын қолданады, бірақ объектілік үлгі Smalltalk тілінен алынған. Осыдан Java тілінің C++ тілімен ұқсастықтары тек қана сыртқы түрде екенін көруге болады. Java көрсеткіштерді (C++, Pascal және

тағы да басқа тілдердің ең қауіпті құралы) қолдамайды, себебі, жадтың жанама адрестерімен типі көрсетілмеген көрсеткіштер арқылы жұмыс істеу мүмкіндігі жадтың қорғанышын елемеуге рұқсат береді. Java тілінде айнымалы арифметикалық есептеудің тәсілдері өзгерген, сондықтан да платформа түрлерінің арасында аралық код үйлесімділігін қамтамасыз ету үшін `strictfp` кілттік сөзі енгізілді. Ол компиляторға айнымалы үтірі бар сандар үшін арифметикалық әрекеттерді кез келген платформада бірдей орындау керек екендігін көрсетеді.

Тіл кластарының жүйелік кітапханасы кластар және пакеттерден тұрады, олар тілдің әртүрлі базалық мүмкіншіліктерін жүзеге асырады. Бұл кітапханаларға қосылған кластардың әдістері JVM–нан Java – бағдарламаның интерпретациясы кезінде шақырылады. Java–да бағдарламаның барлық объектілері динамикалық жадта орналасқан (`heap`) және стектерде сақталынатын объекті сілтемелер арқылы қолжетімді. Бұл шешім жадқа тікелей қолжетімсіздікке мүмкіндік берді, бірақ массив элементтерімен жұмыс істеуді қиындатып жіберді. Java тіліндегі объектілі сілтемелер өздері бағыттталып тұрған объектілердің класы туралы хабардан тұрады. Сондықтан да объектілі сілтемелер дегеніміз, көрсеткіштер емес, олар объектілердің дескрипторлары. Дескрипторлардың болуы JVM–ге код интерпретациясы фазасында типтердің сәйкес келуін тексеруге мүмкіндік береді. Java–да жадты динамикалық бөлу концепциясы да қайта қарастырылған: динамикалық бөлінген жадты босату тәсілдері жоқ. Оның орнына `new` операторының көмегі арқылы көрсетілген жадты автоматты түрде босату жүйесі(қоқыс жинаушысы) іске асырылған.

Java – бағдарламаларда класс спецификациясы мен оның жүзеге асырылуы әрқашан да тек қана бір файлда болады. Java–да көптік мұрагерлік жоқ, тек конструктор бар, бірақ деструкторлар жоқ (қоқысты автоматты түрде жинастыру қолданылады), тілдің резервтелген сөздері бола тұрса да, `goto` операторы және `const` сөзі қолданылмайды.

Java тілінде пайда болған маңызды мүмкіндіктер – интерфейстер мен көп ағымдылық (бағдарлама бөлімдерінің бір уақытта орындалу мүмкіндігі).

Лексикалық негіздер. Түсіндірменің программаның орындалуына ешқандай нәтижені әкелмесе де, оны қолдану бағдарламашыға жеңілдік туғызады. Бір жолды түсіндірме `//` түрінде ашылады.

Резервтелген кілтті сөздер. Резервтелген кілтті сөздер ол арнаулы сөздер индентификаторлар, олар java тілінде кірістірілген типтерді, модификаторларды және программаны басқару құралдарды идентификациялау үшін керек. Қазіргі кезде java тілінде 59 резервтелген сөздер бар. Оларды тек қана мақсатына қарай қолдануға болады сіз резервтелген сөздер тізімін 1.1 кесетеді көре аласыз. Ал айнымалылар, класстар немесе әдістер индентификатор ретінде қолдануға болмайды.

К е с т е 1.1 – Java тілінде резервтелінген сөздер

Резерв сөздер

abstract	boolean	break	byte	byvalue
case	cast	catch	char	class
const	continue	default	do	double
else	extends	false	final	finally
float	for	future	generic	goto
if	implements	import	inner	instanceof
int	interface	long	native	new
null	operator	outer	package	private
protected	public	rest	return	short
static	super	switch	synchronized	this
throw	throws	transient	true	try
var	void	volatile	while	

Айталық byvalue, cast, const, future, generic, goto, inner, operator, outer, rest, var осы сөздер Java–да резервтелінген, бірақ олар қолданылмайды. Java–да әдістердің резервтелінген аттары бар. Оларды 1.2–кестеде көре аласыз.

К е с т е 1.2 – Java–да әдістердің резервтелінген аттары

Резерв аттары				
clone	equals	finalize	getClass	hashCode
notify	notifyAll	toString	wait	

Идентификаторлар. Идентификаторлар класстардың, әдістердің және айнымалыларды атауға үшін қолданылады. Идентификатордың қандайда бір әріптер не сандардың, символдар(\$,_) реттері бола алады. Java тілі регистрге өте тәуелді болады.

Литералдар. Java–да константалар литералды түрде беріледі. Бүтін сандар, жылжымалы нүкте сандары, логикалық нәтижелер, символдарды және жолдарды әр бір жерде орналастыруға болады.

Бүтін литералдар. Бүтін сандар ол бүтін литералдар. Осы тип программаларда көбінесе қолданылады. Әр бір сан –1,2,3,4,15 ол бүтін литерал. Ондық литералдан басқа компьютерде екілік, сегіздік, оналтылық сандар қолданылады.

Жылжымалы нүтелі литералдар. Жылжымалы нүктелі литералдар олар саның нақты түрін айтады.

Логикалық литералдар. Логикалық литералда тек қана екі мәні болады – true (ақиқат) және false (жалған). True және false шығармалар ешқандай сандық нәтижеге ауыспайды. Java да true және false сөздері ешқанда 0 және 1 болмайды.

Символдық литералдар. Java да символдар UNICODE кестесіндегі индекстер. Олар 16 битті мәнді береді, оларды бүтін санға ауыстыруға болады. 1.3–кестеде көрсетілген.

К е с т е 1.3 – Басқару символдар тізбегі

Басқару тізбегі	Сипаттама
\\ddd	Сегіздік символ (ddd)
\\uxxxxx	Оналтылық символ символ UNICODE (xxxxx)
\\'	Апостроф
\\\"	Тырнақша
\\\\	Кері қиғаш сызықша
\\r	Карретканың қайтаруы (carriage return)
\\n	Жолды түсіру (line feed, new line)
\\f	Бетті түсіру (form feed)
\\t	Горизонтальді табуляция (tab)

Оператор – ол қандай да бір әрекетті екі немесе бір аргументтерге орындап нәтижені шығарады. Синтаксикалық тұрғыда операторлар идентификатор және литерал арасында орналасады. 1.4–кестеде операторлар түрлері көрсетілген.

К е с т е 1.4 – Java тілінің операторлары

Операторлар			
+	+=	–	–=
*	*=	/	/=
	=	^	^=
&	&=	%	%=
>	>=	<	<=
!	!=	++	—

К е с т е 1.4 соңы

Операторлар			
==	=	~	?:
	instanceof	[]	

Бөлгіштер. Тек қана бірнеше символдардың тобы Java–программада қолданылады және олар аталған жоқ. Олар программаның сыртқы бетіне, функционалдығына қарапайым бөлгіштер жауап береді. 1.5 кестеде бөлгіштерді көре аласыз.

К е с т е 1.5 – Java тіліндегі бөлгіштер

Символдар	Аты	Не үшін қажет
()	Дөңгелек жақша	Жариялауда және әдісте параметрлердің тізімін белгілейді. Сонымен тапсырманың операция шығарылуын приоритетін есепте белгілейді.
{ }	Фигурлық жақша	Массивтің инициализациялау кезінде оның мәнін бергенде және класста әдісте блокты шектеу.
[]	Квадратты жақша	Массивті жасау кезінде және оның жеке элементтерін пайдалану кезінде қолданылады.
,	үтір	Айнымалыларды жариялау кезінде идентификаторларды бөледі, for циклда операторларды бөлу кезінде қолданылады.

Айнымалылар. Айнымалы бұл java программада ақпаратты сақтау негізгі элемент болып саналады. Айнымалы идентификатордың комбинациямен, типімен және орындалу аймағымен сипатталады.

Айнымалыларды сипаттау. Айнымалыларды сипаттау формасы осындай:

- идентификатор типі[=мәні] [, идентификатор [= мәні 7...];
- тип – бұл кірістірілген типтердің бірі byte, short, int, long, char, float, double, boolean, немесе класс интерфейс аты.

Айнымалылар жазу үлгілері 1.6 кестеде көрсетілген.

К е с т е 1.6 – Айнымалыларды сипаттау.

Мәні	Айнымалылар
Int a, b, c;	a, b, c үш бүтін айнымалыны сипаттаймыз.
Int d = 3, e, f = 5;	Тағыда үш айнымалыны сипаттаймыз d және f мәнін инициализациялаймыз..
byte z = 22;	Z инициализациялау.

Келтірілген мысалда үш айнымалы жасалынады. Тік бұрышты үшбұрышқа сай, содан кейін Пифагор теоремасы бойынша гипотенузаның ұзындығы табылады ол 5 ке тең:

```
class Variables {
public static void main (String args []) {
double a = 3;
```

```
double b = 4;
double c;
c = Math.sqrt (a * a + b * b);
System.out.println ("c = " + c);
} }
```

Қарапайым типтер. Қарапайым типтер java тілінде объектілі–бағытталған болып анықталмайды. Олар алғашқы тілдердегі типтеріне ұқсайды. Java–да сегіз қарапайым типтер бар:– byte, short, int, long, char, float, double және boolean. Оларды төрт бөлікке бөлуге болады:

- оларға byte, short, int и long типтер жатады. Осы типтер бүтін белгі сандар үшін қолданылады;
- жылжымалы нүктемен типтер – float и double. Олар бөлшегі бар сан үшін қолданылады;
- char символдық тип. Символдар таблицадан элементтер үшін қолданылады, мысалы әріптер мен сандар;
- логикалық boolean типі. Бұл арнайы тип, логикалық өлшемдер үшін. Java тілінде кейбір тілдерден айырмашылығы автоматты типтердің бейімделенуі жоқ болады. Типтердің сәкес келмеуі программада ескертуге емес қатені туғызыды. Әр бір тип үшін мәндердің мүмкін жиыны және операциялардың рұқсат түрлері болады.

Бүтін сандар byte – бұл 8–битті тип. Оның диапазоны – 128 ден 127 – дейін byte b.

Егер биттермен манипуляция емес, онда осы типті қолдануға шектеу керек, оның орнына int типін қолдану артық болады.

Short – 16–битті тип. Оның диапазоны – 32768 ден 32767 – дейін. Java–да ең аз қолданыстағы тип оның басты байты бірінші тұрады.

Int типі 32–битті бүтін сандарды ұсыну үшін қолданылады. Оның диапазоны 2147483648–дан 2147483647 дейін. Көбінесе ол жай бүтін екі миллиардқа дейін типтер үшін қолданылады. Бұл тип массивтер мен счетчигер үшін қолдану аясында жақсы келеді.

Long типі 64–битті сандар ұсыну үшін қолданылады. Оның диапазоны кеңістіктегі барлық атомдарды санап шығуға болады.

Астыда 1.7 кестеде бүтін типтердің дәрежелілігі және олардың диапазоны берілді.

К е с т е 1.7 – Бүтін типтердің диапазоны

Аты	Дәрежелілігі	Диапазоны
long	64	–9, 223, 372, 036, 854, 775, 808.. 9, 223, 372, 036, 854, 775, 807
Int	32	–2, 147, 483, 648.. 2, 147, 483, 647
Short	16	–32, 768.. 32, 767

byte	8	-128.. 127
------	---	------------

Жылжымалы нүктесі бар сандар. Жылжымалы нүктелі сандарды жиі нақты сандар айтады. Оны бөлшек бар амалдарды шешу үшін қолданылады. Java-да келесі нақты типтер – float және double. Оларды 1.8 кестеде көре аласыз.

К е с т е 1.8 – Нақты тип

Аты	Дәрежелік	Диапазоны
double	64	1. 7e-308.. 1. 7e+ 308
float	32	3. 4e-038.. 3. 4e+ 038

Егер сізге қарапайым немесе біртекті дәлдікпен айнымалы керек болса онда float типті қолдану керек. Ол 32 битті орын алады.

Double. Екілік дәлдік қажет етілсе біз double кілтті сөзді қолданамыз. Ол 64-битті болады. Трансценденттік sin, cos, sqrt, математикалық функциялар double типінде нәтижені қайтарады.

Типті келтіру (type casting) – бұл C++ нашар қасиеттің бірі, сонымен қатар типті келтіру Java тілінде сақталған. Кейде бізде бір кез келген типіміз бар, бірақта оны бір басқа типке келтіру керек болады. Кейбір типтерге типті келтіру онсызда іске асыруға болады. Java-да типтердің келтіру автоматты түрде орындалады егер типтердің өлшемі сәйкес келсе. Осындай келтіру болады егер byte немесе short типті int типке келтірілсе. Оны кеңейтілу(widening) дейді немесе жоғарыту(promotion), кіші типтегі айнымалы үлкен типтегі айнымалыға ауысады. Егер айнымалыны үлкен типтен кіші типке ауыстырсақ онда кішіреу (narrowing) деп атаймыз:

```
int a = 100;
byte b = (byte) a;
```

Типтердің автоматты келтіруі. Қашан сіз мәнің нәтижесін шығарғанда аралық мәндердің дәлдігі ақырғы мәннен жоғары болу керек жағдайлар да туады:

```
byte a = 40;
byte b = 50;
byte c = 100;
int d = a * b / c;
```

Аралық шығарманың мәні (a * b) byte диапазонынан шығу да мүмкін. Сондықтан Java әр бір өрнектін бөлігінде мәнді int типіне келтіреді.

Типтердің автоматты түрде келтіруі кейде күтпеген жайтағы қателердің шығу хабарламалар шығуы да мүмкін. Мысалы бізбен келтірілген код бірінші көз қарасқа дұрыс, бірақ трансляция фазасында қатені туғызады. Біз 50*2 мәнді byte айнымалысына жазамыз, ол сол айнымалыға сияды. Бірақ int типіне автоматикалық келтіруден біз тронаслятордан қате тұғызған жағдайды

естіміз. Өйткені int типінен byte типіне келтірген жағдайда дәлдік жоғалу мүмкін:

```
byte b = 50;  
b = b * 2;
```

Incompatible type for =. Explicit cast needed to convert int to byte (үйлесімді емес тип =. Int типті byte типіне тура келтіру қажет).

Түзетілген мәтін:

```
byte b = 50;  
b = (byte) (b * 2);
```

Егер өрнекке byte, short және int типі жазылса, онда типтің шектен аспау үшін ол барлығы int-ке деін ұлғаяды. Егер өрнекте Long типті айнымалы болса онда нәтиже long типіне ауысады.

Java JDK платформасында бағдарламаларды құру. Java тілінде түпкі файл ол бір тексттік файл бір немесе бірнеше класс сипаталып жазылады. Java трансляторы программаның түпкі коды java кеңейтілуіндегі файлдарда сақталғаның жариялайды. Трансляциялау кезіндегі код әр бір классқа жеке шығу файлға жазылады, класстың атымен сәйкес келеді және class кеңейтілуінде болады.

Алдымен біз мысалға “Hello World” программаны жазайық. Содан кейін барлық негізгі лексикалық элементтерді қарастырайық. Оларға пробел, сілтеме, кілтті сөздер, идентификаторлар литералдар, операторлар жатады.

Сонымен java да бірінші бағдарламамыз:

```
class HelloWorld {  
public static void main (String args []) {  
System.out.println ("Hello World"); } }
```

Java тілі барлық программалық код аталған класстардың ішінде болғаның қажет етеді. Келтірілген мысал HelloWorld.java файлында жазылғаның қажет етеді. Кодты трансляциялау үшін Java – javac трансляторы қажет:

Транслятор HelloWorld.class процессордан әмбебап байт кодты файлын құрады. Файлды іске асыру үшін бізге java іске қостыру ортасын қостыру керек.

Осы жолда class резервтелген сөз қолданылған. Ол трансляторға жаңа классты сипаттайтымыз деп айтады. Класстың толығымен сипаттауымыз фигурлық жақшалар ішінде сипатталады.

Осындай бірінші көзқарасқа қиын жол java да құрастыру кезінде өте керек міндет болады. Өйткені java да глобальді функциялар жоқ болады. Осы жол көп кездесетіндіктен біз оны қарастырайық.

Public. Бұл – бағдарламашыға кез келген әдістің көрінетіндігімен басқаруға мүмкіндік берген және кез келген айнымалы рұқсаттың түрлендіргіші. Осы

мысалда `public` рұқсат модификаторы `main` әдісі әр бір классқа көрінеді және ашық болады.

`Static`. Келесі кілтті сөз ол – `static`. Осы класстың әдістері локальді не статикалық айнымалылармен ғана істей алады.

`Void`. Сізде әдістерді қолдану барысында нәтижені қандай да бір типте шығаруын қажет етеді. Мысалы `int` бүтін, `float` нақты. Біздің мысалда экранға жай ғана жолды шығару керек.

`Main`. `main` классыз `java` трансляторы программаны іске қоса алады, ал `java` интерпреторы онсыз программаны қоспайды.

Осы жолда `println` әдісі `out` объектісінде орындалады.

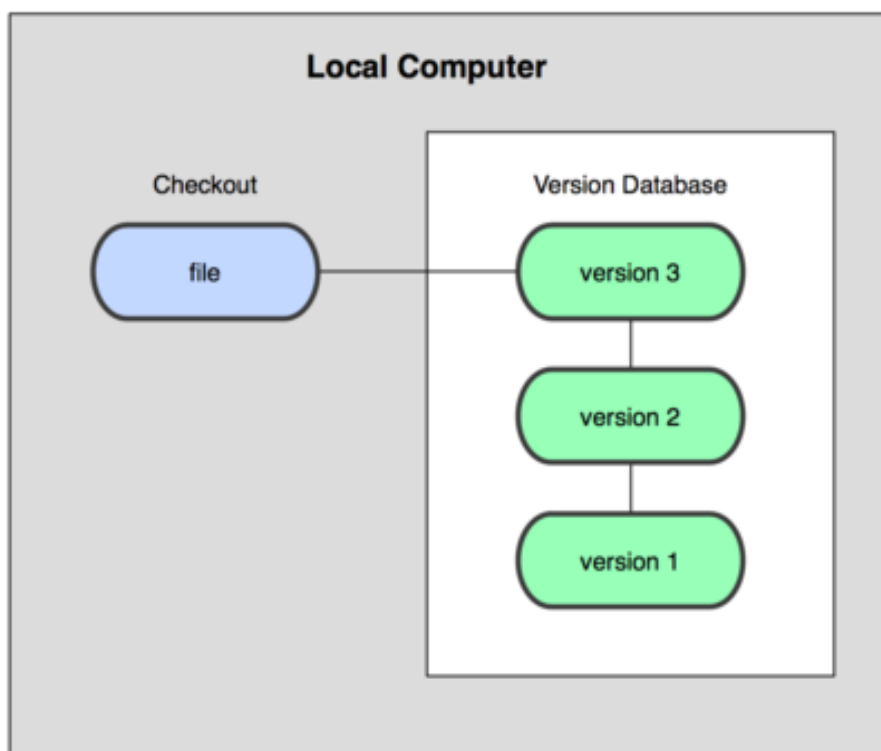
1.7 Git файл нұсқаларын басқарудың таратылған жүйесі туралы

Нұсқаларды бақылау жүйелері туралы. Нұсқаларды басқару жүйесі (ағылш. `Version Control System`, `VCS` немесе `Revision Control System`) – өзгермелі ақпаратпен жұмыс істеуді жеңілдетуге арналған бағдарлама. Ол бір немесе бірнеше файлдардағы өзгісті одан әрі бұл файлдардың нақтылы ескі нұсқаларына қайтып келуге болатындай етіп есепке алатын жүйе.

`VCS` жеке файлдарды бұрынғы түрге қайтаруға, барлық жобаны бұрынғы жағдайға қайтаруға, уақытпен болып жататын өзгеруді қарауға, кім соңғы өзгеріс жасағанын анықтауға мүмкіндік береді. Егер `VCS` пайдаланған жағдайда файлды бұлдіріп немесе жоғалтып алған жағдайда, бәрін қалпына келтіруге болады.

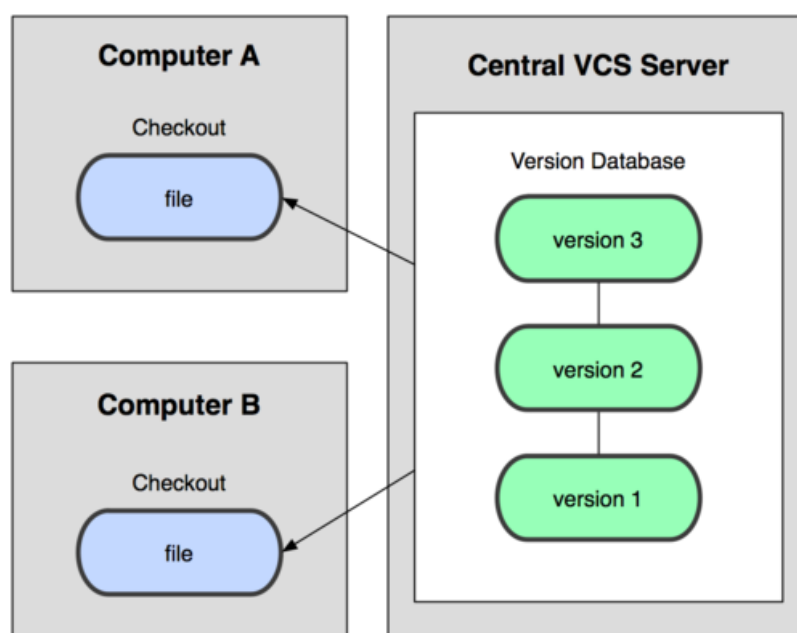
Нұсқаларды бақылаудың жергілікті жүйелері. Әдетте нұсқаларды бақылау файлдарды басқа директорияға көшіру арқылы ғана жүргізіледі (әдетте, директория атауына ағымдағы күнді қоса). Мұндай тәсілдеме кең таралған, бірақ ол өз қарапайымдылығынан қателіктерді жиі болдырады. Төменде 1.5 суретте жергілікті `VCS` сұлбасы берілген.

Нұсқаларды бақылаудың орталықтандырылған жүйелері. Келесі негізгі мәселе басқа компьютерлердегі бағдарламашылармен бірігіп жұмыс жасау қажеттілігі болды. Оны шешу үшін нұсқаларды бақылаудың орталықтандырылған жүйелері жасалды.



Сурет 1.5 – Жергілікті VCS сұлбасы

Мысалы, CVS, Subversion және Perforce сияқты бақылау жүйелерінде барлық файлдарды сақтайтын орталық сервер және одан файлдардың көшірмелерін алып отыратын клиенттер бар. Көптеген жылдар бойы бұл нұсқаларды бақылау жүйелері үшін стандарт болды. 1.6 суретте GitHub желісінің орталықтандырылған VCS сұлбасы көрсетілген.



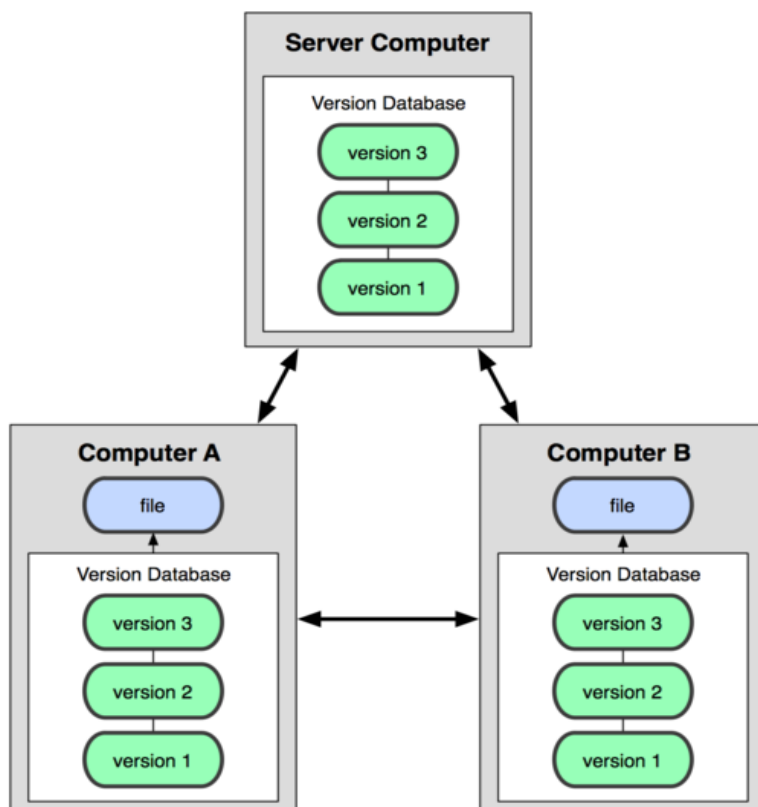
Сурет 1.6 – Орталықтандырылған VCS сұлбасы

Мұндай тәсілдеме көптеген артықшылықтарға ие, әсіресе жергілікті VCS ге қарағанда. Мысалға, жобанда кім қандай өзгеріс жасағаны белгілі. Администраторларда қолданушылардың рұқсат деңгейін шектеу мүмкіндігі бар және орталықтандырылған VCS ді басқару әр клиенттегі жергілікті VCS ді басқаруға қарағанда әлдеқайда оңай.

Дегенмен мұндай тәсілдемеде бірнеше кемшілік бар. Орталықтандырылған сервер барлық жүйенің осал жер болып табылады. Егер сервер бір сағатқа өшірілсе, онда бағдарламашылар осы сағаттың ішінде өзара әрекет ете алмайды, және ешкім өз жұмысын сақтай алмайды. Егер орталық сервер дискі бүлінсе және резерв көшірме жоқ болса, пайдаланушылардың жұмыс машинасында сақталған бірнеше жұмыс нұсқаларынан басқа жобаның барлық тарихы жоғалады. Нұсқаларды бақылаудың жергілікті жүйелері де осы кемшілікке ие: егер жобаның барлық тарихы бір жерде сақталса, сіз барлық мәліметті жоғалтуға тәуекел етесіз.

Нұсқаларды бақылаудың таратылған жүйелері. Бұл мәселені нұсқаларды бақылаудың таратылған жүйелерін шешеді. Git Mercurial, Vazaar немесе Darcs сияқты жүйелерде клиенттер файлдардың соңғы нұсқаларын ғана емес, барлық репозиторийларды толық көшіріп алады.

Сондықтан жұмыс сервері пайдасыз болған жағдайда, кез келген клиент репозиторийы деректерді қалпына келтіру үшін серверге керіле алады. Клиент файлдардың жаңа нұсқаларын алған кезде әрбір рет ол барлық деректердің толық көшірмесін құрады. 1.7 суретте кескіні көрсетілген.



Сурет 1.7 – Таратылған VCS сұлбасы

Git тарихы. Linux ядросы – үлкен ашық жоба. Linux ядросының (1991–2002) өзгерістерінің көптеген бөлігі патчтар және архивтелген файлдар түрінде тарайды. Жоба 2002 жылы BitKeeper нұсқаларды бақылаудың таратылған жүйесіне көшті.

2005 жылы Linux ядросы қоғамдастығы мен BitKeeper жүйесін жасаушы компания арасында қатынас бұзылды және өнімді тегін пайдалануын құқығы жойылды. Бұл Linux қоғамдастығын (және оны ойлап тапқан Линус Торвальдсты) өз нұсқаларды бақылау жүйесін жасауға итермеледі. Жаңа жүйеге қойылған негізгі талаптар келесі:

- жылдамдық;
- дизайн қарапайымдылығы;
- сызықты емес жобаны қолдау (мың қатарлас бұтақтар);
- толық таратылған жүйе;
- Linux ядросы сияқты үлкен жобалармен тиімді жұмыс жасау мүмкіндігі(жылдамдық және деректердің көлемі бойынша).

Git 2005 жылы жасалғаннан бері өзінің бастапқы қасиеттерін сақтай отырып, пайдалануда ыңғайлы әрі оңай бола түсті.

1.8 GitHub жобалар хостингы веб сервисі туралы

GitHub – жобалардың хостингы мен оларды бірігіп жасаудың ең үлкен веб–сервисі. Git нұсқаларының бақылау жүйесіне негізделген және бұрындары Logical Awesome компаниясы болып танылған GitHub, Inc компаниясымен Ruby on Rails пен Erlang те жасалынған. Жасаушылары – Крис Ванстрас, PJ Хиетт және Том Престон–Вернер.

Шығыс коды ашық жобаларға сервис тегін және оларға (SSL’ды қоса алғанда) барлық мүмкіндіктерді береді, ал жеке жобаларға әртүрлі таривті жоспарларды ұсынады.

Сервистің ұраны – «Social Coding» – қазақ тіліне «Кодты бірге жазайық» деп аударуға болады. Ал жейделерде мүлдем басқа сөз орамын жазады: «Fork you» («Тармақтан!»).

Мүмкіндіктері. Сервисті жасаушылар GitHub–ты «жасаушыларға арналған әлеуметтік желі» деп атайды. Кодты орналастырумен қатар қатысушылар өзара байланысып, бір–бірінің жасаған түзетулеріне түсінік жазып, таныстарының жаналықтарын бақылай алады. Git’тің кең ауқымды мүмкіндіктерінің арқасында бағдарлама жасаушылар өз репозиторийларын біріктіре алады. Ол үшін GitHub ыңғайлы интерфейсті ұсынады және әрбір қатысушының үлесін шежіре түрінде кескіндей алады.

Жобалар үшін жеке парақтар, шағын Викилер және қателерді бақылайтын жүйе бар. Сервистің өзінде біршама тілдердің синтаксисі жарық болатын жобалардың файлдарын көруге болады. Ақылы тарифтік жоспарларды қолданған кезде қолданушылардың шектелген ортасына ғана қол жетімді болатын репозиторийлерді жасауға болады.

Сервисте 2012 жылдың 5 желтоқсанынан бастап оның веб–интерфейсі арқылы өз репозиториіне жаңа файлдарды тікелей қосу мүмкіндігі қосылған.

Жобалардың кодын Git арқылы көшіріп қана қоймай, сайттан кәдімгі мұрағатты жүктеуге болады.

Git'тен басқа сервис кодты SVN және Mercurial арқылы алуды және өзгертулер енгізуді қолдайды.

Бұрында Ruby-жобалар сервистің RubyGems-репозиториінде автоматты түрде жарияланатын, бірақ 2009 жылдың қазан айынан бастап GitHub бұл сервистен бас тартты.

Оған қоса сайтта код фрагменттерін тез жариялауға арналған pastebin-сервисі (gist.github.com) бар.

Танымалдылығы. Бірінші жеке репозиторий 2008 жылдың 12 қаңтарында жасалынып шығарылды. 2011 жылдың аяғына қарай жобада 1 миллионнан астам қатысушы және 2 миллионнан астам репозиторийлер тіркелген.

2008 жылдың аяғында GitHub «Үздік стартап-дебют» марапатына ие болды.

GitHub сервисі Ruby-жасаушылар арасында өте танымал және жиі қолданылып келеді. Көптеген ауқымды және маңызды жобалар өздерінің ресми репозиторийлерін осы сервиста орналастырады.

Facebook, Twitter, HP webOS, Yahoo, Per, Erlang, Scala, Ruby on Rails, CyanogenMod, PHP, JUnit, jQuery, Prototype, MooTools, Microsoft IronRuby, osCommerce, Valve.

2 Арнайы бөлім

Бұл бөлімде Android платформасы үшін мобильді қосымшаны жобалау және құру.

2.1 АЭЖБУ мобильді қосымшасының жалпы сипаттамасы

Негізгі бағдарлама студенттер мен талапкерлерге қолжетімді және қолайлы түрде ақпаратты алуға арнап жасалған ақпараттық қосымша. Қолданушы өзіне қажетті мәліметтерді мобильді қосымша арқылы біле алады. Бағдарлама интерфейсі қолданушы студенттер мен талапкерлерге қолайлы түрде жасалған.

2.2 Қажетті құралдарды дайындау

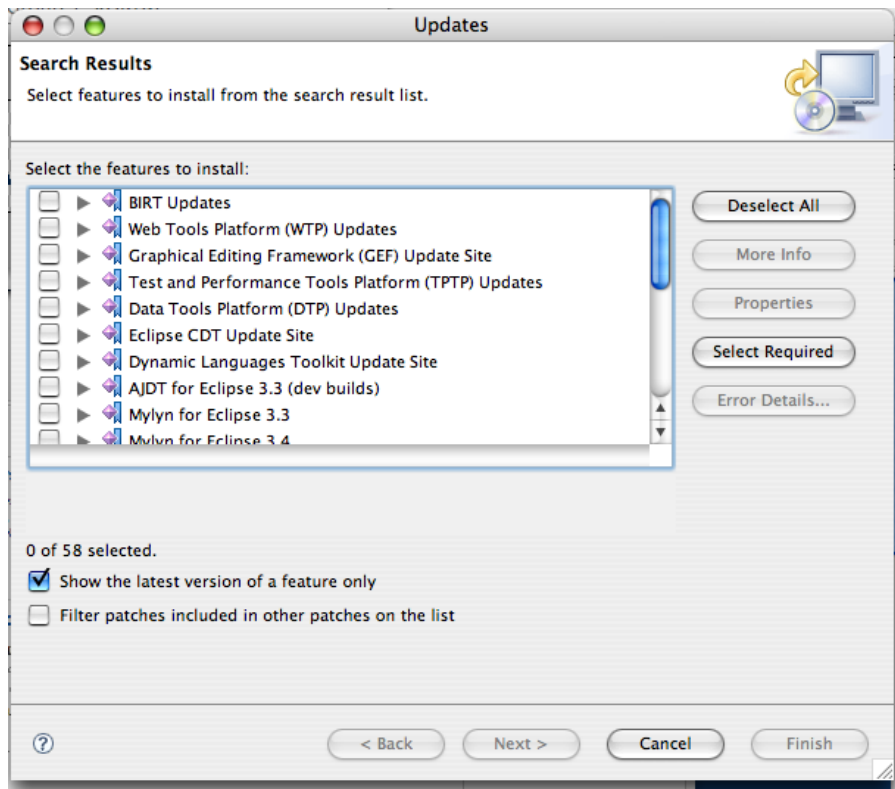
Java Development Kit (JDK) құралын орнату. JDK құралынның 7-ші нұсқасын [<http://www.oracle.com/technetwork/java/javase>] сайтынан жүктеп алып, орнату қажет.

Eclipse IDE бағдарламалау ортасын орнату. [<http://Eclipse IDE.org>] сайтынан Eclipse IDE соңғы 7.3 нұсқасын жүктеп алу қажет. Орнату үшін Windows операциялық жүйесіне арналған Eclipse IDE-7.3rc2-windows.exe файлы жүргіземіз (Сурет 2.1).



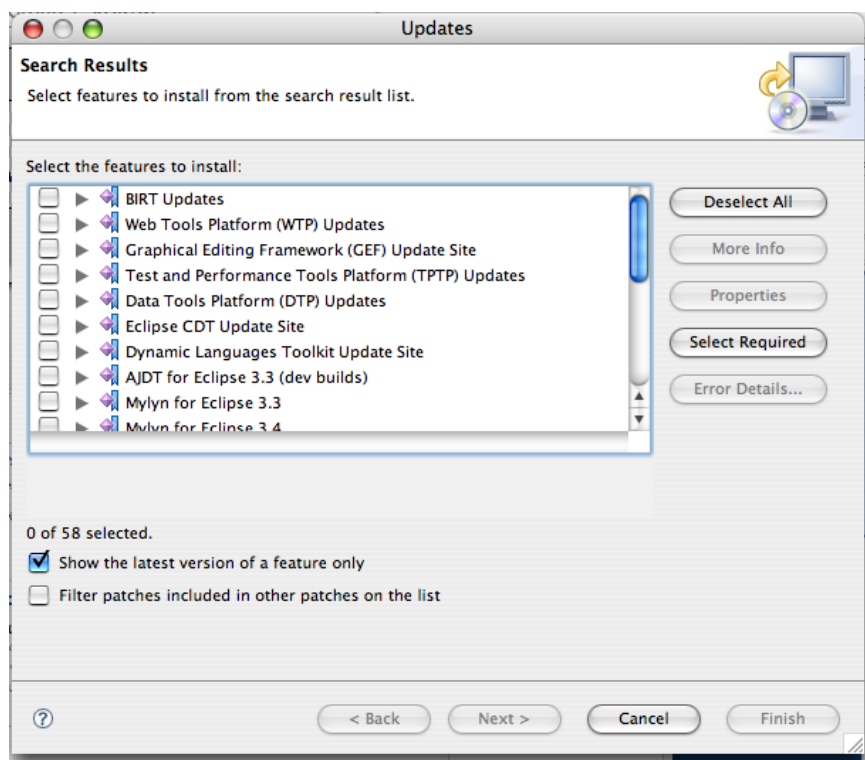
Сурет 2.1 – Eclipse IDE бағдарламалау ортасын орнату терезесі

Келесі қадамда орнатылатын пакеттер тізімін көреміз (Сурет 2.2).



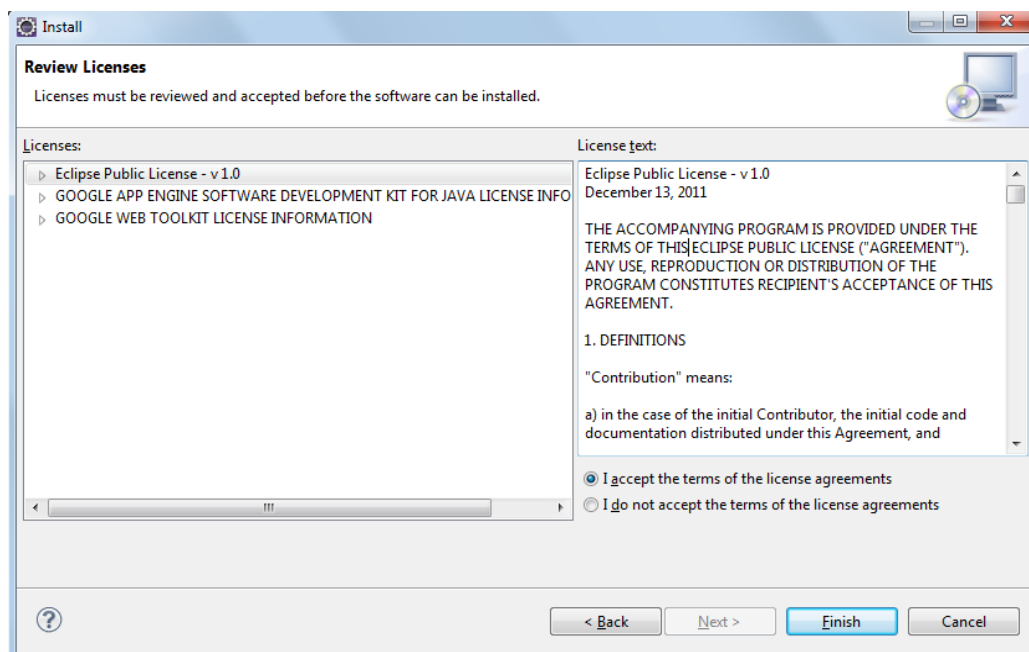
Сурет 2.2 – Eclipse IDE орнатылатын пакеттер тізімі терезесі

Customize батырмасын басып, пакеттер тізімін өзгерту мүмкіндігі бар (Сурет 2.3).



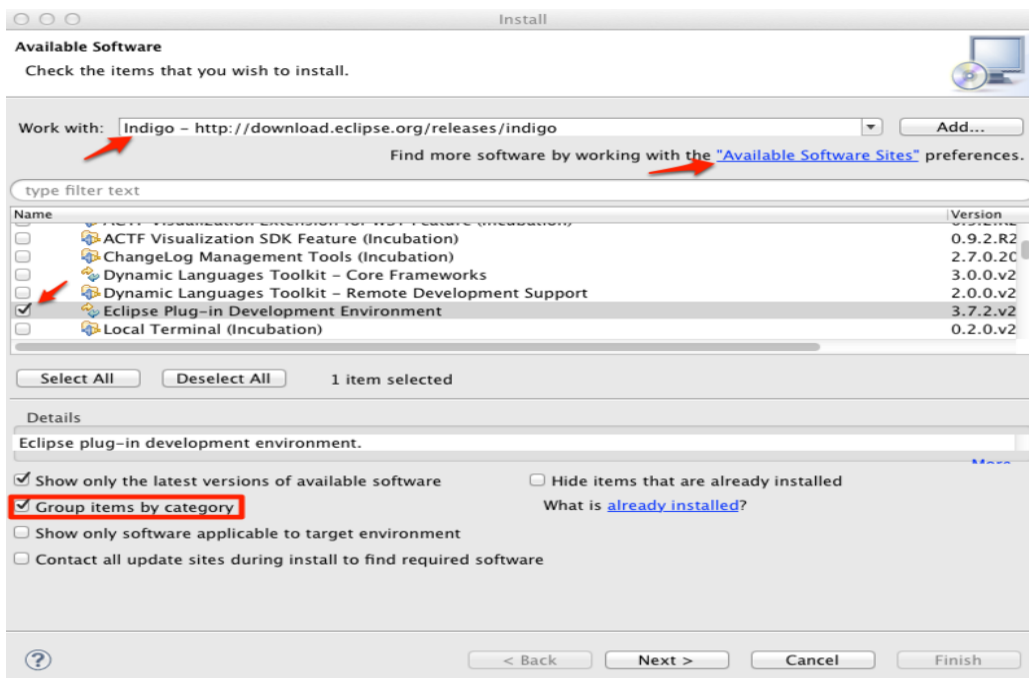
Сурет 2.3 – Eclipse IDE орнатылатын пакеттер тізімін өзгерту терезесі

Келесі қадамда қолданушы лицензиясын қабылдаймыз (Сурет 2.4).



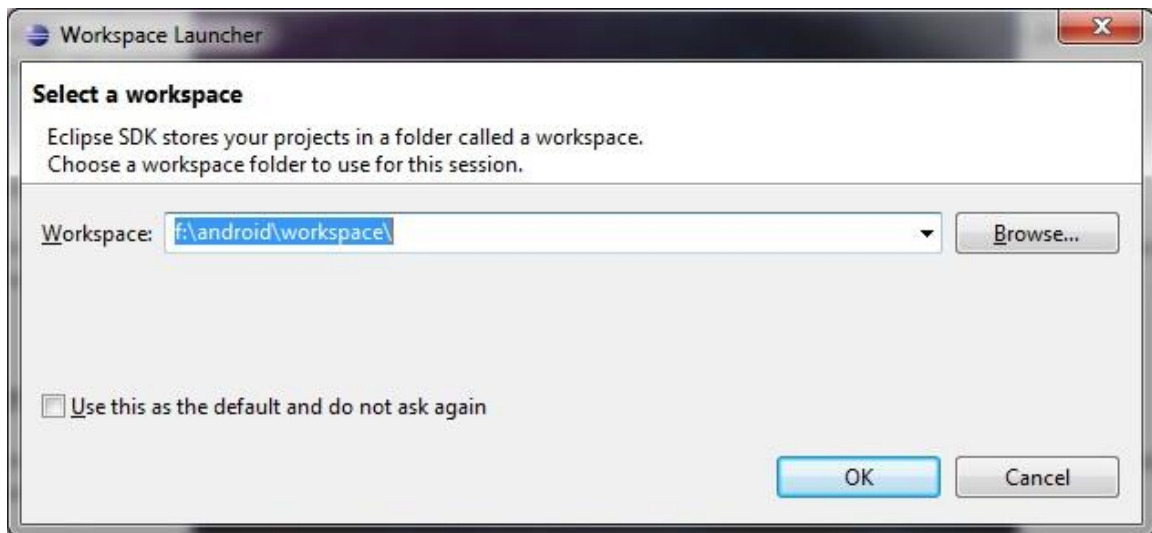
Сурет 2.4 – Eclipse IDE қолданушы лицензиясын қабылдау терезесі

Келесі қадамда JUnit құралының қолданушы лицензиясын қабылдаймыз (Сурет 2.5).



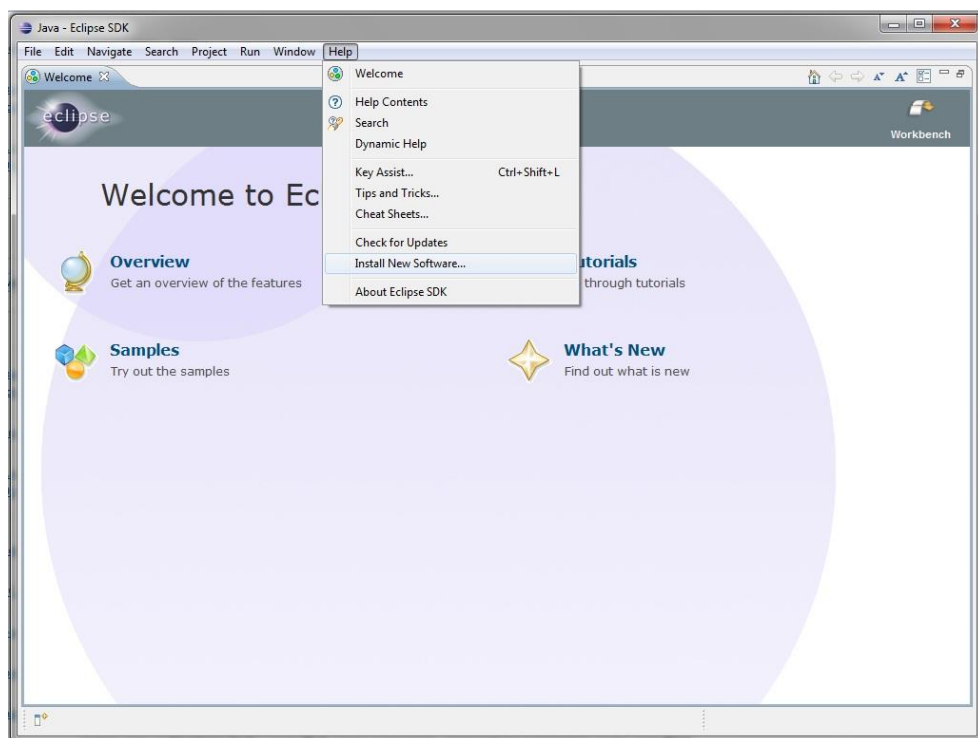
Сурет 2.5 – JUnit қолданушы лицензиясын қабылдау терезесі

Келесі қадамда Eclipse IDE орнату директориясын және JDK директориясын көрсетеміз (Сурет 2.6).



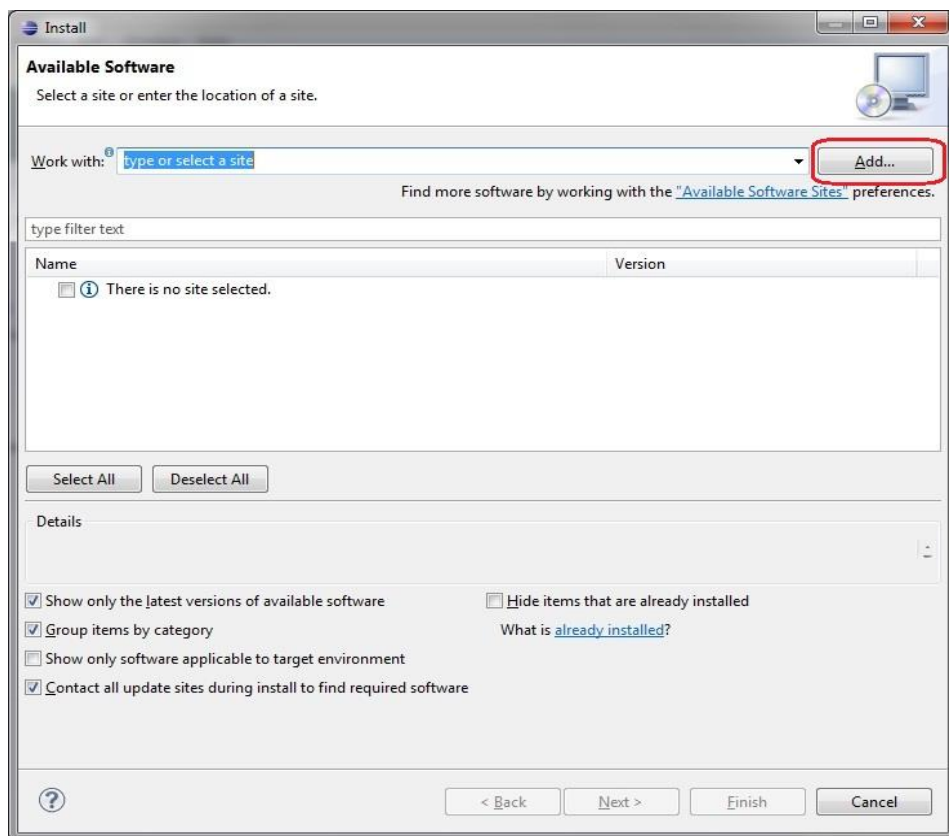
Сурет 2.6 – Eclipse IDE орнату директориясын және JDK директориясын көрсету терезесі

Келесі қадамда GlassFish серверінің орнату директориясын көрсетеміз (Сурет 2.7).



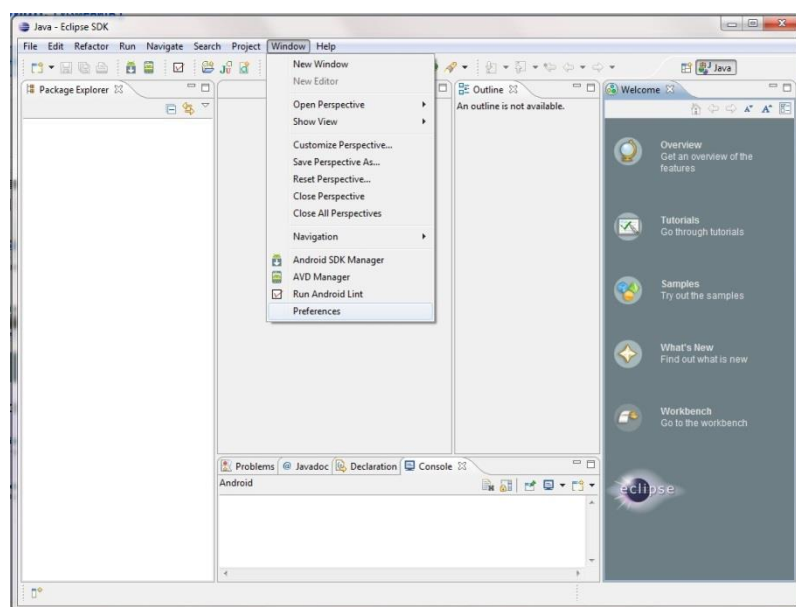
Сурет 2.7 – GlassFish серверінің орнату директориясын көрсету терезесі

Жаңартуларға тексеру қажет жағдайда Check for Updates нұсқасын белгілеп, Install батырмасын басамыз (Сурет 2.8).



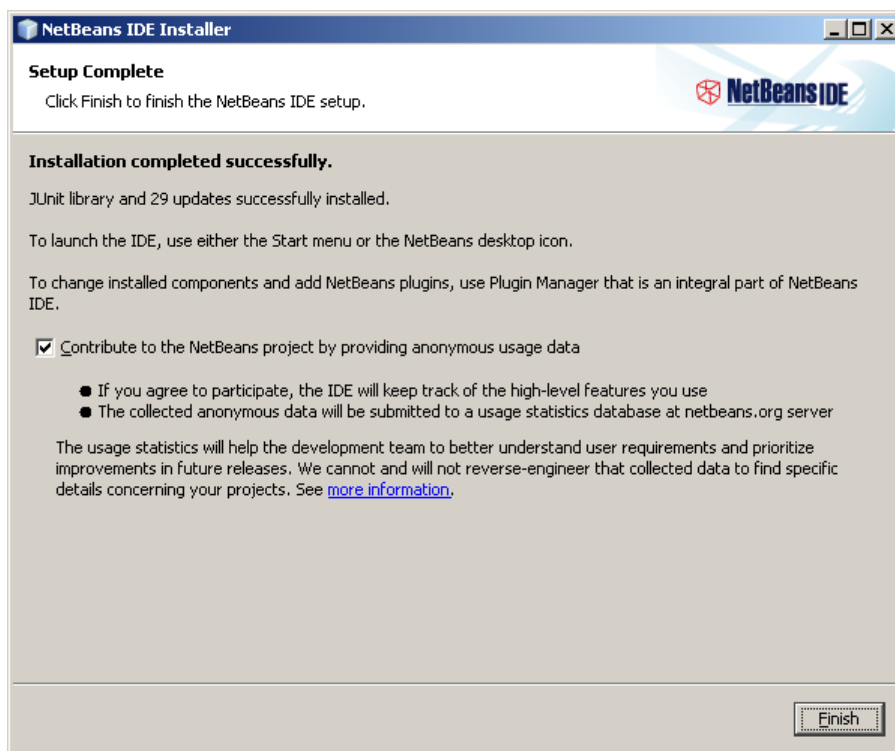
Сурет 2.8 – Check for Updates нұмқасын таңдау

Келесі қадамда Eclipse IDE ортасын орнату көрсетілген (Сурет 2.9).



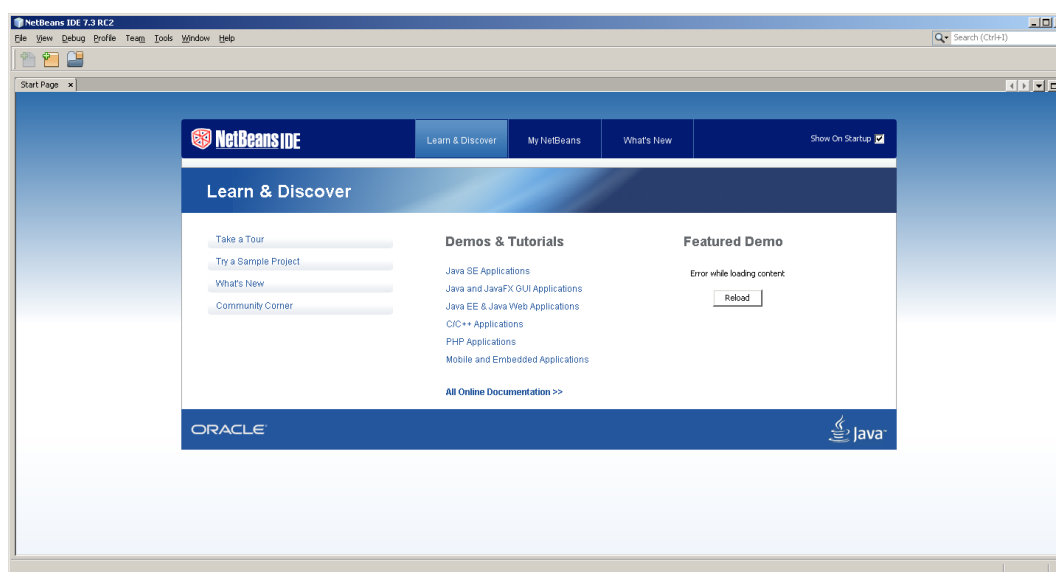
Сурет 2.9 – Eclipse IDE ортасын орнату барысы

Келесі суретте Eclipse IDE ортасының аяқатлуы (Сурет 2.10).



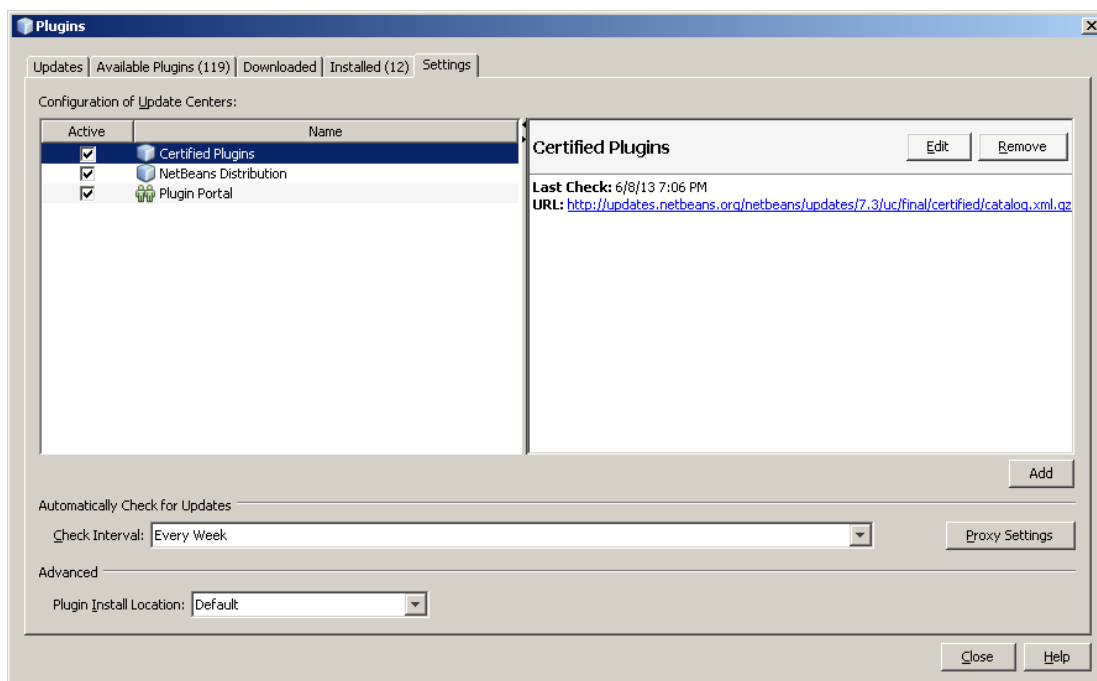
Сурет 2.10 – Eclipse IDE ортасын орнатудың аяқталуы

Eclipse IDE платформасының ортасы (Сурет 2.11).



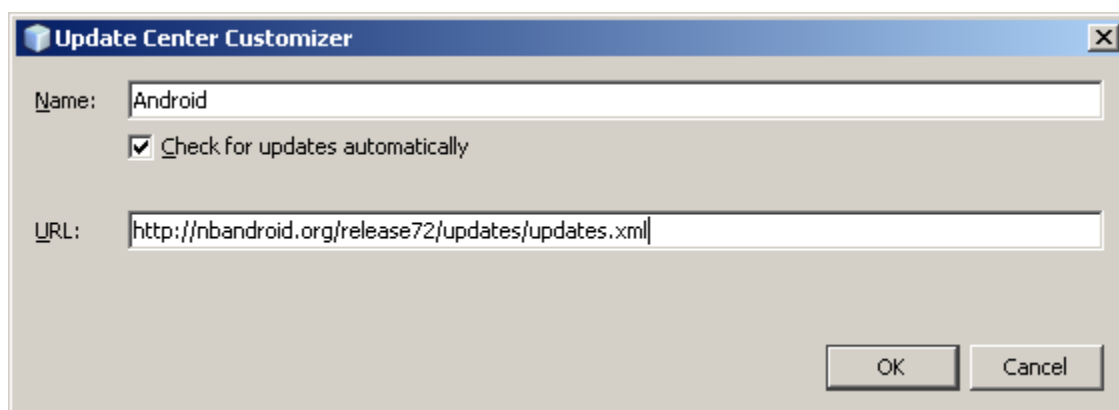
Сурет 2.11 – Eclipse IDE ортасы

Android платформасына қосымша құруды бастамастан бұрын, плагинін орнату қажет. Ол үшін бас мәзірдің Tools > Plugins > Settings терезесіне өтіп, жаңа конфигурация құрамыз (Сурет 2.12).



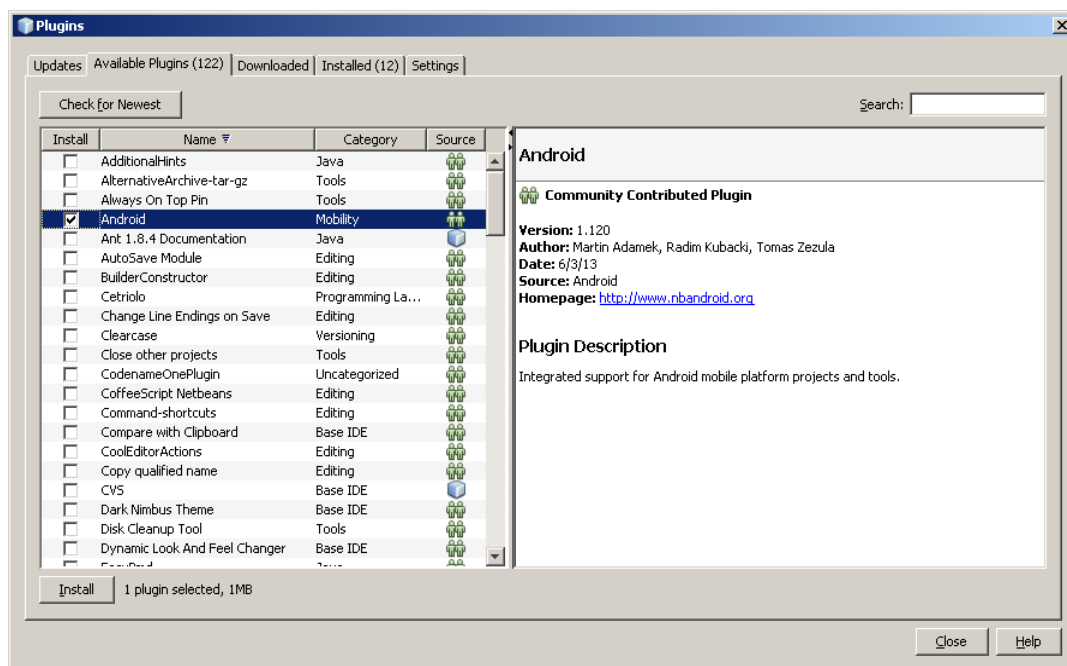
Сурет 2.12 – Бас мәзірдің Tools > Plugins > Settings терезесі

Android отрасында жаңа конфигурация құру (Сурет 2.13).



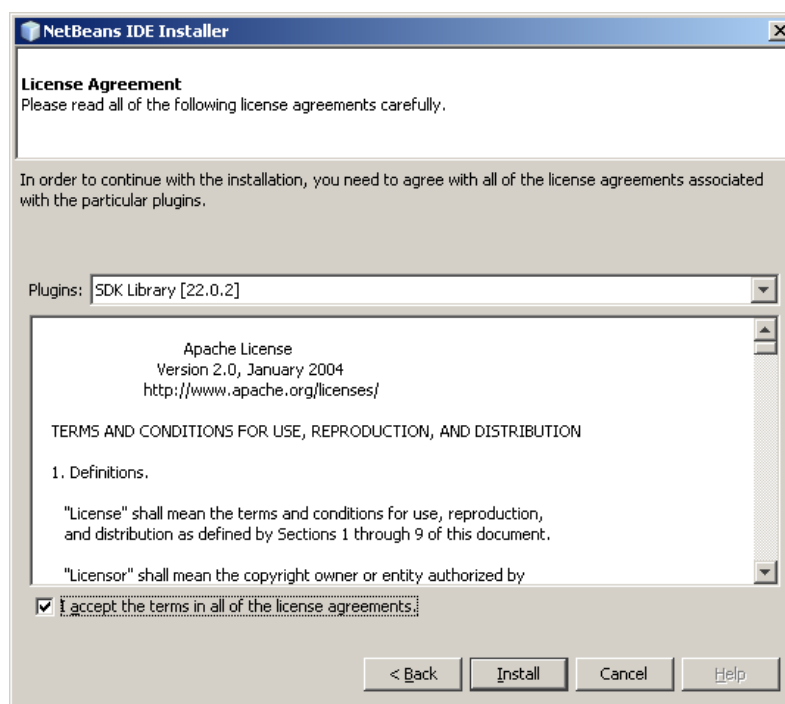
Сурет 2.13 – Жаңа конфигурация құру

Одан кейін Tools > Plugins > Available Plugins терезесіне өтіп, Android плагинін орнатамыз (Сурет 2.14).



Сурет 2.14 – Android плагинін орнату

Android плагинін орнату кезінде қолданушы лицензиясын қабылдау қажет (Сурет 2.15).



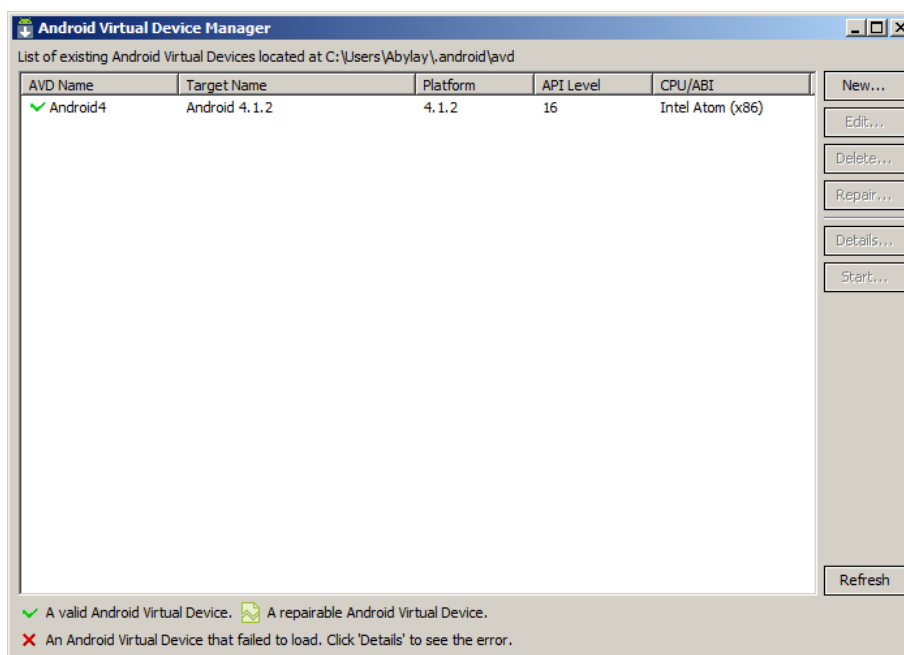
Сурет 2.15 – Android плагинінің қолданушы лицензиясы

Android SDK құралын орнату. Android SDK құралын <http://developer.android.com/sdk/index.html> сайтынан жүктеп алу қажет. Орнату үшін Windows операциялық жүйесіне арналған installer_r21.1-windows.exe файлы жүргіземіз.

2.3 Қосымшаның Java Eclipse IDE платформасында жүзеге асырылуы

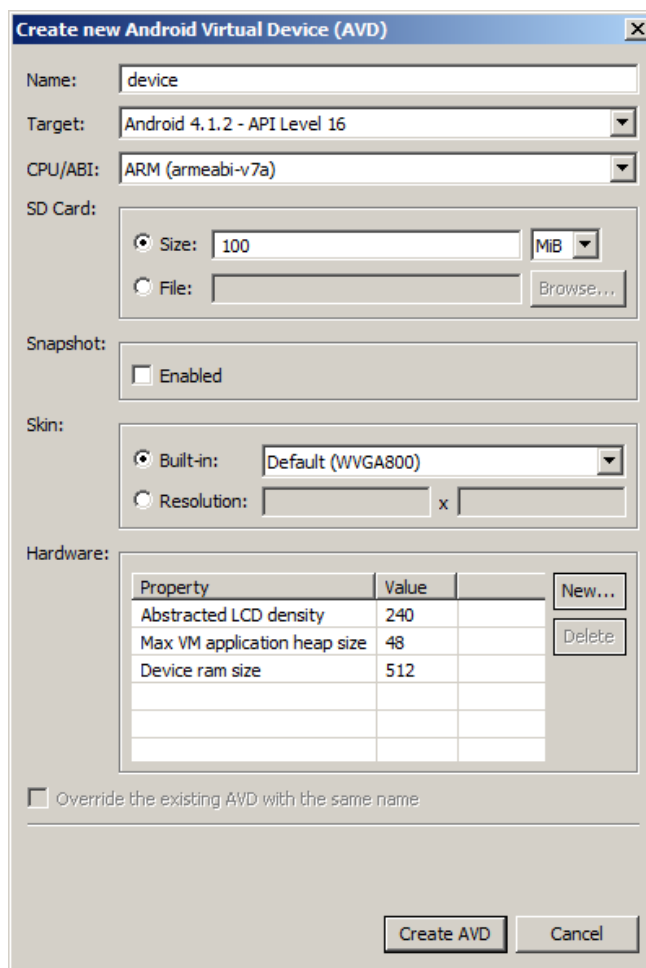
Қосымшаны құрмас бұрын қосымшаларды тестілеуге арналған виртуалды Android-құрылғы құрамыз. Android Virtual Device (AVD) бұл Android операциялық жүйесі негізіндегі виртуалды смартфон. Android Virtual Device артықшылығы қосымшаны әртүрлі смартфондарға орнатып, тестілеу мүмкіндігінде.

Ол үшін бас мәзірдің Tools > AVD Manager терезесіне өтеміз (Сурет 2.16).



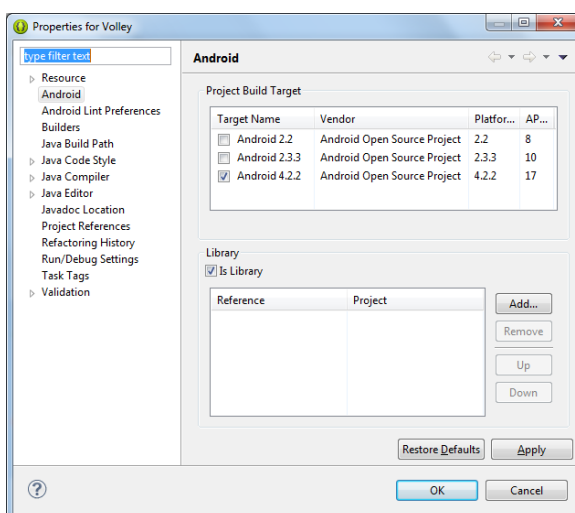
Сурет 2.16 – AVD Manager терезесі.

Жаңа құрылғы құру үшін New батырмасын басып, Name (құрылғы атауы), Target (құрылғы Android нұсқасы), CPU/ABI (процессор), SD Card Size (сыртқы жады өлшемі) параметрлерін толтырып, батырмасын басамыз (Сурет 2.17).



Сурет 2.17 – Жаңа құрылғы құру

Еclipse IDE ортасында жаңа жоба құру үшін бас мәзірдің File > New Project терезесіне өтеміз. Android бөлімінде Android Project жоба нұсқасын таңдаймыз (Сурет 2.18).



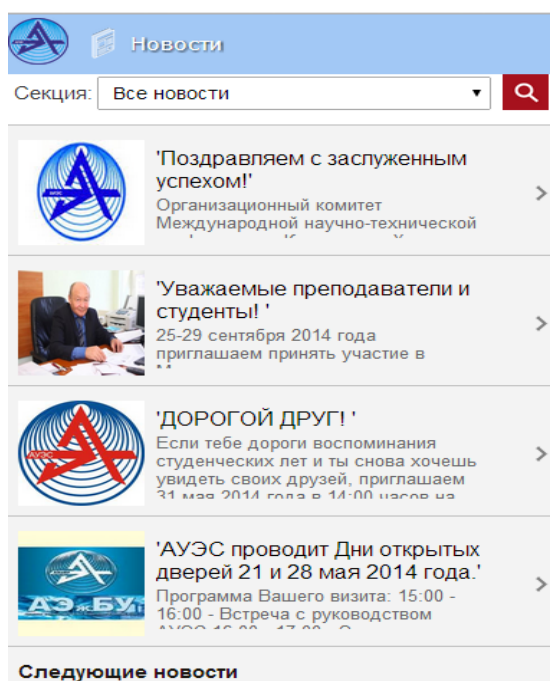
Сурет 2.18 – Android жаңа жобасын құру

Қосымша құрылғаннан кейін бастапқы беттің кескіні жасалады. Басты бет мәзірлер терезесінен тұрады (Сурет 2.19).



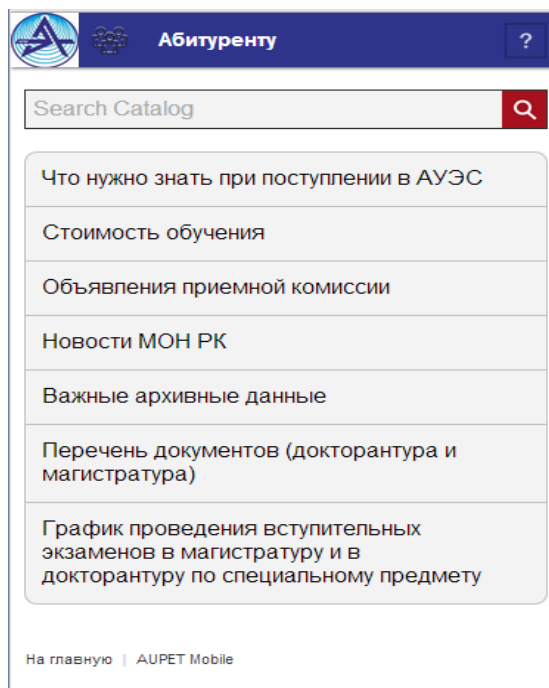
Сурет 2.19 – Қосымша ашылу кезіндегі басты беті

Басты бет интерфейсінде мәзірлер терезесінен жаңалықтар терезесін таңдаймыз (Сурет 2.20). Жаңалықтар терезесін басу арқылы соңғы жаңалықтар тізімінің андатпаларын көруге болады.



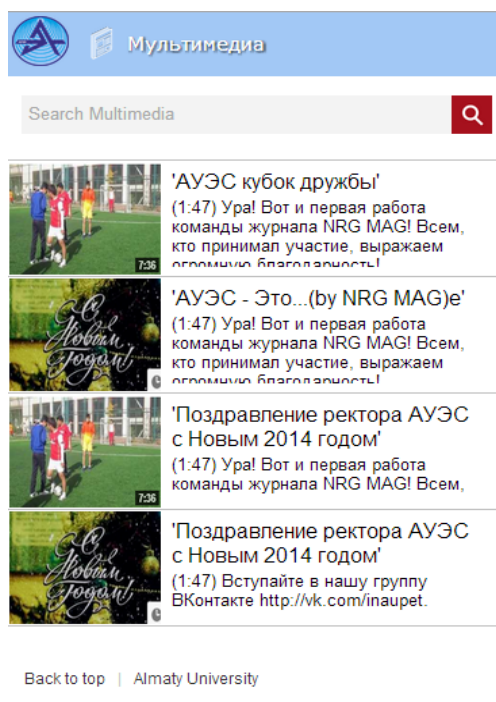
Сурет 2.20 – Жаңалықтар беті

Талапкерге терезесін басу арқылы талапкерге арналған бетке кіре аламыз (Сурет 2.21). Бұл бетте талапкерге керекті ақпараттар көрсетілген.



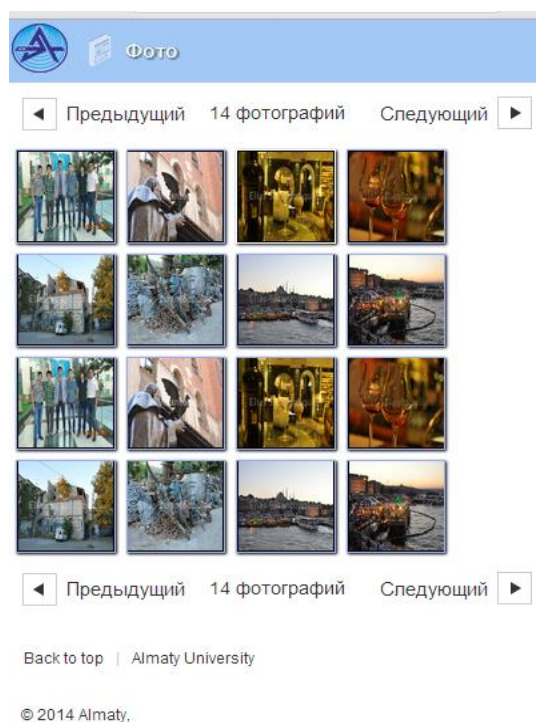
Сурет 2.21 – Талапкер беті

АЭЖБУ–дегі соңғы бейнебаяндарды көру үшін мультимедиа терезесін басу арқылы бейнебаяндар бетіне көшеміз (Сурет 2.22).



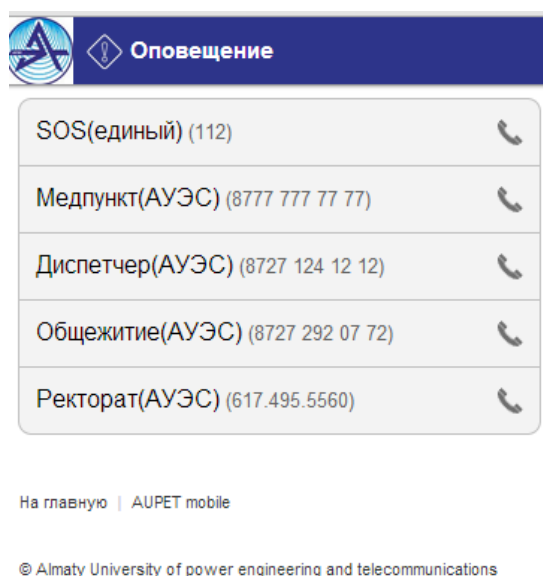
Сурет 2.22 – Бейнебаяндар беті

Өзекті және қызықты болған суреттер фото галлерейға жүктеледі (Сурет 2.23). Жүктелген суреттерді анимация түрінде қарауға болады.



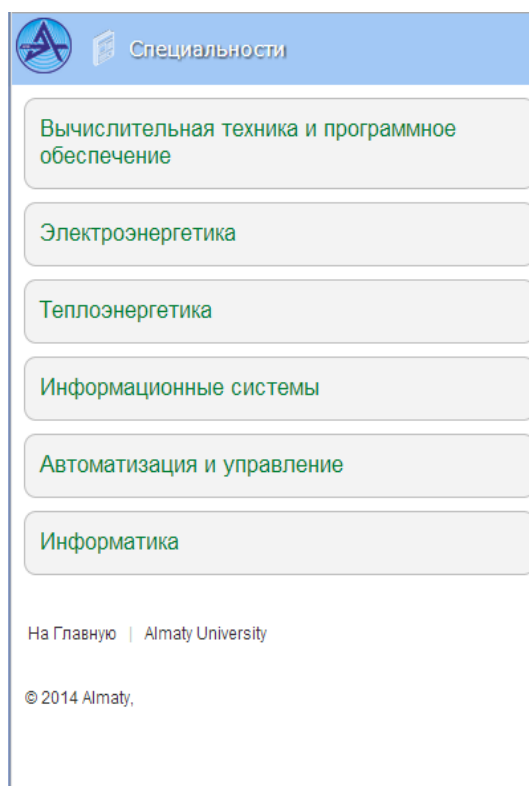
Сурет 2.23 – Фото галлерейа беті

Қолданушыға шұғыл жағдайда қоңырау шалу қажеттілігі туындаған кезде хабарлау бетіне өту арқылы керекті телефон номерлерін қолдана алады (Сурет 2.24).



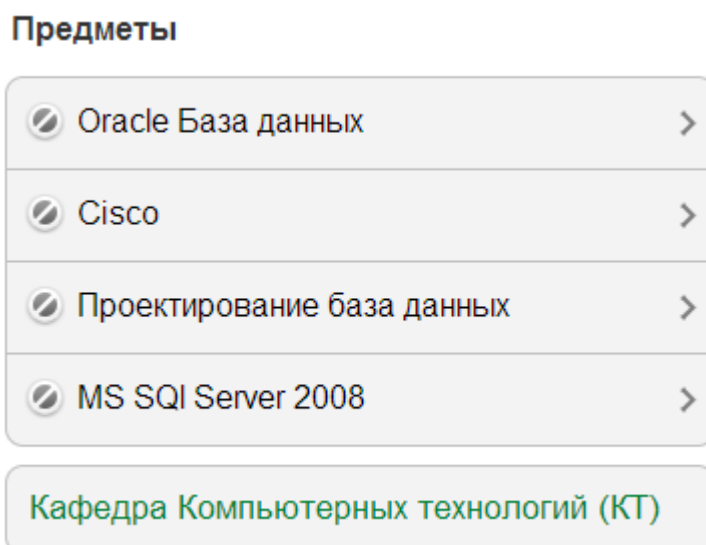
Сурет 2.24 – Хабарлау беті

Қолданушы АЭЖБУ–дағы мамандықтар тізімін көру үшін мамандықтар бетіне барады (Сурет 2.25). Мамандықтар бетінде АЭЖБУ–дағы мамандықтар тізімі көрсетілген. Қажетті мамандықты таңдау арқылы толық ақпарат біле алады және қай кафедраның құрамына кіретінін қарай алады.



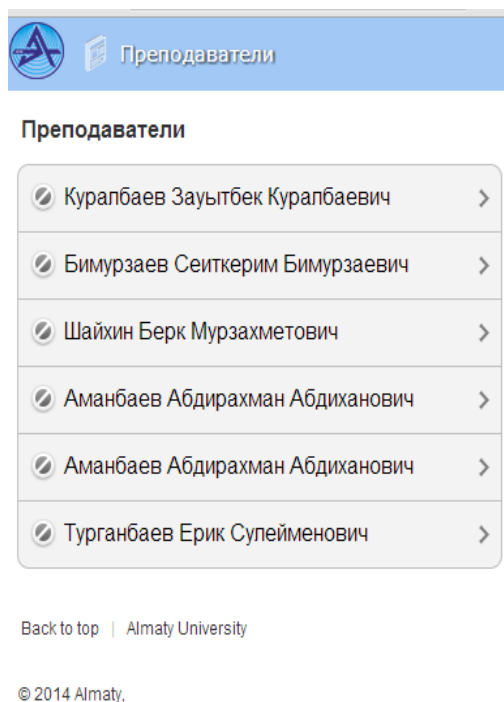
Сурет 2.25 – Мамандықтар беті

Бір мамандықты таңдау арқылы келесі бетке өтеміз (Сурет 2.26). Мамандықтың сабақтар тізімін оның кафедрасын қарай аламыз.



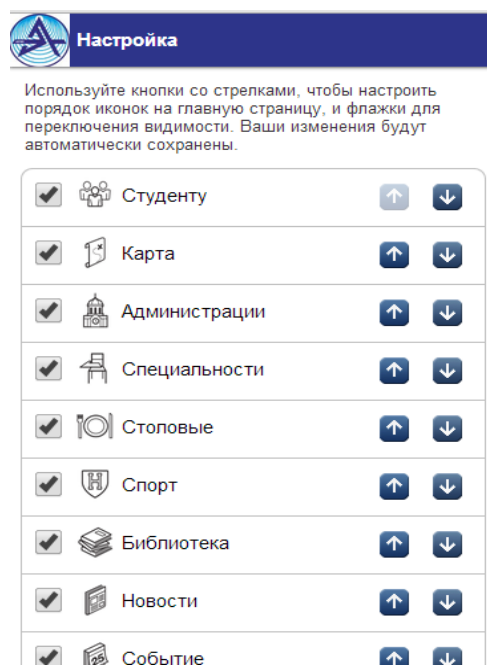
Сурет 2.26 – Сабақтар беті

Кафедраны шерту арқылы оқытушылар тізімін көре алады (Сурет 2.27).



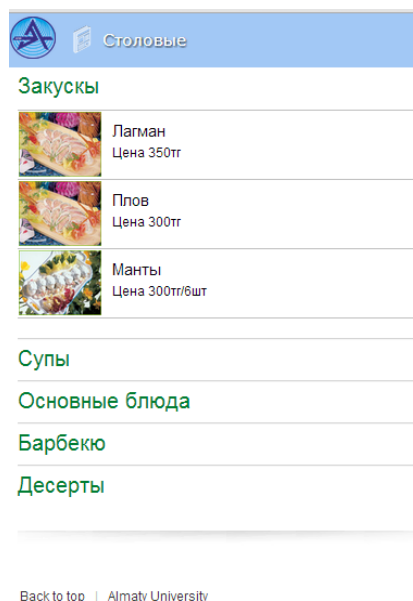
Сурет 2.27 – Оқытушылар беті

Баптаулар бетінде қолданушы өзіне қажетті терезелерді өзгерту арқылы қолайлы интерфейс құра алу мүмкіндіктері жүзеге асырылған (Сурет 2.28).



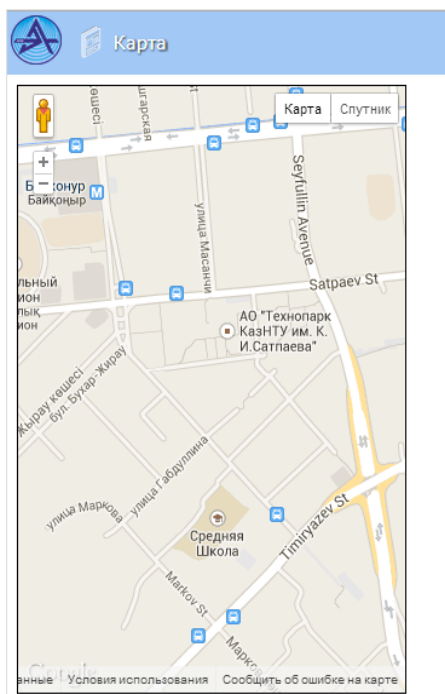
Сурет 2.28 – Баптау беті

Қолданушы күнделікті асханалардағы ас мәзірін көргісі келсе асханалар бетіне өте алады (Сурет 2.29).



Сурет 2.29 – Асханалар беті

АЭЖБУ–ін қолданушы картадан қарау қажеттілігі туындағанда карта бетіне бара алады (Сурет 2.30).



Сурет 2.30 – Карта беті

Бұл мобильді қосымшада өзгеде қолданушы үшін анықтамалық ақпараттар беретін терезелер қарастырылған.

3 Тіршілік қауіпсіздігі

3.1 Оператордың жұмыс жағдайын талдау

Дипломдық жұмыста студенттердің Алматы Энергетика және Байланыс Университетінің мобильді қосымшасын жобалау қарастырылған. Бағдарламаны құру және жобалау Android платформасында жүзеге асырылады. Біз жұмыс орны ретінде мекеме бөлмесін қарастырамыз. Қарастырылып отырған бөлмеде төрт қызметкер жұмыс істейді, олардың әрқайсысының өз жұмыс орны бар. Басты инженер—бір адам, инженер—бір адам, және екі оператор бар. Барлығына тәулікте сегіз сағаттық жұмыс жұмыс күні орнатылған және барлығы күндізгі ауысымда аптасына бес күн жұмыс істейді.

Оператордың жұмыс орны деп іскердің өндірістік процесті жүргізу және бақылау мақсаты үшін жабдықталған кеңістік бөлігін айтады. Сонымен қатар, жұмыс орыны болып, оның еңбек қызметін іске асыратын және басқару органы мен қосымша жабдықпен, ақпаратты көрсету құралдарымен жабдықталған жүйедегі орыны саналады. Оператордың жұмыс орынын құрастырғанда келесі шарттардың маңызы зор. Атап айтар болсақ: жұмыс істеуші адам үшін жұмыс кеңістігінің жеткіліктігі, ол техникалық жабдықтардың қызмет етуіне және пайдалануына барлық қажетті орын ауыстыруын және қозғалуын жүзеге асыруына мүмкіндігі болуы жұмыс орнында дірілдеткіштер мен шуыл көздерінен қорғаныс шараларының қамтамасыз етілуі. Дербес компьютермен жұмыс жасайтын орындықтардың құрастырылуы техникалық талаптарға сәйкес болуы керек. Жұмыс отырғышы көтерілетін—бұрылатын және биіктігі мен бұрыштары бойымен реттелетін болуы керек. МЕСТ 12.1.005–88 «Жұмыс орнының ауасы, санитарлық—гигиенаның жалпы талаптары» 3.1 – кестеде келтірілген. Оператордың жасайтын еңбегі бірінші категорияға жатады (Ia).

К е с т е 3 . 1 – Организмнің энергошығыны бойынша жұмыс категориясы

Жұмыс категориясы	Категория	Адам организмнің энергия шығыны, Ккал/сағ.	Жұмыс сипаттамасы
Жеңіл физикалық жұмыс I a	I a	138–172	Жұмыс отыру күйінде өтеді

Жұмыс бөлмесін талдау барысында жұмыс аумағының қалыпты микроклиматтық шарттарын міндетті түрде қарастыру қажет. Мұндай әмбебап жүйе ретінде автономдық кондиционерлер болып табылады. Кондиционерді орнату алдында арнайы есептеулер жасау керек. Сол есептеулердің қорытындысымен сипаттамасы және талаптарына сай кондиционер таңдалады. Бөлмені кондиционерлеу төменде келтірілген.

3.2 Жұмыс орнының микроклимат жағдайы ,өрт қауіпсіздігі

Микроклиматтың күйін қадағалау операторлар залында қолайлы жағдайды орнатуға көмектеседі. Ал жұмыс орнының қолайлы жағдайларында адамдардың жұмыс істеу қабілеттері жақсарады, жұмысшылардың ауруға шалдығуы азаяды сонымен қатар, еңбек өндірісі артады. Жұмыс орнындағы нормаланған климаттық параметрлер 3.2 кестеде келтірілген.

К е с т е 3.2 – Температураның, ылғалдылық пен ауа қозғалысының нормалары

Жыл мезгілі	Ауа температурасы, °С			Салыстырмалы ауа ылғалдылығы, %		Ауа қозғалысының жылдамдығы, м/с	
	Қолайлы	Рұқсат етілген		Қолайлы	Рұқсат етілген	Қолайлы	Рұқсат етілген
		Жоғарғы	Төменгі				
Суық	21–23	24	20	40–60	75	0,1	<0,1
Жылы	22–24	28	22	40–60	60 – 70	0,2	0,1–0,3

Жазғы микроклиматтың қысқыға қарағанда, ауа баптауы сәл жоғарырақ. Кесте бойынша, ауа температурасы 23–25° ылғалдылығы 30–50%, ауа қозғалысының жылдамдығы 0,2м/с. 55% дейінгі ауа ылғалдылығы құрғақ деп есептеледі, 56–70% – деңгейінде орташа құрғақ, 71–85% – орташа ылғалды және 85% жоғары – тым ылғалды. Ең үлкен ылғалдылық пен салыстырмалы ылғалдылық арасындағы айырым қанығу тапшылығы деп аталады. Ауаның температурасы 23–24⁰С – де ауа қозғалысының жылдамдығы 0,15 м/с дейін болған жөн. Адамның өмір тіршілігіне байланысты ауа құрамының өзгеруі, оған берілген көмір қышқыл газымен есептеледі – CO₂. Бөлмедегі CO₂ шоғырлануы 3.3 – кестеде келтірілген.

К е с т е 3.3 – Бөлме ауасындағы көмір қышқыл газының шоғырлануы

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

CO₂ жұтатын ауада артық не кем болуы ағзаға біркелкі зиянды. CO₂, жетімді шоғырлануы 0,03% кем болса, онда аталған органдардың жұмысы бұзылады, CO₂ > 1,5%–тен артық болса, онда наркотикалық әсері болады, бас ауырады және т.б. Егер дем алатын ауада CO₂= 0,5–1,5% мәнінде болса, онда ағзаға оның елеулі әсері болмайды. Ал ең қолайлы шоғырлану шамасы CO₂= 0,04–0,5% сәйкес келеді. Қолданыстағы санитарлық нормалар бойынша бір адамға 20–60 м³/сағ таза ауа қажет.

Келтірілген шамалар адам организміне ыңғайлы нормаларға сай келмейді. Сондықтан операторлар бөлмесінде ауаны кондиционерлеу мәселесі қарастырылған.

3.3 Жұмыс орнының табиғи және жасанды жарықталуы, электр қауіпсіздігі

Табиғи жарықтандыру терезелерден түсетін жарық арқылы іске асады. Табиғи жарықтандыру коэффициентінің (ТЖК) шамасы жоғары дәлдік категориясының жұмысын орындау кезіндегі нормативті деңгейге сәйкес.

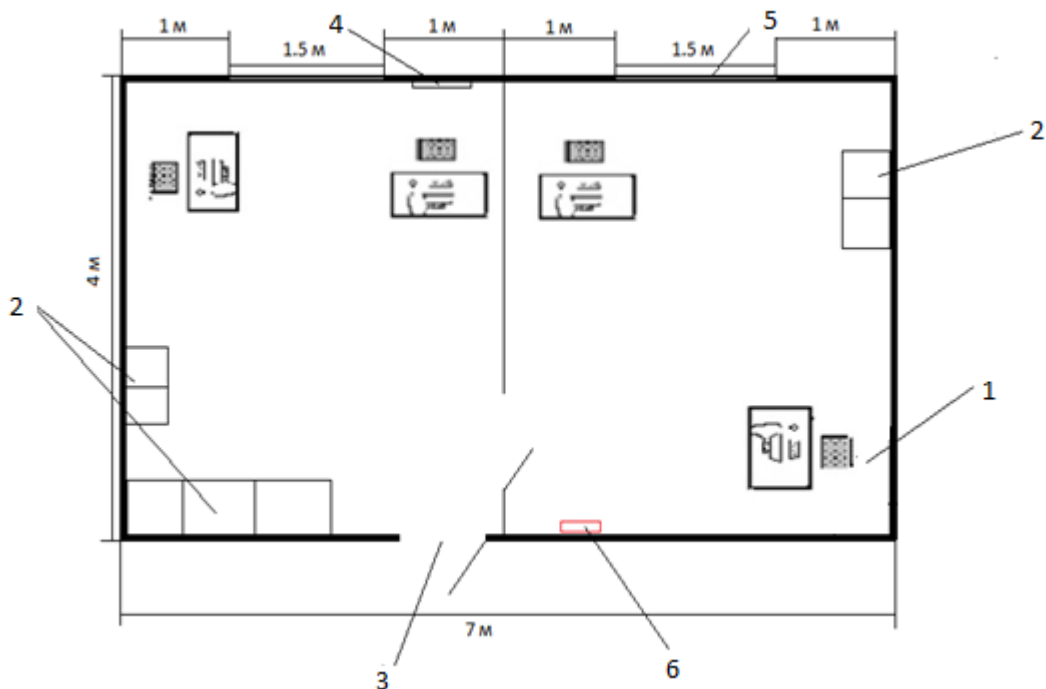
Берілген бөлmemіз барлық электр қауіпсіздігі қанағаттандырады, яғни бөлме ылғал емес 60% аспайды, тақтай еденнен жасалған. Жерге тұйықтау және нөлдеу сымдарының қимасы, олардың қосылыстары мен жалғанымдарының бүтіндігі және беріктігі тексерілген.

3.4 Бөлме жоспары

Жұмыс орнының мінездемесі:

- дербес компьютер құрылғысынан шу аз мөлшерде шығады, сондықтан да үлкен мөлшердегі шу зияны жоқ. Өрт қауіпсіздігі бөлмедегі өрт сөндіргіш құралы және автоматты жылу сезгіш арқылы қамтамасыз етілген;
- бөлменің өлшемдері (бөлме): ұзындығы 7 м, ені 4 м, биіктігі 3 м;
- жарық өткізетін материалдың түрі – шыны парағы, екіқабатты;
- күннен қорғанатын жабдықтар – жиналатын жазюли және шторлар;
- екі терезе 1,5 – 1,2;
- қабырғаның ішкі түрі – ақшыл сары;
- есік 1·2.

Оператор залының жоспары төменде 3.1 суретте көрсетілген.



Сурет 3.1 – Оператор залының жоспары

1 – Операторлардың жұмыс орындары.

2 – Шкаф.

3 – Есік жақтауы.

4 – Кондиционерлер орны.

5 – Терезе жақтаулары.

6 – Өрт сөндіргіш балоны.

Жұмыс істеу кезіндегі қолданылатын құрылғының мінездемесі:

- Intel(R) core i7 CPU E7800 @ 3.0 GHz 3.0 GHz, 8 ГБ RAM / HDD 1000 Gb, 802.11b/g;
- монитор Samsung 2013 luminal, диагональ 15.6”;
- қоректену: айнымалы кернеу 220–250 В, жиілігі 50 Гц. Қуат 400 Вт.

3.5 Жасанды жарықтандыруды есептеу

Жұмыс орнындағы табиғи жарықтану аз мөлшерде болғандықтан біз біз жасанды жарықтануды орнатуға көшеміз. Бөлмедегі жасанды жарықтану люминисценттік жарық көзінің жүйесі арқылы ұйымдастыруды қарастырамыз. Қарастырылып жатқан жарық көзінің жүйесінің жарық беруі жоғары болу керек (75 лм/Вт және одан да жоғары), жұмыс істеу уақытының мерзімі (10000 сағатқа дейін), және сонымен қатар жарықтың сәуле шашуының спектрлік құрамы табиғиға жуық болу керек, бұл жағдай жарық жіберуінің жақсы болуын қамтамасыз етеді.

Бұл әдістің мақсаты η коэффициентін анықтауға негізделген, бұл коэффициент есептелетін жазықтыққа түсетін жарық ағынына тең мән. Бұл коэффициенттің мәнін бөлменің геометриялық параметрлерін байланыстыратын кестесіне

аламыз (болме индексі i), сонымен қатар оптикалық сипаттамаларын аламыз (болме тобесінің шағылысу коэффициенті $\rho_{\text{пот}}$, дуалдың $\rho_{\text{ст}}$ және еденнің $\rho_{\text{п}}$). Болме индексі i келесі формула арқылы анықталады

$$i = \frac{A \cdot B}{h \cdot (A + B)} \quad (3.1)$$

мұндағы A – бөлме ұзындығы;
 B – бөлме ені;
 h – есептелген биіктік

$$h = H - h_p \quad (3.2)$$

$$h = 3 - 0,8 = 2,22 \text{ м}$$

Есептелген биіктіктің анықталған мәнінен болме индексін анықтаймыз

$$i = \frac{7 \cdot 4}{2,22 \cdot (7 + 4)} = 1,13$$

Шағылысу коэффициентінің мәндерін келесідей деп аламыз

$$\begin{aligned} \rho &= 70\% , \\ \rho_{\square} &= 50\% , \\ \rho_{\cdot} &= 30\% \end{aligned}$$

Анықталған бөлме индексі және таңдап алынған шағылу коэффициенттері арқылы η коэффициентін анықтаймыз, ол келесідей болады $\eta = 60\%$.

Кіші қысымды газоразрядтық шамдар ең үнемді болып саналады (ЛД, ЛДЦ, ЛХБ және т.б.). Газоразрядтық шамдардың кейбір техникалық сипаттамалары 3.4 кестеде келтірілген.

К е с т е 3.4 – Кіші қысымды газоразрядтық шамдардың кейбір сипаттамалары

Номиналдық қуат, Вт	НОМИНАЛДЫ ЖАРЫҚТЫҚ АҒЫМ ЛМ., ШАМДАР ТИПІ					Шамның өлшемі, мм.	
	ЛДЦ	ЛД	ЛХБ	ЛТБ	ДБ	Диаметр	Штырьдың ұзындығы
15	500	590	675	700	760	27	451,6
20	820	920	935	975	1060	40	604,0

К е с т е 3.4 соңы

Номиналдык куат, Вт	НОМИНАЛДЫ ЖАРЫҚТЫҚ АҒЫМ ЛМ., ШАМДАР ТИПІ					Шамның өлшемі, мм.	
	ЛДЦ	ЛД	ЛХБ	ЛТБ	ДБ	Диаметр	Штырьдың ұзындығы
30	1450	1640	1720	1720	2100	27	908,8
40	2100	2340	3000	3000	3120	40	1213,6
65	3050	3570	3820	3980	4650	40	1514,2
80	3740	4070	4440	4440	5220	40	1514,2

Бөлмені жарықтандыру үшін люминесценттік газразрядтық ЛД шамдарын таңдаймыз, бұл шамдардың қуаты 65 Вт және номиналды жарық ағыны 3570 лм. ЛДР типті шамдар (2·60 Вт): ұзындығы 1,24 м, ені 0,27 м, биіктігі 0,10 м. Жоғарыда қарастырылып кеткен есептеулеріміздің арқасында болмедегі шамдар санын анықтаймыз

$$N = \frac{E \cdot r \cdot S \cdot z}{n \cdot \Phi \cdot \eta} \quad (3.3)$$

мұндағы E берілген минималды жарықтану;

$E = 300$ лк., ойткені көру жұмысының разряды $n = 5$;

r – қор коэффициенті, $r = 1,3$ (болмеде дербес компьютерлермен жұмыс істеу үшін);

S – жарықтандырылатын аудан, $S = 28 \text{ м}^2$;

z – нормаланған жарықтандыруды сипаттайды, $z = 1,1$ (люминесценттік шамдар үшін);

n – шам ішіндегі лампалар саны.

Сонымен шамдар саны

$$N = \frac{300 \cdot 1,3 \cdot 28 \cdot 1,1}{2 \cdot 3750 \cdot 0,6} \approx 4$$

Шамдарды орналастыру келесідей шамалармен анықталады:

$H = 3 \text{ м}$ – болме биіктігі;

$h_c = 0,25 \text{ м}$. – шамдардың жабындыдан ара қашықтығы.

Шам мен еден арасындағы ара қашықтық

$$h_{\Pi} = H - h_{\text{сы}} \quad (3.4)$$

$$h_{\Pi} = H - h_c = 3 - 0,25 = 2,75 \text{ м}$$

h_p = есептік жазықтығының биіктігі 0,7 м (ДК мен жұмыс ісеуге арналған болмелерге байланысты).

Есептік биіктік

$$h = h_n - h_p \quad (3.5)$$

$$h = 2,75 - 0,7 = 2,05$$

ЛДР типті шам (2·60 Вт). Ұзындығы 1,24 м, ені 0,25 м, биіктігі 0,10 м.
Көрші тұрған шамдар арасындағы арақашықтықтың формуласы

$$\lambda = \frac{\alpha}{n} (1,2 \div 1,4) \quad (3.6)$$

$$\alpha = 1,2 \cdot 2,75 = 3\text{ м}$$

Шамнан дуалға дейінгі арақашықтық

$$l = (0,25/0,3) \cdot \alpha \quad (3.7)$$

$$l = 0,25 \cdot 3 = 0,75\text{ м}$$

L – көрші шамдар арасындағы ара қашықтық (люминесценттік шамдар қатарында). L_a (бөлменің ұзындығы бойынша) = 1,76 м. L_b (бөлменің ені бойынша) = 3 м. l – шеткі шамдардан алшақтық немесе шамдар қатарынан дуалға дейінгі қашықтық,

$$l = 0,3 - 0,5L$$

$$l_a = 0,5L_a, l_b = 0,3L_b$$

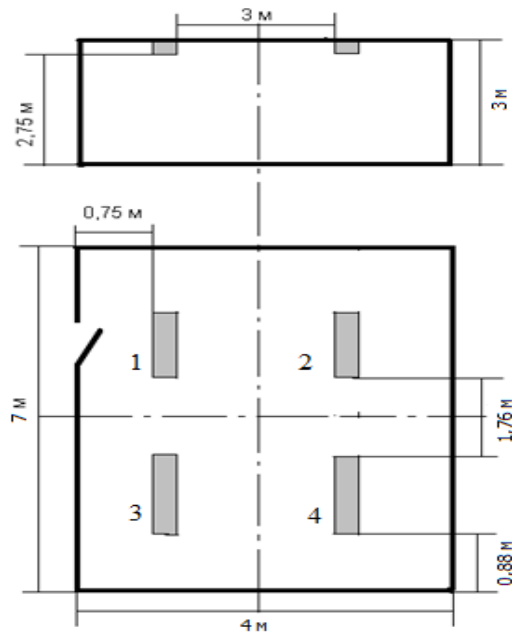
$$l_a = 0,88\text{ м}, l_b = 0,75\text{ м}$$

Люминисценттік шамдарды жұмыс бөлмесінде қатар қатар орналастырып қою кепілді.

Жарық ағының есептеу үшін коэффициентті қолдану әдісі, горизонтальды жазықтықтардың бетіндегі жалпы жарықтану біріңғай таралуы үшін қолданады. Шамдардың орналасу сұлбасы 3.2 суретте көрсетілген.

Жарықтануды меншікті қуат әдісі арқылы анықтаудың басты ерекшелігі мен маңыздылығы бар, шамның түріне байланысты орнатылуы, жұмыс жазықтығының үстінен шамның іліну биіктігі, сонымен қатар жұмыс бөлмесінің горизонталь жазықтығындағы жарықтанудың меншікті қуат мәнімен анықталуында.

Меншікті қуат деп орнатылған шамның қуатының жарықтандыруға қажетті ауданға қатынасын айтады ($\text{Вт}/\text{м}^2$), әр түрлі шамдар үшін қуаттың мәні арнайы анықтамалық кітаптарда беріледі.



Сурет 3.2 – Шамдардың орналасу сұлбасы

Жарықпен қамтамасыз еті үшін ЛДР типті шамдарды бір біріне паралель екі қатарға және әр қатарға екі шамнан қойып орнатамыз

$$N = 2 \cdot 2 = 4$$

СНиП 2.4–79 әдебиетіне байланысты шамның меншікті қуаты $\omega = 15 \text{ Вт/м}^2$
Бір шамның қуаты, Вт

$$\rho = \frac{\omega \cdot S}{N} \quad (3.4)$$

$$\rho = \frac{15 \cdot 30}{4} = 112$$

Әр шамда екі лампа бар, олардың әр қайсысының қуаты

$$\rho_s = \frac{112}{2} = 56 \text{ Вт}$$

3.6 Табиғи жарықты есептеу

Бөлменің ұзындығы $a = 7 \text{ м}$, ені $b = 4 \text{ м}$, биіктігі $h = 3 \text{ м}$. Жұмыс бетінің еден деңгейінен биіктігі

$$h_{pn}, h_{pn} = 0,7 \text{ м}$$

Терезелер биіктігі

$$h_{но}, h_{но} = 1,2 \text{ м}$$

$$h_o, h_o = 1.5 \text{ м}$$

Жұмыс орының бөлмесі IV сағат белдеуде орналасқан – Алматы қаласы. Көлеңке түсіретін ғимараттар жоқ.

Жұмыс орны бөлменің қабырғасынан 0,5м жерде орналасқан. Минималды жарықтандыру терезеден 5м қашықтықтағы нүктеде болады.

Терезенің толық ауданы S_0 , м^2 , осы формуламен анықтаймыз

$$100 \cdot \frac{S_0}{S_n} = \frac{e_n \cdot \eta_0}{\tau_0 \cdot r_1} \cdot k_{зд} \cdot k_3, \quad (3.5)$$

$$S_0 = \frac{S_n \cdot e_n \cdot \eta_0}{100 \cdot \tau_0 \cdot r_1} \cdot k_{зд} \cdot k_3 \quad (3.6)$$

мұндағы S_n – бөлменің ауданы м^2 ;

e_n – Ортақ жарықтандыру жарық көздері 3.5– кестесінен тандаймыз

$$e_n^{IV} = e_n \cdot m \cdot c$$

мұндағы $m = 0,8$;

$c = 0,75$ – IV сағаттық белдеу үшін

$$S_n = a \cdot b \quad (3.7)$$

$$S_n = a \cdot b = 7 \cdot 4 = 28 \text{ м}^2$$

К е с т е 3.5 – m коэффициентінің мәні

жарықтану белдеуінің климаты– IV 50^0 солтүстік ұзындығы және оңтүстік (Алматы, Караганда)	Ғимараттың сыртқы қабырғасынан	Тікбұрышты және трапециялық шамдарға	Зенитті шамдарға
	0,75	0,8	0,7

$e_n = 1,2$ жұмыстағы үлкен дәлдіктер III разрядтағы көрініс жұмысы

$$e_n^{IV} = 1,2 \cdot 0,8 \cdot 0,75 = 0,72$$

Бөлменің типін ескере отырып, осыдан коэффициенті табамыз $k_3 = 1,2$ (жұмыс істеу орны қарастырылады).

τ_0 – жарық өткізудің толық коэффициенті

$$\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4$$

$\tau_1 = 0,5$ (бос орынды шынылы блок);

$\tau_2 = 0,6$ (екі қабатты ағаштан жасалған тоқыма);

$\tau_3 = 0,8$ (темірбетонды аркалар);

$\tau_4 = 1$ (жиналатын жалюзилер және шторлар);

$$\tau_0 = 0,5 \cdot 0,6 \cdot 0,8 \cdot 1 = 0,24;$$

η_0 – терезенің жарықтану мінездемесі.

Анықтаймыз η_0

$$l = B - 1$$

$$l = 4 - 1 = 3 \text{ м}$$

$$\frac{L}{l} = \frac{L}{B-1} = \frac{7}{3} = 2,33$$

$$h_{\text{расч}} = h_{\text{но}} + h_o - h_{\text{рт}} \quad (3.8)$$

$$h_{\text{расч}} = 1,2 + 1,5 - 0,7 = 2$$

$$\frac{B}{h_{\text{расч}}} = \frac{4}{2} = 2$$

Бөлмедегі орташа шағылу коэффициент $\rho_{\text{СР}} = 0,5$, тек бір жағындағы жарықты қарастырамыз

$$\frac{l_{\text{рт}}}{B} = \frac{0,5}{4} = 0,125$$

$$r_1 = 1,05$$

$k_{зд}$ – іске асыру кезіндегі жарық көздерінің шаңдануы мен жарамсыз болуын ескеретін қор коэффициенті $k_{зд} = 1$.

Терезелердің толық ауданын есептейміз

$$S_0 = \frac{28 \cdot 1,35 \cdot 10 \cdot 1 \cdot 1,2}{100 \cdot 0,24 \cdot 1,05} = 18 \text{ м}^2$$

Жарық өтуінің ауданы тең

$$S_{сп} = 18 \text{ м}^2$$

Осыған орай жарық өту ауданы ($2,33 \cdot 2 \cdot 1,5 = 7 < 18$) жұмыс орындағы қажетті еңбек жағдайларын қамтамасыз етпейді. Осы мақсатта еңбектің ең қолайлы параметрі 7·4·3 қамтасамыз ететін, $e_N = 0,84$ көру жұмысының III мінездемесі, сонда жұмыс орнында табиғи жарықты және жасанды жарықты қолданады.

4 Техникалық экономикалық негізделуі

4.1 Жоба сипаттамасы

Ұсынылып отырған қызметтің сипаты мен артықшылықтары

Universal mobile app – бұл жоғарғы оқу орындарына арналып жасалынған мобильді қосымша. Қазіргі таңда смартфондар мен планшеттердің ауқымды түрде қолдануы жоғарғы және орта оқу орындарын жарнамалау және ондағы студенттер мен барлық қызметкерлерге тез, қолайлы түрде ақпарат таратудың бір түрі болып табылады. Осы факторларды ескере отырып оқу орындарын жарнамалауға және қолайлы түрде ақпаратпен қамтамасыз ететін Android операциялық жүйесіне арналған мобильді қосымша жасалынды. Мобильді қосымша алғашқы уақытта 5 жоғарғы оқу орнына сатылады деп жоспарланады, себебі мобильді қосымшаның тиімділігін және аз уақыт ішінде өзін-өзі ақтайтына байланысты сұранысқа ие болады деп ойластырылуда.

Тұтас сегмент мен позициялау

Тұтас сегмент – бұл қазіргі техникалық прогрестің мүмкіндігін толық пайдаланып жұмыс істеуді қажет ететін электроэнергетикалық компаниялар жүйесі.

Қызметті SWOT – талдау

4.1 кестеде SWOT – талдау көрсетілген.

Кесте 4.1 – Қызметті SWOT – талдау

Артықшылықтары	Кемшіліктері	Мүмкіншілігі	Қауіпі
Бұл қызмет түрі елдегі оқу орындарының ешқайсысында қолданылмауы.	Оқу орындарының шектеулі болуы	Қосымша табыс көзі	Бағдарлама рұқсатсыз көшірілуі мүмкін

Маркетинг – микс қызметтері

Маркетинг–микс маркетингтік құралдардың жиынтығы, бұл арқылы компания тауарлар немесе қызметтер сұранысына ықпал етуге тырысады.

Маркетинг – микс бұл белгілі бір тауарға анықталған баға бойынша белгілі бір орында қозғалыс арқылы сұранысты қалыптастыратын жиындардың (стратегия) жүйесі.

Айта кету керек, стратегия тек компанияның белгілі бір мақсатта нарықтың кез келген сегментінде қозғалуына арналған ұмтылысы және стратегия жалпы, бөлінбейтін сипатқа ие. Жағдайды басқару мен оны назардан тыс шығармау үшін тек нақты орындалу уақыты анықталған жиындарды іске асыру керек, осы жиындарға жауапты адамдарды, қажетті ресурстарды

(қаржылық, адами және т.б.) басқару қажетіске асыру қажет. Міне сондықтан таңдалған стратегия үшін маркетинг–микс деп аталатын арнайы жиындар іске асырылады. Осы маркетингтің тактикалық деңгейі болып табылады.

Ереже бойынша, стандартты маркетинг–микс жиындарды төрт құраушы тактикалық іс бойынша сипаттайды. Маркетинг жиынтығы (marketing–mix) – бұл фирма біртұтас нарықтың оң жауабын алу үшін қолданатын басқаруға мүмкіндік беретін маркетингтік факторлардың жиыны. Маркетинг–микстің міндеті – біртұтас нарықта потенциалды тұтынушылардың қажеттерін өтеумен ғана шектелмейтін, сонымен қоса ұйымның әсерін жоғарлатуға тырысатын жиынтықтарды (mix) құрастыру.

Продукт (Product) – бұл фирманың тұтас нарыққа ұсынатын «бұйымдар мен тауарлар» жиынтығы.

Баға (Price) – тұтынушылардың тауарды алу үшін төлеуі қажет ақша суммасы.

Дистрибуция (Place) – тауар барлық бірыңғай тұтынушыларға қолжетімді болу үшін жүзеге асырылатын мүмкіндігі көп іс.

Қозғалу, даму (Promotion) – фирманың өзінің тауарлары туралы ақпарат таратуы мен тұтынушыларды осы тауарларды сатып алуға үгіттейтін мүмкіндігі зор іс–шаралар.

4.2 кестеде маркетинг – микс элементтері көрсетілген.

К е с т е 4.2 – Маркетинг – микс элементтері

Маркетинг-микс элементтері	Қажетті істер мен талдаулар
Product	
Жоғары оқу орындарына арналған мобильді қосымша қызметі.	Жалпы тапсырманы шешу келесі бөлімдерден тұрады: 1) Пайдаланушыға қолайлы интерфейс жүйесін ұйымдастыру; 2) Пайдаланушы арнайы жасалған мобильді қосымшаға кіріп өзіне қажетті ақпаратты ала алады.
Price	
Базалық бағасы	Мекенің қажеттілік ерекшеліктерін ескере отырып ақысы алынады.
Promotion	
	Басында уақытша жарнамасыз жұмыс істеу, яғни тікелей сату

Маркетинг-микс элементтері	Қажетті істер мен талдаулар
Place	
Оқу орны 1, Оқу орны 2, Оқу орны 3, Оқу орны 4, Оқу орны 5.	Алматы қаласындағы жоғары оқу орындарының админстраторларын хабарландыру, келіссөз жүргізу.

4.2 Бағдарламалық қамтаманың өңделуінің еңбек өнімділігінің есептелуі

Еңбекке кеткен шығынның базалық көрсеткіштері мына формула бойынша есептеледі

$$Q = q \cdot c \quad (4.1)$$

мұндағы q – бағдарламалық өнімдегі (бастапқы команда) операторлар саны, менің жұмысымда операторлар саны 1700-ге тең, атап өткен q коэффициентінің мәнін кестеден таңдап алуға болады;

c – бағдарлама күрделілігінің коэффициенті, ол менде 1 ге тең, себебі менің жобамның бағдарламалық тілі жоғарғы тілге жатады және күрделілік тобы үшінші топ, ал жаңалық дәрежесі бойынша В тобына, яғни типтік шешім қолданған бағдарламаларды дайындау тобына жатқандықтан осы мәндердің қиылысуынан арнайы кестеден алынды; Осы әдіспен базалық көрсеткішті табамыз

$$Q = 1700 \cdot 1 = 1700$$

Ары қарай бағдарламалық өнімді әзірлеуге кететін уақытты есептеу керек. Уақыт адам-сағатпен есептеледі, ал T_d нақты істелген уақытпен алынады, ал қалған кезеңнің уақыты Q командасының шартты санына байланысты есептік жолмен анықталады,

Бағдарламалық өнімін дайындауға кеткен әр кезеңнің уақытын анықтаймыз: $T_{ПО}$ (мақсат сипатын дайындау уақыты), нақтылы деректер бойынша алынады және келесі мәнге тең деп алынады (3-тен 5 күнге дейін, 8 сағаттан)

$$T_{ПО} = 24 \text{ адам / сағ}$$

1) T_O (мақсат сипаттамасы уақыты) келесі формуламен анықталады

$$T_O = Q B / (50 K) \quad (4.2)$$

$$T_O = 1700 \cdot 1,5 / 50 \cdot 0,8 = 63,75 \text{ адам/сағ}$$

мұндағы B – мақсат есебі өзгерісінің коэффициенті;

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

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

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

$$T_A = Q / (50 K) \quad (4.3)$$

$$T_A = 1700 / (50 \cdot 0,8) = 42,5 \text{ адам/сағ}$$

3) T_{BC} (блок – сұлба құруға кеткен уақыт) T_A сияқты 4.3 формуламен есептеледі.

4) T_H (бағдарламаның тілінде жазуға кеткен уақыт) келесі формуламен анықталады

$$T_H = Q \cdot 1,5 / (50 K) \quad (4.4)$$

$$T_H = 1700 \cdot 1,5 / (50 \cdot 0,8) = 63,75 \text{ адам/сағ}$$

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

$$T_{II} = Q / 50 \quad (4.5)$$

$$T_{II} = 1700 / 50 = 34 \text{ адам/сағ}$$

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

$$T_{OT} = Q \cdot 4,2 / 50 K \quad (4.6)$$

$$T_{OT} = 1700 \cdot 4,2 / 50 \cdot 0,8 = 178,5 \text{ адам/сағ}$$

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

$$T_D = 24 \text{ адам / сағ}$$

Еңбек шығындарының сомасы еңбек шығынының құрама сомасы ретінде 4.7 формуламен есептеледі

$$T = T_{II} + T_{TO} + T_A + T_{BC} + T_H + T_{II} + T_{OT} + T_D \quad (4.7)$$

$$T = 24 + 63,75 + 42,5 + 42,5 + 63,75 + 34 + 178,5 + 24 = 473 \text{ адам/сағ}$$

4.3 Бағдарламалық қамтаманы өңдеуге кеткен шығынның есептелуі

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

Негізгі еңбек құны

$$Z_{\text{осн}} = t_{\Sigma} \cdot TC / (t_{\text{cp}} \cdot 8) \quad (4.8)$$

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

t_{cp} – айдағы орташа күн саны, 21 күнге тең, жұмыс күніндегі 8 сағатқа көбейтіледі. Тарифтік мөлшерлеме минималды өлшемдегі еңбек ақы 01.01.2014ж бастап ҚР–да 19966 теңгені құрады;

TC – тарифтік қойылым = $19966/8 = 2495,5$ теңгеге тең болды;

t_{Σ} – (4.7) формуласы бойынша 473–ге тең.

Осылайша негізгі еңбек ақы

$$Z_{\text{осн}} = 473 \cdot 2495,75 / 21 = 56213,8 \text{ теңге}$$

Қосымша еңбек ақы негізгі еңбек ақының 20%–ын құрайды. (4.9) формуласы бойынша анықталады

$$Z_{\text{доп}} = 0,2 \cdot Z_{\text{осн}} \quad (4.9)$$

$$Z_{\text{доп}} = 0,2 \cdot 56213,8 = 1242,8 \text{ теңге}$$

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

$$\text{ФОТ} = Z_{\text{доп}} + Z_{\text{осн}} \quad (4.10)$$

4.4 Қосымша шығындар статьясы

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

Қондырғының құны бағдарламалық қамтама өңделуінің өзіндік құнына кірмегенімен, қосымша шығындар статьясындағы есептеулерде қолданылады. ЭЕМ – на бағдарлама жазу кезінде құрылғы ретінде дербес компьютер қолданылады. Құны $C_{\text{обор}} = 325000$ теңге (дербес компьютермен қосымша жабдықтардың қазіргі уақыттағы нарықтағы теңгемен берілген құны алынды).

Материалдар мен көмекші бөлшектер шығыны, бағдарламалық өнімді жазу барысын

$$C_{\text{МжК}} = 0,015C_{\text{обор}} \quad (4.11)$$

$$C_{\text{МжК}} = 0,015 \cdot 325000 = 4875 \text{ теңге}$$

$$C_{\text{ТО}} = 0,025 C_{\text{обор}} \quad (4.12)$$

$$C_{\text{ТО}} = 0,025 \cdot 194900 = 8125 \text{ теңге}$$

Амортизациялық аударым өнімнің физикалық және моральдық жарамсыздығына байланысты сол өнімді толық қалпына келтіру үшін жиналатын ақша қоры болып есептеледі. Амортизациялық аударым амортизация нормасы бойынша орнатылады, (4.13) формуласы бойынша есептеледі

$$A = \frac{C_{\text{обор}} \cdot H_A \cdot N}{100 \cdot 12 \cdot t} \quad (4.13)$$

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

H_A – амортизация нормасы;

N – жұмыс орындалуына кеткен күннің саны;

t – дербес компьютерді қолдануға кеткен жалпы уақыт

$$N = T/24 = 473/24 = 19,71 \text{ күн}$$

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

Дербес компьютерде жалпы жұмыс істеу уақыты мына формуламен есептеледі

$$t = T_A + T_{\text{БС}} + T_H + T_{\text{П}} + T_{\text{ОТ}} \quad (4.14)$$

$$t = T_A + T_{\text{БС}} + T_H + T_{\text{П}} + T_{\text{ОТ}} = 42,5 + 42,5 + 63,75 + 34 + 178,5 = 361,25 \text{ адам/ сағ}$$

Амортизация нормасы (4.15) формуласы бойынша анықталады

$$H_A = \frac{C_{\text{обор}} - C_{\text{ликв}}}{T_{\text{норм}} \cdot C_{\text{обор}}} \cdot 100\% \quad (4.15)$$

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

$$C_{\text{ликв}} = 0,05 \cdot C_{\text{обор}} \quad (4.16)$$

$$C_{\text{ликв}} = 0,05 \cdot 325000 = 16250 \text{ теңге}$$

$T_{норм}$ – нормативті қызмет ету өтілі (дербескомпьютер үшін ҚР да $T_{норм}=4жыл$) осыдан

$$H_A = \frac{C_{обор} - C_{ликв}}{T_{норм} \cdot C_{обор}} \cdot 100\% \quad (4.17)$$

$$H_A = \frac{325000 - 16250}{4 \cdot 325000} \cdot 100\% = 23,75\%,$$

$$A = 325000 \cdot 23,75 \cdot 19,71 / 10012 \cdot 361,25 = 350,94 \text{ теңге}$$

Бағдарламалық камтаманы жазу кезіндегі электроэнергияның шығынын қоса есептеу керек. Электроэнергияның құны мына формула бойынша есептеледі

$$C_{ЭЭ} = M \cdot k_3 \cdot F_{эф} \cdot C_{кВт-ч} \quad (4.18)$$

мұндағы M – ЭВМ қуаты (450 Вт);

k_3 – жүктеу коэффициенті (0.8);

$C_{кВт.ч}$ – 1 кВт-сағ электроэнергияның құны, оның мәні ҚР – да 14,65 теңгеге тең;

$F_{эф}$ – жұмыс уақытының эффективті фонды, мына формула бойынша есептеледі

$$F_{эф} = D_{ном} \cdot d \cdot (1 - f / 100) \quad (4.19)$$

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

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

$f = 2,6\%$ – ЭВМ – ды жөндеуге кеткен жоспарланған уақыт.

$$F_{эф} = 258 \cdot 8 \cdot (1 - 2,6 / 100) = 2010$$

Берілген коэффициенттер мен параметрлер бойынша электроэнергияның құны

$$C_{ЭЭ} = 0,45 \cdot 0,8 \cdot 2010 \cdot 14,65 = 10601 \text{ теңге}$$

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

$$\Delta_3 = t_{\Sigma} \cdot C_3 / F_{эф}, \quad (4.20)$$

$$C_3 = C_{ЭЭ} + C_{ТО} + A_{жыл} \quad (4.21)$$

$$C_{\Sigma} = C_{\Sigma\Sigma} + C_{TO} + A_{\text{жыл}} = 10601 + 8125 + 350,94 = 19076,94 \text{ теңге}$$

мұндағы $F_{\Sigma\phi}$ – жұмыс уақытының эффективті фонды;

$t_{\Sigma'}$ – ЭВМ–нің есепті шешу үшін жалпы қолданылған уақыты, бірақ бұл жерде тек компьютердегі жұмыс уақытын ескере отырып анықталған

$$t_{\Sigma'} = t_{\text{пр}} + t_{\text{отл}} + t_{\text{д}}, \quad (4.22)$$

$$t_{\Sigma'} = t_{\text{пр}} + t_{\text{отл}} + t_{\text{д}} = 42,5 + 178,5 + 24 = 245 \text{ сағ}$$

Сәйкесінше, ЭЕМ–ның жалпы жылдық эксплуатационды шығын құны

$$\Sigma_3 = 245 \cdot 19076,94 / 2010 = 2325,3 \text{ теңге}$$

Уақыт коэффициенті мына формула бойынша есептеледі

$$w = \frac{\Sigma_3}{C_{\Sigma}} \quad (4.23)$$

$$w = \frac{2325,3}{19076,94} = 0,12$$

Осылайша, жалпы эксплуатационды шығынды уақыт коэффициентін ескеріп коррекциялаймыз (дұрыстаймыз).

Электроэнергияға шығын

$$C_{\Sigma\Sigma}^* = C_{\Sigma\Sigma} \cdot w \quad (4.24)$$

$$C_{\Sigma\Sigma}^* = 10601 \cdot 0,12 = 1292,2 \text{ теңге}$$

Амортизационды аударымы

$$A_{\text{жыл}}^* = A_{\text{жыл}} \cdot w \quad (4.25)$$

$$A_{\text{жыл}}^* = 350,94 \cdot 0,12 = 42,77 \text{ теңге}$$

Сонымен қатар, (4.26) формуласы бойынша есептелетін жұмыс жалақысына тәуелді шығындар бар. Осындай шығындар қатарына социалды сақтандыруға кететін аударымдар мен жүктеме аударымдар кіреді.

Әлеуметтік салыққа кететін аударымдар бүкіл жалақының 11% құрайды [11%, ҚР ның Салық Кодексі], (4.26) формуласы бойынша анықталады

$$CC = (\text{ФОТ} - 0,1 \cdot \text{ФОТ}) \cdot 0,11 \quad (4.26)$$

$$CC = (67456,6 - 0,1 \cdot 67456,6) \cdot 0,11 = 6678,2 \text{ теңге}$$

Жүктеме шығындар қызмет көрсету мен басқаруға қатысты, құрылғының эксплуатациясы мен басқа да өндіріс процесін қамтамасыз ететін қосымша шығындар болып табылады, еңбек жалақысы фондының 50% құрайды, (4.27) формуласы бойынша анықталады

$$C_{\text{накл}} = 0,5 \cdot \text{ФОТ} \quad (4.27)$$

$$C_{\text{накл}} = 0,5 \cdot \text{ФОТ} = 0,5 \cdot 67456,6 = 33728,3 \text{ теңге}$$

4.5 Өзіндік құн нәтижесінің кестесі

Бағдарламалық қамтаманы өңдеуге кеткен жалпы шығын еңбек жалақысы фондының, эксплуатационды шығынның, социалды сақтандыру, жүктеме шығындар мен материал және құрамалардың қосындысына тең болады. Бағдарламалық қамтаманы өңдеудің қорытынды құны 4.3 кестеде көрсетілген.

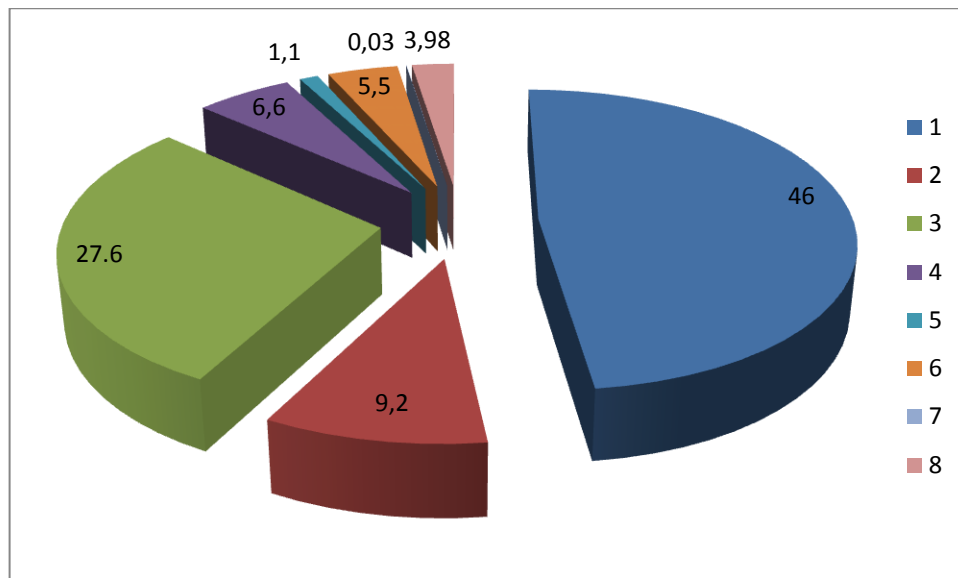
К е с т е 4.3 – Өзіндік құн нәтижесінің кестесі

Шығын статьялары		Сумма, теңге	Жалпы суммадан процентпен
ФЗП	$Z_{\text{осн}}$	56213,8	46
	$Z_{\text{доп}}$	11242,8	9,2
Жүктеме шығындар, $C_{\text{накл}}$		33728,3	27,6
Әлеуметтік сақтандыру, CC		6678,2	5,46
Эксплуатационды Шығындар	$C_{\text{ээ}}^*$	1292,2	1,05
	$C_{\text{то}}$	8125	6,65
Материалдар мен комплектілер, $C_{\text{мик}}$		4875	3,98

К е с т е 4.3 соңы

Шығын статьялары		Сумма, теңге	Жалпы суммадан процентпен
Барлығы		122198	100

4.1 суретте өнімнің өзіндік құны көрсетілген.



Сурет 4.1 – Өнімнің өзіндік құнының диаграммасы

4.6 Бағдарламаның бағасын есептеу

Өнімнің өзіндік құны 122198 теңге болды, онда бір енгізу объектісіне арналған жүйенің толық өзіндік құны

$$C_{\text{пс}} = 122198 / 5 = 24439,6 \text{ теңге}$$

4.7 Минималды бағаны, кірісті және пайданы есептеу

Өнімнің минималды бағасы келесі формуламен есептеледі

$$Ц_{\text{min}} = (C_{\text{пс}} + Ц_{\text{мд}} + Ц_{\text{з}})(1 + r/100) \quad (4.28)$$

$$Ц_{\text{min}} = (24439,6 + 190 + 140) \cdot 1,21 = 29971,2 \text{ теңге}$$

мұндағы $Ц_{\text{мд}}$ – мғниттік жазба бағасы

$Ц_{\text{з}}$ – жазба бағасы;

r – Өзіндік құнға қатынасымен алынған тиімділіктің жоспарланған пайызы, $r = 21\%$.

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

$$Ц_{\text{min}_{\text{НДС}}} = Ц_{\text{min}} + Н_{\text{ндс}} \cdot Ц_{\text{min}} \quad (4.29)$$

$$Ц_{\text{min}_{\text{НДС}}} = 29971,2 + 0,12 \cdot 29971,2 = 33567,74 \text{ теңге}$$

Кіріс келесі формуламен есептеледі

$$V_{\text{реал}} = Ц_{\text{min}}_{\text{НДС}} \cdot V_{\text{объем}} \quad (4.30)$$

$$V_{\text{реал}} = 33567,74 \cdot 5 = 167838,76 \text{ теңге}$$

мұндағы $V_{\text{объем}}$ – бағдарламаны жүзеге асыру көлемі.
Жалпы пайда келесі формуламен есептеледі

$$\Pi = V_{\text{реал}} - V_{\text{реал}} \cdot Н_{\text{ндс}} - C_{\text{разраб}} \quad (4.31)$$

$$\Pi = 167838,76 - 20140,64 - 122198 = 25500,12 \text{ теңге}$$

мұндағы НДС – қосылған құн салығы (12%).
Таза пайда келесі формуламен есептеледі

$$\Pi_{\text{чист}} = \Pi - Н_{\text{прибыль}} \quad (4.32)$$

$$\Pi_{\text{чист}} = 25500,12 - 0,2 \cdot 25500,12 = 20400,1 \text{ теңге}$$

мұндағы $Н_{\text{прибыль}}$ – пайдаға салынатын салық (ҚР да $Н_{\text{прибыль}} = 20\%$)

4.8 Бағдарламалық өнімді иемденудегі кәсіпорындардың бірізгідегі шығындарын есептеу

Кәсіпорындардың бірізгідегі шығындары келесі бөлімдерден тұрады:

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

Жүйе құны 33567,74 теңге (4.28 формуласы). Көлік шығындары жүйе құнының 20% құрайды

$$A_{\text{кш}} = Ц_{\text{min}}_{\text{НДС}} \cdot 0,2 \quad (4.33)$$

$$A_{\text{кш}} = Ц_{\text{min}}_{\text{НДС}} \cdot 0,2 = 33567,74 \cdot 0,2 = 6713,55 \text{ теңге}$$

Жұмысшыларды жүйемен жұмыс істеуге оқыту шамамен екі сағатта консалтингті фирманың маманы көмегімен сағатына 2500–3000 теңгедей болып табылады.

Соған сәйкес оқу бағасы: $t_0 = 2$ сағ

$$A_{\text{жо}} = 2500 \cdot t_0 = 2500 \cdot 2 = 5000 \text{ теңге}$$

Кесте 4.4 де кәсіпорынның бірізгідегі шығындарын есептеулерінің нәтижесі көрсетілген.

Кесте 4.4 – Ақпаратты жүйені енгізуге арналған бірмезгілдегі шығындардың есептелу нәтижесі

Шығындар түрлері	Сумма, теңге
Жүйе құны	33567,74
Көлік шығындары	6713,55
Жұмысшыларды оқыту бағасы	5000
Қорытынды:	45281,28

4.9 Қолдану аясындағы жылдық бірмезгілдегі шығындарды есептеу

Негізгі шығындар түрлері:

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

Есептеу жылына бір рет ПК операторымен 2500 теңге/сағ (жоғарыда берілген) сағаттық мөлшерде бір сағат көлемінде жүргізіледі

$$2500 \cdot 1 = 2500 \text{ теңге}$$

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

$$(2500 - 2500 \cdot 0,1) \cdot 0,11 = 247,5 \text{ теңге}$$

Электрэнергияға кететін шығындар

$$0,8 \cdot 14,65 = 11,72 \text{ теңге}$$

мұндағы 0,8 – жұмсалған қуат мөлшері, кВт–сағ;

14,65 – ҚР 1 кВт–с электрэнергияның орташа құны, теңге.

Есептеулерді 4.5 кестеге енгіземіз.

Кесте 4.5 – Кәсіпорынның жылдық бірмезгілдегі шығындарын есептеу нәтижесі

Шығындар түрлері	Сумма, теңге
Еңбекке төленетін қор	2500
Әлеуметтік салыққа аударымдар	247,5
Электрэнергияға төлем	11,72
Қорытынды:	2759,2

4.10 Ақпараттық өнімді енгізуден алынған табысты есептеу

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

Тапсырыс беруші–кәсіпорын ақпараттық өнімді сатып алғаннан кейін оны қолдану уақытында пайда табады.

Кіріс көзі ретінде ақпараттық өнімде ақылы сервистерді және жарнамаларды қою арқылы пайда табады. Жарнамаларды қою ақпараттық өнімді танымалдығына байланысты болады. Ақпараттық өнімді қолданушылар аудиториясына байланысты болады. Ақпараттық өнімде орнатылған жарнама баннерін әрбір 100–300 көрсетілім де қолданушы бір рет басқанда 18,2 теңге мен 91 теңге аралығында пайда түседі.

Айына орташа есеппен ақпараттық өнім 50000–100000 рет баннерде көрсетілім жасай алады. Сонда 4550 теңге айына пайда түседі.

Статистика googlemarket сервисінен алынды

$$\Phi = 18,2 \cdot (50000/200) \cdot 12 = 54600 \text{ теңге,}$$

$$\Delta \mathcal{E}_k = 54600 - 2759,2 = 51840,8 \text{ теңге}$$

4.11 Пайда мен шығындарды есептеу

Пайда мен шығындарды есептеу 4.6 кестеде көрсетілген.

К е с т е 4.6 – Бір жылдағы пайда мен шығындарды есептеу

Аты	Жыл
Ақпараттық өнімнен түсетін пайда, теңге	51840,8
Ағымдағы шығындар, теңге	2759,2
Шартты пайда, теңге	25500,12
Пайдаға салынатын салық, теңге	5100
Таза пайда, теңге	20400,1

4.12 Ақшалай құралдардың қозғалысы

Төмендегі 4.7 кестеде ақшалай тәсілдерінің қозғалысы берілген.

Кесте 4.7 – Ақшалай тәсілдердің қозғалысы, теңге

Аты	2014	2015	2016
Бірмезгілдік шығындар кВ мың, теңге	45281,28		

Жеке кәсіпкерлік пайда		20400,1	20400,1
Ақпараттық өнімнен түсетін пайда		51840,8	51840,8
Ағымдағы шығындар		2759,2	2759,2
Пайдаға салынатын салық		10368,16	10368,16
Таза табыс	-45281,28	41482,64	41482,64
Дисконттау коэффициенті (20% мөлшерінде)	1	0,833	0,694
Таза дисконтталған табыс (ТДТ)	-45281,28	34546,7	28788,95
ТДТ өспелі нәтижесімен	-45281,28	-10734,58	18054,37

$$K_{д2} = 1/(1+\alpha)^1 = 1/(1+0,2)^1 = 0.833,$$

$$K_{д3} = 1/(1+\alpha)^2 = 1/(1+0,2)^2 = 0.694$$

мұндағы α – инфляция коэффициенті;

K_d – дисконттау коэффициенті.

Таза дисконтталған табыс (ТДТ) = Таза табыс Дисконттау коэффициенті.

Таза дисконтталған табыс

$$(ТДТ) = 41482,64 \cdot 0.833 = 34546,7 \text{ теңге,}$$

$$41482,64 \cdot 0,694 = 28788,95 \text{ теңге}$$

$$ТДТ_2 \text{ (өспелі нәтижесімен)} = ТДТ_1 + \text{Таза дисконтталған табыс (ТДТ)}_2$$

4.13 Экономикалық тиімділікті есептеу

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

Шығындары бірізгілікте тек жобаның басында күрделі салымдар (C_0) ретінде іске асатын жобалар үшін NPV келесі формуламен есептеледі

$$NPV = \sum_{i=1}^n \frac{B_i}{1+r} - C_0$$

мұндағы B_i – i -ші жылдағы жобадан алынатын пайда;

r – дисконттеу мөлшері

$$NPV(20\%) = \left(\frac{41482,64}{(1+0,2)^1} + \frac{41482,64}{(1+0,2)^2} \right) - 45281,28 = 18054,37 \text{ теңге} > 0$$

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

4.13.2 Пайда индексін есептеу (Profitability index, PI)

PI жобаның салыстырмалы пайдасын көрсетеді. Бұл көрсеткіш келесі формуламен есептеледі

$$PI = \frac{NPV}{C_0}$$

$$PI = (18054,37/122198) \cdot 100\% = 0,147 \cdot 100\% = 14,7\%$$

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

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

$$IRR = r$$

мұндағы $NPV = 0$

$$IRR = r_1 + \frac{f(r_1)}{f(r_1) - f(r_2)} \cdot (r_2 - r_1)$$

$$r_b = 25\%$$

$$PV_1 = 41482,64 / (1 + 0,25) = 33186,11 \text{ теңге,}$$

$$PV_2 = 41482,64 / (1 + 0,25)^2 = 26548,88 \text{ теңге,}$$

$$NPV(25,0\%) = (33186,11 + 26548,88) - 45281,28 = 14453,71 \text{ теңге,}$$

$$NPV(20\%) = 18054,37 \text{ теңге,}$$

$$NPV(25,0\%) = 14453,71 \text{ теңге,}$$

$$IRR = 0,2 + (18054,37 / (18054,37 - 14453,71)) \cdot (0,25 - 0,2) = 45\%$$

4.13.4 Өтімділік периодын есептеу (Payback period, PBP)

Өтімділік периоды мына формуламен есептеледі

$$T_{\text{ок}} = t + \frac{C_0 - \sum_{i=1}^n B}{B_{i+1}}$$

мұндағы C_0 – жоба басындағы салым;

B_i – i -ші жылдағы жобадан алынатын пайда.

Сонда біздің өтімділік периодымыз

$$T_{\text{ок}} = 2 + \frac{45281,28 - 34546,7}{28788,95} = 2,37 = 2 \text{ жыл } 4 \text{ ай}$$

Бұл жобаның өтімділік периоды 2 жыл 4 айға тең, яғни жобамыз осы уақытта өз – өзін ақтайды.

Қорытынды

Дипломдық жұмысты орындау барысында, заманауи мобильді технологиялардың адам өмірінің әртүрі салаларына қарқынды түрде еніп жатқаны және мобильді қосымшаларды қолдану айтарлықтай тиімді екені белгілі болды. Қазіргі таңда мобильді телефондар мен планшеттерді өндіруші компаниялардың басым бөлігі Android операциялық жүйесіне арнап өндіретіндігі, тұтыну бағасы қолжетімді және тұтынушыларға пайдалану қолайлы екендігі анықталды. Осыған орай студенттер мен талапкерлерге ақпаратты қолайлы және қолжетімді түрде беретін Алматы энергетика және байланыс университетіне арналған мобильді қосымша құру туралы шешім қабылданды.

Алматы энергетика және байланыс университетіне арналған мобильді қосымшаны жасауға қажетті технологияларға талдау жасалды. Жұмысты орындауға JavaEclipse IDE бағдарламалау ортасының негізі, оның негізгі мақсаты мен мобильді қосымшалар жасау мүмкіншіліктері қолданылды. Сонымен қатар Java бағдарламалау тілі, ондағы класстар, қасиеттер ашылып жазылды. Android SDK құралы көмегімен мобильді қосымшалар жасау және оның жұмыс істеу принциптері зерттелді. Қосымшаны жүзеге асыру кезінде Git файл нұсқаларын басқарудың таратылған жүйесін қолдану, оны GitHub жұмыстар хостингы веб сервисіне жариялау қарастырылды. Деректер қорымен жұмыс жасау ондағы мәліметтерге қол жеткізу және өңдеу үшін сонымен қатар сервермен байланыс ретінде PHP бағдарламау тілі қолданылды.

.Мобильді қосымшаның интерфейсі құрастырылды. Қосымшаның Java Eclipse IDE платформасында жүзеге асырылу қадамдары сипатталған.

Қойылған мәселелердің толық шешімін бағалау. Алға қойылған мақсат орындалды.

Жұмыстың нәтижелерін нақты қолдану бойынша ұсыныстар мен шығыс мәліметтері зерттелді. Жұмыс бойынша Алматы энергетика және байланыс университетіне арналған мобильді қосымшасы құрылды.

Дипломдық жұмыс нәтижесінде құралған бағдарламаның негізгі жетістігі мобильді қосымшалардың маңызы артып отырғаны, қолданушыларға анықтамалық түрде ақпараттарды ұсыну.

Пайдаланылған әдебиеттер тізімі

- 1 Сайт URL: <http://www.canalys.com/pr/2011/r2011013.html>.
- 2 Сайт URL: <http://developer.android.com/sdk/index.html>.
- 3 Сайт URL: http://www.mutualmobile.com/wpcontent/uploads/2011/03/MM_Android_Design_Guidelines.pdf.
- 4 Burnette E. Hello, Android. Introducing Google's Mobile Development Platform. – Pragmatic Programmers, LLC, 2008.
- 5 Сайт URL: <http://mobile.tutsplus.com/series/learn-java-android/>.
- 6 DiMarzio J. Android, A. Programmers Guide. – The McGraw–Hill, 2008.
- 7 Сайт URL: <https://Eclipse IDE.org>.
- 8 Сайт URL: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

- 9 Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. 2–е изд. – М.: ДМК Пресс, 2007.
- 10 Голощапов А.Л. Google Android: Программирование для мобильных устройств. – СПб.: БХВ–Петербург, 2011.
- 11 Хашими С., Коматинени С., Маклин Д. Разработка приложений для Android. – Питер, 2011.
- 12 Герберт Шилдт. Java: полное руководство = 8-е изд. The Complete Reference.- М.: «Вильямс», 2012.
- 13 Android 2: программирование приложений для планшетных компьютеров и смартфонов:[пер. С англ]/Рето Майер.-М.Эксмо 2011.
- 14 Харди Б.,Филлипс Б.Программирование под Android:Для профессионалов-СПб.:Питер, 2014.
- 15 Эккель Б. Философия Java.Библиотека программиста.4-е изд. - СПб.:Питер, 2009.
- 16 Сайт <http://developer.android.com/design/index.html>.
- 17 Рик Роджерс, Медниекс Зигурд и др. «Android. Разработка приложений» 2010г.
- 18 Shane Conder, Lauren Darcey «Android Wireless Application Development» Sept. 7, 2009.
- 19 Dave MacLean и др. «Pro Android 2» Paperback – March 15, 2010).

Қосымша А

```

<?php
defined('AUES') or die('Access denied');
function get_title_news(){
$query='SELECT news_id,title,anons,img from news limit 4';
$res = mysql_query($query);
    $news = array();
    while($row = mysql_fetch_assoc($res)){
        $news[] = $row;
    }
    return $news;
}
function get_text_news($news_id){
$query="SELECT news_id,title,text from news where news_id=$news_id";
$res = mysql_query($query);
$news_text = array();
$news_text = mysql_fetch_assoc($res);

```

```

    return $news_text;
}
function get_social(){
$query="SELECT social_id,name,text,icon,ssylka from social";
$res=mysql_query($query);
$social=array();
while($row=mysql_fetch_assoc($res)){
    $social[]=$row;
}
return $social;
}
function get_info(){
$query="SELECT info_id,title from info";
$res=mysql_query($query);
$info=array();
while($row=mysql_fetch_assoc($res)){
    $info[]=$row;
}
return $info;
}
function get_info_text($info_id){
$query="SELECT info_id,title,text from info where info_id=$info_id";
$res=mysql_query($query);
$info_text=mysql_fetch_assoc($res);
return $info_text;
}
function get_multimedia(){
$query="SELECT m_id,title,text,img from video";
$res=mysql_query($query);
$multimedia=array();
while($row=mysql_fetch_array($res)){
    $multimedia[]=$row;
}
return $multimedia;
}
function get_video($m_id){
$query="SELECT date,title,text,url from video where m_id=$m_id";
$res=mysql_query($query);
$video=mysql_fetch_assoc($res);
return $video;
}
function get_dining(){
$query="SELECT d_id,title,time from dinig";
$res=mysql_query($query);
$dining=array();
while($row=mysql_fetch_array($res)){
    $dining[]=$row;
}
return $dining;
}
function get_list_cat(){
$query="SELECT mc_id,name from menu_cat";
$res=mysql_query($query);
$fdining=array();

```

А қосымшаның жалғасы

```

}
return $multimedia;
}
function get_video($m_id){
$query="SELECT date,title,text,url from video where m_id=$m_id";
$res=mysql_query($query);
$video=mysql_fetch_assoc($res);
return $video;
}
function get_dining(){
$query="SELECT d_id,title,time from dinig";
$res=mysql_query($query);
$dining=array();
while($row=mysql_fetch_array($res)){
    $dining[]=$row;
}
return $dining;
}
function get_list_cat(){
$query="SELECT mc_id,name from menu_cat";
$res=mysql_query($query);
$fdining=array();

```

```

while($row=mysql_fetch_array($res)){
    $fdining[]=$row;
}
return $fdining;
}
function get_menu(){
    $date=date("w");
    $query="SELECT menu_cat_view.name as name,price,img from menu_cat_view,menu_day
where menu_cat_view.mcv_id=menu_day.mcv_id and menu_day.day_id=$date and
menu_cat_view.fmc_id=1";
    $res=mysql_query($query);
    $getmenu=array();
    while($row=mysql_fetch_array($res)){
        $getmenu[]=$row;
    }
    return $getmenu;
}
function get_hostel_inf(){
    $query="SELECT host_id,name,address from hostel";
    $res=mysql_query($query);
    $hostel=array();
    while($row=mysql_fetch_array($res)){
        $hostel[]=$row;
    }
    return $hostel;
}
}

```

А қосымшаның жалғасы

```

function get_hosteldetail($host_id){
    $query="SELECT host_id,name,address,facultet,phone,etaj,img,rules from hostel where
host_id=$host_id";
    $res = mysql_query($query);
    $hosteldetail = array();
    $hosteldetail = mysql_fetch_assoc($res);
    return $hosteldetail;
}
<?php
defined('AUES') or die('Access denied');
session_start();
require_once MODEL;
require_once 'functions/functions.php';
$news = get_title_news();
$view = $_GET['view'];
switch($view)
{
case('news'):
    $news = get_title_news();
    break;
case('fnews'):
    $news_id = abs((int)$_GET['news_id']);
    $news_text = get_text_news($news_id);
    break;
case('alert'):
    break;
case('social'):
    $social=get_social();
    break;
}
}

```

```

case('info'):
    $info=get_info();
break;
case('infotext'):
    $info_id=abs((int)$_GET['info_id']);
    $info_text=get_info_text($info_id);
break;
case('multimedia'):
    $multimedia=get_multimedia();
break;
case('video'):
    $m_id=abs((int)$_GET['m_id']);
    $video=get_video($m_id);
break;
case('dining'):
    $dining=get_dining();
break;
case('fdining'):
    $mc_id=abs((int)$_GET['mc_id']);
    $fdining=get_list_cat();
    $getmenu=get_menu();

```

А қосымшаның жалғасы

```

break;
case('hostel'):
    $hostel=get_hostel_inf();
break;
case('hosteldetail'):
    $host_id=abs((int)$_GET['host_id']);
    $hosteldetail=get_hosteldetail($host_id);
break;
case('abiturient'):
break;
case('student'):
break;
case('admin'):
break;
case("")
default:
    $view='index';
}
require_once TEMPLATE.'index.php';

```

```

<?php defined('AUES') or die('Access denied'); ?>
<?php if(!$view):?>
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<link rel="stylesheet" type="text/css" href="<?=TEMPLATE?>css/style.css" />
<script type="text/javascript" src="<?=TEMPLATE?>js/functions.js"></script>
<title>Almaty University</title>
</head>
<body>
<div class="karkas">
<div class="head">
    

```

```

</div>
<div class="menu">
<div class="nonfocal" id="searchformcontainer">
<form method="get" action="/home/search"><fieldset class="inputcombo emphasized">
<div class="searchwrapper">
<input class="forminput" type="text" id="filter" name="filter" placeholder="Search Almaty University"
title="Search Harvard Mobile Web" value="" onfocus="androidPlaceholderFix(this);">
</div>
<input class="combobutton" id="sch_btn" src="<?=TEMPLATE?>images/search-button.png"
type="image" alt="Search" title="Search">
<input type="hidden" name="_b" value="[{"&quot;t&quot;:&quot; Home&
quot;,&quot;lt&quot;:&quot;Home&quot;,&quot;p&quot;:&quot;index&quot;,&quot;a&quot;:&quot;:&quot;}&
quot;]">
</fieldset> </form> </div> </div>
<div class="content-main">
<div class="slogan-main">
<div class="slogan">
<div>
<a href="?view=news"></a>
<a href="?view=news"><h2>Новости</h2></a>
</div>
</div>
<div class="slogan">
<div>
<a href="#"></a>
<a href="#"><h2>Библиотека</h2></a>
</div>
</div>
<div class="slogan">
<div>
<a href="#"></a>
<a href="#"><h2>Специальности</h2></a>
</div>
</div><div class="slogan">
<div>
<a href="#"></a>
<a href="#"><h2>Спорт</h2></a>
</div>
</div><div class="slogan">
<div>
<a href="#"></a>
<a href="#"><h2>Фото</h2></a>
</div>
</div>
<div class="slogan">
<div>
<a href="?view=multimedia"></a>
<a href="?view=multimedia"><h2>Мультимедиа</h2></a>
</div>
</div>

```

А қосымшаның жалғасы


```

    <div class="slogan">
        <div>
            <a href="?view=dining"></a>
            <a href="?view=dining"><h2>Столовые</h2></a>
        </div>
    </div>

    <div class="slogan">
        <div>
            <a href="?view=hostel"></a>
            <a href="?view=hostel"><h2>Общежитие</h2></a>
        </div>
    </div>
    <div class="slogan">
        <div>
            <a href="?view=student"></a>
            <a href="?view=student"><h2>Студенту</h2></a>
        </div>
    </div>

    <div class="slogan">
        <div>
            <a href="?view=abiturient"></a>
            <a href="?view=abiturient"><h2>Абитуриенту</h2></a>
        </div>
    </div>

    <div class="slogan">
        <div>
            <a href="?view=admin"></a>
            <a href="?view=admin"><h2>Администрации</h2></a>
        </div>
    </div>

    <div class="slogan">
        <div>
            <a href="?view=map"></a>
            <a href="?view=map"><h2>Карта</h2></a>
        </div>
    </div>

        А қосымшаның жалғасы

    <div class="slogan">
        <div>
            <a href="?view=info"></a>
            <a href="?view=info"><h2>Инфо АУЭС</h2></a>
        </div>
    </div>
    <div class="slogan">

```

```

        <div>
            <a href="?view=social"></a>
            <a href="?view=social"><h2>Social</h2></a>
        </div>
    </div>
    <div class="slogan">
        <div>
            <a href="?view=alert"></a>
            <a href="?view=alert"><h2>Оповещение</h2></a>
        </div>
    </div></div>
</div>
    <div class="footer">
        <div class="footer-main">
            <div class="footer-f">
                <div>
                    <a href="#"></a>
                    <a href="#"><h2>Факультеты</h2></a>
                </div>
            </div>
            <div class="footer-f">
                <div>
                    <a href="#"></a>
                    <a href="#"><h2>Полный сайт</h2></a>
                </div>
            </div>
            <div class="footer-f">
                <div>
                    <a href="#"></a>
                    <a href="#"><h2>Настройка</h2></a>
                </div>
            </div>
            <div class="footer-f">
                <div>
                    А қосымшаның жалғасы
                    <a href="#"></a>
                    <a href="#"><h2>О нас</h2></a>
                </div>
            </div>
        </div>
    </div>
    <div class="footer-copy">
        <p>&copy; 2014 Almaty University of Power Engineering & Telecommunications</p>
    </div>
</div>
</body>
</html>

```

```

<?php else:?>
<?php include $view. '.php' ?>
<?php endif:?>
<?php

defined('AUES') or die('Access denied');
define('PATH', 'http://localhost/auес/');
define('MODEL', 'model/model.php');
define('CONTROLLER', 'controller/controller.php');
define('VIEW', 'views/');
define('TEMPLATE', VIEW.'auес/');
define('PRODUCTIMG', PATH.'userfiles/all_img/baseimg/');
define('GALLERYIMG', PATH.'userfiles/product_img/');
define('SIZE', 1048576);
define('HOST', 'localhost');
define('USER', 'root');
define('PASS', 'root');
define('DB', 'auес');
define('TITLE', 'Мобиьльная версия сайта АУЭС');
define('ADMIN_EMAIL', 'admin@ishop.com');
define('PERPAGE', 9);
define('ADMIN_TEMPLATE', 'templates/');
mysql_connect(HOST, USER, PASS) or die('No connect to Server');
mysql_select_db(DB) or die('No connect to DB');
mysql_query("SET NAMES 'UTF8'") or die('Cant set charset');
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<link rel="stylesheet" type="text/css" href="<?=TEMPLATE?>css/style.css" />
<title>Документ без названия</title>
</head>
<div class="helpon">
  <div class="breadcrumbs-homepage">
    А қосымшаның жалғасы

    <li>
      <a href="<?=PATH?>" class="homelink" title="Almaty University">
</a>
      </li>
      <li>
        
      </li>
      <li class="li_news">
        Новости
      </li>
    </ul>
  </div>
<div class="nnonfocal" id="ssearchformcontainer">
<form method="get" action="/home/search"><fieldset class="inputcombo eemphasized">
<div class="searchwrapper">
<input class="fforminput" type="text" id="ffilter" name="ffilter" placeholder="Поиск новости"
title="Search Multimedia" value="" onfocus="androidPlaceholderFix(this);">
</div>
<input class="ccombobutton" id="ssch_btn" src="<?=TEMPLATE?>images/search-button.png"
type="image" alt="Search" title="Search">

```

```

<input type="hidden" name="_b" value="">
</fieldset>
</form>
</div>
<ul class="results">
<?php if($news):?>
<?php foreach($news as $new):?>
<li class="story"><a href="?view=fnews&news_id=?=$new['news_id']?">

<div class="ellipsis">
<div class="title">'<?=$new['title'];?'</div>
<div class="smallprint"> <?=$new['anons'];?'>
</div>
</div>
</a>
</li>
<?php endforeach;?>
<?php else:~?>
<p>Пока новостей нет</p>
<?php endif;?>

<li class="pagerlink">
<a href="">Следующие новости</a></li>

```

А қосымшаның жалғасы

```

<div id="footerlinks">
<a href="#top">На Главную</a> | <a href="/home/">
Almaty University </a>
</div>
<div id="footer">
<div class="copyright">
<a href="">© 2014 Almaty, </a>
</div>
</div>
</div>
</div>
</div>
<body>
</body>
</html>
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<!--[if IE]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
<link rel="stylesheet" type="text/css" href="=?=TEMPLATE?>css/style.css" />
<title>Multimedia</title>
</head>
<body>
<div class="helpon">
<div class="breadcrumbs-homepage">
<ul>
<li>
<a href="=?=PATH?" class="homelink" title="Almaty University">
</a>
</li>

```

```

    <li>
    
    </li>
    <li class="li_news">
    Мультимедиа
    </li>
</ul>
</div>
<div class="nnonfocal" id="ssearchformcontainer">
<form method="get" action="/home/search"><fieldset class="iinputcombo eemphasized">
<div class="searchwrapper">

```

А қосымшаның жалғасы

```

<input class="fforminput" type="text" id="ffilter" name="ffilter" placeholder="Search Multimedia"
title="Search Multimedia" value="" onfocus="androidPlaceholderFix(this);">
</div>
<input class="ccombobutton" id="ssch_btn" src="<?=TEMPLATE?>images/search-button.png"
type="image" alt="Search" title="Search">
<input type="hhidden" name="_b" value="">
</fieldset>
</form>
</div>
<ul class="results" id="videoList">
<?php if($multimedia): ?>
<?php foreach($multimedia as $key):?>
<li class="video">
<a href="?view=video&m_id=<?=$key['m_id']?>" class="">
<div class="eellipsis" id="eellipsis_0">
<div class="title">
<font>
<font>'<?=$key['title']?>'</font>
</font>
</div>
<div class="ssmallprint">
<font>
<font><?=$key['text']?> </font>
</font>
</div>
</div>
</a>
</li>
<?php endforeach;?>
<?php else:?>
<p>Пока записей нет</p>
<?php endif;?>
</ul>
<div id="footerlinks">
    <a href="#">На главную</a> | <a href="<?=PATH?>">AUPET Mobile</a>
</div>
<div id="footer">
    <div class="copyright"><a href="http://www.aipet.kz">&copy; 2014 Almaty University of Power
engineering and Telecommunications</a></div>
</div>
</div>

```

```
</body>
</html>
<!DOCTYPE HTML>
<html> <head>
```

А қосымшаның жалғасы

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<link rel="stylesheet" type="text/css" href="<?=TEMPLATE?>css/style.css" />
<title>Документ без названия</title>
</head>
<div class="helpon">
  <div class="breadcrumbs-homepage">
    <ul>
      <li>
        <a href="index.html" class="homelink" title="Almaty University">
</a>
        </li>
        <li>
          
          </li>
          <li class="li_news">
            Dining
          </li>
        </ul>
      </div>
<div id="locations">
<div class="nonfocal listhead">
<h3>Столовые</h3>
</div>
<ul class="nav">
  <?php if($dining):?>
  <?php foreach ($dining as $key):?>
<li class="closed">
<a href="?view=fdining" class=""><?=$key['title']?><div class="smallprint">Режим работы:
<?=$key['time']?></div>
</a>
</li>
<?php endforeach;?>
<?php else:?>
  <p>dddd</p>
<?php endif;?>
</ul>
</div>
<div id="footerlinks">
  <a href="#">На главную</a> | <a href="<?=PATH?>">AUPET Mobile</a>
</div>
<div id="footer">
```

А қосымшаның жалғасы

```
<div class="copyright"><a href="http://www.aipet.kz">&copy; 2014 Almaty University of Power
engineering and Telecommunications</a></div>
</div>
</div>
```

```

<body>
</body>
</html>
@charset "utf-8";

*{
    padding:0; margin:0;
}
img {
    border::none;
}
body {
    background:#fff; font-size:13px; font-family:Arial, Helvetica, sans-serif; color:969696; line-
height:1.3;
}
h1, h2, h3, h4, h5, h6{
    color:2e2c2d; font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
}
.karkas{
    max-width:100%; margin:0 auto;
}
.head{
    padding:13px 30px 20px 0; overflow:hidden;
}
    .logo{
        margin:0 auto 0 auto; display:block;
    }
    #searchformcontainer {
background-color: #333;
padding: 10px;
margin: 0;
}
#searchformcontainer form {
margin: 0;
padding: 0;
}
form {
margin: 18px 0 12px 0;
padding: 0;
border:1px solid #587ab1;
position: relative;
}
form {
display: block;
margin-top: 0em;
}
.inputcombo {
border: none;
}
.emphasized {
margin: 0;
padding: 0;
}
.inputcombo {
height: 30px;
}
.inputcombo {

```

```

position: relative;
overflow-x: hidden;
}
fieldset {
display: block;
-webkit-margin-start: 2px;
-webkit-margin-end: 2px;
-webkit-padding-before: 0.35em;
-webkit-padding-start: 0.75em;
-webkit-padding-end: 0.75em;
-webkit-padding-after: 0.625em;
min-width: -webkit-min-content;
}
.searchwrapper {
right: 30px;
}
.searchwrapper {
position: absolute;
left: 0;
}
media="all"
div, p, li, a {
word-wrap: break-word;
}
div {
display: block;
}

.emphasized .forminput {
background-color: #666;
color: #eee;
}

.inputcombo .forminput, .emphasized .forminput {
box-sizing: border-box;
-webkit-border-radius: 0;
-moz-border-radius: 0;
-o-border-radius: 0;
border-radius: 0;
border: none;
height: 30px;
line-height: 20px;
width: 100%;
padding: 4px 5px 6px 5px;
}
.forminput {
position: absolute;
left: 0;
right: 34px;
-webkit-appearance: textfield;
}
.forminput {
border-color: #333333;
font-size: 14px;
}
input, textarea, select {
font-size: 100%;
}

```



```

}
.inputcombo .combobutton, .emphasized .combobutton {
width: 30px;
height: 30px;
-webkit-border-radius: 0;
-moz-border-radius: 0;
-o-border-radius: 0;
border-radius: 0;
border: none;
}
.inputcombo .combobutton {
position: absolute;
z-index: 10;
top: 0;
right: 0;
}
.content-main{
padding:30px 0;
}
.slogan-main{
overflow:hidden;
}
.slogan{
float:left; width:16.5%; margin:0 0 30px 0;
}
.slogan div a img{
width:64px; height:64px; display:block; margin:0 auto;
}
.slogan div a h2{
font-size:12px; text-align:center; color:#666; text-decoration:none;
}
.footer{
padding:25px 30px; background:#333;
A қосымшаның жалғасы
}
.footer-main{
overflow:hidden;
}
.footer-f{
float:left; width:25%;
}
.footer-f div a img{
width:32px; height:32px; display:block; margin:0 auto;
}
.footer-f div a h2{
font-size:12px; text-align:center; color:#FFF; text-decoration:none;
}
.footer-copy {
background:#333; padding:12px 0;
}
.footer-copy p{
text-align:center; color:#FFF; font-size:11px;
}
.helpo{
}
.homelink img{

```

```
padding:1px 0 0 3px;
}
.breadcrumbs-homepage{ padding:3px 2px 1px 3px; background-color:#a2c8f5;
}
    .moduleicon{
        margin:0 0 3px 5px;
    }
    .breadcrumbs-homepage ul li {
        display: inline-table;
    }
.li_news {
vertical-align: top;
padding-top:12px; color:#FFF; font-weight:bold; font:15px Verdana, Geneva, sans-serif;
word-spacing:1px;
    text-shadow:1px 1px 2px #000;
}
    .header {
padding:0 0;
}
.header form {
display: inline;
margin: 0;
padding: 0;
}
}
```

А қосымшаның жалғасы

```
table[Attributes Style] {
border-top-width: 0px;
border-right-width: 0px;
border-bottom-width: 0px;
border-left-width: 0px;
border-spacing: 0px;
}
table {
display: table;
border-collapse: separate;
border-color: gray;
}
tbody {
display: table-row-group;
vertical-align: middle;
border-color: inherit;
}
tr {
display: table-row;
vertical-align: inherit;
border-color: inherit;
}
#category-switcher .formlabel {
width: 62px;
text-align: right;
font-size: 15px;
padding-right: 0.25em;
}
#category-switcher td {
vertical-align: middle;
```

```

height: 32px;
padding: 0;
}
th, td {
line-height: 1em;
}
#category-switcher .inputfield select, #category-switcher .inputfield input[type=text] {
width: 99.9%;
margin: 0;
}
.header form select {
line-height: normal;
padding: 4px 6px;
font-size: 14px;
}
option {
font-weight: normal;
font: inherit;
}

```

А қосымшаның жалғасы

```

#category-switcher #category-form .togglefield {
width: 40px;
}
#category-form .toggle-search-button {
padding: 0 .5em 0 .5em;
}
input, textarea, select {
font-size: 100%;
}
input[Attributes Style] {
width: 32px;
height: 30px;
}
.results {
padding: 0;
margin: 10px 0;
border-bottom: 1px solid #c0c0c0;
position: relative;
background-color: #f3f3f3;
border-color: #c0c0c0;
}
.story {
height: 88px;
margin: 0;
padding: 1px;
overflow: hidden;
position: relative;
}
.results li.story .thumbnail {
float: left;
margin: 0 8px 0 0;
width: 88px;
height: 88px;
background: transparent url("/modules/news/images/news-placeholder.png") no-repeat left top;
background-size: 5.5em;
}

```

```

}
a img {
border: none;
}
.nav, .results, .nav a, .nav a:visited, .nav span.nolink, .results a, .results a:visited {
color: #000;
}
.results a {
display: block;
position: relative;
padding: 9px 25px 9px 7px;
margin: -7px;

```

А қосымшаның жалғасы

```

background-image:url(file:///E:/Sabaktar_2010-2014/saabak/diplom/images/action-arrow.png);
background-repeat: no-repeat;
background-position: right;
line-height: 1.2em;
text-decoration: none;
}
.results li {
list-style-type: none;
border-top: 1px solid #c0c0c0;
padding:0 11px 0 0;
position: relative;
}
.results li.story a .ellipsis {
overflow: hidden;
margin: 6px 25px 6px 0;
line-height: 1.2em;
height: 76px;
-webkit-border-radius: 0;
border-radius: 0;
}
.results li div.smallprint {
padding-top: .25em;
}
.results li.pagerlink a {
color: #1A1611;
font-weight: bold;
background-image: none;
}
.smallprint {
font-weight: normal;
font-size:13px;
line-height: 1.214em;
}
.smallprint {
color: #666;
}
.title{ font-size:14px;
}
.pagerlink a {
font-size:15px;
}
.footerlinks {

```

```
color: #ccc;
}
#footerlinks {
clear: both;
margin: 2.5em 6px 1.5em 6px;
font-size: 12px;}
```

А қосымшаның жалғасы

```
#footer, #footerlinks a, #footerlinks a:visited, a.copyright, a.copyright:visited, div.copyright a,
div.copyright a:visited {
color: #333;
}
#footerlinks a {
text-decoration: none;
padding: 6px;
}
#footer {
clear: both;
min-height: 22px;
padding-top: 10px;
margin: 1.5em 10px 1.5em 10px;
font-weight: normal;
}
#footer {
font-size: 12px;
line-height: 1.167em;
}
div, p, li, a {
word-wrap: break-word;
}
div {
display: block;
}
.nonfocal.listhead {
margin-bottom: -0.2em;
}
.nonfocal {
margin: 12px;
}
.nonfocal, .nonfocal .legend, .formlabel {
color: #333;
}
h3 {
font-family: Helvetica Neue, Helvetica, Arial, sans-serif;
font-size: 100%;
font-weight: bold;
}
h3 {
display: block;
-webkit-margin-before: 1em;
-webkit-margin-after: 1em;
-webkit-margin-start: 0px;
-webkit-margin-end: 0px;
}
.nav {
padding: 0;
margin: 9px;
```

border: 1px solid #c0c0c0;

А қосымшаның жалғасы

```
-webkit-border-radius: 9px;
border-radius: 9px;
position: relative;
}
.nav, .results, .nav a, .nav a:visited, .nav span.nolink, .results a, .results a:visited {
color: #000;
}
.nav, .results {
background-color: #f3f3f3;
border-color: #c0c0c0;
}
#locations li.closed {
background-image:url(file:///E:/Sabaktar_2010-2014/saabak/diplom/images/locations-status-
closed.png)
}
#locations li {
background-position: 8px 10px;
background-repeat: no-repeat;
padding-left: 35px;
}
.nav li:first-child {
border-top: 0;
}
.nav li {
list-style-type: none;
padding: 11px;
position: relative;
}
.nav li, .results li, ul.focal li {
border-color: #c0c0c0;
}
li {
display: list-item;
text-align: -webkit-match-parent;
}
.nav li div.smallprint {
padding-top: 2px;
}
```