

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

ҚОРҒАУҒА ЖІБЕРІЛДІ

Кафедра меңгерушісі

PhD, доцент

Т.С. Картбаев

«___» _____ 2019 ж.

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: Android платформасында «Z Reader» мобильдік қосымшасын құру

Мамандығы: 5B070400 – «Есептеу техникасы және бағдарламалық камтамасыз ету»

Орындаған: Аманжол А.Б. Тобы: ВТук-16-3

Ғылыми жетекші: ф.-м.ғ.к., доцент Калижанова А.У.

Кенесшілер:

Экономикалық бөлім: э.ғ.к., профессор Ж.Г. Арнбаева
«31» 05 2019 ж.

Өміртіршілік қауіпсіздігі: т.ғ.д., профессор Т.Т. Қайым
«29» 05 2019 ж.

Есептеу техникасын қолдану: аға оқытушы Ж.С. Айтқұлов
«31» 05 2019 ж.

Норма бақылаушы: аға оқытушы К. Мұқапал
«06» 06 2019 ж.

Сын-пікір беруші: т.ғ.д., профессор У.А. Тукеев
«___» _____ 2019 ж.

Алматы 2019

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

Басқару жүйелері және ақпараттық технологиялар институты

IT-инжиниринг кафедрасы

Мамандығы 5B070400 – «Есептеу техникасы және
бағдарламалық қамтамасыз ету»

Дипломдық жобаны орындауға берілген
ТАПСЫРМА

Білім алушы Аманжол Алмас Болатұлы

Жобаның тақырыбы: «Z Readear» мобильдік қосымшасын құру

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

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

Дипломдық жобаның бастапқы мәліметтері (зерттеу (жоба) нәтижелерінің талап етілген параметрлері мен объектінің бастапқы мәліметтері): Дипломдық жобада Андроид операциялық жүйесі бар смартфондарға арналған музыкалық плеер құру қарастырылған. Бағдарламалық қосымша құру барысында Android studio, Java технологиялары қолданылды.

Дипломдық жобада қарастырылған мәселелер тізімі немесе дипломдық жобаның қысқаша мазмұны:

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

Графикалық материалдар тізімі (міндетті сызбалар дәл көрсетілуі тиіс):
19 кесте, 54 сурет ұсынылған.

Ұсынылатын негізгі әдебиеттер:

1 Нотон Java. Справочное руководство. Все, что необходимо для программирования на Java / Нотон, Патрик. - М.: Бином, 2015. - 448 с.

2 Майер, Рето Android 4. Программирование приложений для планшетных компьютеров и смартфонов / Рето Майер. - М.: Эксмо, 2013. - 81 с.

3 Роджерс, Рик Android. Разработка приложений / Рик Роджерс и др. - М.: ЭКОМ Паблшерз, 2010. - 400 с.

Дипломдық жобаның бөлімдеріне қатысты белгіленген кеңес берушілер

Бөлімдер	Кеңесшілер	Мерзімі	Қолы
Экономикалық бөлім	Аренбаева Ж.Г.	04.03.19 - 31.05.19	<i>Аренбаева</i>
Өміртіршілік қауіпсіздігі	Қайым Т.Т.	04.03.19 - 29.05.19	<i>Қайым</i>
Программалық камтама	Айтқұлов Ж.С.	02.04.2019 - 31.05.2019	<i>Айтқұлов</i>
Норма бақылау	Мукапил К.	04.04.2019 10.05.2019	<i>Мукапил</i>

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекшіге ұсыну мерзімдері	Ескерту
Талдау бөлімі	24.10.18 - 28.12.18	<i>орындау</i>
Жобалау бөлімі	03.01.19 - 15.02.19	<i>орындау</i>
Жүзеге асыру және тестілеу бөлімі	24.02.19 - 12.04.19	<i>орындау</i>

Тапсырманың берілген күні «29» маусым 2018 ж.

Кафедра меңгерушісі _____ Т.С. Картбаев

Жобаның ғылыми жетекшісі *Калижанова* А.У. Калижанова

Тапсырманы орындауға алған білім алушы *Аманжол* А.Б. Аманжол

Андатпа

Дипломдық жобаның алғашқы бөлімінде смартфондарға және компьютерлерге арналған графикалық интерфейсті қосымша құрастыру негізінде музыкалық плеер қосымшасы құрылды. Бағдарламалық Z Reader Music Player қосымшасын құру барысында «Android studio», «Java» технологиялық программалары қолданылды. Қосымшада мобильдік құрылғы арқылы музыкалық композицияларды жүктеу, тыңдау, инструменталдық биттерді пайдалану арқылы жаңа әуен жазу, әуендерді көшірмеден өткізу мүмкіндіктері қарастырылған.

Z Reader Music Player қосымшасы логикалық байланысқан бөлімдерден құралған, қосымша интерфейсі пайдаланушыға қолайлы, қосымшалық өнім әр түрлі форматты файлдармен жұмыс жасауға оңтайландырылған.

Екінші тарауда, қосымшаны құру және оны іске асыру жолдары жасалды.

Үшінші және төртінші тарауда, техника-экономикалық және өміртіршілік қауіпсіздігі бөлімдері қарастырылған.

Аннотация

В первой части дипломного проекта создано приложение – музыкальный плеер на основе дополнительного построения графического интерфейса для смартфонов и компьютеров. При создании программного приложения Z Reader Music Player использованы технологические программы как «Android studio», «Java». В приложении предусмотрена возможность загрузки, прослушивания музыкальных композиций с помощью мобильных и компьютерных устройств, написания новой мелодии с использованием инструментальных бит, копирования мелодий и выявления плагиатов.

Приложение Z Reader Music Player состоит из логически связанных разделов, дополнительный интерфейс подходит любому пользователю, продукт оптимизирован для работы с файлами различных форматов.

Во второй главе были разработаны пути создания и реализации приложения.

В третьей и четвертой главе предусмотрены разделы технико-экономической и безопасности жизнедеятельности.

Annotation

In the first part of the graduation project created an application – music player based on the additional construction of a graphical interface for smartphones and computers. When creating the z Reader Music Player software application, technological programs such as "Android studio", "Java" are used. The application provides the ability to download, listen to music using mobile and computer devices, writing a new melody using instrumental beats, copying melodies and identify plagiarism.

Z Reader Music Player application consists of logically linked sections, an additional interface is suitable for any user, the product is optimized to work with files of different formats.

In the second Chapter have been developed with the creation and implementation of the application.

The third and fourth Chapter provides sections of technical, economic and life safety.

Мазмұны

Кіріспе	8
1. Смартфондарға арналған графикалық интерфейсті қосымша құру негіздері	10
1.1 Смартфондар. Смартфондар платформасы туралы жалпы түсінік	16
1.2 Android платформасы	18
1.3 Android studio бағдарламасы және орнатылымы	20
1.4 Z Reader Music Player қосымшасы	23
1.5 Z Reader Music Player қосымшасының жұмыс істеу принципі: дерекқор файлы, түзету, бөліп алу	27
1.6 Z Reader Music Player қосымшасының жұмыс істеу принципі: көрініс, транспорт, тректер, құру	28
1.7 Z Reader Music Player қосымшасының жұмыс істеу принципі: эффекттер, анализ, құралдар, анықтамалық	29
2. Қосымшаны құру және оны іске асыруы	31
2.1 Бағдарламалық жүйені құрудың бастапқы кезеңі. Қажетті құралдарды дайындау	31
2.2 Қосымшаның Android studio бағдарламасында жүзеге асуы	45
3. Техника-экономикалық негізделуі	52
3.1 Бағдарламалық жасақтама әзірлеудің күрделілігі	52
3.2 ПП әзірлеуге жұмсалатын шығындар есебі	53
3.3 Жұмысқа қажетті еңбек ресурстары	55
3.4 ПП-ның мүмкін болатын (келісілген) бағасы	58
4. Өміртіршілік қауіпсіздігі	60
4.1 Жарықтандырудың негізгі сипаттамалары	60
4.2 Жұмыс орындағы микроклимат параметірлеріне талдау жасау	66
4.3 Электр тораптарындағы қауіпті талдау	67
4.4 Электр қондырғыларындағы қорғаныс шаралары	71
Қорытынды	76
Пайдаланылған әдебиет тізімі	77
А қосымшасы	78
Ә қосымшасы	80

Кіріспе

Қазіргі кезде кез келген салада мобильдік құрылғыларда жұмыс жасайтын бағдарламалық қосымшалардың қажеттігі зор екеніне еш күмән келтіре алмаймыз. Қарапайым ғана, әр екінші адамзаттың қолынан табылатын гаджеттеріміздің ішінде бір емес, он бес-жиырма шақты мобильдік қосымшалар енгізілген.

Әлем бойынша тараған операциялық жүйелердің бірі – Android платформасы. Android – ашық бастапқы коды бар алғашқы тегін операциялық жүйе болып табылады. "Жасыл роботта" – телефондар, планшеттер, теледидарлар, ақылды сағаттар және басқа да гаджеттер жұмыс істейді, ал жақын арада автокөліктер Android көмегімен басқарылатын кез де алыс емес-ті. Андроид OS өте белсенді дамып келе жатқан Google компаниясына тиесілі. Бұл өнім пайдалануға ыңғайлы ғана емес, ол қол жетімді қосымшалардың үлкен жиынтығымен мақтана алады, жүйе түрлі мүмкіндіктерге ие болып келеді. Пайдаланушы көптеген дизайн элементтерін өзгерте алады. Сондай бір қосымшаларың бірі, қоғамымызда кез келген жеке тұлға қолданатын әуен тыңдау, жазу, ойдан шығару қосымшаларын жатқыза аламыз. Бүгінгі тақырыбымызда сол жайында сөз қозғамақпыз. Заман талабына сай қосымшалар өте аз, жеке тұлғамен немесе тұлғалармен ойлап табылып, әрленіп, заңдастырылған, әрі қарапайым халыққа қолдануға ыңғайлы болатын қосымша болуы шарт.

Еліміздің әрбір азаматының қосымшаларды пайдалану арқылы, әуен тыңдау, жазу, ойдан шығару жүйесін компьютерлендірудің ойдағыдай жүзеге асып келе жатқандығын практика жүзінде бақыладық. Сондай бір қосымшалардың бірі – Z Reader Music Player қосымшасы. Бұл қосымшаның бірден бір ерекшеліктері көп. Мысалға алсақ, қарапайым бағдарлау сілтемелерін пайдалана отырып, тұлғаның әуесқой әуенін де жазуға болады. Мүмкін жеке бір тұлға битбоксмен (beatbox) айналысатын болса, оның ойындағы әуенге жетіспей тұрған элементтер мен эффектілерді қосып, толыққанды су жаңа өнім алуға болады.

Менің жазған дипломдық жобамда толыққанды смартфондар платформасы туралы жалпы түсінік қамтыла отырып, смартфондар әдеттегі ұялы телефондардан бөгде әзірлеушілермен бағдарламалық қамтамасыз етуді әзірлеу үшін ашылған жеткілікті дамыған операциялық жүйенің болуымен ерекшеленетіні жайлы ақпаратпен қамтылады. Қосымша қолданбаларды орнату – смартфондардың функционалдығын қалыпты ұялы телефондармен салыстырғанда айтарлықтай жақсартуға мүмкіндік беретінін барлығымыз да білеміз. Алайда соңғы уақытта күнделікті тұтынатын, әрбір екінші азаматтың телефондары мен смартфондары арасындағы шекара бұрынғыдай смартфондарға тән, мысалы, электрондық пошта және HTML-браузер, сондай-ақ көпмақсаттылық сияқты жаңа телефондар бұрыннан келе жатқан функционалдылыққа ие болды.

XXI ғасыр – жаңа технологиялар дәуірінде Android платформасы –

смартфондарға, планшеттерге, электрондық кітаптарға, сандық ойнатқыштарға, қол сағаттарына, фитнес білезіктерге, ойын тіреулеріне, ноутбуктерге, нетбуктерге, смартбуктерге, Google Glass көзілдіріктеріне, теледидарларға және басқа құрылғыларға арналған операциялық жүйе ретінде қолданамыз. Осы Android платформасы арқылы қарапайым күнделікті қолданатын қосымшаларымыз, соның ішінде тыңдайтын әуендерімізді арнайы қосымшалар арқылы тыңдап, әрі әуесқой қызығушылық танытып, өзіміздің әуен шығаратын мүмкіндіктеріміз жетіп жатады. Бұл ойнатқыш болашақта стильді түрлі түсті тақырыптарды таңдау мүмкіндігімен қарастырылатын, сондай-ақ қарапайым және сол уақытта өте стильді көрініс тауып ерекшеленетін қосымша. Форматтардың бір бөлігі бойынша қосымша MP3 және WAV, OGG, MIDI және FLAC ойнатады.

Z Reader Music Player қосымшасында біз күні бойы олар туралы айта алатын жеке ерекшеліктер көп. Алдын ала орнатылған пресеттер, төмен және жоғары жиілікті реттегіштері бар эквалайзер және тонды және дыбыс деңгейін реттеу үшін жеке қойындылар бар. Егер сіз бәрін жақсы баптасаңыз, FLAC пішімін тыңдау кезінде әсіресе байқалатын дыбыс сапасын жақсарта аласыз. Ол сондай-ақ mp3, mp4/m4a (alac), ogg, WMA, flac, wav, ape, wv, tta, mpc және aiff пішімдерін ойнатады. Егер сіз белгілі бір музыкаға ұйықтап қалғыңыз келсе, онда сіз үшін жақсы жаңалық бар, бұл жерде ұйқы таймері бар, ол сіз ұйықтауға және бүкіл батареяны разрядтайтын музыканы қосулы қалдырыңыз туралы алаңдамауға мүмкіндік береді. Z Reader Music Player қосымшасы плагиімен ән мәтіндерін іздеуге болады. Осы секілді басқа қосымшаларға қарағанда өте көп артықшылықтары бар екенін ескере кеткім келеді.

1 Смартфондарға арналған графикалық интерфейсті қосымша құру негіздері

Пайдаланушының графикалық интерфейсі (ПГИ) – (ағылшын тілінде graphical user interface, GUI) - пайдаланушыға дисплейде ұсынылған интерфейс элементтері (мәзір, түймелер, белгішелер, тізімдер және т.б.) графикалық суреттер түрінде орындалған пайдаланушы интерфейсіннің бір түрі. Сондай-ақ, басқару графикалық қабығы деп те аталады. Пайдаланушының графикалық интерфейсі – пайдаланушы интерфейсіннің бөлігі болып табылады және визуализацияланған ақпарат деңгейінде пайдаланушымен өзара әрекеттесуін анықтайды.

Жалпы пайдаланушының графикалық интерфейсі

- қарапайым: типтік экрандық пішіндер және стандартты интерфейс элементтері;

- шынайы-графикалық, екі өлшемді: интерфейсіннің стандартты емес элементтері және қолданба құралдарымен немесе бөгде кітапханамен іске асырылған түпнұсқа метафорлар;

- үш өлшемді болып классификацияланады.

Пайдаланушының графикалық интерфейсіннің артықшылықтары:

- графикалық интерфейсмен жаңадан бастап келе жатқан адам компьютермен тез-ақ үйренісіп кетеді, тек аздаған қабілеті болса әрине.

- графиканы өңдеу бағдарламалары арасында ол әзірге жалғыз құрылымы болып табылады.

Java-да графикалық интерфейсдерді жасау үшін 2 негізгі пакет бар (Graphics User Interface). Бұл Abstract Windows Toolkit (AWT) және Swing. AWT Операциялық жүйе виджеттерін пайдаланады, сондықтан бұл қор тезірек жұмыс жасайды. Осы тьюториалда біз Swing қорының негізгі элементтерін қарастырамыз және мысал ретінде қарапайым интерфейс (GUI) жасаймыз. Интерфейс компонентін топтастыру үшін контейнерлер (Container) қолданылады. Құру үшін негізгі контейнер үшін контейнер қосымшасы жиі пайдаланылады, мысалы JFrame (бар JWindows және JApplet). JFrame-ден іздеу оңай, мысалы, көптеген әдістерге қол жеткізуге болады:

setBounds(x, y, w, h) - терезенің жоғарғы сол жақ жоғарғы биіктігінің координаттарын, сондай-ақ оның ені мен биіктігін көрсетеді.

setResizable(bool) - терезе өлшемін өзгертуге болатынын көрсетеді.

setTitle(str) - терезенің атауын белгілейді.

setVisible(bool) - терезені көрсетеді.

setDefaultCloseOperation(operation) - терезені жабу кезінде орындалатын әрекетті көрсетеді.

Java бағдарламалау тілі және есептеу платформасы болып табылады, ол 1995 жылы алғаш рет Sun Microsystems компаниясымен шығарылған. Java жылдам, жоғары қорғаныс деңгейімен және сенімділігімен ерекшеленеді. Портативті компьютерлерден деректер орталықтарына дейін, ойын консолінен ғылыми әзірлемелер үшін пайдаланылатын супер компьютерлерге

дейін, ұялы телефондарға дейін интернет-Java желісі барлық жерде тааралған.

Тілдің негізгі ерекшеліктері. Java бағдарламалары виртуалды машинасымен (JVM) орындалатын Java байт-кодына — байт кодын өңдейтін және интерпретатор ретінде жабдыққа беретін нұсқауға арналған бағдарламамен көрсетіледі. Java тұмары болып бағдарламаны орындаудың тәсілінің артықшылығы байт-кодтың операциялық жүйе мен жабдықтан толық тәуелсіздігі болып табылады, бұл тиісті виртуалды машина бар кез келген құрылғыда Java-қосымшаларды орындауға мүмкіндік береді. Java технологиясының тағы бір маңызды ерекшелігі-бағдарламаның орындалуы виртуалды машинамен толық бақыланатын икемді қауіпсіздік жүйесі болып табылады. Бағдарламаның белгіленген өкілеттілігінен асатын кез келген операциялар (мысалы, деректерге рұқсатсыз қол жеткізу немесе басқа компьютерге қосылу әрекеті) дереу тоқтатуды талап етеді. Виртуалды машина тұжырымдамасының кемшіліктеріне – жиі өнімділіктің төмендеуі жатады. Бірқатар жетілдірулер Java-да бағдарламаларды орындау жылдамдығын арттырды: тікелей бағдарлама (JIT-технология) жұмыс істеп жатқанда машина кодында сынып нұсқаларын сақтау мүмкіндігімен байт-кодты машиналық кодқа трансляциялау технологиясын қолдану, стандартты кітапханаларда платформалық-бағытталған кодты (native-код) кеңінен пайдалану, байт-кодты жылдам өңдеуді қамтамасыз ететін аппараттық құралдар (мысалы, ARM архитектурасының кейбір процессорлары қолдайтын Jazelle технологиясы).

Java қосымшасының жаңару тарихы:

JDK 1.0 Java әзірлеу 1990 жылы басталды, бірінші ресми нұсқасы — Java 1.0, - 1996 жылдың 21 қаңтарында ғана шығарылды.

JDK 1.1 Екінші нұсқа 1997 жылы 19 ақпанда шығарылды. Accessibility Кітапханасы. Java 2D. Drag-and-drop технологиясын қолдайды. Жапон, қытай және корей тілдерінде енгізуді қолдидуды қосағанда, Unicode толық қолдауы. Бірнеше танымал форматтағы дыбыс файлдарын ойнатуды қолдау. CORBA технологиясын толық қолдау. JIT-компилятор, жақсартылған өнімділік. JDK құрал-саймандық құралдарын жетілдіру, оның ішінде Java-бағдарламаларды профильдеуді қолдайды.

J2SE 1.2 Шығарылған күні 1998 жылғы 8 желтоқсан. Playground кодтық атауы. Бұл жағдайда шатасуы бар. Кітап, мысалы, *beginning Java 2 by Ivor Horton* (Mar 1999), іс жүзінде J2SE 1.2 (бұрынғы атауы — Java 2) бойынша шығарылды. Сонымен қатар осы күнге дейін мұндай кітаптар жарияланады, мысалы: Х. М. Дейтел, П. Дж. Дейтел, С. И. Сантри. *Java бағдарламалау технологиялары 2. Бөлінген қосымшалар* (2011). Белілі болғандай, Java 2 келесі релиздермен Тарихи ауыстырылып, осы сияқты кітаптар атауларының Java-ның қандай нұсқасы шын мәнінде жазылғаны туралы түсінікке бағдарланады. Егер J2SE 1.2 Java 2 деп есептелсе, ал Java 2 үшін кітаптар авторлары JDK 7 қабылдайды, бұл толық шатасуға әкеледі.

J2SE 1.3 Шығарылған күні 2000 жылғы 8 мамыр. Kestrel кодтық атауы.

J2SE 1.4 Шығарылған күні 2002 жылғы 6 ақпан. Merlin кодтық аты.

Java 5.0 ерекшелігі 30 қыркүйек 2004 жылы шығарылды, Tiger кодтық атауы. Бұл нұсқада ресми индекстеу өзгеріілді, Java 1.5 орнына Java 5.0 деп атау дұрыс. Ішкі sun индекстеу бұрынғы қалды-1.X. Бұл үшін "Update" сөзі немесе "u" әрпі қолданылады, мысалы, Java Development Kit 5.0 Update 22. Жаңартуларға қателерді түзету және API, JVM-ге шағын қосымшалар кіруі мүмкін.

Бұл нұсқада әзірлеушілер тілге бірқатар түбегейлі толықтырулар енгізді:

- аталатын түрлері (ағыл. enum). Бұрын болмаған Java типтері C++ аналогы бойынша ресімделген, бірақ қосымша мүмкіндіктер қатары бар;

- атап айтқанда, Java толық классы болып табылады, яғни конструктор, өрістер, әдістер, оның ішінде жасырын және дерексіз болуы мүмкін;

- аудару интерфейстерді іске асыра алады;

- тізімдер үшін атау бойынша тип мәндерін, символдық мәндерді, атауларға сәйкес келетін, нөмір мен мән арасындағы түрлендірулерді, оның атап айтқанда типін тексеруге мүмкіндік беретін кіріктірілген әдістер бар;

- аннотация-бағдарламаның мәтініне кодты орындауға тікелей әсер етпейтін, бірақ Код және оның орындалуы туралы әртүрлі мәліметтерді алу үшін пайдалануға жол беретін метадеректерді қосу мүмкіндігі. Аннотацияланған кодты пайдалану үшін бір уақытта құрал шығарылды. Аннотацияларды қолданудың бірі-Java-кодына арналған тест модульдерін жасауды жеңілдету;

- жалпыланған бағдарламалау құралдары (ағылш. GPON технологиясы (GPON) — Ada[21] және Eiffel тілдеріне ұқсас механизм (кейінірек C# - да пайда болды, с++ шаблондарынан принципті ерекшеленеді), ол еркін Объектілік типтегі өрістер мен параметрлермен кластар мен әдістерді жасауға мүмкіндік береді. Осы тетікті пайдалану арқылы Java стандартты кітапханасы топтамаларының жаңа нұсқалары іске асырылды;

- параметрлердің анықталмаған саны бар әдістер;

- autoboxing / Unboxing-Java скаляр түрлері мен Тиісті орау түрлері (мысалы, int — Integer арасында) арасындағы Автоматты түрлендіру. Мұндай мүмкіндіктің болуы кодты қысқартады, өйткені айқын жағдайларда үлгілердің анық өзгеруін орындау қажеттілігін жоққа шығарады.

Статикалық өрістер мен әдістерді импорттауға рұқсат етілген. Тілге Нысандар жинағы бойынша цикл енгізілді (итератор, ағыл. foreach). Бастапқы кодта түсініктеме бойынша құжаттаманы автоматты түрде ресімдеу үшін пайдаланылатын Javadoc-түсініктемелерді пайдалану енгізілді. Java SE 6.

Болжам нұсқасы 11 Желтоқсан 2006 жыл, Mustang кодтық атауы болды. Ресми индекстеу өзгертілді-күтілген 6.0 нұсқасы 6 ретінде көрсетіледі. Java 5.0 сияқты минорлы өзгерістер, мысалы, Java Standard Edition Development Kit 6 Update 27. Мынадай өзгерістер енгізілді:

- коллекциялар-коллекцияның екі жағынан жұмыс істейтін кезекті ұйымдастыру үшін интерфейстер қосылды; жақын сәйкестіктер бойынша іздеуді ұйымдастыратын; элементті күту кезінде өзін бұғаттайтын. Аталған интерфейстерді іске асыратын жаңа сыныптар ұйымдастырылған;

- жапон императорлық күнтізбесінің қолдауы қосылды (қазіргі Григориан және будда күнтізбелерімен қатар);
- желі арқылы жіберу мүмкіндігі бар қысылған деректерді оқу және жіберу үшін кластар бар. Мұрағаттағы файлдар санына (бұрын 64 Кб), файл атауының ұзындығына (бұрын 256 таңба) және бір уақытта ашылған файлдар санына (бұрын 2000 дана) шектеулер алынды;
- кэшіті басқару жүйесі ұйымдастырылып, HTTP сұрауында "no-cache" параметріне қолдау қосылды;
- JConsole, JVM графикалық мониторингі, ресми қолдау құралы болды;
- Java HTTP Server, қажетті функционалдық қасиеттерімен толық HTTP серверін жасауға мүмкіндік береді;
- есептеу жылдамдығы 70%-ға өсті, енгізу-шығару операцияларының жылдамдығы екі есе өсті;
- Swing-OpenGL және DirectX жұмыс қабілеттілігі жақсартылған; LCD-де мәтінді өңдеу; GIF форматындағы файлдармен жұмыс істеу үшін GifWriter қосылған;
- көптеген қателер түзетілді.

Java FX

Шығарылған күні 8 қазан 2013 жыл.

JavaFX 2.2 Java SE 7 update 6 құрамына кірді. 11-ші нұсқадан бастап модуль JDK-дан бөлек келеді.

Java ME Embedded

Шығарылған күні 10 Қазан 2013 жыл. Micro Edition кодтық атауы.

Java SE 7

Нұсқаның шығарылымы 2011 жылдың 28 шілдесінде, Dolphin кодтық атауы болды. Java Standard Edition 7 соңғы нұсқасына бұрын жоспарланған барлық өзгерістер енгізілмеді. Даму жоспарына сәйкес ("Б"жоспары), жаңа енгізілімдерді қосу екі бөлікке бөлінеді: Java Standard Edition 7 (лямбсыз-есептеулер, Jigsaw жобасы және Coin жобасының жақсартуларының бір бөлігі) және Java Standard Edition 8 (қалған бөлігі), 2012 жылдың соңына белгіленген.

Java Standard Edition 7 (Java Platform, Standard Edition 7) атауын алған жаңа нұсқада көптеген қателерді түзетуден басқа, бірнеше жаңалықтар ұсынылды. Мысалы, Java Standard Edition 7 эталондық іске асыру ретінде JDK проприетарлық пакеті емес, оның OpenJDK Ашық іске асырылуы пайдаланылды, ал платформаның жаңа нұсқасының хабарламасы Oracle инженерлерінің Java әлемдік экожүйесінің қатысушыларымен, JCP комитетімен (Java Community Process) және OpenJDK қоғамдастығымен тығыз ынтымақтастықта дайындалды. Java Standard Edition 7 эталондық іске асырудың Oracle бинарлық файлдары OpenJDK кодтық базасы негізінде жиналған, эталондық іске асыру GNU ClassPath ерекшеліктерімен бірге GPLv2 лицензиясымен толық ашық. Басқа жаңалықтарға Coin жобасы аясында дамитын Java шағын тілдік жақсартулар жиынтығын біріктіру

жатады, Ruby, Python және JavaScript сияқты динамикалық типиялануы бар бағдарламалау тілдерін қолдау, JAXP 1.4, JAXB 2.2 А және JAX-WS 2.2 және т.б. қамтитын жаңартылған XML-стек URL кластарын жүктеуді қолдау қосылды.

Java Standard Edition 7 релизі шыққанға дейін 5 күн бұрын әдепкі бойынша қосылған және Java виртуалды машинасын крахқа әкелетін циклдарды ыстық оңтайландыруда бірнеше елеулі қателер анықталды. Oracle мамандары қысқа мерзімде табылған қателіктерді түзете алмады, бірақ олар екінші жаңартуда (Java 7 Update 2) және жартылай бірінші жаңартуда түзетіледі деп уәде берді.

Жаңалықтар тізімі:

- динамикалық түрленген тілдерді қолдау (InvokeDynamic) — динамикалық түрленген тілдерді қолдау үшін JVM (байт код семантикасы), Java тілі кеңейтуі;

- Class файлдарын қатаң тексеру-51 (Java Standard Edition 7) немесе одан кейінгі нұсқадағы class файлдары typechecking-верификатормен тексерілуі тиіс; JVM ескі верификаторға ауыспауы тиіс;

- Java (Project Coin) тілінің синтаксисін өзгерту — жалпы бағдарламалау есептерін жеңілдетуге арналған Java тіліндегі ішінара өзгерістер.

- класты жүктеушінің модификациясы (class-loader) — класты жүктеудің иерархиялық емес топологиясындағы тұйық жағдайларды болдырмау;

- URLClassLoader ашық ресурстарды жабу;

- коллекцияларды жаңарту (JSR 166);

- Unicode 6.0 Қолдау;

- пайдаланушы тілі мен пайдаланушы интерфейсінің тілі бөлімі-пайдаланушы интерфейсінің тілінен локалды бөлу үшін тілдерді өңдеуді жаңарту;

- Java платформасына арналған Жаңа I/O интерфейстері (nio.2);

- JDBC 4.1 және Rowset 1.1 пайдалану.

Java SE 8

Нұсқасы 19 наурыз 2014 жыл өтті. Octopus кодтық атауы.

Жаңалықтар тізімі:

- лямбда-өрнектерді толыққанды қолдау;

- әдепкі функцияны қолдау үшін интерфейстердегі default кілт сөзі;

- сілтемелер әдістері мен конструкторлар;

- функционалдық интерфейстер (предикаттар, жеткізушілер);

- коллекциялармен жұмыс істеу үшін ағындар (stream).

Java SE 9

Jigsaw жобасы аясында модульдік жүйені іске асырудағы қиындықтарға байланысты, нұсқаның хабарламасы, бастапқыда 2016 жылдың 22 қыркүйегінде жоспарланған, бірнеше рет кейінге қалдырылды.

Соңғы күн нұсқаның ресми күні болды.

Жаңалықтар тізімі:

- Jigsaw интеграциясы, оның аясында Java 9 платформасына арналған модульдік жүйе жасалған және JDK 9 қолданылған;
- операциялық жүйе үдерістерімен өзара әрекеттесуді жақсарту үшін Process API жаңарту. Жаңарту дәлелденуі бұл әзірлеушілерге көбінесе тура келді, жазу платформозависимый коды үшін осындай міндеттер;
- HTTP/2 және веб-сокеттерді қолдайтын жаңа HTTP клиенті; Ескірген HttpURLConnection класын ауыстыруға арналған;
- сығылған жолдар: егер жол мазмұны мүмкіндік берсе, ол Latin-1 (таңбаға бір байт) кодталуы мүмкін; String класының нақты данасының кодтамасын таңдау барлық жолдарда бар айнымалы-ту мағынасында көрсетіледі;
- SHAKE128 және SHAKE256 басқа, NIST FIPS 202 SHA-3 араластыру алгоритмдерін қолдау. Басқа криптографиялық функциялар үшін негіз ретінде SHA-3 қолдану алгоритмдері тиісті стандарттардың жоқтығына байланысты іске асырылмаған.

Java SE 10

Күні: 20 наурыз 2018 жыл.

Жаңалықтар тізімі:

- Жаңалықтардың ресми ішінара тізімі мен релиз жоспары OpenJDK сайтында орналасқан;
- Var кілт сөзінің көмегімен белгіленген жергілікті айнымалылардың типтерін шығару;
- Жаңа жинағыштарды әзірлеуді оңайлату үшін қоқыс жинаушының мөлдір интерфейсін жасау;
- қоқыстарды жинаудың параллель толық циклін жүзеге асыру есебінен G1 көп ағынды қоқыстарды жинаудың кідіріс уақыты азайды;
- жаңа эксперименталды JIT-Graal компиляторы, Ahead-of-Time компиляция мүмкіндігін береді; әдепкі бойынша өшірулі, тек Linux / x64-те жұмыс істейді;
- Java SE және JDK нұсқаларын уақыт бойынша басқару схемасына жақындату мақсатында нөмірлеу жүйесін жаңарту.

Java SE 11

Жаңалықтардың ресми ішінара тізімі мен релиз жоспары OpenJDK сайтында орналасқан. Күні-25 Қыркүйек 2018 жыл.

Жаңалықтар тізімі:

- оптикалық кабельдің негізгі сипаттамалары;
- оптикалық кабельдің негізгі параметрлері;
- Epsilon-қоқыс жинайтын жаңа қоқыс жинаушы, мүлдем айналыспайтын қоқыс жинаушы;
- Epsilon пайдалану кезінде бөлінетін жад шегінен асып кету JVM жұмысының аяқталуына әкеледі;
- HTTP клиенті Java 9-да эксперименталды ретінде енгізілген HTTP/2 қолдайтын стандартталған;

- Unicode стандартының 10-ші нұсқасын қолдау;
- TLS 1.3 протоколын қолдау.

Басқарудың негізгі элементтері:

JLabel - тіркелген мәтінді көрсету элементі;

JTextField - қарапайым edit-box;

JButton - қарапайым батырма(button);

JCheckBox – таңдау элемент (аналог checkbox);

JRadioButton - радио батырма.

Көріп отырғаныңыздай, барлығы қарапайым және қисынды болып келеді.

Басқару элементтерін көрсету кезінде арнайы менеджер - LayoutManager қолданылады. Барлық LayoutManager-лерде элементтерді жою және қосу әдістері бар.

FlowLayout – элементтерді дәйекті көрсету үшін қолданылады. Егер элемент нақты жолға қойылмаса, ол келесіде көрсетіледі.

GridLayout – бірдей ұяшықтары бар кесте түрінде элементтерді көрсету.

BorderLayout – 5 элементтен артық емес көрсету кезінде қолданылады. Бұл элементтер фрейм шеттерінде және центрте орналасқан: North, South, East, West, Center.

BoxLayout – қатар немесе баған түрінде элементтерді көрсетеді.

GridBagLayout – әр виджеттің орналасу орны мен өлшемін тағайындайды. Бұл ең күрделі, бірақ ең тиімді көрініс.

Оқиғаларды өңдеуге назар аудару қажет. Ол үшін Event Listeners деп аталады.

1.1 Смартфондар. Смартфондар платформасы туралы жалпы түсінік.

Смартфон (ағылш. smartphone-ақылды телефон) – қалта дербес компьютердің функциясымен толықтырылған ұялы телефон. Сондай-ақ коммуникатор (ағылш. communicator, PDA phone) - ұялы телефонның функциясымен толықтырылған қалта дербес компьютер. Ұялы телефондарда әрқашан қосымша функциялар (калькуляторлар, күнтізбелер) болса да, уақыт өте келе зияткерлік үлгілер шығарылды, мұндай модельдердің өсіп келе жатқан функционалдығы мен есептеу қуатын көрсету үшін "смартфон" терминін енгізді. Қалта компьютеріні танымалдылығының өсу дәуірінде олар ұялы телефон функцияларымен шыға бастады, мұндай құрылғылар коммуникаторлар деп аталды. Қазіргі уақытта смартфондар мен коммуникаторларға бөлу өзекті емес, екі термин де толық пайдаланушы интерфейстері және ұялы телефонның дамыған радио — интерфейстері бар бір шағын әмбебап компьютерді білдіреді.

Қазіргі уақытта коммуникаторлар мен смартфондар арасында айқын ажырату жоқ, себебі құрылғылардың екі класының функционалдығы шамамен бірдей. Әр түрлі сарапшылар мен өндірушілер бұл терминдерді әр түрлі

түсіндіреді. Жиі "тарихи көзқарас" деп аталады, Ол мыналардан тұрады: егер құрылғы өз шежіресін ҚҚП — дан алып келсе — бұл коммуникатор, ал егер ұялы телефондардан болса-бұл смартфон. Бұл тәсіл шеңберінде коммуникаторлар әдетте Apple iOS, Windows Phone, Open webOS немесе Android операциялық жүйесінің басқаруында жұмыс істейтін сенсорлық экраны бар құрылғылар (пернетақтамен толықтырылуы мүмкін) болып табылады. Тек QWERTY - және/немесе сандық пернетақтаны (телефон аналогы) енгізу үшін қолданатын Windows Mobile құрылғылары смартфондар деп аталады. Symbian OS басқаратын құрылғылардың көпшілігі смартфондарға (Nokia 9xxx, Nokia E90 және басқа да сериялардан басқа) жатады. Басқа жағдайларда құрылғыны позициялау өндірушіге байланысты.

Сондай-ақ, мамандардың бір бөлігі коммуникаторлар мен смартфондарды толық өлшемді (QWERTY) пернетақтаның (виртуалды немесе физикалық) болуына немесе болмауына сәйкес бөледі.

Смартфондардың кезекті даму кезеңі Apple фирмасынан iPhone ұялы телефонының нарығына сәтті шыққаннан кейін басталды. Смартфон ретінде орналасқан осы құрылғының операциялық жүйесі маркетингтік пайымдаулардан функционалдық түрде кесілген. Осылайша, бөгде өндірушілердің бағдарламаларын орнату мүмкіндігі шектелді, көп беріктік бөлігінде шектеулер болды. Дегенмен, сәтті дизайн мен сауатты жылжыту саясатының арқасында бұл құрылғы мод заң шығарушысы болды және ауытқусыз құрылғылар үшін жаңа стандарттар орнатты. Егер 2000-шы жылдардың ортасында көптеген коммуникаторлар мен смартфондар экранының көлемі 320×240 нүктенің рұқсаты бар 2,4-2,8 дюймді құраса, қазіргі уақытта 1920×1080 нүктенің рұқсаты бар 5 экран типтік болды.

Смартфондар әдеттегі ұялы телефондардан бөгде әзірлеушілермен бағдарламалық қамтамасыз етуді әзірлеу үшін ашылған жеткілікті дамыған операциялық жүйенің болуымен ерекшеленеді (кәдімгі ұялы телефондардың операциялық жүйесі бөгде әзірлеушілер үшін жабық). Қосымша қолданбаларды орнату смартфондардың функционалдығын қалыпты ұялы телефондармен салыстырғанда айтарлықтай жақсартуға мүмкіндік береді.

Алайда соңғы уақытта "әдеттегі" телефондар мен смартфондар арасындағы шекара бұрынғыдай смартфондарға тән, мысалы, электрондық пошта және HTML-браузер, сондай-ақ көпмақсаттылық сияқты жаңа телефондар (ең арзан модельдерден басқа) бұрыннан келе жатқан функционалдылыққа ие болды.

Мобильді қосымшаларды әзірлеушілер әрқашан жаңа бағдарлама құрылатын платформаны анықтаудан бастап жұмыс істей бастайды. Құзыретті сарапшылардың пікірінше, бүгінгі таңда ең сәтті және сұранысқа ие болып келесі мобильді құрылғылар: Android, Windows Phone, iOS және Symbian платформалары саналады.

Android платформасы Wi-Fi Direct, NFC қолдайды, мультимедиалық файлдарды ағындық режимде қабылдауға мүмкіндік береді. Сондай-ақ, смартфондар мен планшеттік компьютерлерге осы операциялық жүйесі

базасындағы USB порты: камералар, ТВ тюнерлер, флеш жады тасымалдаушылары арқылы қосыла алады. Орнатылған және жаңартылған қауіпсіздік жүйесі сертификатталмаған және зиянды қолданбаларды қабылдауға мүмкіндік бермейді. Android 4.3 нұсқасында (Jelly Bean) кез келген Bluetooth құрылғыларымен гарнитурасыз байланыс орнатуға мүмкіндік беретін Bluetooth Smart мүмкіндіктеріне қолдау көрсету және OpenGL ES 3.0 қосымшаларымен жұмыс істеуге мүмкіндік бар.

Android Go платформасының маңызды артықшылықтары:

- Бос орын көп. ОС қосымша сервистерден, функциялар мен қосымшалардан айырылғандықтан, ол әлдеқайда аз салмақ. Бұл ішкі жадта қымбат бос орынды үнемдеуге мүмкіндік береді;

- Қажетсіз бағдарламалардың болмауы. Бұған дейін айтылғандай, платформада Google-да ең маңызды емес деп санаған қосымшалар алынып тасталды. Осының арқасында жаңа адамдар софттың көп мөлшерде шатастыра алмайды;

- Мінсіз оңтайландыру;

- Жақсы жүргізілген оңтайландырудың көмегімен жүйе тіпті ашық әлсіз құрылғыларда да жылдам жұмыс істейді. Сіз жоғары жылдамдықты платформаны темірге байланыстырмай аласыз;

- Трафикті үнемдеу. "Оське" тұтынылатын трафик шығынын үнемдеу режимі енгізілген. Ол әдепкі бойынша іске қосылған, сондай-ақ мегабайттарды үнемді жұмсауға көмектеседі.

- Толық Google Play. Мұнда ең аз шектеулері бар танымал сандық дүкеннің толық нұсқасы орналасқан. Бұл ретте кейбір қосымшалар "Go" деп белгіленеді. Қол жетімді бағдарламалардың мұндай нұсқалары ең жеңіл және оңтайландырылған болып табылады.

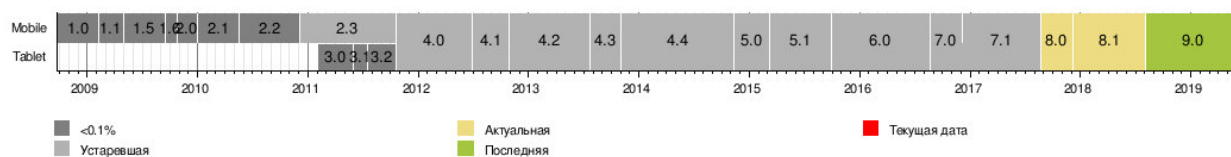
1.2 Android платформасы

Android – смартфондарға, планшеттерге, электрондық кітаптарға, сандық ойнатқыштарға, қол сағаттарына, фитнес білезіктерге, ойын тіреулеріне, ноутбуктерге, нетбуктерге, смартбуктерге, Google Glass көзілдіріктеріне, теледидарларға және басқа құрылғыларға арналған Операциялық жүйе (2015 жылы автомобиль ойын-сауық жүйелеріне және тұрмыстық роботтарға қолдау көрсетілді).

Linux ядросына негізделген және Google-дан Java виртуалды машинасын өз іске асыру. Бастапқыда Android, Inc компаниясы әзірлеген., содан кейін Google сатып алды. Кейіннен Google қазір платформаны қолдау және одан әрі дамытумен айналысатын Open Handset Alliance (ОНА) альянсын құруға бастама жасады. Android Google кітапханалары арқылы құрылғыны басқаратын Java-қосымшаларды жасауға мүмкіндік береді. Android Native Development Kit Си және басқа тілдерде жазылған кітапханаларды және қосымшалардың компоненттерін бүлдіруге мүмкіндік береді. 2014 жылдың екінші тоқсанында сатылған смартфондардың 86 % - ында

Android операциялық жүйесі орнатылды. 2017 жылдың мамыр айында әзірлеушілерге арналған конференцияда Google Android тарихында 2 млрд-тан астам Android құрылғысы іске қосылғанын жариялады.

Обновления версий Android



1-сурет. Android нұсқаларының жаңаруы

Android басқаруында жұмыс істейтін алғашқы құрылғы HTC смартфоны HTC Dream (T-Mobile ұялы операторымен T-Mobile G1 деп аталатын ресми түрде шығарылған) болды. Смартфондар мен планшеттерден Басқа Android операциялық жүйесін басқа құрылғыларға орнатады. Мысалы, 2009 жылдың соңында Android-та жұмыс істейтін алғашқы фото рамка сатылымда пайда болды. 2011 жылдың маусым айында Blue Sky итальяндық компаниясы Android операциялық жүйесінің басқаруымен I'm Watch интеллектуалды қол сағаттарын шығаруды жариялады. 2012 жылдың тамыз айында Nikon әлемдегі алғашқы фотокамераны ұсынды. "Google Nexus" сериясында смартфондар мен планшеттер ғана емес, Android және Nexus Player жұмыс істейтін Nexus Q медиаплеері де бар. 2012 жылдың қазан айында компанияның атқарушы директоры Ларри Пейдж Android базасындағы 500 миллионнан астам смартфондар мен планшеттер белсендірілгендігін хабарлады, сондай-ақ күн сайын осы операциялық жүйе базасында 1,3 миллион құрылғы іске қосылатынын мәлімдеді. 2013 жылдың қыркүйек айының басында әлемде Android-та миллиардтан астам құрылғылар іске қосылғаны туралы жарияланды. 2015 жылдың 29 қыркүйегінде CEO Google Сундар Пичай Android негізіндегі құрылғыларды пайдаланушылар саны 1,4 млрд-тан асқанын атап өтті.

Бағдарламалық қамтамасыз ету.

Android қосымшаларын жасау Java тілінде жүргізуге болады (Java 1.5 төмен емес). Eclipse — android Development Tools (ADT) үшін плагин бар, ол 3.3—3.7 Eclipse нұсқаларына арналған. Сондай-ақ, IntelliJ IDEA үшін Android-қосымшаларды әзірлеуді жеңілдететін плагин бар[64], және NetBeans IDE әзірлеу ортасы үшін. Сонымен қатар, Motodev Studio for Android — Google SDK-мен тікелей жұмыс істеуге мүмкіндік беретін Eclipse базасында кешенді даму ортасы бар.

2009 жылы ADT-ке қосымша Android Native Development Kit (NDK) жарияланған болатын. NDK жылдамдыққа сыни код учаскелерін әзірлеу үшін пайдалану ұсынылады.

2013 жылы Google JetBrains ұсынған IntelliJ IDEA негізінде Android Studio әзірлеудің жаңа ортасын ұсынды.

2013 жылы Embarcadero RAD Studio — XE5 шығарылды. Android платформасына арналған нативті қосымшаларды әзірлеу мүмкіндігі. Android қолданбасын жасау процесі Android құрылғысынан басқа қосымша құрылғыларды талап етпейді (негізінде эмулятормен де жұмыс істеуге болады).

Android 4.4-те Dalvik виртуалды машинасын ART (Android Runtime) ауыстыру мүмкіндігі пайда болды. ART жоғары жүктеу жылдамдығымен ерекшеленеді. Жадты оңтайландыру тетігі пысықталды.

Android 5-те Material Design тұжырымдамасына негізделген дизайн қайта суреттелген, Project Volta энергия үнемдеу режимі қосылды, Dalvik орнына машинаны таңдау жоғалып кетті ART.

Android 6-да Doze энергия шығынының зияткерлік режимі және бұрыннан қолданылмайтын App Standby бағдарламалары аясында Интернетке шығуға және жұмыс істеуге тыйым салу пайда болды.

Артықшылықтары.

Кейбір шолушылар Android өзінің бәсекелестерінің бірі Apple iOS, веб-серфинг, Google сервистерімен интеграция сияқты бірқатар ерекшеліктерде өзін жақсы көрсететінін айтады. Сондай-ақ, Android, iOS қарағанда, ашық платформа болып табылады, бұл көптеген әзірлеушілердің функцияларын жүзеге асыруға мүмкіндік береді.

"Тексерілмеген көздерден" (мысалы, жад картасынан) бағдарламаларды орнатуға бастапқы тыйым салғанына қарамастан, бұл шектеу аппараттың баптауларында штаттық құралдармен ажыратылады, бұл интернет қосылысынсыз телефондар мен планшеттерге бағдарламаларды орнатуға мүмкіндік береді, сондай-ақ барлық ниет білдірушілерге Android үшін қосымшаларды тегін жазуға және өз аппаратында тестілеуге мүмкіндік береді.

Android ARM, MIPS, x86 сияқты түрлі аппараттық платформаларға қол жетімді.

Google Play қосымша дүкендері бар: Amazon Appstore (ағылш.), Opera Mobile Store, Yandex.Store, GetUpps!, Mobogenie, F-Droid, Imobile Market, Meizu Appstore.

1.3 Android studio бағдарламасы және орнатылымы

Android studio бағдарламасы және орнатылымы
Android Studio – бұл 2013 жылдың 16 мамырында Google I/O конференциясында анонсталған Android платформасымен жұмыс істеу үшін біріктірілген даму ортасы (IDE). IDE 2013 мамыр айында жарияланған 0.1 нұсқасынан бастап, содан кейін 2014 жылдың маусым айында шығарылған 0.8 нұсқасынан бастап бета-тестілеу сатысына өтті. 1.0 бірінші тұрақты нұсқасы 2014 жылдың желтоқсанында шығарылды, сонда Eclipse үшін Android Development Tools (ADT) плагинін қолдау тоқтатылды. JetBrains компаниясынан IntelliJ IDEA бағдарламалық жасақтамасына негізделген Android Studio - Android қосымшаларын әзірлеудің ресми құралы.

Бұл жұмыс ортасы Windows, OS X және Linux үшін қол жетімді. 17 мамыр 2017, Google I / O жыл сайынғы конференциясында, Google Java және C++ қосу Android платформасына арналған ресми бағдарламалау тілі ретінде Android Studio-да қолданылатын Kotlin тілін қолдауды жариялады.

Ерекшеліктері.

Жаңа мүмкіндіктер әрбір жаңа Android Studio нұсқасымен пайда болады. Қазіргі уақытта келесі мүмкіндіктер бар:

- кеңейтілген макеттер редакторы: WYSIWYG, Drag-and-Drop көмегімен UI компоненттерімен жұмыс істеу қабілеті, экранның бірнеше конфигурациясында макетті алдын ала қарау функциясы;
- Gradle негізделген бағдарламаларды құрастыру;
- құрастырудың әртүрлі түрлері және бірнеше генерация .apk файлдар;
- код рефакторингі;
- статикалық код талдағышы (Lint), өнімділік, нұсқалардың үйлесімсіздігі және т.б. мәселелерін табуға мүмкіндік береді;
- кірістірілген ProGuard және қолданбаларға қол қою үшін утилитасы;
- негізгі макеттер мен Android компоненттерінің үлгілері;
- Android Wear және Android TV үшін бағдарламаларды дамыту;
- Google Cloud Messaging және App Engine сервистерімен интеграцияны қамтитын Google Cloud Platform қолдау;
- Android Studio 2.1 Android N Preview SDK қолдайды, бұл әзірлеушілер жаңа бағдарламалық платформаға арналған қосымшаны жасау бойынша жұмысты бастай алады;
- Android Studio 2.1 жаңа нұсқасы жаңартылған Jack компиляторымен жұмыс істей алады, сондай-ақ Java 8 қолдауы мен жетілдірілген Instant Run функциясын алды;
- Platform-tools 23.1.0 бастап Linux үшін тек 64 биттік;
- Android Studio 3.0 стандарты бойынша JetBrains IDE-ге негізделген Kotlin тілінің құралдарын қамтиды.

Kotlin тілінің құралдарын қамтиды.

Android Studio орнату.

Ең алдымен, сізде JDK (Java Development Kit) орнатылғанына көз жеткізу керек. Бұл Java-да әзірлеу үшін міндетті компонент, ал Android-та әзірлеу Java — да және Android-та әзірлеу үшін жүргізіледі.

JDK осы сілтеме бойынша жүктеп алуға болады (ресми сайт, өзіңіз, JDK тегін) ең жаңа JDK таңдаңыз, және дәл JRE емес, JDK!

Енді Android Studio орнату өтіңіз.

Бастау үшін, Android Studio жүктеу қажет. Бір құрылғыда барлық қажетті болады өзі, Android Emulator, Android SDK. Жинақта жоқ нәрсе, инсталлятор өзі шыңдайды.

Тікелей орнатуға өтеміз. Онда ерекше ештеңе болмайды — инсталлятор диалогы. Процесте тек бір маңызды сұраққа жауап беру керек, және бұл опциональды сұрақ. Бұл жерде, орнатушы студияны қайда қою керектігін және SDK қайда қою керектігін сұрайды. SDK мұқият болу керек.

Скриншоттан кейін SDK орнату үшін дискідегі кем дегенде 3.2 GB орын қажет. Бұл минимум, шын мәнінде, орын көп қажет, өйткені біраз уақыттан кейін сіз жаңартылған SDK-ны жүктеп алу керек. сондықтан, егер сіз болашақта орын жетілікті деп сенімді емес болса-орынды кең дискіге өзгерту жақсы. Содан кейін стандартты бірнеше рет "әрі қарай" батырмасын басу керек және Android Studio орнату аяқталды.

Android Studio бірінші рет іске қосылғанда конфигурацияны импорттау туралы стандартты сұрақ қояды. Әдепкі бойынша сол чекбокс таңдалып, "OK" батырмасын басыңыз. Осыдан кейін Android Studio Android SDK жүктеу бастайды. Бұл біраз уақыт алуы мүмкін. Егер жүктеу сәтсіз болса, IDE тағы бір рет көріңіз — "Retry" түймесін басыңыз. Жүктеу аяқталғаннан кейін "Finish" түймесін басыңыз. Негізінде, бұл орнату аяқталды, бірақ мен кейбір баптауларды жасауды ұсынар едім:

- біріншіден, мен тақырыпты қараңғы ("Darcula") деп өзгертуге кеңес беремін;

- екіншіден, «ширек нөмірін көрсету» және «көрсету әдісінің бөлгіштері» деген белгі қойыңыз;

- үшіншіден, автотолтыруды орнатыңыз. Мұны орындау үшін «Кескіннің сезімтал аяқтауында» «Жоқ» дегенді таңдаңыз.

Неге дәл солай екенін түсіндіремін. Әдепкі параметр автодөңдеу тек бірінші әріп дұрыс тіркелімде терілген жағдайда ғана іске қосылады. None параметрі сіз кодты тере бастағанына қарамастан, автоматты түрде толтыруды туындататын болады.

1.4 Android studio бағдарламасы және орнатылымы

Android –Java бағдарламалану интерфейсіне ие Linuxқа негізделген тегін операциялық жүйе. Android, OpenGL кітапханасын пайдалана отырып, 2D және 3D-графикасын ұстанады, сонымен қатар деректерді SQLite деректер қорында сақтайды.

Әр Android-қосымшасы өзінің меншікті процесінде қосылады. Сондықтан қосымша, басқа қосылған қосымшалардан сақталған, және дұрыс жұмыс жасамайтын қосымшалар кедергісіз басқа қосымшаларға кесірін тигізбейді. Activity (Белсенділік, Қызмет) — Activity классымен ұсынылған және XML-файл түрінде белгіленген қолдану интерфейсі экранын ұсынады. (белгіленуді бағдарламалау арқылы да жасауға болады, бірақ қазір олай жасамайды). Android-қосымшасы бірнеше формада тұруы мүмкін және қосымшаларды орындауда өзара қосылыса алады.

- Intents (Намерения) — асинхрондық хабарлама, бұл дегеніміз басқа қызметтерден функция алу мүмкіншілігін тудырады. Қосымша қызметке тікелей сұраныс жасай алады немесе Androidтан тіркелген қызметтер сұрай алады.

- Мысалы, қосымша, Intent арқылы деректер алу үшін немесе браузерді қосу үшін сұрауы мүмкін.

– Views (Представление, Түр, Басқару элементі, Компонент) — виджеттермен класс құратын қолданыс интерфейсі. Бұлар өзімізге таныс басқару элементтері: кнопкалар, мәтін кеңістігі, жалаушалар, қосқыштар және т.б.

– Services (Қызметтер) — қолданыс интерфейсын қолданбай фондық талаптарды орындай алады. (мысалы, әуенді ойнату). Олар қолданушыны Android ескерту жүйесі арқылы ескертіп отырады.

– Content Provider (Контент-провайдерлері) — қосымшаларға деректерді контент-провайдер арқылы жеткізеді. Сіздің қосымшаңыз басқа қосымшалармен дерек алмаса алады.

– Broadcast Receiver (Хабарламаларды кеңінен тарататын қабылдағыштар) — жүйелік хабарламаларды қабылдайды, жүйе жағдайының өзгеруіне де әсер ете алады. Қосымша белгілі бір анықталған қабылдағыш ретінде де тіркеле алады, егер ондай хабарлама болған жағдайда ғана қоысла алады.

Егер сіз өзіңіз Android басқармасын пайдалан отырып телефондарға қосымшалар ойлап тапқыңыз келсе, онда сізге жақсылап дайындалып, жұмысқа арналған барлық аспаптарды дайындап алу қажет.

1.4 Z Reader Music Player қосымшасы

Ұялы телефондардың дамуына, олардың технологиялық өсуіне қарай, музыкалық плеерлер жеке құрылғылар ретінде өте танымал. Мобильді құрылғылар жадына (Android/iOS) музыка гигабайттарын (mp3, flac, vorbis және т.б.) жүктеуге және оны жақсы сапада тыңдау болады. Егер дыбыс сапасы ұнамаса-эквалайзерді ширату қиын емес және сол арқылы дыбысты қанық етуге болады. Бұл ойнатқыш болашақта стильді түрлі түсті тақырыптарды таңдау мүмкіндігімен қарастырылатын, сондай-ақ қарапайым және сол уақытта өте стильді көрініс тауып ерекшеленетін қосымша. Форматтардың бір бөлігі бойынша қосымша MP3 және WAV, OGG, MIDI және FLAC ойнатады. Дыбысты реттеу үшін түрлі баспақтар мен бас реттегіші бар кіріктірілген эквалайзер бар. Ыңғайлы свайппен басқарылатын, папкалар бойынша тректерді ойнату мүмкіндігі және орындаушылар, жанрлар және басқа да өлшемдер бойынша жылдам іздеу мүмкіндігі болашақта қарастырылатын жаңа плеер. Ойнату прогресі композицияның қажетті сәттерін іздеуді жеңілдететін спектрограмма түрінде көрсетілетін болады. Онда дыбысты жақсарту үшін көптеген мүмкіндіктер бар, сондықтан музыканы әдетте басқаша реттеуге болады. Сонымен қатар, ол эквалайзермен (32 пресеттермен бірге), бас күшейту, тегтер редакторы, виджеттер және тіпті MIDI ойнату сияқты қарапайым әсерлермен жабдықталған. Тегін және ақылы нұсқалар дерлік бірдей. Ақылы нұсқа жарнаманы алып тастайды және тақырып қосады.

Дыбыс сигналын өңдеу-дыбыс сигналдарын электрондық манипуляциямен байланысты сигналды өңдеу болып табылады. Дыбыс

сигналдары дыбыс толқындарының электрондық көріністері – қысулар мен сиретулерден тұратын ауа арқылы өтетін бойлық толқындар екенін ескеру қажетпіз. Дыбыс сигналдарындағы энергия, әдетте децибелдер ретінде өлшенеді. Аудио сигналдар сандық немесе аналогтық пішімде ұсынылуы мүмкін болғандықтан, өңдеу кез келген аймақта жүргізілуі мүмкін. Аналогтық процессорлар тікелей электр сигналында жұмыс істейді, ал цифрлық процессорлар оның сандық көрінісінде математикалық түрде жұмыс істейді.

Аналогтық дыбыс сигналы электр кернеуімен немесе ауадағы дыбыстық толқындарға "ұқсас" токпен ұсынылған үздіксіз сигнал болып табылады. Аналогтық сигналды өңдеу осыдан кейін электр тізбектері арқылы ток кернеуін немесе ағымын немесе міндетін өзгерту жолымен үздіксіз сигналды физикалық өзгертуді қамтиды.

Тарихи, кең таралған цифрлық технологиялар пайда болғанға дейін аналогты сигналмен манипуляциялаудың жалғыз әдісі болды. Компьютерлер мен бағдарламалық қамтамасыз ету анағұрлым қабілетті және қол жетімді болған кезден бастап, сигналдарды цифрлық өңдеу тандау әдісі болды. Алайда, музыкалық қосымшаларда аналогтық технология жиі қажет, себебі ол жиі сандық сүзгілермен ойнату қиын бейсызық пікірлерді шығарады.

Цифрлық көрініс әдетте екілік сандардың символдар тізбегі түрінде дыбыс сигналын білдіреді. Бұл сигналдарды цифрлық сигналдық процессорлар, микропроцессорлар және жалпы мақсаттағы компьютерлер сияқты сандық схемаларды пайдалана отырып өңдеуге мүмкіндік береді. Сигналдарды цифрлық өңдеу әдістері сигналдарды аналогтық өңдеуге қарағанда әлдеқайда күшті және тиімді.

Z Reader Music Player қосымшасында біз күні бойы олар туралы айта алатын жеке ерекшеліктер көп. Міне, ол ұсына алатын кейбір мүмкіндіктерге қысқаша шолу ұсынамыз. Алдын ала орнатылған пресеттер, төмен және жоғары жиілікті реттегіштері бар эквалайзер және тонды және дыбыс деңгейін реттеу үшін жеке қойындылар бар. Егер сіз бәрін жақсы баптасаңыз, FLAC пішімін тыңдау кезінде әсіресе байқалатын дыбыс сапасын жақсарта аласыз. Ол сондай-ақ mp3, mp4/m4a (alac), ogg, WMA, flac, wav, ape, wv, tta, mpc және aiff пішімдерін ойнатады. Егер сіз белгілі бір музыкаға ұйықтап қалғыңыз келсе, онда сіз үшін жақсы жаңалық бар, бұл жерде ұйқы таймері бар, ол сіз ұйықтауға және бүкіл батареяны разрядтайтын музыканы қосулы қалдырыңыз туралы алаңдамауға мүмкіндік береді. Z Reader Music Player қосымшасы плагиімен ән мәтіндерін іздеуге болады. Тректердің басында және соңында тыныштық режиміне қоюға болады. Орыс тілінің қолдауы бар.

Z Reader Music Player қосымшасы ұялы музыка ойнатқышынан күтуге болатын барлық мүмкіндіктерге негізделген, оның ішінде аудио және бейне қолдау, подкаст қолдау, санаттар бойынша навигация, пайдаланушы үшін ыңғайлы интерфейс, күшейту коэффициентін теңшеу, эквалайзер және тіпті ұйқы таймері қосылған. Жанр, орындаушы, альбом немесе атау бойынша тректерді іздеу мүмкіндігі бар. Бірақ Windows үшін Z Reader Music Player қосымшасын қолдансаңыз, Wi-Fi арқылы синхрондау мүмкіндігі сізге

пайдалы болуы мүмкін. Мұнда сіз плейлисттерді синхрондау, ойнату саны, рейтинг, коллекцияға ән қосу күні және компьютерде бағдарлама сақтайтын басқа да метадеректер аласыз. Жақында плеер смартфон функцияларын автомобильдің мультимедиалық жүйесіне айналдыруға арналған Android Auto бағдарламасын қолдай бастады. Енді сіз сүйікті әндерді тыңдап, жолда ләззат ала аласыз.

Бізге бірде белгілі, бірде белгілі бола бермейтін әр түрлі дыбыс әсерлерін қарастырайық. Дыбыс әсерлері – дыбыс сигналын өзгертуге арналған жүйелер. Өңделмеген дыбыс метафорикалық құрғақ деп аталады, ал өңделген дыбыс ылғалды деп аталады.

- кідіріс немесе эхо – үлкен залда немесе үңгірде реверация әсерін имитациялау үшін бастапқы сигналға бір немесе бірнеше кідіріс сигналдары қосылады. Эхо ретінде қабылданады, кідіріс шамамен 35 миллисекунд және одан жоғары болуы керек. Талап етілетін ортада дыбысты ойнатудан басқа, эхо әсері сандық немесе аналогтық әдістермен іске асырылуы мүмкін. Аналогтық Эхо-әсерлер таспалы кідірістер немесе шөміш құрылғылар арқылы іске асырылады. Көптеген кідірген сигналдарды араластыру кезінде реверберация әсері туындайды. Қорытынды дыбыс үлкен бөлмеде әсер етеді.

- flanger-ерекше дыбыс жасау үшін бастапқы сигналға баяу өзгеретін кідірісі бар кідірілген сигнал қосылады (әдетте 10 миллисекундан кем). Бұл әсер енді DSP көмегімен электрондық тәсілмен орындалады, бірақ бастапқыда екі синхрондалған магнитофонда бір жазбаны ойнату, содан кейін сигналдарды бірге араластыру арқылы жасалған. Машиналар синхрондалған дейін, қоспа көп немесе аз қалыпты естіледі, бірақ оператор ойыншылардың бірінің фланеціне саусақты орналастырса (яғни, "flanger"), бұл машина баяулайды, және оның сигналы фазалық тарақ Сүзгіш әсерін жасау арқылы өзінің әріптесімен фазадан түседі. Оператор саусақты алғаннан кейін, ойыншы шебермен фазаға оралғанға дейін жылдамдады, және ол болған кезде фазаның әсері жиілік спектрі бойынша жоғарыға жылжыды. Бұл фазалау жоғары-төмен Регистр ырғақты орындалуы мүмкін.

- фазер-ерекше дыбысты жасаудың тағы бір тәсілі; сигнал бөлінеді, бөлік фазалық жылжуды алу үшін барлық жиіліктердің ауыспалы сүзгісімен сүзіледі, содан кейін сүзілмеген және сүзілген сигналдар тарақ сүзгісін алу үшін араласады. Фазердің әсері бастапқыда флангердің әсерін қарапайым іске асыру болды, өйткені кідірістер аналогтық жабдықпен іске асыру үшін қиын болды.

- қайырмада-кідірілген сигнал нұсқасы бастапқы сигналға қосылады. Кідіріс эхо ретінде қабылданбау үшін қысқа болуы керек, бірақ 5 мс жоғары естілуі керек. Егер кідіріс тым қысқа болса, ол undelayed сигналымен бұзады және кері әсерін жасайды. Жиі кідірген сигналдар бірнеше дауыстың әсерін шынайы жеткізу үшін биіктігі бойынша сәл жылжиды.

- теңестіру-түрлі жиіліктік жолақтар талап етілетін спектралдық сипаттамаларды алу үшін әлсірейді немесе күшейтіледі. Біркелкі туралауды

пайдалану (жиі "эквалайзер" ретінде қысқартылған) жазба тонының сапасын "жұқа теңшеу" үшін пайдаланылуы мүмкін; экстремалды туралауды пайдалану, мысалы, белгілі бір жиіліктің күшті қысқаруы ерекше әсерлерді тудыруы мүмкін.

- сүзу-тегістеу сүзу формаларының бірі болып табылады. Жалпы мағынада жиілік диапазондары төмен жиілікті, жоғары жиілікті, жолақтық немесе жолақтық сүзгілердің көмегімен астын сызылуы немесе әлсіреуі мүмкін. Дауыс жолағын сүзу телефон әсерін еліктеуі мүмкін, себебі телефондар жолақ сүзгілерін пайдаланады.

- түбіт қорабын пайдалану сияқты шамадан тыс жүктеу әсерлері роботталған дауыстарды имитациялау үшін немесе бұрмаланған радиотелефондық трафикті имитациялау үшін (мысалы, "Жұлдызды соғыс" ғылыми-фантастикалық фильміндегі жойғыш ұшқыштардың арасындағы радиоболт) бұрмаланған дыбыстарды өндіру үшін пайдаланылуы мүмкін. Овердрайвтің ең негізгі әсері оның абсолюттік мәні белгілі бір шектен асып кеткен кезде сигналды ажырату болып табылады.

- тонның биіктігінің жылжуы-бұл әсер сигналды тонның биіктігі бойынша жоғары немесе төмен жылжытады. Мысалы, сигнал октаваға жоғары немесе төмен жылжытылуы мүмкін. Бұл әдетте әрбір нотаға емес, барлық сигналға қолданылады. Бастапқы сигналды ығыстырылған дубликаттармен араластыру бір дауыстан үйлесім жасай алады. Тонның биіктігін жылжытудың басқа қолданылуы тонның биіктігін түзету болып табылады. Мұнда музыкалық сигнал сандық өңдеу әдістерін пайдалана отырып, дұрыс үнге теңшейді. Бұл әсер караоке-машиналарда және жиі фальшиво әншілерге көмектесу үшін қолданылады. Ол сондай-ақ Cher ' s Believe және Madonna Die Another Day сияқты поп-әндерде эстетикалық әсер үшін әдейі қолданылады.

- уақыт бойынша созылу-тонның жылжуын толықтыру, яғни дыбыс сигналының жылдамдығын оның биіктігіне әсер етпей өзгерту процесі.

- резонаторлар-берілген жиіліктердегі гармоникалық жиілік мазмұнын көрсетеді. Олар параметрлік эквалайзерлерден немесе тежелу негізінде тарақ сүзгілерден жасалуы мүмкін.

- роботталған дауыстық әсерлер актердің дауысы синтезделген адам дауысы ретінде естіледі.

- модуляция-алдын ала анықталған сигналға қатысты тасымалдау сигналының жиілігінің немесе амплитудасының өзгеруі. Сондай-ақ, амплитудалық модуляция ретінде белгілі сақиналы модуляция-бұл ғылыми фантастикада кең қолданылатын, доктор кімдердің алыс жерлерімен белгілі әсер.

- қысу-динамикадағы кездейсоқ ауытқуларды болдырмау үшін динамикалық дыбыс ауқымын азайту. Сығу деңгейі дыбыс амплитудасына әсер етпей, деректер көлемі азаятын дыбыс қысумен шатастырмау керек.

- 3D аудио effects-стерео базадан тыс дыбыс орны.

- кері Эхо - сигнал кері бағытта жұмыс істеп тұрғанша дыбыс сигналын реверсиялау және Эха жазу және / немесе кідірту арқылы жасалатын дауысталу әсері. Артқа бұрын ойнаған кезде, соңғы Жаңғырық дыбыстың алдында естіледі. Led Zeppelin Джимми Пейдж бұл әсерді "Whole Lotta Love"фильмінде пайдаланды.

- толқын өрісінің синтезі-виртуалды акустикалық орталарды құру үшін дыбыстың кеңістіктік рендеринг әдісі

1.5 Z Reader Music Player қосымшасының жұмыс істеу принципі: дерекқор файлы, түзету, бөліп алу

Z Reader Music Player қосымшасының ішінде ең алғашқы бетінде дерекқор файлы енгізілген. Оның ішінде түрлі жұмыс істеу жолдары қарастырылған. Осы қойынды арқылы келесі әрекеттерді орындай аласыз:

- жобаны жасау – Ctrl+N жүйесі арқылы жүзеге асады;
- жобаны ашу – Ctrl+O жүйесі арқылы жүзеге асады;
- соңғы файлдар батырмасы арқылы өшірілмеген, жақында жасалынған іс-әрекеттерге көшу қарастырылған;
- жобаны жабу – Ctrl + W жүйесі арқылы жүзеге асады;
- жобаны сақтау батырмасы арқылы қалай, қандай файлға және қандай тақырыппен сақтау әдістері ұсынылған;
- экспорт арқылы дайын өнімді басқа құрылғыларға сақтау қарастырылған;
- импорт арқылы дайын немесе дайын емес өнімді басып, өзгерту қарастырылған;
- бет параметрлері;
- басып шығару;
- жабу.

Дерекқор файлдарынан кейін Z Reader Music Player қосымшасында түзету функциясы енгізілген. Оның ішінде:

- жобаны болдырмау – Ctrl + Z жүйесі арқылы жүзеге асады;
- жобаны қайталау Ctrl + Z жүйесі арқылы жүзеге асады;
- бөліп алу – Ctrl + X жүйесі арқылы жүзеге асады;
- өшіру – Ctrl + K жүйесі арқылы жүзеге асады;
- көшіру – Ctrl + C жүйесі арқылы жүзеге асады;
- қою – Ctrl + V жүйесі арқылы жүзеге асады;
- дубликат – Ctrl + D жүйесі арқылы жүзеге асады;
- арнайы өшіру батырмасы арқылы өшірілмеген, жақында жасалынған іс-әрекеттерге көшу қарастырылған;
- фрагменттің шекаралары;
- белгілер;
- белгіленген аудиолар – бірнеше сақалған әуендер болған жағдайда;

- метадеректер – бұл пайдаланылатын деректер туралы субканалды ақпарат;

- параметрлері берілген.

Түзету функциясынан кейін бөліп алу функциясы жұмыс атқарады, оның ішінде:

- барлығы;

- барлығын жаңалау;

- тректер;

- аудан;

- спектр;

фрагменттің шекаралары қойылған.

1.6 Z Reader Music Player қосымшасының жұмыс істеу принципі: көрініс, транспорт, тректер, құру

Көрініс функциялары өнімнің сыртқы бетіне әсерін жүргізеді:

- масштаб батырмасы арқылы медиафайл көп жадын алып отырмас үшін, қысу қосымшасы қарастырылған;

- тректің өлшемі;

- белгілі бір уақытқа дейін жіберу;

- тарихы батырмасы арқылы белгілі-бір әуеннің жасалу жолын оқуға болады;

- караоке;

- панельдер;

- экстра-меню;

- клиппинг әдісі – күшейткіштің шығыс кернеуінен қуат кернеуінің шегі асып кеткен кезде сигнал амплитудасын шектеуде көрінетін дыбыстың бұрмалау нысаны.

Транспорт функцияларының қызметтері:

- ойнату;

- жазу;

- айналдыру батырмасы арқылы тезірек алдыға айналдыру функциясы қарастырылған;

- курсор;

- ойнату ауданы;

- аудио құрылғыларды қайта іздеу;

- транспорт опциялары.

Тректер функциялары:

- жаңасын қосу;

- микс;

- тректерді өшіру;

- дауысын қосу/сөндіру;

- паронамалау - камераның тік немесе көлденең осьтің айналасында бір мезгілде бірқалыпты бұрылысымен кино - немесе бейнетүсірілім. Үлкен кеңістікті немесе орын ауыстыратын нысандарды түсіру үшін қолданылады.;

- тректерді түзеу компьютерлік бағдарламасы болып табылады. Жеке алынған файлдың және тұтас пакеттік файлдар тобының (мысалы, кез келген музыкалық альбомда) әуеннің дыбысын жөндеу;

- тректерді сұрыптау;
- тректерді қадамдасыру.

Құру функциясы:

- плагиндер менеджері бұл – плагиндерді орнатуға, жаңартуға және жоюға мүмкіндік беретін плагин. Орталық орналасумен XML файлы жүктейтін және өңдейтін плагиндер тізімін және орнатылған плагиндер тізімін қамтиды;

- DTMF-тон – тоналды жиынтық, тоналды сигнал. Телефон нөмірін теру үшін пайдаланылатын екі тонналы көп жиілікті аналогтық сигнал;

- тыныштық;
- тон;
- чирп-сигнал – сызықтық жиіліктік модуляциясы бар сигнал;
- шу;
- барабан-риссета;
- плак-звук дауыс синтезі;
- ритм-трек.

1.7 Z Reader Music Player қосымшасының жұмыс істеу принципі: эффекттер, анализ, құралдар, анықтамалық

Эффекттер қамтитын тізім:

- плагиндер менеджері бұл – плагиндерді орнатуға, жаңартуға және жоюға мүмкіндік беретін плагин. Орталық орналасумен XML файлы жүктейтін және өңдейтін плагиндер тізімін және орнатылған плагиндер тізімін қамтиды;

- соңғы эффекті қайталау;

- EQ(эквалайзер) - жиіліктік сипаттамаларына (биіктікке, дыбыс тембріне) байланысты амплитудасын таңдап түзетуге мүмкіндік беретін жоғары класты стереофониялық кешендер (Hi-Fi) құрамындағы радиоэлектрондық құрылғы немесе компьютерлік бағдарлама;

- автоприглушение;
- басстар және вч;
- вау-вау;
- қалпына келтіру;
- тонның биіктігін өзгерту;
- жылдамдықты өзгерту;
- темпті өзгерту;

- инвертизациялау - бұл нақты деректерді есеп деректерімен салыстыру жолымен әуеннің ішіндегі бит және басстардың бар-жоғын және жай-күйін тексеру;

- бұрмалау;

- компрессор - бұл дыбыстық сигналдың динамикалық диапазонын азайтатын электрондық құрылғы немесе компьютерлік бағдарлама; басқаша айтқанда, компрессор ең тыныш және ең қатты дыбыс арасындағы тар айырмашылықты жасауға мүмкіндік береді. Көбінесе компрессорлар кәсіби дыбыстық жабдыққа жатады, өйткені оларды тұрмыстық салада өте сирек кездестіруге болады;

- нормализация;

- тыныштықты жою;

- бірқалыпты созу;

- қайталау;

- шуды басу;

- радиалды созылу;

- реверберация – бұл әдіс кез-келген дыбыс жабық кеңістікте естілетін кезде пайда болады, соның нәтижесінде қабырғалардың беттерінің шағылысуы үлкен мөлшерде эхо тудырады, содан кейін дыбыс қабырғалар мен ауамен дыбыс толқындарының жұтылу себебі бойынша баяу өшеді.;

- реверс;

- шертулерді жою;

- күшейткіш;

- фейд-кіріс;

- фейд-шығыс;

- фейзер;

- жаңғырық;

- SC4;

- вокодер – бай спектрі бар еркін сигнал негізінде сөйлеу синтезінің құрылғысы. Байланыс жүйесінің радиолиниясының жиіліктік ресурстарын үнемдеу мақсатында әзірленген;

- кідіру;

- тректердің кроссфейді – бұл бір бүтін екі тректі жоюға мүмкіндік беретін қарапайым музыкалық бағдарлама;

- фрагменттердің кроссфейді.

Анализ жасау қызметіне:

- плагиндер менеджері бұл – плагиндерді орнатуға, жаңартуға және жоюға мүмкіндік беретін плагин. Орталық орналасумен XML файлы жүктейтін және өңдейтін плагиндер тізімін және орнатылған плагиндер тізімін қамтиды;

- контраст;

- спектр графигі әртүрлі меншікті жиіліктегі осцилляторлардың қатарының жоғары немесе белгіленген жауап беру (жылжу, жылдамдық немесе жылдамдату) кестесі;

- клиппингіні ұйымдастыру;
- үлес іздеуші;
- дыбыс іздеу техникасы;
- тыныштық поисковигі.

Құралдар ішіне:

- макрос;
- макрос қолдану;
- скришот;
- өнімділік сынағын бастау;
- Найквист командасының панелі – кешенді координаттарда кесте

түрінде желілік стационарлық динамикалық жүйенің жиіліктік үнін ыңғайлы ұсыну;

- плагиндер менеджері бұл – плагиндерді орнатуға, жаңартуға және жоюға мүмкіндік беретін плагин. Орталық орналасумен XML файлы жүктейтін және өңдейтін плагиндер тізімін және орнатылған плагиндер тізімін қамтиды

- сэмпл деректерін импорттау салыстырмалы түрде аз сандық дыбыс фрагменттерді басқа қосымшаларға сақтау;

- тұрақты аралықтардың белгілері;
- найквист плагиндерінің орнатушысы;

- сэмпл деректерін экспорттау салыстырмалы түрде аз сандық дыбыс фрагменттерді басқа қосымшалардан енгізу.

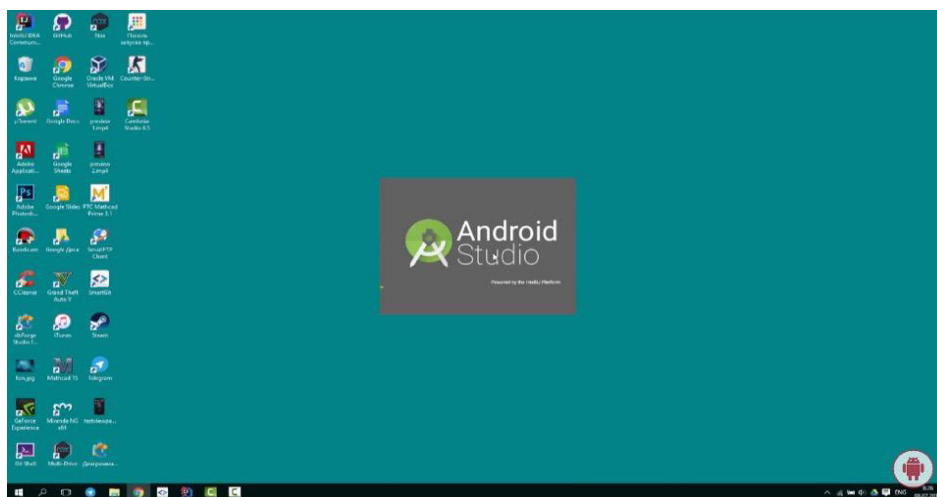
Анықтамалық:

- қысқаша мәлімет;
- көшбасшылық;
- диагностика;
- жаңарту жүйесін тексеру;
- Audacity жайлы.

2 Қосымшаны құру және оны іске асыруы

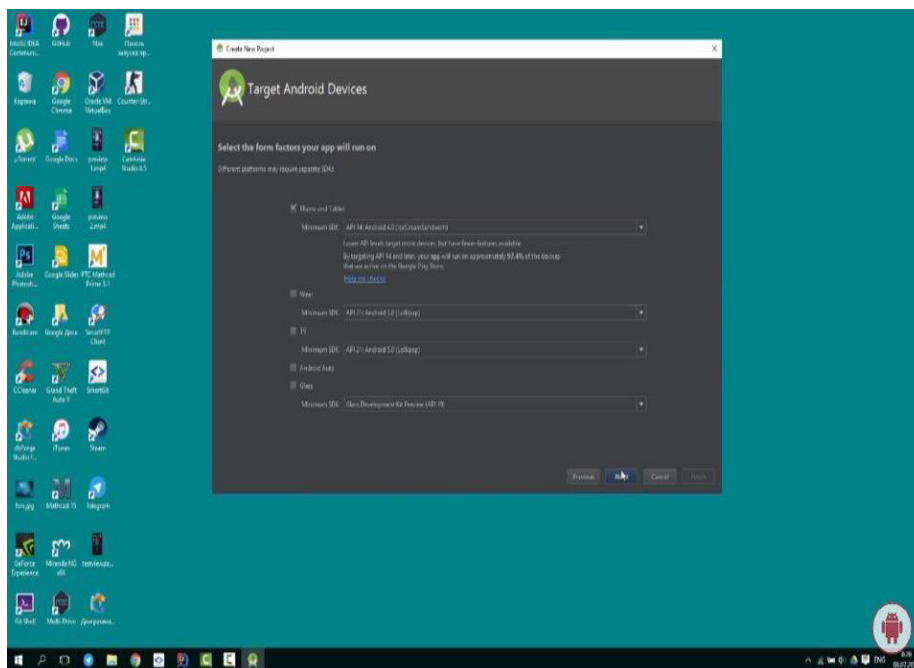
2.1 Бағдарламалық жүйені құрудың бастапқы кезеңі. Қажетті құралдарды дайындау

Бағдарламамен жұмыс істеу алдында, Android studio бағдарламасын қосу керек. (2.1-сурет).

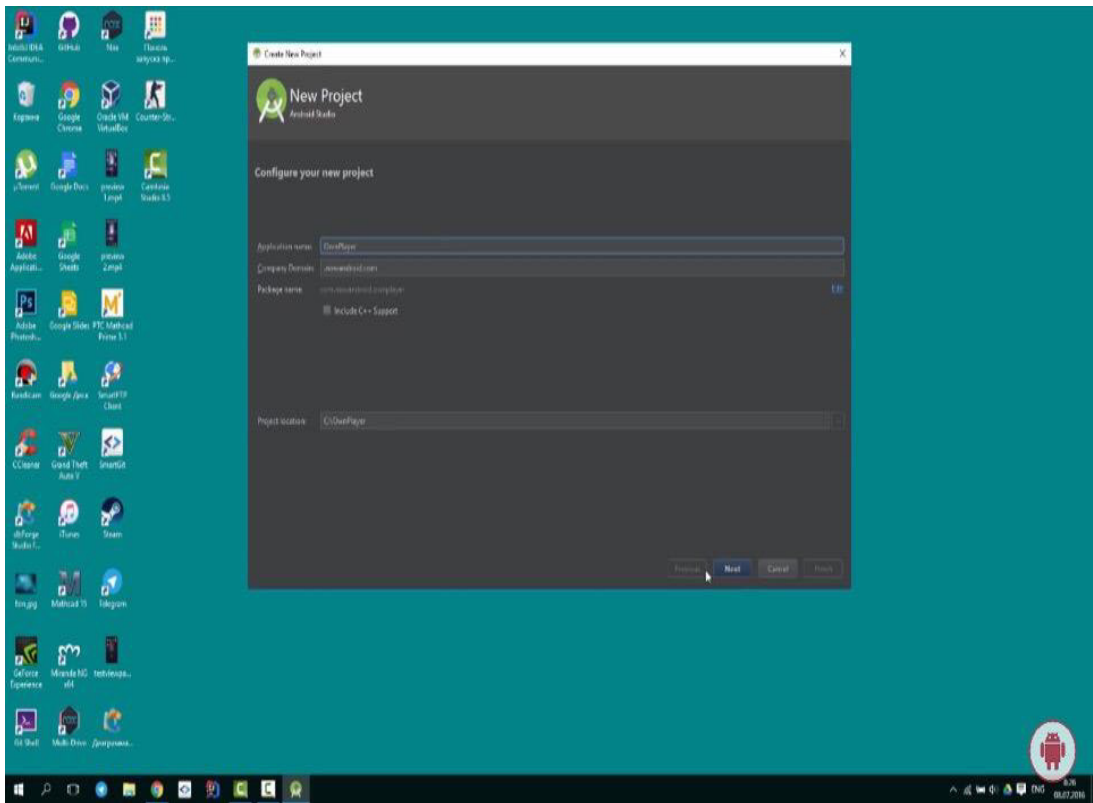


2.1-сурет – Android studio бағдарламасын қосу

Android studio бағдарламасында жоба құрудың бастапқы кезеңдерінің 2 әдісі бар. (2.1 және 2.2-суреттер)

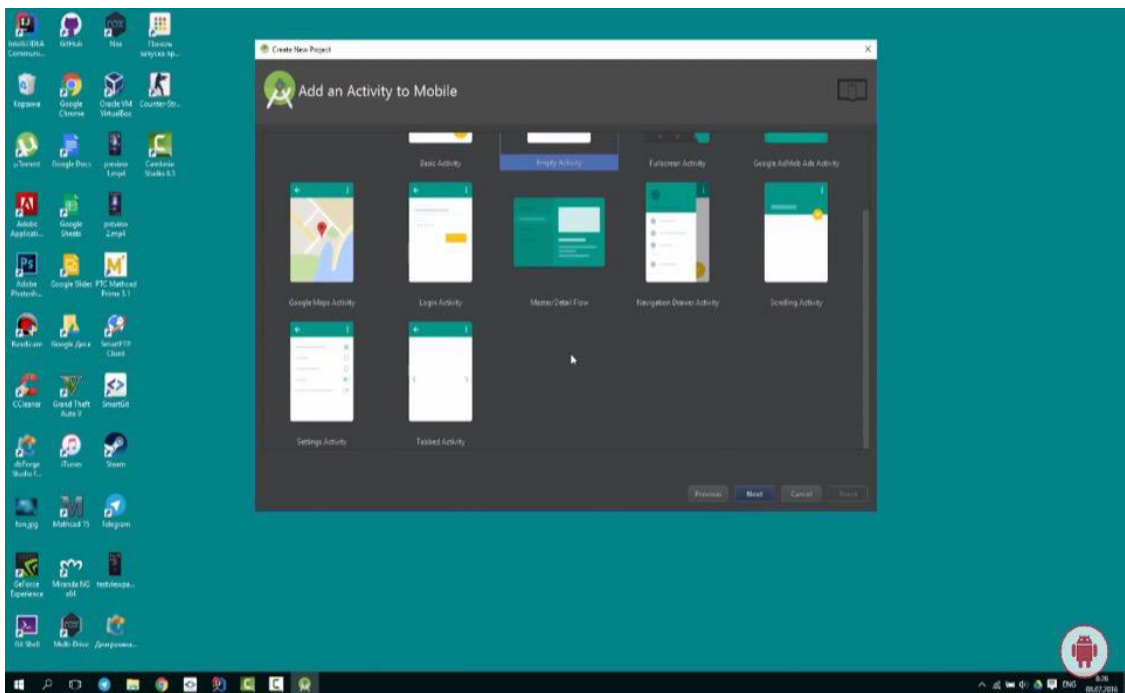


2.2-сурет – Android studio бағдарламасында жоба құру

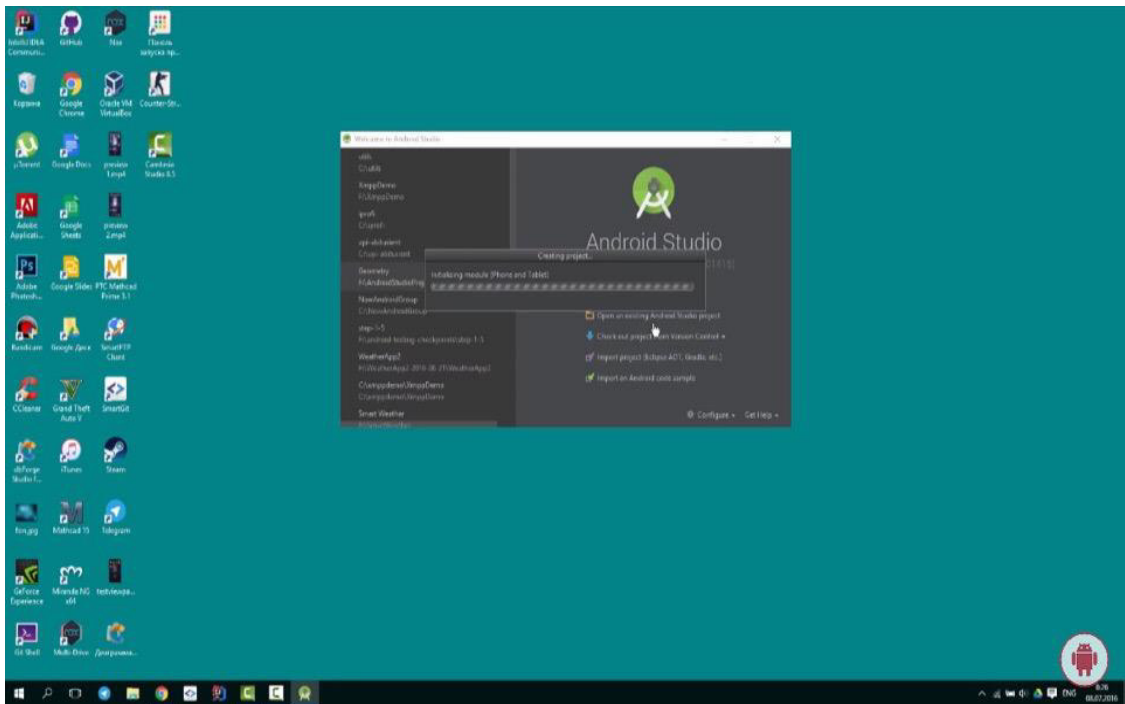


2.3-сурет – Android studio бағдарламасында жоба құру

Бағдарламалау ортасы ұсынған жобалау интерфейсін таңдау мүмкіндігі болады.

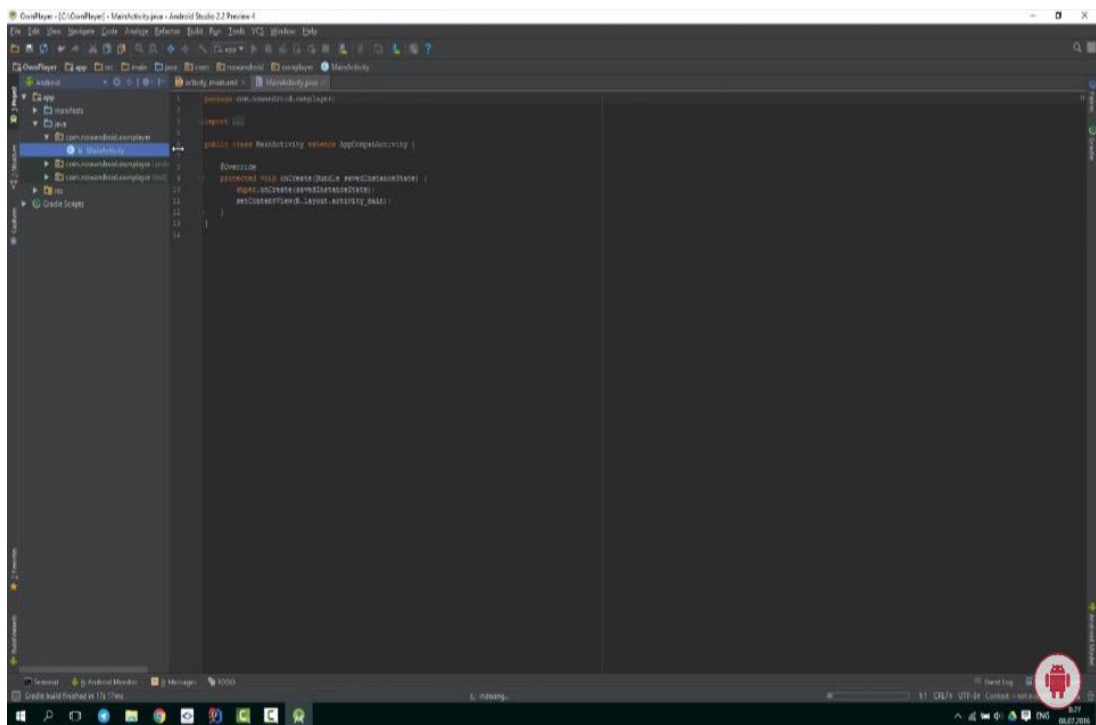


2.4-сурет – Жобаның интерфейсін таңдау беті

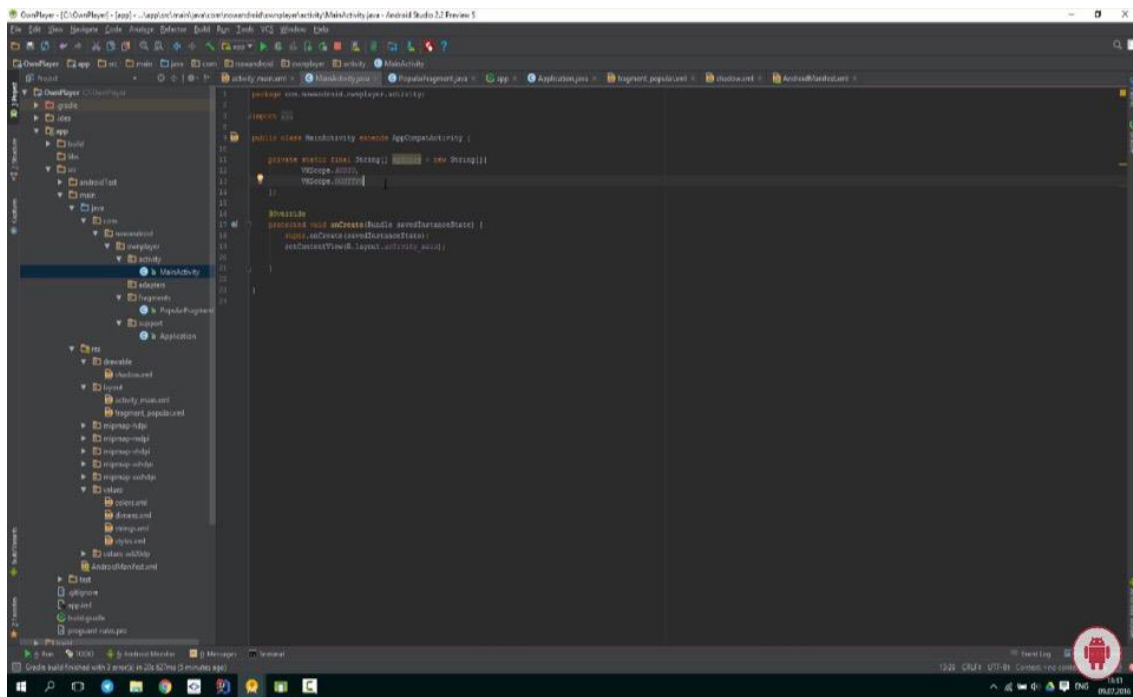


2.5-сурет – Android studio бағдарламасының компоненттерінің жүктелуі

Android studio бағдарламасында әдістемелік нұсқаулармен, деректермен әлеуметтік желілердегі сілтемелер арқылы ғаламтор жүйесінде танысуға да болады.

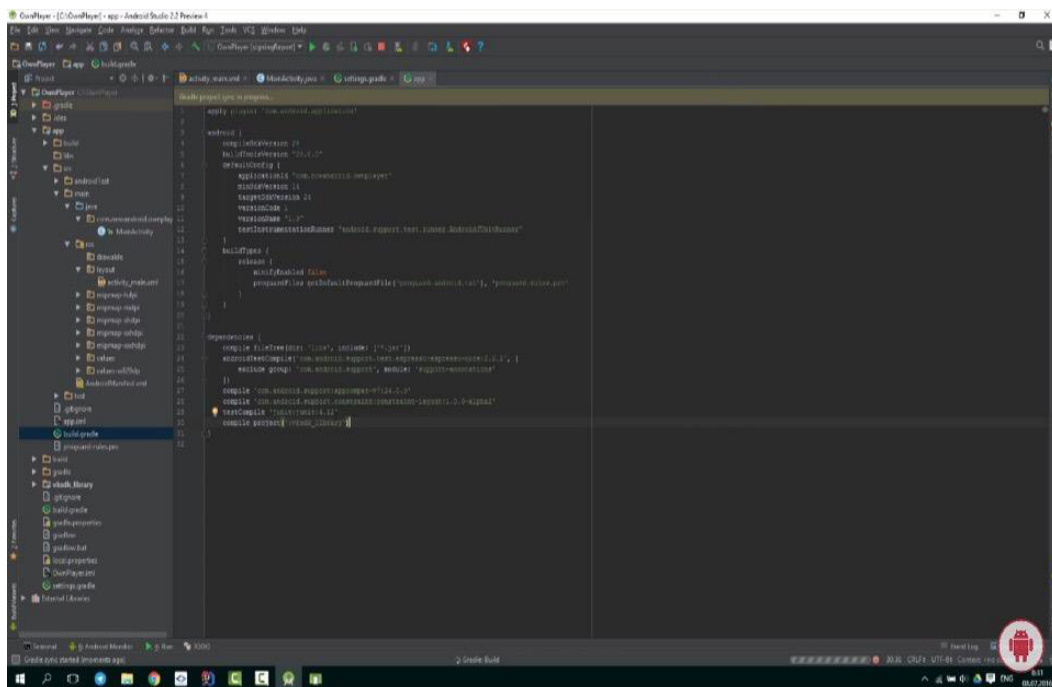


2.6-сурет – Программаны жасауға қолданылатын класстар

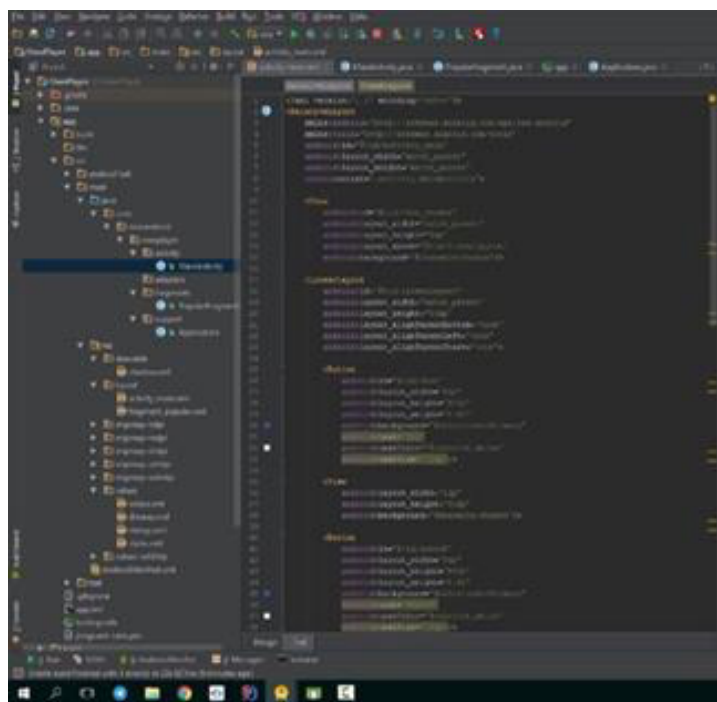


2.7-сурет – Класстардын жалғасы

Android studio бағдарламасында арқылы қосымша интерфейсін құруға болады. Жаңа жоба терезесінде қосымшаның атын және орналасу орынын көрсетеміз. Келесі кезекте жобаны құру терезесі ашылғаны. (2.8-сурет).



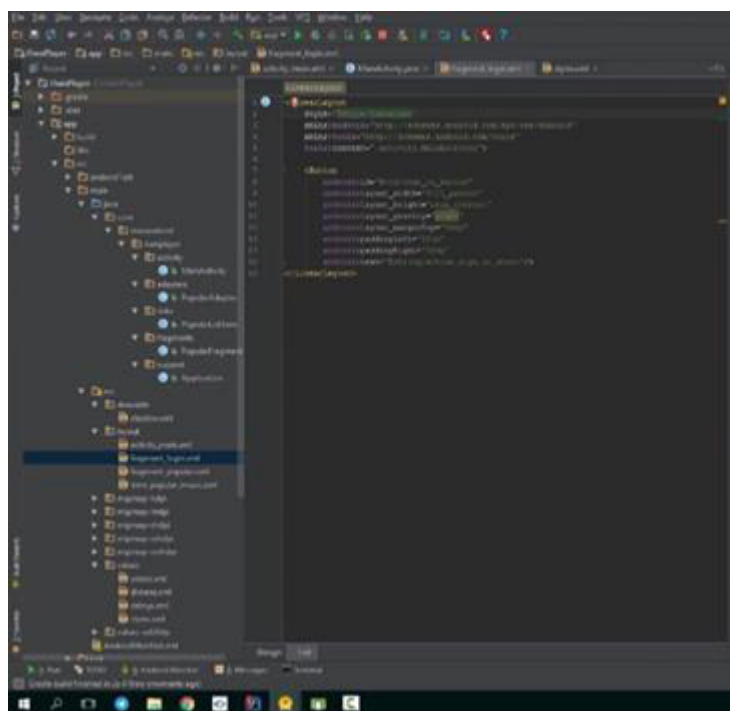
2.8-сурет – Эмулятордағы бастапқы беті



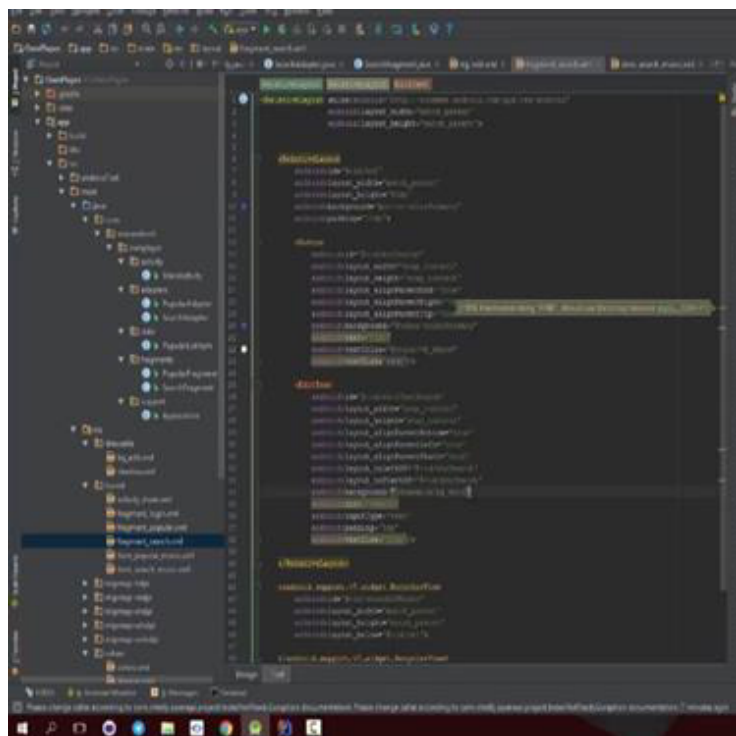
2.9-сурет – Жаңа жоба жасау беті

Бағдарламаны қосқаннан кейін Start a new Android Studio Project батырмасын басқанда, бағдарламалық жүйені құру терезесі ашылады (2.9-2.10 суреттер).

Мекеме талабы бойынша Android платформасы таңдалды. Бірақ бұл бағдарламада басқада платформалар таңдауға болады.

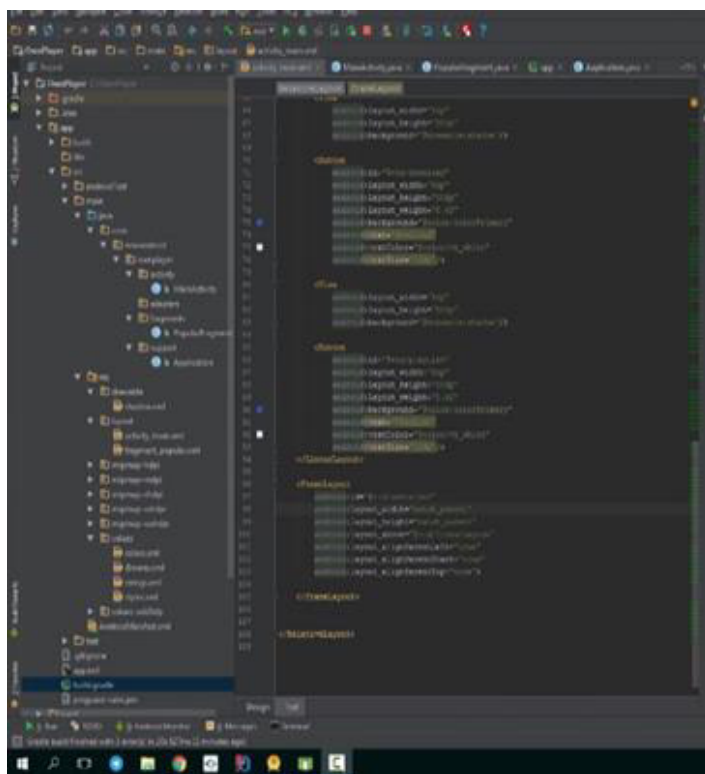


2.10-сурет – Жаңа жоба жасау беті

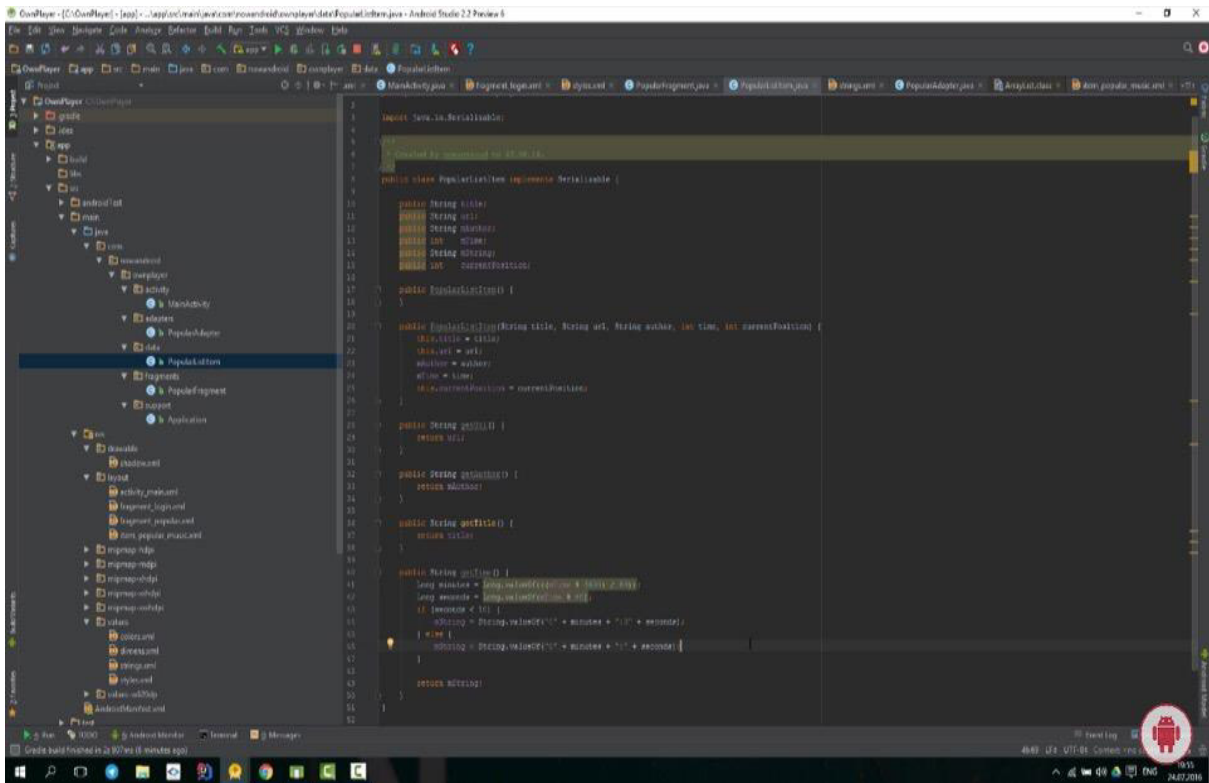


2.13-сурет – Негізгі бет интерфейсінің элементтерін орналастыру

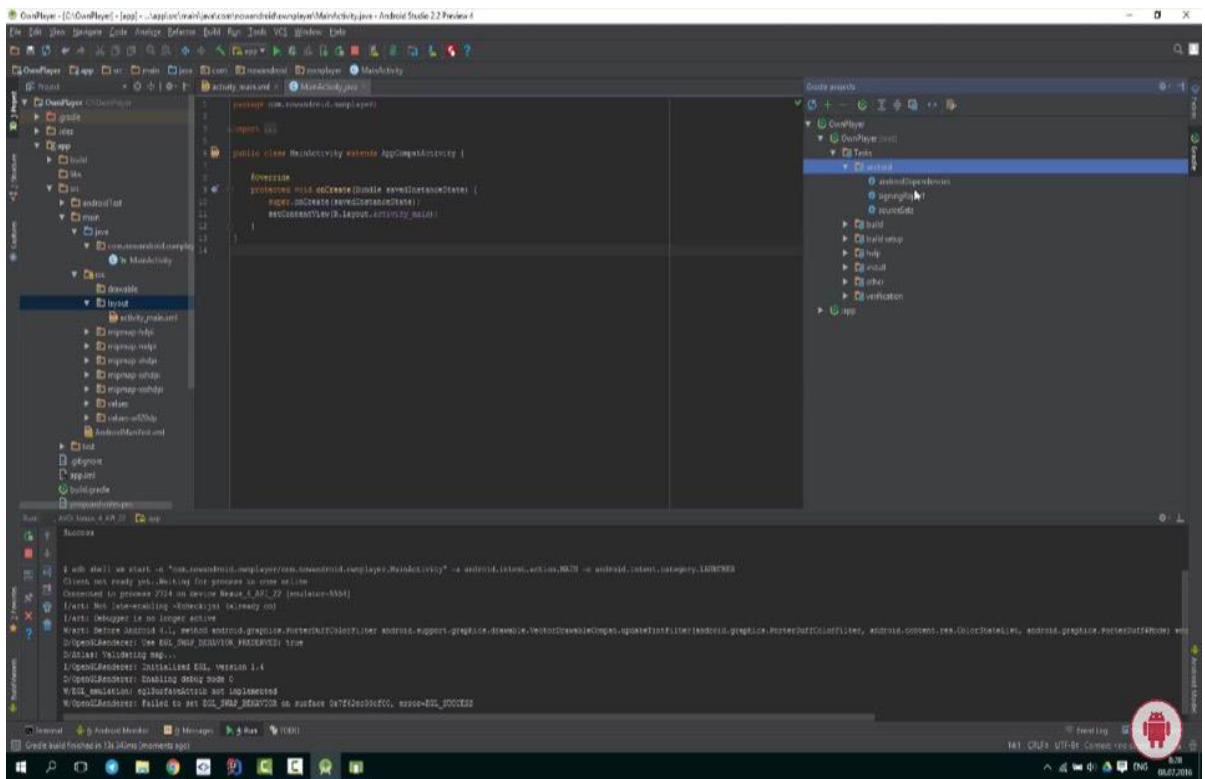
Келесі қадамда қосымша интерфейсінің элементтерін орналастыру және бағдарламалау орындалады.



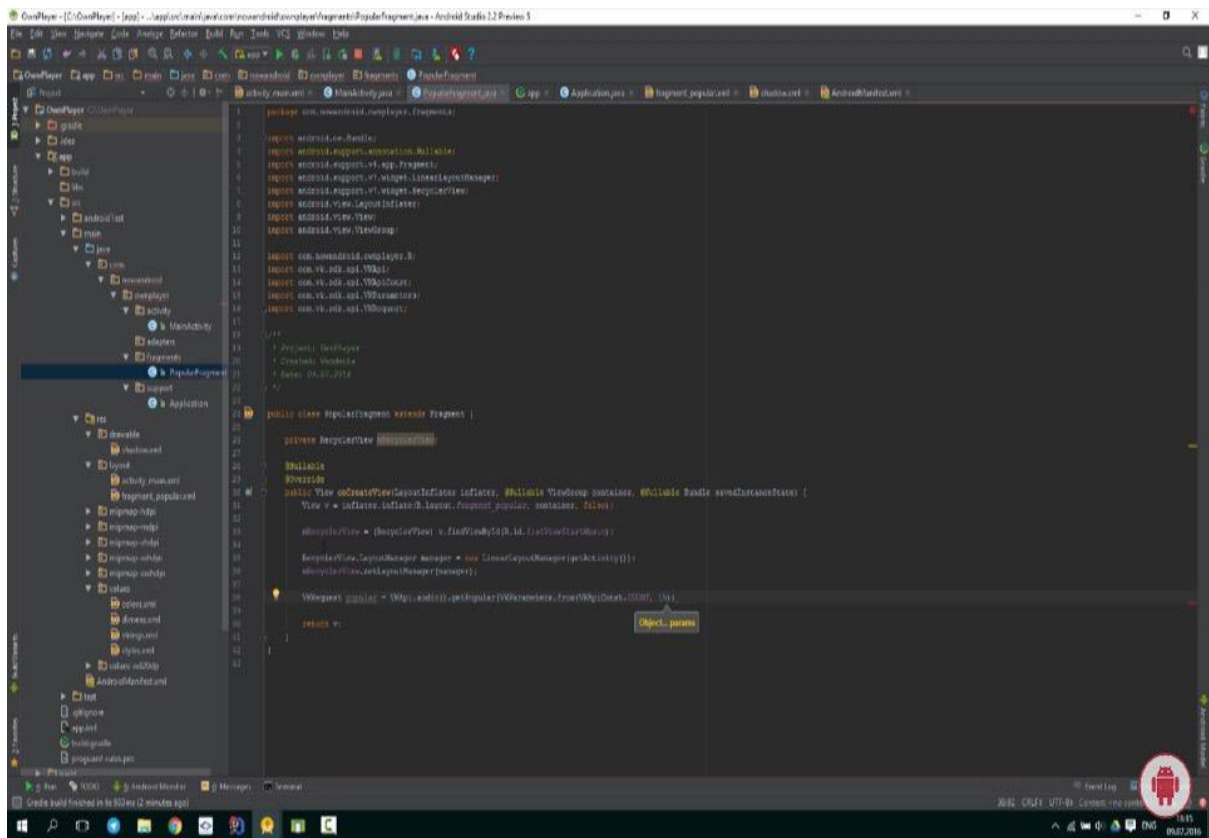
2.14-сурет – Элементтер құру бағдарламасының фрагменті



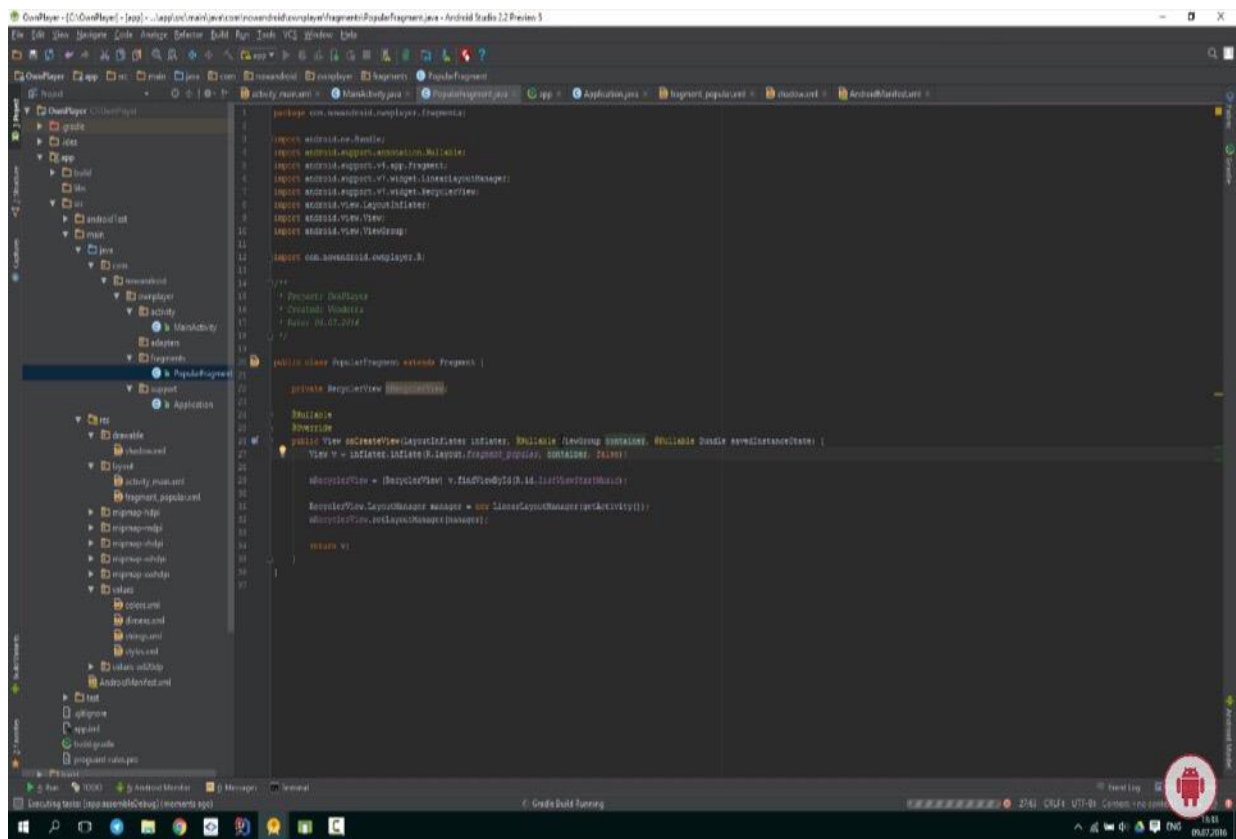
2.15-сурет – Қосымша бастапқы элементтерінің орналасуы



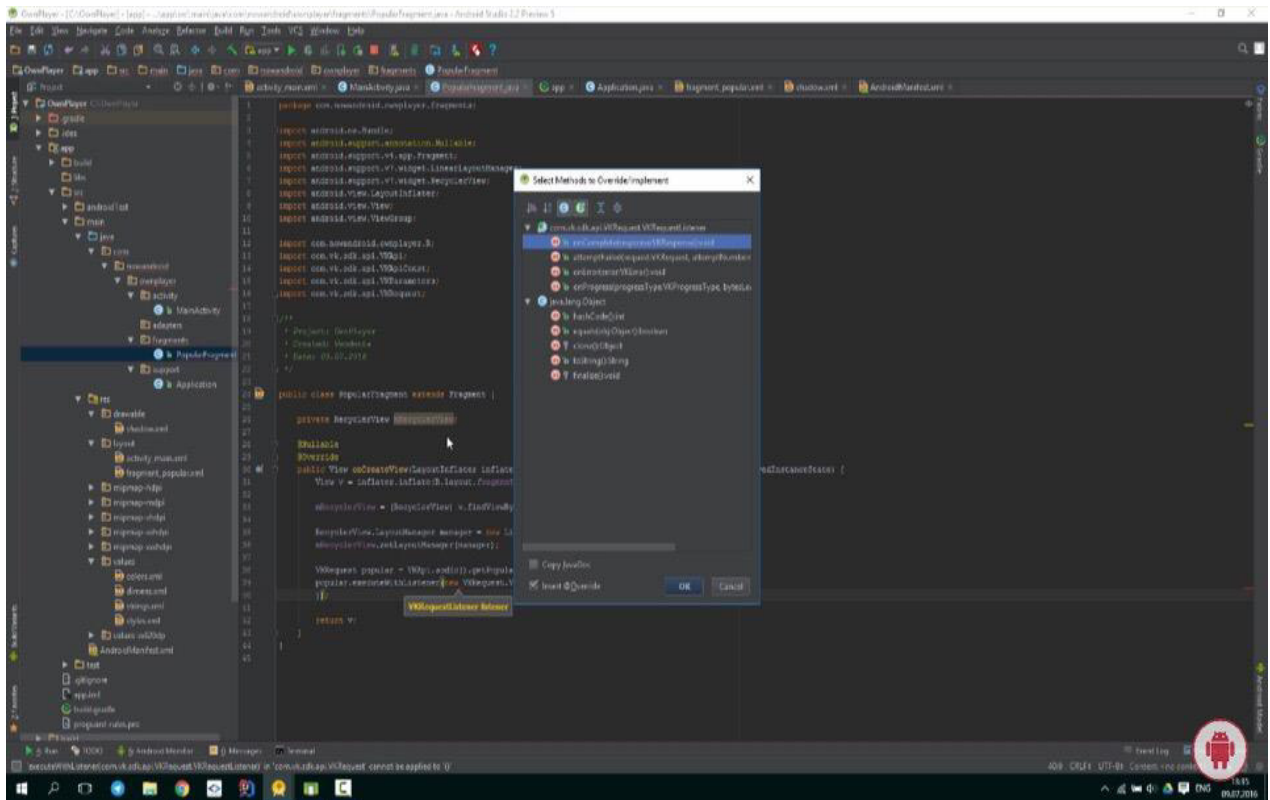
2.16-сурет – Элементтерді интерфейске енгізу



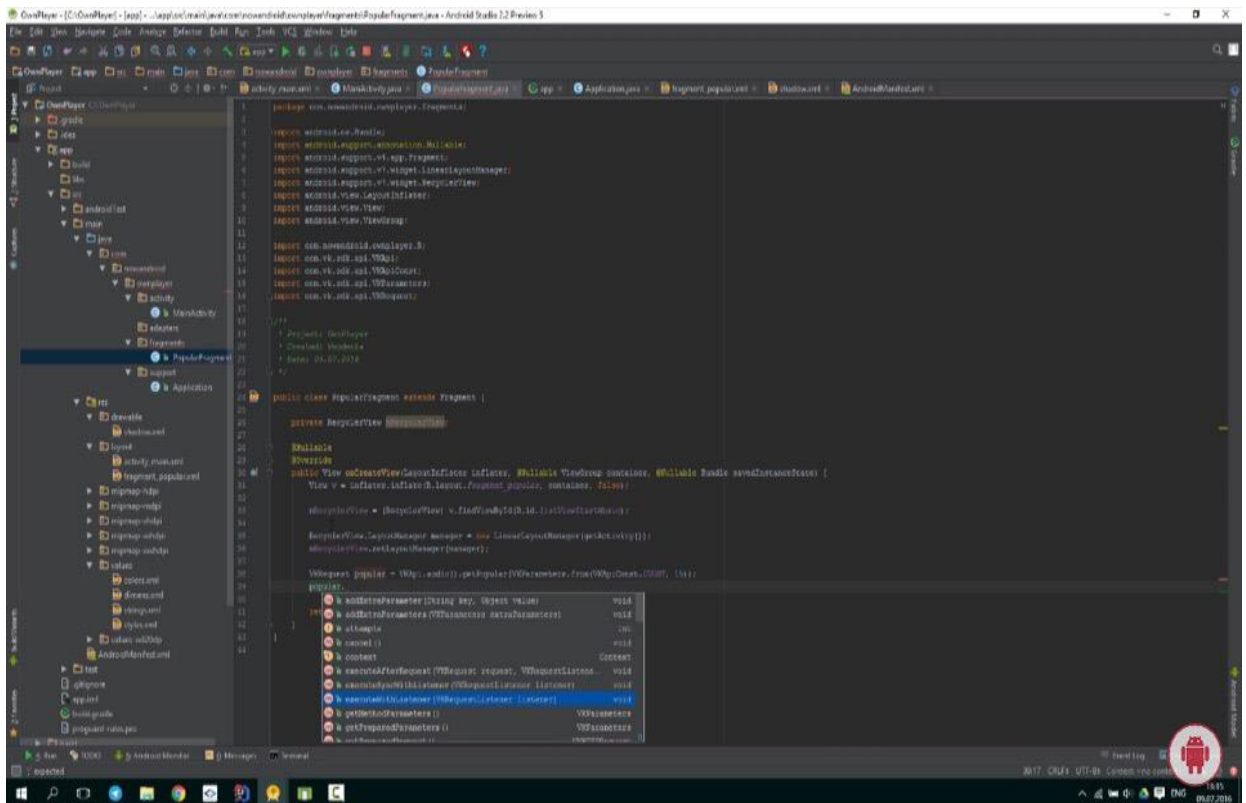
2.17-сурет – Қосымшаны бағдарламалау



2.18-сурет – Класстар арқылы программаға мәзір енгізу



2.19-сурет – Программаға мәзір енгізілді



2.20-сурет – Мәзірді енгізу жолы берілген

```

<activity
    android:name="com.example.musicplayer.MainActivity"
    android:label="@string/app_name"
    android:launchMode="singleTop"
    android:screenOrientation="portrait" >

```

2.21-сурет – Интерфейсті және атқару функцияларын орындауда Java тілі қолданылды.

```

<activity
    android:name="com.example.musicplayer.MainActivity"
    android:label="@string/app_name"
    android:launchMode="singleTop"
    android:screenOrientation="portrait" >

```

2.22-сурет – Екінші жолы

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:background="#FF330000"
    tools:context=".MainActivity" >

    <ListView
        android:id="@+id/song_list"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" >
    </ListView>

</LinearLayout>

```

2.23-сурет –Қосымшаны бағдарламалау жобасының іздеу жолы

```

<menu xmlns:android="http://schemas.android.com/apk/res/android" >

    <item
        android:id="@+id/action_shuffle"
        android:icon="@drawable/rand"
        android:orderInCategory="1"
        android:showAsAction="always"
        android:title="Shuffle"/>

    <item
        android:id="@+id/action_end"
        android:icon="@drawable/end"
        android:orderInCategory="2"
        android:showAsAction="always"
        android:title="End"/>

</menu>

```

2.24-сурет – Java тілі қолданылады

```

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import android.net.Uri;
import android.content.ContentResolver;
import android.database.Cursor;
import android.widget.ListView;

```

2.25-сурет – Қолданылған Java тілі технологиялық бағдарламасы арқылы батырманың интерфейсін және атқаратын функциясының сипаты .

```

if(musicCursor!=null && musicCursor.moveToFirst()){
    //get columns
    int titleColumn = musicCursor.getColumnIndex
        (android.provider.MediaStore.Audio.Media.TITLE);
    int idColumn = musicCursor.getColumnIndex
        (android.provider.MediaStore.Audio.Media._ID);
    int artistColumn = musicCursor.getColumnIndex
        (android.provider.MediaStore.Audio.Media.ARTIST);
    //add songs to list
    do {
        long thisId = musicCursor.getLong(idColumn);
        String thisTitle = musicCursor.getString(titleColumn);
        String thisArtist = musicCursor.getString(artistColumn);
        songList.add(new Song(thisId, thisTitle, thisArtist));
    }
    while (musicCursor.moveToNext());
}

```

2.26-сурет – java тілінің жалғасы

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:onClick="songPicked"
    android:orientation="vertical"
    android:padding="5dp" >

    <TextView
        android:id="@+id/song_title"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textColor="#FFFFFFF99"
        android:textSize="20sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/song_artist"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textColor="#FFFFFFF99"
        android:textSize="18sp" />

</LinearLayout>

```

2.27-сурет – Қолданылған тілдер


```

public class SongAdapter extends BaseAdapter {

    @Override
    public int getCount() {
        // TODO Auto-generated method stub
        return 0;
    }

    @Override
    public Object getItem(int arg0) {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public long getItemId(int arg0) {
        // TODO Auto-generated method stub
        return 0;
    }

    @Override
    public View getView(int arg0, View arg1, ViewGroup arg2) {
        // TODO Auto-generated method stub
        return null;
    }
}

```

2.28-сурет – Қосымшаны бағдарламалау

```

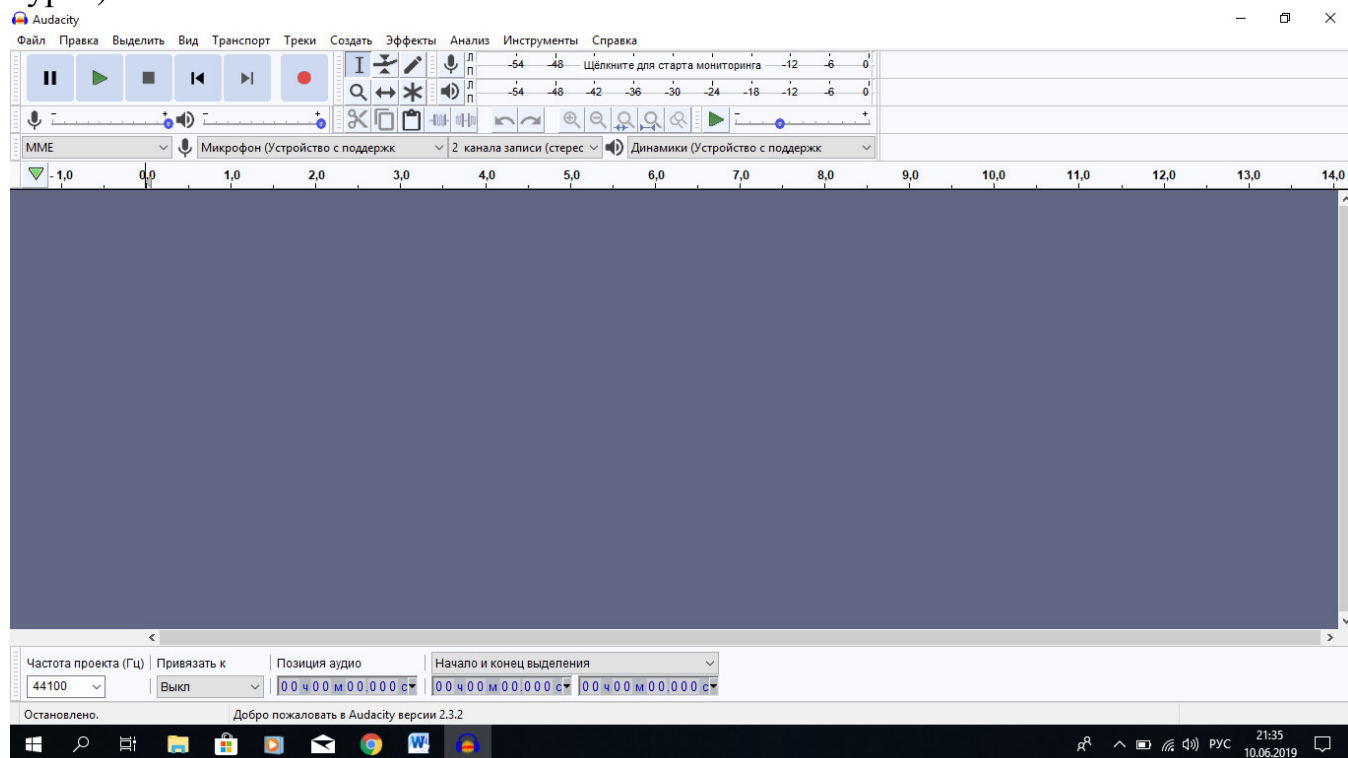
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    //map to song layout
    LinearLayout songLay = (LinearLayout)songInf.inflate
        (R.layout.song, parent, false);
    //get title and artist views
    TextView songView = (TextView)songLay.findViewById(R.id.song_title);
    TextView artistView = (TextView)songLay.findViewById(R.id.song_artist);
    //get song using position
    Song currSong = songs.get(position);
    //get title and artist strings
    songView.setText(currSong.getTitle());
    artistView.setText(currSong.getArtist());
    //set position as tag
    songLay.setTag(position);
    return songLay;
}

```

2.29-сурет – Қолданылған тілдер

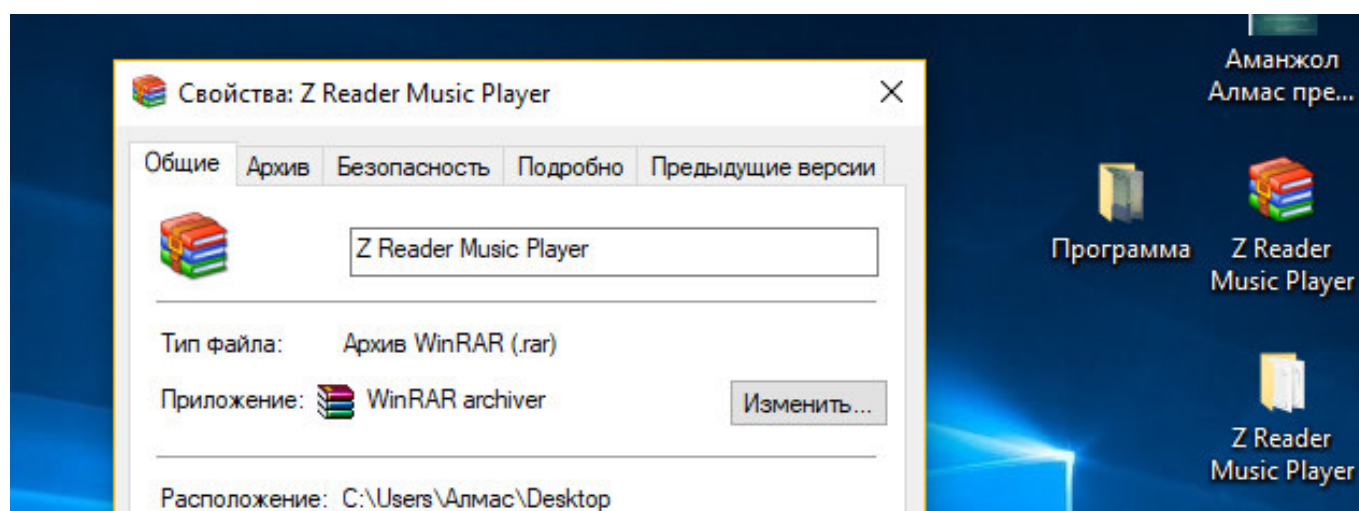
2.2 Қосымшаның Android studio бағдарламасында жүзеге асуы

Қосымшаны жасағанан кейінгі бастапқы интерфейсінің бейнесі (2.30-сурет).



2.30-сурет – Бастапқы интерфейсі

Қосымшаны қосу үшін келесі пішімге ауыстырылады. (2.31-сурет).




2.31-сурет – Z Reader Music Player

Свойства: Z Reader Music Player



Общие Архив Безопасность Подробно **Предыдущие версии**

 Предыдущие версии можно получить из истории файлов или точек восстановления.

Версии файлов:

Имя	Дата изменения
Предыдущие версии не обнаружены	

2.32-сурет – «Z Readear»

Z Reader Music Player.rar - WinRAR

Файл Команды Операции Избранное Параметры Справка

Добавить Извлечь... Тест Просмотр Удалить Найти Мастер Информация Вирусы Комментарий Защита SFX

Z Reader Music Player.rar\Z Reader Music Player - RAR 5.0 архив, размер исходных файлов 70 644 411 байт

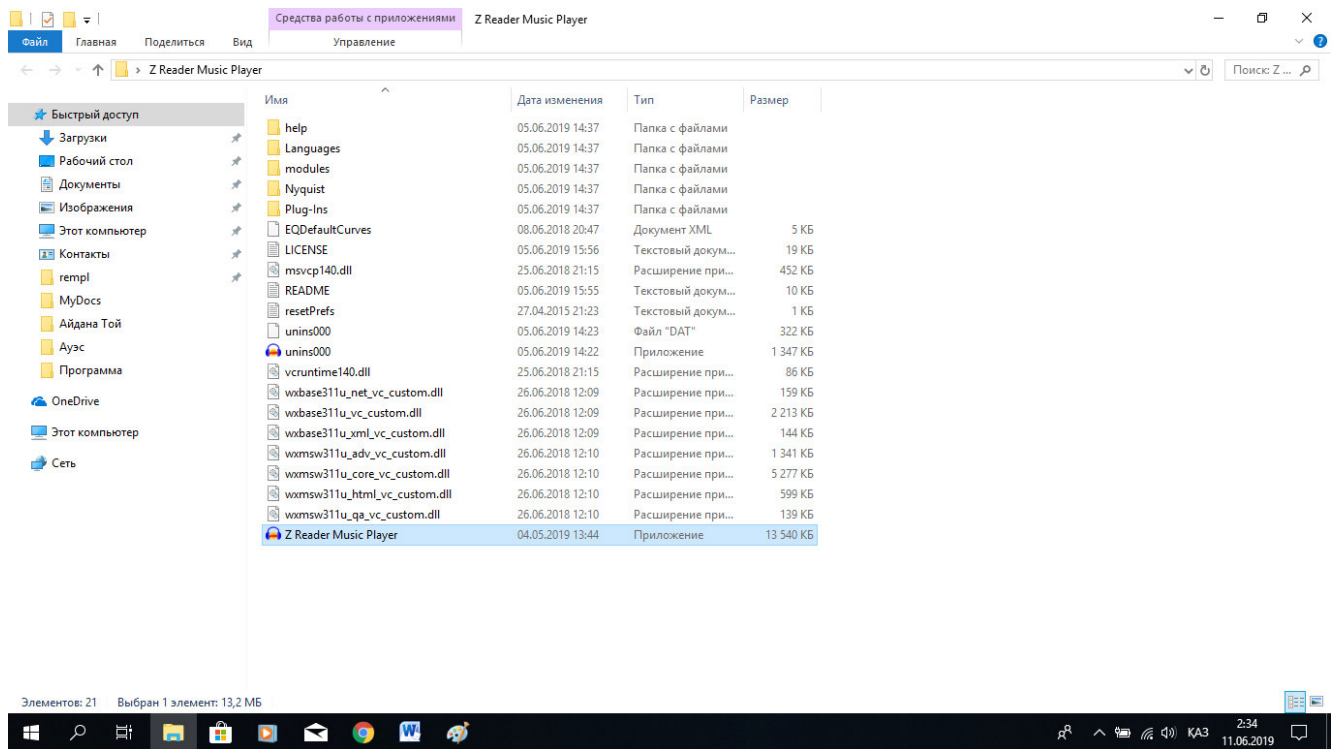
Имя	Размер	Сжат	Тип	Изменён	CRC32
..			Папка с файлами		
help			Папка с файлами	05.06.2019 14:37	
Languages			Папка с файлами	05.06.2019 14:37	
modules			Папка с файлами	05.06.2019 14:37	
Nyquist			Папка с файлами	05.06.2019 14:37	
Plug-Ins			Папка с файлами	05.06.2019 14:37	
EQDefaultCurves.xml	4 691	832	Документ XML	08.06.2018 20:47	26D4874C
FirstTime.ini	27	27	Параметры конф...	05.06.2019 14:23	E7347C28
LICENSE.txt	18 955	7 110	Текстовый докум...	05.06.2019 15:56	DBFDF3C9
msvcpl140.dll	462 120	115 366	Расширение при...	25.06.2018 21:15	BA424B52
README.txt	9 946	3 589	Текстовый докум...	05.06.2019 15:55	B896358C
resetPrefs.txt	129	114	Текстовый докум...	27.04.2015 21:23	B9D8AE17
unins000.dat	329 085	20 281	Файл "DAT"	05.06.2019 14:23	DC30C729
unins000.exe	1 379 033	504 400	Приложение	05.06.2019 14:22	1CCC60DF
vcruntime140.dll	87 352	43 212	Расширение при...	25.06.2018 21:15	05EE9826
wxbase311u_net_vc_custom.dll	162 304	51 589	Расширение при...	26.06.2018 12:09	9A5B1105
wxbase311u_vc_custom.dll	2 266 112	572 232	Расширение при...	26.06.2018 12:09	498B10A8
wxbase311u_xml_vc_custom.dll	147 456	56 892	Расширение при...	26.06.2018 12:09	C5F0FC37
wxmsw311u_adv_vc_custom.dll	1 372 672	334 612	Расширение при...	26.06.2018 12:10	9117CCF7
wxmsw311u_core_vc_custom.dll	5 403 136	1 422 530	Расширение при...	26.06.2018 12:10	A1EB477C
wxmsw311u_html_vc_custom.dll	612 864	162 538	Расширение при...	26.06.2018 12:10	E8668307
wxmsw311u_qa_vc_custom.dll	141 824	40 948	Расширение при...	26.06.2018 12:10	FCC80633
Z Reader Music Player.exe	13 864 960	3 505 900	Приложение	04.05.2019 13:44	8A50CD0D

Выбрано: 13 864 960 байт в 1 файле

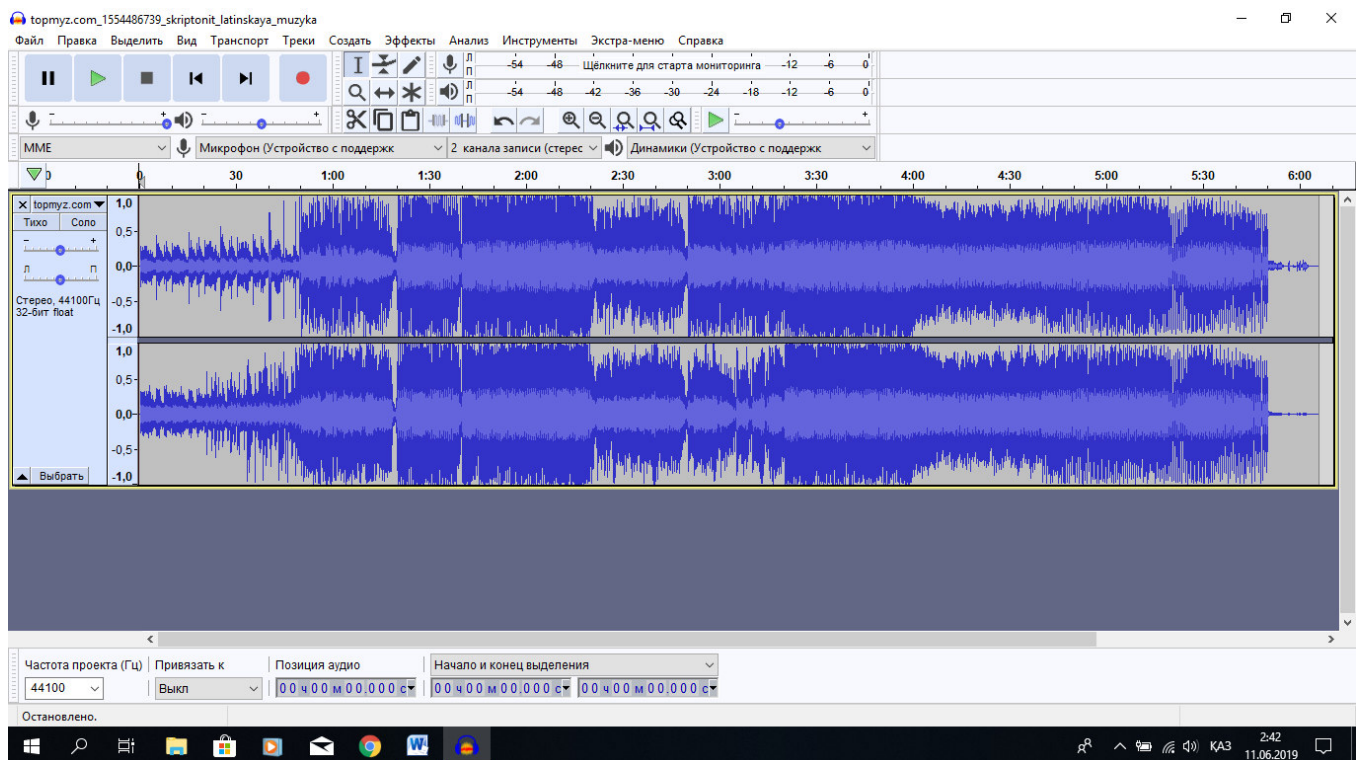
Всего: 5 папок и 26 262 666 байт в 17 файлах

2:33 11.06.2019

2.33сурет – «Z Readear»

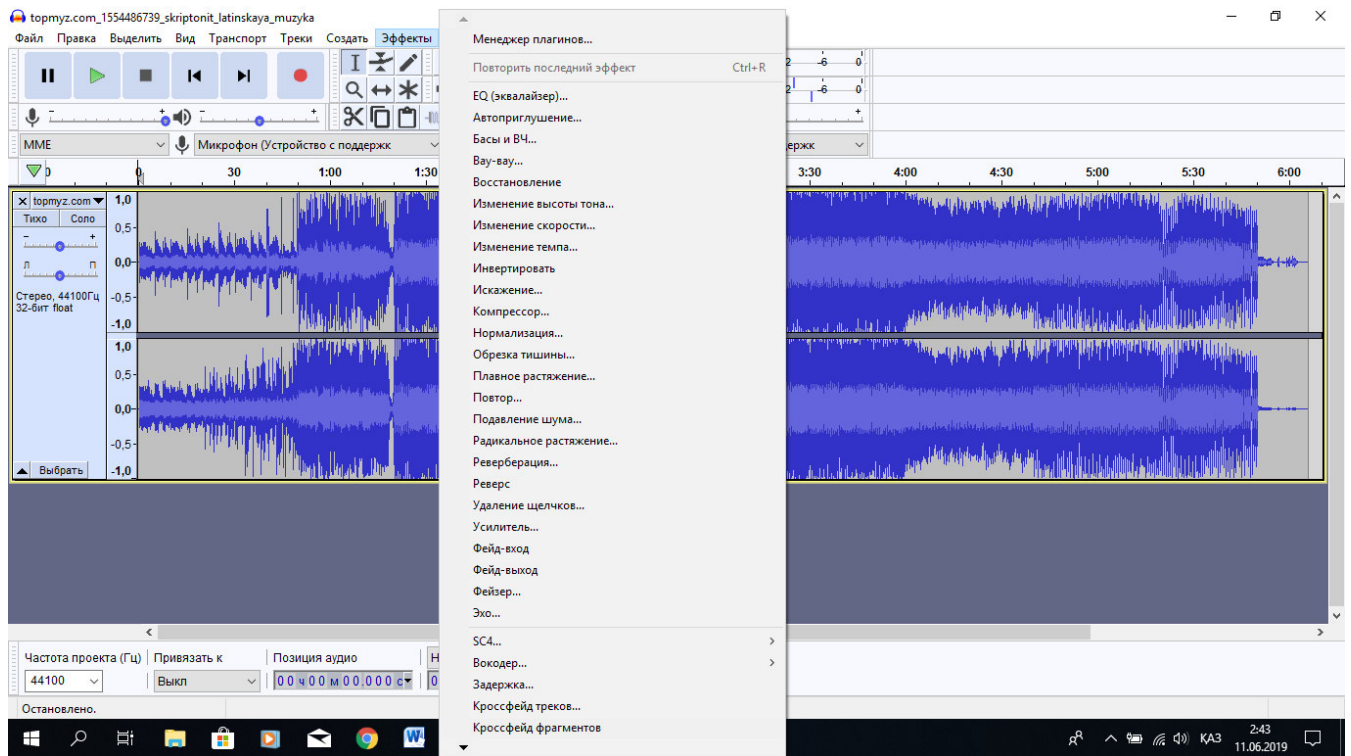


2.34-сурет – «Z Reader Music Player»



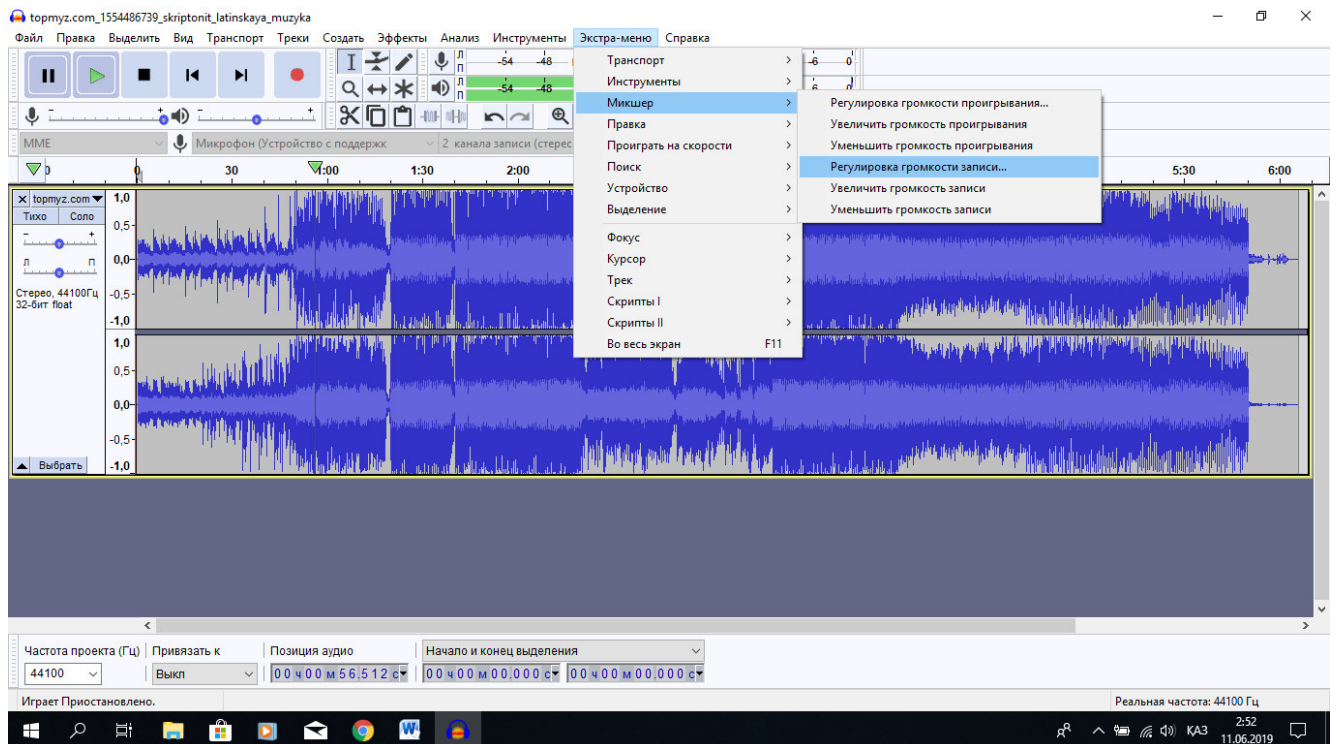
2.35-сурет – «Z Reader» терезесі

Барлық параметрлер таңдалғаннан кейін әннің орындалуын келесі суреттен көруге болады (2.36-сурет).

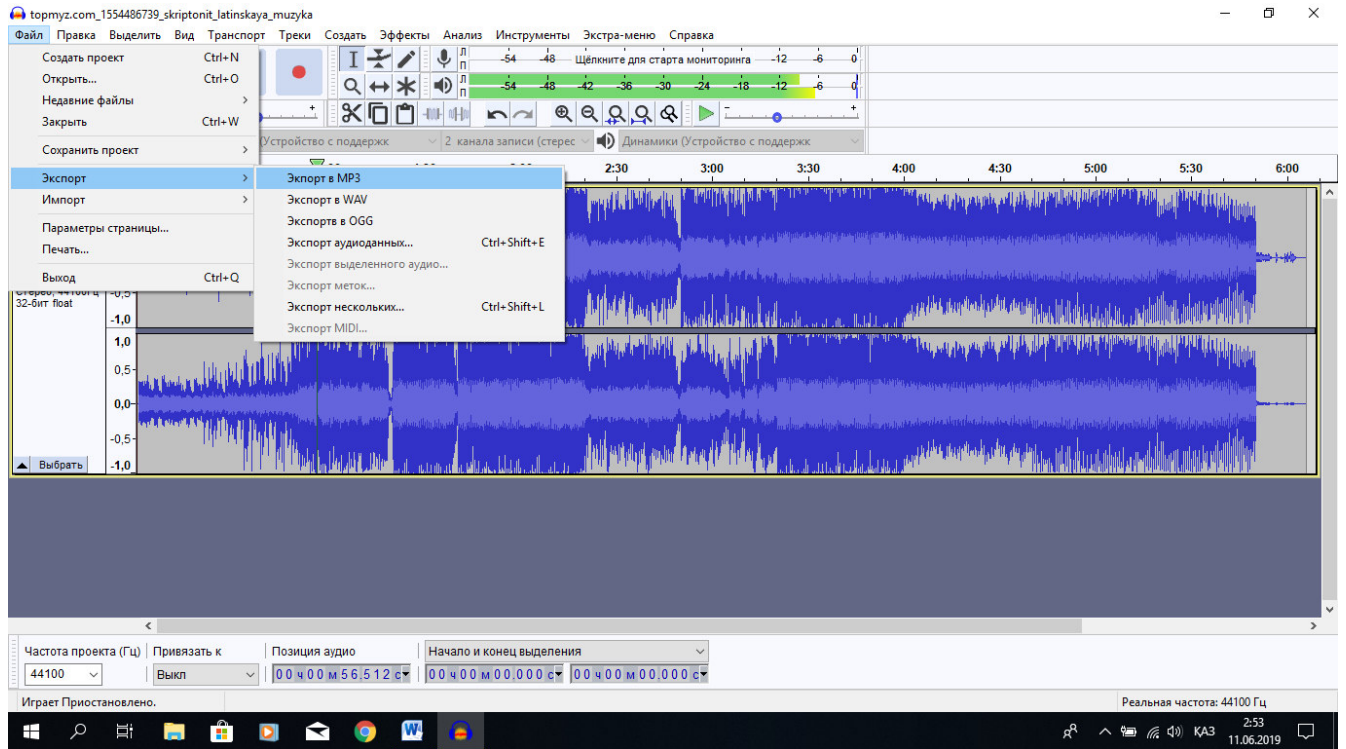


2.36-сурет – Музыкалық плеер интерфейсі

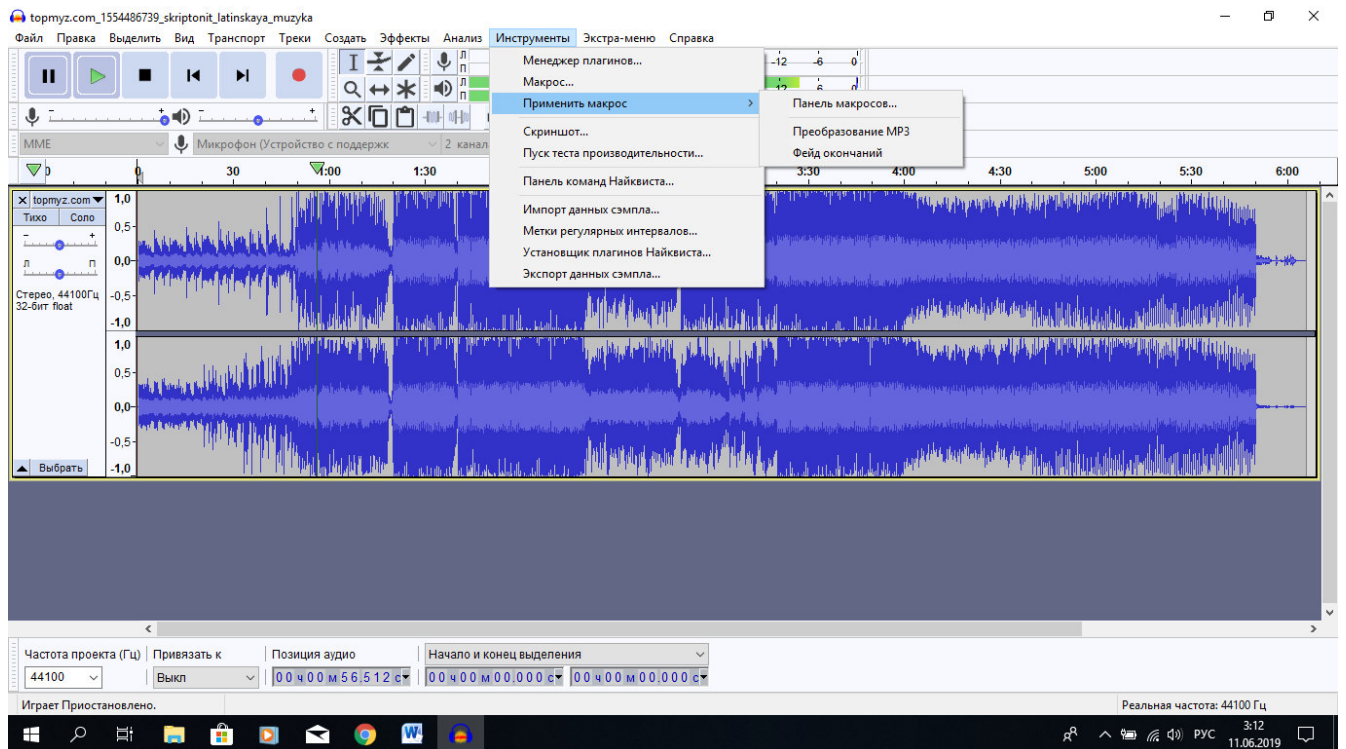
Таңдалған ән не музыкалық композициямен бірнеше әрекеттер жасау мүмкіндігі ұйымдастырылған (2.37-сурет).



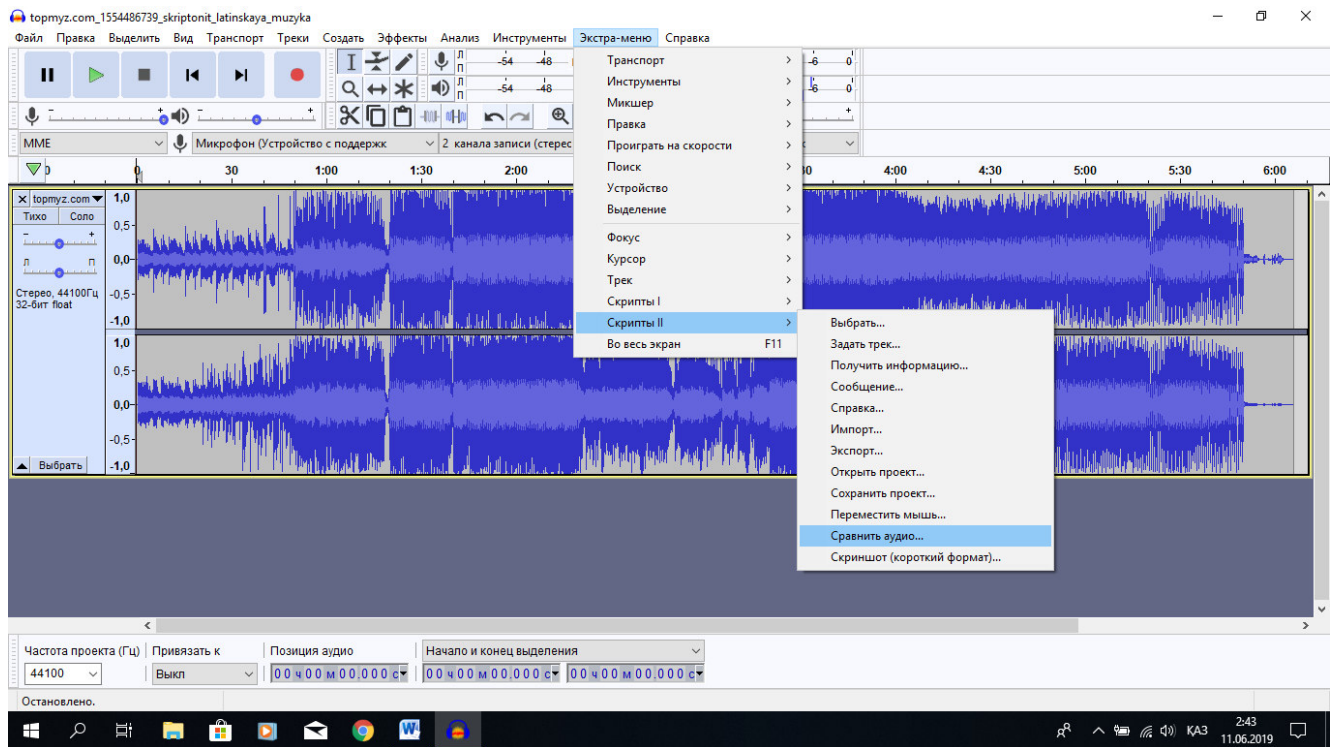
2.37-сурет – Музыкалық плеер интерфейсі



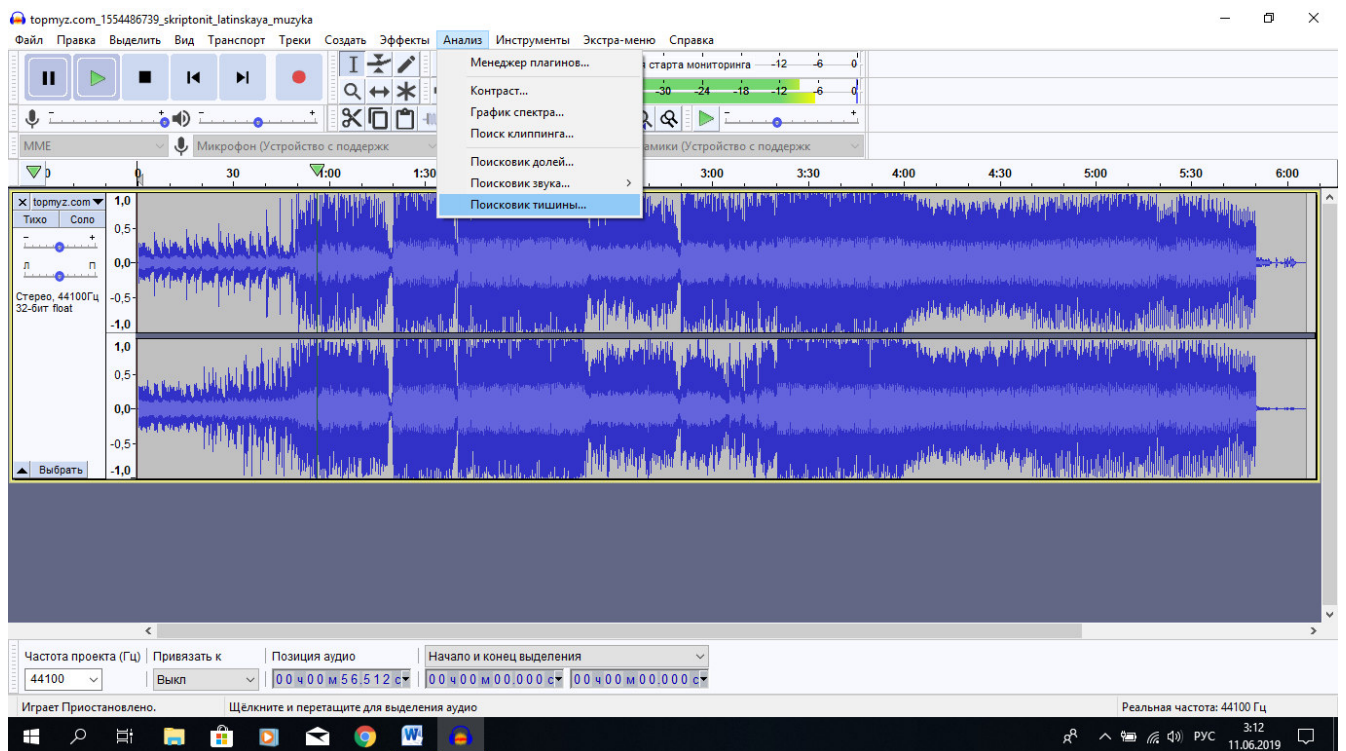
2.38-сурет – Музыкалық плеер интерфейсі



2.39-сурет – Музыкалық плеер интерфейсі



2.40-сурет – Музыкалық плеер интерфейсі



2.41-сурет – Музыкалық плеер интерфейсі

Бағдарламалық қамтама бірнеше операциялық жүйелерде жұмыс жасау мүмкіндігі бар. Құрылған жобаның бағдарламасы А және Ә қосымшасында келтірілген.

Z Reader музыкалық ойнатқышының сипаттамасы (Music Player Z Reader):

-Z Reader музыкалық ойнатқышы - жақсы сапалы смартфон ойнатқышы және компьютерлік қосымшасы;

-ойнатқыш визуализацияның 6 түрі, теңшелетін эквалайзер және көптеген басқа параметрлер бар;

- музыканы орнатуға және тыңдауға болады.

Ерекшеліктері:

- сәнді дизайн;
- топтамалық композициялар;
- ойнату тізімдерін басқарыңыз;
- плагиатқа өлеңді тексереді
- өз өлеңді жасауға болады;
- барлық белгілі дыбыстық файл пішімдерін қолдайды.

3 Техника-экономикалық негізделуі

3.1 Бағдарламалық жасақтама әзірлеудің күрделілігі

Бұл дипломдық жобаның тақырыбы - «Андройд платформасындағы Z Readear Music Player қосымшасының мобильдік әзірлемесі».

Дипломдық жобаның экономикалық негіздемесінің мақсаты - ақпараттық жүйені құруға және іске асыруға қажетті шығындарды есептеу және талдау, оны жобалау мен пайдалану шығындарын зерттеу; шығындарды өтеу шарттары мен мерзімін анықтау. Осы қосымшаны жобалаудың ұтымдылығын дәлелдеу.

Бұл тарауда зерттеу жобасының коммерциялық құндылығының экономикалық есептеуі жасалған. Есептеулер ақпараттық жүйені құру және енгізу шығындарын ескерген.

Жобаны іске асыру үшін қаржы, материалдық және еңбек ресурстары қажет; Жобалау жоспары мен жобаны іске асыру 1 айда (4 апта) 3 кезең бойынша қарастырылған.

Бірінші кезең:

- міндетті белгілеу;
- жұмыс мазмұнын әзірлеу.

Екінші кезең:

- программалық жасақтаманы құру.

Үшінші кезең:

- тестілеу;
- откладка;
- енгізу.

Кесте 3.1 – Жұмысты кезеңі мен түрі бойынша бөлу және олардың еңбек қарқынын бағалау

Әзірлеменің кезеңі ПП	Осы кезеңдегі жұмыс түрі	ПП әзірлемесінің еңбек қарқыны, адам× с.
Бірінші кезең	Тапсырмалардың берілуі; Жұмы мазмұнын әзірлеу	45
Екінші кезең	Программалық жасақтаманы құру	97
Үшінші кезең	Тестілеу; Түзету; Енгізу.	50
БАРЛЫҒЫ диплом жұмысын орындауы		192

3.2 III әзірлеуге жұмсалатын шығындар есебі

Кесте 3.2 – Программалық жасақтама мен жабдықтарға жұмсалатын шығындар

Ресурстардың атауы	Өлшеу бірлігі	Пайдаланылатын материалдың көлемі	бірлік бағасы, тг	Сомма, тг
Lenovo Core i5 3320M Ноутбугі	шт	1	69000	69000
HP Deskjet Ink Advantage 2515 Лазерлік принтер	шт	1	42000	42000
БАРЛЫҒЫ программалық жасақтама мен жабдықтарға жұмсалатын шығындар				111000

Кесте 3.3 – Материалды шығындар

Материалды ресурс атауы	Өлшем бірлігі	Жұмсалған материалдардың саны	Бірлік бағасы, тг	Сомасы, тг
Қағаз	дана	3	1100	3300
Картридж	дана	1	2500	2500
Барлығы материалдық ресурстарға жұмсалған шығындар				5800

Материалдық ресурстарға жұмсалған шығындардың жалпы сомасы (Z_M) мына формула бойынша анықталады:

$$Z_M = \sum_{i=1}^n P_i \times C_i \quad (5.1),$$

мұндағы P_i - материалдық қорлардың i - түріндегі шығыны, нақты бірліктер;

C_i - материалдық қорлардың i - түріндегі бірлігі үшін бағасы, тг;

i – материалдық ресурстардың түрі;

n - материалдық ресурстардың саны.

$$Z_M = \sum_{i=1}^n P_i * C_i = ((1100 \times 3) + 2500) = 5800 \text{ тг.}$$

Электр энергиясына жұмсалған шығындарды 3.4 кестесінде келтірілген үлгіде есептеу қажет.

Электр энергиясына жұмсалған шығындардың жалпы сомасы (Z_E) төмендегі формула бойынша есептеледі:

$$Z_3 = \sum_{i=1}^n M_i * K_i * T_i * C \quad (5.2)$$

мұндағы M_i - i - электржабдығының құжаттық қуаты, кВт;

K_i - i - электржабдығының қуатын пайдалану коэффициенті
($K_i=0.7, 0.9$ қолданылады);

T_i - ПП әзірлеудің барлық кезеңі үшін i - электржабдығының жұмыс уақыты с;

C – электр энергиясының бағасы, тг/кВт*с;

i – электр жабдығының түрі;

n - электр жабдығының саны.

$$Z_{31} = 0,45 \cdot 0,7 \cdot 192 \cdot 18,32 = 999,7 \text{ тг.} \approx 1107,99 \text{ тг.}$$

$$Z_{32} = 0,33 \cdot 0,7 \cdot 2 \cdot 18,32 = 7,6 \text{ тг} \approx 8,46 \text{ тг.}$$

$$Z_3 = Z_{31} + Z_{32} = 1107 + 8 = 1116,45 \text{ тг.}$$

Кесте 3.4 – Электр энергиясына жұмсалатын шығындар

Жабдықтардың атауы	Құжаттық қуаты, кВт	Қуатты пайдалану коэффициенті	ПП әзірлеуге арналған жабдықтардың жұмыс уақыты,с	Электр энергия бағасы, тг/кВт*с	Сомасы, тг
Lenovo Corei5 3320M Ноутбүгі	0,45	0,7	192	18,32	1007,99
HP Deskjet Ink Advantage 2515 Лазерлік принтер	0,33	0,7	2	18,32	8,46
Барлығы электр энергиясына жұмсалған шығын					1116,45

3.3 Жұмысқа қажетті еңбек ресурстары

Жалақы төлемі шығындарының жалпы сомасын есептеу үшін $Z_{тр}$ 3.3 формуласын пайдаланамыз:

$$Z_{тр} = \sum_{i=1}^n ЧC_i \cdot T_i, \quad (3.3)$$

мұндағы n – қосымшаларды әзірлеушілердің саны;
 $ЧС_i$ - i қызметкердің сағаттық мөлшерлемесі, тг;
 T_i – ПП әзірлемесінің еңбек қарқындылығы, адам×с;
 i – жұмыскер санаты.

ПП әзірлемесінің еңбек қарқындылығы 3.1 кестесінде берілген деректер бойынша анықталады.

Әзірлеуші инженердің сағаттық мөлшерлемесі әзірлеме қарқындылығының -192 с 900 (тг/с) құрайды.

Негізгі жалақы есебінің нәтижелері 3.5. кестеде берілген.

Кесте 3.5– Негізгі жалақы шығындарын есептеу нәтижелері

Жұмыс атауы	Орындаушы	Еңбек қарқындылығы норма-сағат	1сағатқа жалақы, тг/сағ	Жалақы сомасы, тенге
Тех. Тапсырма	Әзірлеуші	9	900	8100
Моделдеу		36		32400
Программалау		97		87300
Тестілеу		34		30600
Енгізу		16		14400
Барлығы		181		172800

Сонымен қатар әлеуметтік салық төлемдерін есептеу қажет, ол жұмыскер табысының 9,5% (ҚР СК 485 бабына сәйкес құрайды. Әлеуметтік салымдар келесі формула бойынша анықтаймыз:

$$З_{сзi} = (З_{тр} - З_{по}) \cdot 0,095, \quad (3.4)$$

мұндағы $З_{по}$ – зейнетақы шегерімі, еңбек ақының жалпы сомасының 10%-ы, тенге. Зейнетақы шегерімдерін келесі формула бойынша анықтаймыз:

$$З_{по} = З_{тр} \cdot 0,1 \quad (3.5)$$

3.5 және 3.9 формулаларын пайдалана отырып төмендегі нәтижені аламыз:

$$З_{по} = 172800 \cdot 0,1 = 17280 \text{ тенге}$$

$$З_{сзi} = (172800 - 17280) \cdot 0,095 = 14774 \text{ тенге}$$

Жылдық амортизациялық шегерімдер сомасы төмендегі формула бойынша анықталады:

$$A = \text{бастапқы құны} \cdot \text{амортизация нормасы} / 100 \quad (3.6)$$

Амортизациялық шегерімдер 5.6. кестесінде көрсетілген.

НҚ амортизациялаудың жылдық нормасы ҚР СК бойынша немесе НҚ-ны пайдалы қолданудың мүмкін болатын мерзіміне байланысты қабылданады:

$$N_{Ai}=100/T_{Ni}, \quad (3.7)$$

мұндағы T_{Ni} - i - ОФ пайдаланудың мүмкін мерзімі, жыл.

Кесте 3.6 – Негізгі қорларды амортизациялау (ОФ)

Жабдық және ПО атауы	ПО және жабдық құны, тг	Амортизациялаудың жылдық нормасы, %	Жабдық және ПО тиімді пайдалану мерзімі, жыл	Жылын ааморт изациялау сомасы, тг	Айына амортизациялау сомасы, тг
Lenovo Core i5 3320M Ноутбүгі	69 000	20	5	13 800	1150
HPDeskjetInk Advantage2515 Лазерлік принтер	42 000	20	5	8 400	700
БАРЛЫҒЫ негізгі қор амортизациясы					1850

$$N_{A_{об}}= 100 / 5 = 20;$$

$$A_{н}=(69\ 000 * 20) / 100 = 13\ 800 \text{ тг};$$

$$A_{м}=(42\ 000 * 20) / 100= 8\ 400 \text{ тг};$$

Бір айға амортизациялау сомасы = $A / 12$.

Бір айға амортизациялық шегерімдер сомасы 1850 тг. тең

«Басқада шығындар» бабына коммуналдық төлемдерді, кеңсе және басқа шаруашылық шығындарын қоса алғанда жалдау шығындары енгізілген. Бір айға үй-жайды жалдау құны 118 000 тг. Тең (бұл сомаға коммуналдық қызметтер де кіреді)

Жалдау ақысы мына формул бойынша есептеледі:

$$AP= Ca*S, \quad (3.8)$$

мұндағы Ca – жалдау ақысы;

S – 1 айға жалдау құны.

$$AP = 118\ 000 * 1 = 118\ 000 \text{ тг.}$$

Интернет шығындары, оған төленетін ақы төлемі 6300 тг тең:

$$P_{и} = 6300 * 1 = 6300 \text{ тг.}$$

Басқа шаруашылық шығындар 5000 тг құрайды;

Басқа шығындар = 118000 + 6300 + 5000 = 129300 тг.

Жеке баптар бойынша алынған деректер негізінде 5.7 кестесінде ПП әзірлеуге жұмсалған шығындар сметасы келтірілген.

Кесте 3.7 – ПП әзірлеуге жұмсалған шығындар сметасы

Шығындар баптары	Сомасы, тг
Материалдық шығындар, оның ішінде:	111000
- материалдар	5800
- электр энергиясы	1008
2 Еңбек ақы шығындары.	172800
3 Әлеуметтік шығындар шегерімдері.	14774
4 Негізгі қорларды амортизациялау.	1850
5 Басқа шығындар.	129300
Смета бойынша БАРЛЫҒЫ	436640,45

3.4 ПП-ның мүмкін болатын (келісілген) бағасы

ПП-ның мүмкін болатын (келісілген) бағасы тапсырыс беруші (тұтынушы) мен орындаушының экономикалық мүддесіне жауап беретін деңгейде оны орындау тиімділігі, сапасы мен мерзімін ескере отырып белгіленеді.

Қолданбалы ПП-ға арналған келісілген баға (КБ) мына формула бойынша анықталады:

$$Ц_{д} = Z_{нир} \times \left(1 + \frac{P}{100}\right), \quad (3.9)$$

Мұндағы $Z_{нир}$ – ПП әзірлеуге жұмсалған шығындар (3.7 кестесінен), тг;

P – ПП оңтайлығының орташа деңгейі. %

$$Ц_{д} = Z_{нир} \times \left(1 + \frac{P}{100}\right) = 436640,45 \times \left(1 + \frac{20}{100}\right) = 523968,54 \text{ тг.}$$

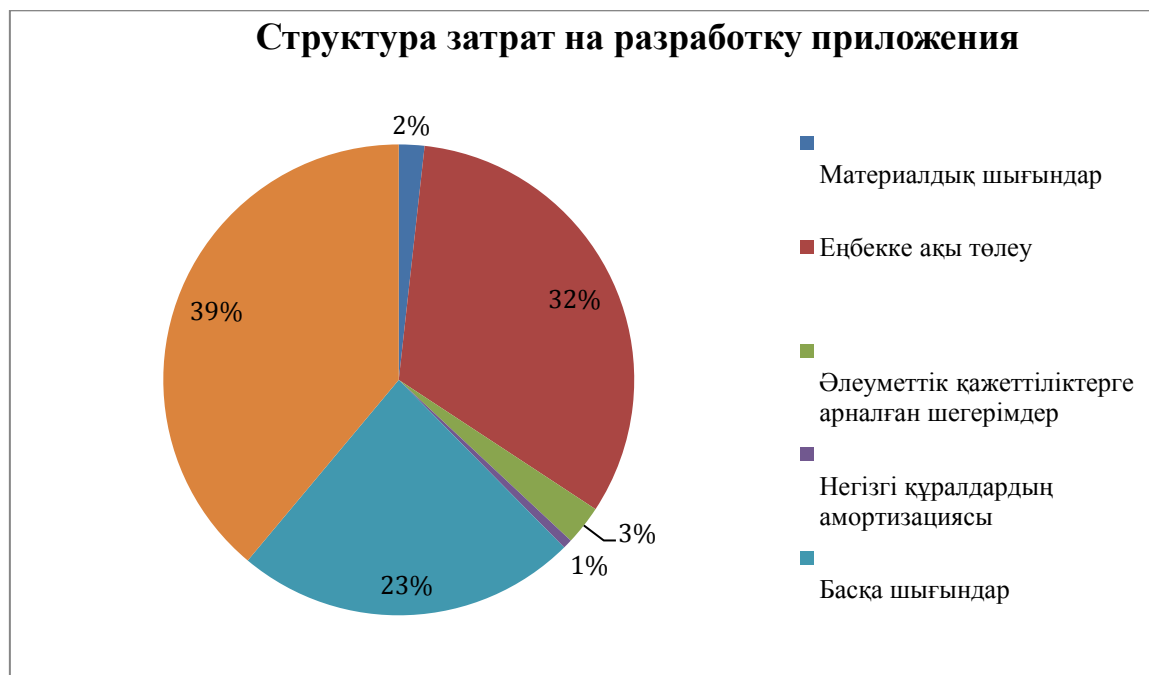
Бұдан әрі жүзеге асыру бағасы қосымша құн салығын (ҚҚС) ескере отырып анықталады, ҚҚС мөлшерлемесі ҚР СК заңында белгіленеді.

2019 жылға ҚҚС мөлшерлемесі 12% көлемінде белгіленген.

Жүзеге асыру бағасы ҚҚС ескере отырып мына формула бойынша есептеледі:

$$Ц_p = Ц_d + Ц_d \times НДС. (3.10)$$

$$Ц_p = Ц_d + Ц_d \times НДС = 523\,968,54 + 523\,968,54 * 0,12 = 125\,752,4496$$



Сурет 3.1 - Өтінішті әзірлеу құнының құрылымы

4 Өміртіршілік қауіпсіздігі

4.1 Жарықтандырудың негізгі сипаттамалары

Өндірістік жарықтандыру сандық және сапалық көрсеткіштермен сипатталады. Сандық көрсеткіштер негізгі жарықтехникалық өлшемдермен сипатталады, оларға жарық ағыны, жарық күші, жарықтану және айқын жарықтану жатады.

Сапалық көрсеткіштер көру жұмысындағы жағдайларды анықтайды, олар: фон, айқындылығы, соқырлық көрсеткіштігі, жарықтану толықсыма коэффициенті, дисконфорт көрсеткіші.

Жарық ағыны (F)—сәуле энергиясының қуаты. Жарық ағыны люменмен (лм) өлшенеді.

Жарық күші (I) - жарық ағынының кеңістіктегі қысымы. Жарық күші дене бұрышының жарық ағынына қатынасымен анықталады, өлшем бірлігі

$$I = \frac{F}{w}, \quad (4.6)$$

кандела (кд).

мұнда I—жарық күші, кд;

F—жарық ағыны; лм;

w—дене бұрышы, ср.

Дене бұрышы (w) косинуспен шектелген кеңістіктің жартысы, ол сфераның ортасындағы шыңға ие. Дене бұрышы аудан (S) қатынасымен анықталады.

Жарықтылық (E) - жарық ағынының беттік тығыздығы. Ол беттік элементке түсетін жарық ағынының сол элемент ауданына қатынасымен анықталады және люкста (лк) өлшенеді.

$$E = \frac{F}{S}, \quad (4.7)$$

мұнда E—жарықтылық, лк;

F -беттік элементке түсетін жарық ағыны;

S - беттік элементтің ауданы, м².

Беттік айқындылық (B) – көз қабылдайтын жарық техникалық шама

$$B = \frac{I}{S \cdot \cos \alpha}, \quad (4.8)$$

мұндағы B—беттік айқындылық, кд/м²;

I—жарық күші, кд;

S—шағылысатын беттің ауданы, м²;

α - шағылысу бағытымен беттік жазықтың арасындағы бұрыш.

Фон—қаралатын ажырату объектісіне ескертілген беттік.

Фон мынадай болып саналады: ашық – беттің сәулелену коэффициенті 0,4 тен көп болса, орташа - 0,2 ден 0,4 дейін, қараңғы - 0,2-ге дейін.

Объект контрасты төмендегі теңдеумен анықталады.

$$K = |L_o - L_\phi| / L_\phi, \quad (4.9)$$

мұнда L_o - ажырату объектісінің айқындылығы, кд/м²;

L_ϕ - фон айқындылығы, кд/м².

Өндірісті жарықтануға қойылатын негізгі жарық техникалық талаптар:

а) жұмыс орындарын жарықтандыру деңгейі орындалатын жұмыстың сипатына сәйкес болуы қажет;

б) уақыт бойынша тұрақты және бір қалыпты жарықтануы;

в) қатты жылтырау мен айқындылықтың жоқ болуы;

г) жұмыс орындарында диффузиялық - жарық ағыны болуы керек (яғни көлеіңкесіз жарықтандыру).

Сонымен бірге, жасанды жарықтандыруды қамтамасыз етуге қосымша қойылатын талаптар: табиғи жарыққа жақын тиімді спектральді құрылым; жүйе зиянды факторлардың пайда болуын болдырмауы керек; электр және өрт қауіпсіздігін сақтау.

Өндірістік жарық СНиП- II-4-79 бойынша нормаланады, мұнда табиғи жарық КЕО (табиғи жарықтандыру коэффициенті) бойынша нормаланады.

$$e = \frac{E_{внутри}}{E_{наружи}} \cdot 100, \quad \%$$

Қазақстан үшін (IV жарық белдеуі) $e^{IV} = e^{III} \cdot m \cdot c$, мұндағы c , m - табиғи жарықтанудың түрін және климаттық зонаны ескеретін коэффициенттер. ТЖК (КЕО) көру жұмыс разряды мен жарықтану жүйесіне байланысты анықталады.

Табиғи жарықтануын есептеу үшін жарық тесуінің ауданын мына теңдеумен анықтаймыз:

$$100 \cdot \frac{S_0}{S_n} = \frac{e \cdot k \cdot \eta_0}{\tau_0 R_1} * k_{здания},$$

Қабырғадан түсетін жарық үшін

$$100 \cdot \frac{K_{фонар}}{S_n} = \frac{e \cdot k \cdot \eta_{фонаря}}{\tau_0 \cdot R_2 K_{фонаря}}.$$

Төбеден түсетін жарық үшін

мұнда S_n - бөлме ауданы;

e - нормалық КЕО;

k - запас коэффициенті, ол технологиялық процеске қарап СНиП бойынша алынады;

τ_0 - жарық өткізу коэффициенттері (төрт факторға байланысты: $\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4$; τ_1 - шыны сапасын ескереді; τ_2 - өткел түрі; τ_3 - қаптаудың түрі; τ_4 - күннен қорғайтын экрандардың барын ескеретін коэффициент);

R_1 - КЕО ның жоғарлауын ескеретін коэффициент;

R_2 - қабырға мен еденнің шағылуы арқылы КЕО ның жоғарлауын ескеретін коэффициент;

η_0 - терезенің жарық сипаттамасы;

$K_{зд}$ - жанында орналасқан ғимараттарды ескеретін коэффициент;

η_ϕ - фонарьдың жарық сипаттамасы;

K_{ϕ} - фонарьдың типін ескеретін коэффициент;

S_0 - терезенің ауданы;

S_{ϕ} - фонарьдың ауданы.

Жасанды жарықтандыру жарық көздері арқылы жасалады, олар екі топқа бөлінеді: қыздыру шамдары, газ разрядты шамдар.

Қыздыру шамдары- пайдалануы қарапайым, арзан өндіріледі, пульсация коэффициенті көп емес (3-ке дейін), кең аралықты температура және қысымда жұмыс істеуге мүмкіндік береді. Кемшілігі-ПӘК-і өте кіші (7%), жарық беруі төмен (25 лм/Вт-қа дейін), аз мерзімде қызмет етеді (2000 сағ), спектральді құрамы тиімді емес, жарықтандыруы бір қалыпты емес.

Газоразрядты шамдар - екі типті болады: төмен қысымды (түскі жарық), жоғары қысымды (ДРЛ, ДРН). Артықшылығы - үлкен мерзімде қызмет етеді (13 мың сағат), жарық беруі жоғары (25 ÷ 125 лм/Вт), жарықтандыруы бір қалыпты, спектральды құрамы тиімді болады. Кемшілігі: қымбат, қоршаған ортаның ластауы.

Жасанды жарықтандыру жұмыс разрядына байланысты жарықтылықпен (Е), жарықтандыру жүйесі және жарық көздерінің түріне байланысты СНиП II – 4 – 79 бойынша нормаланады.

Жасанды жарықтандыруды жарық ағынының пайдалану коэффициенті әдісімен (жалпы жарықтану), нүктелік (аралас) және меншікті қуат әдісі арқылы есептеледі.

4.1 Адам организміне электр тоғының әсер етуі

Электр қауіпсіздігі дегеніміз – ол, электромагниттік өрістің, статикалық электрленудің, электрлік доға мен электр тоғының зиянды және қауіпті әсерінен адамдарды қорғауды қамтамасыз ететін ұйымдастырылған және техникалық жұмыстар мен шаралардың жүйесі.

Егер адамның екі нүктесі арасында потенциалдар айырмасы болса, онда адам денесі арқылы электр тоғы жүреді. Адам бір уақытта жанасқан екі нүктелік тоқ тізбегі арасындағы кернеу -жанасу кернеуі деп аталады.

Дене арқылы жүретін электр тоқ адамға жылулық, биологиялық және электролиттік әсер етеді.

Тоқтың жылулық әсері электр энергиясының жылуға айналуында сезіледі және ол терінің, тканның және қан тамырларының қызуын тудырады.

Тоқтың биологиялық әсері тоқтың бұлшық еттер арқылы жүруінде оның қысқаруын тудырады.

Тоқтың электролиттік әсері қан құрамының өзгеруіне алып келеді.

Электр тоғына түсіп қалғанда төмендегі зақымдалулар болуы мүмкін:

күйіп қалу, терінің металдануы, электр белгілері, электроофтальмия, электр соққысы, механикалық зақымдалулар:

- электр күйігі электр тоғының жылулық әсерінде пайда болады.

Электр доғасының әсері нәтижесінде пайда болатын күйік өте қауіпті болып табылады, өйткені оның температурасы +3000-6000° С аралығында болады;

- терінің металдануы электр тоғының әсерінен металдың майда бөлікшелері теріге сіңуі нәтижесінде болады. Соның нәтижесінде терінің электр өтімділігі жоғарылайды, яғни оның кедергісі күрт төмендейді.

Электр белгілері деп, тоқ жүретін бөліктермен тығыз байланыста болғанда, яғни оны қысып ұстағанда теріде сұр немесе ақшыл – сары түсті дақтың қалуын айтамыз.

Электроофтальмия дегенде электр доғасының ультрафиолеттік сәулесі әсерінен көздің сыртқы қабатының зақымдалуын түсінеміз.

Электр соққысы болғанда, адам организмі жалпы зақымданады, яғни нерв және жүрек тамырларының бұзылуы, бұлшық еттерінің тырысуы пайда болады.

Механикалық зақымдалулар (тканның бөлшектенуі, сынықтар) адам бұлшық еттерінің тырысуы және де электр тоғының әсерінен төбеден құлау нәтижесінде болады.

Электр тоғынан зақымдалу сипаттамасы тоқтың тегі мен мәніне, оның жүріп өту жолына, әсер ету ұзақтығына, адамның жеке физиологиялық ерекшелігіне және оның зақымдалған кезіндегі жағдайына байланысты болады.

Көп жағдайларда тоқтың тегі мен мәні зақымдалу сипаттамасын анықтайды. Өндірістік жиіліктегі (50 Гц) кернеуі 500 В дейінгі айнымалы тоқтағы электр қондырғылары тұрақты тоққа қарағанда өте қауіпті. Бұл адам организмі клеткаларындағы күрделі биологиялық процестерге байланысты болады. Тоқ жиілігі өскен сайын, зақымдалу қаупі азаяды. М: бірнеше жүздеген кГц жиілікте электр соққысы сезілмейді.

Тоқтардың мәндеріне байланысты адам организміне әсер етуін былай бөлуге болады: сезінуші, жібермейтін және фибрилляциялық.

Адам айнымалы тоқтың (50 Гц) әсер етуін 0,5 тен 1,5 мА аралығында сезінеді, ал тұрақты тоқта – 5 тен 7 мА дейін. Бұл жағдайда, адамның саусақтары қалтырап, дірілдейді; тұрақты тоқта тері қыза бастайды. Бұл тоқтар бастапқы (пороговый) сезіну тоқтары деп аталады.

Жібермейтін тоқтарда қолдың бұлшық еттерінің тырысуы болады, яғни адам өз еркімен қолын тоқ жүретін бөліктерден ала алмайды. Оның мәндері айнымалы тоқ үшін – 10-15 мА, ал тұрақты тоқ үшін – 50-80 мА. Тоқтың ары қарай өсуі жүрек қан тамырларының зақымдалуына алып келеді. Дем алуы қиындайды және тоқтайды, жүрек жұмысы өзгереді.

Фибрилляциялық тоқ жүрек фибрилляциясын, яғни жүрек тамырларының әлсізденуін, түршігуін және тырысуын пайда етеді. Фибрилляция нәтижесінде жүректен бүркелетін қан қажетті өмір сүру органдарына бармайды және ең бірінші кезекте миды қанмен қамтамасыз ету бұзылады. Қан бармаған адам миы ары кетсе 5-8 минут өмір сүреді, сондықтан да бұл жағдайда зақымдалған адамға тез және өз уақытында алғашқы көмек көрсету керек. Фибрилляция тоғының мәндері –80 нен 5000 мА аралығында болады.

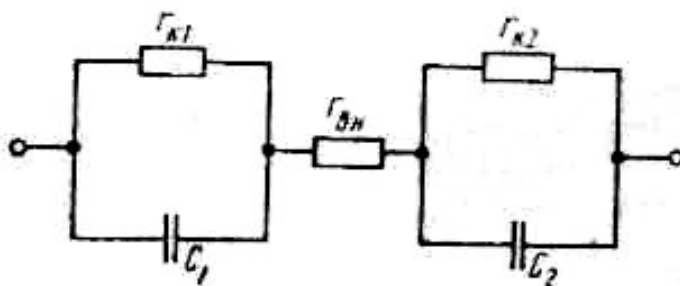
Адам денесі арқылы өтетін электр тоғының мәніне және тегіне байланысты зақымдалу сипаттамасы 4.1- кестеде көрсетілген. Тоқтың жүру жолы зақымдалу сипатына бір шама әсерін тизізеді. Өте қауіпті жол – жүрек, бас миы және тыныс алу кеудесі арқылы өтетін жол болып табылады. Электр травматизмін талдауда «оң қол – аяқ», «қол – қол» жолдары жиі кездеседі, ал «бас – аяқ», «бас – қол», «аяқ – аяқ» жолдары сирек кездеседі.

Адам организмі арқылы өтетін тоқтың мәні берілген кернеуге және дене кедергісіне байланысты болады. Кернеу қаншалықты көп болса, соншалықты көп ток адам арқылы өтеді.

4.1-кесте. Адам денесі арқылы өтетін электр тоғының мәніне және тегіне байланысты зақымдалу сипаттамасы

I, mA	50 Гц айнымалы токта	Тұрақты токта
0,6-1,5	Сезіну пайда болуы, қол саусақтарының дірілі	Сезілмейді
5 – 7	Қолдың қалтырауы	Сезіну пайда болуы, Терінің қызуы
8–10	Қолды электродтан әрең алуға болады, бірақ сүйектер қатты ауырады	Қызудың көтерілуі
20-25	Қолды электродтан тартып алу мүмкін емес, тыныс алу қиындайды	Бұлшық еттің кішкене қысқаруы
50-80	Тыныс алу тоқтайды, жүрек фибрилляциясы басталады	Қатты қызу, қолдағы бұлшық еттің қысқаруы, тыныс алу қиындайды
90-100	Жүрек соғуы (әсер ету уақыты 3 с көп) және тыныс алу тоқтайды	Тыныс алу тоқтайды

Адамның кедергісі тұрақты емес және ол әр түрлі факторлаға байланысты болады. М: терінің жағдайы, жанасу тығыздығы және шамасы, берілген кернеудің мәні және тоқтың әсер ету уақыты. Электр тоғының әсері астында тұрған адам денесінің кедергісін 4.1 – суретіндегі эквивалентті сұлба түрінде көрсетуге болады. r_{R1} кедергісі ток кірген жердегі тері бетінің кедергісі мысалы, қолдың кедергісі.



4.1 – сурет. Электр тоғының әсері астында тұрған адам денесі кедергісінің эквивалентті сұлба

r_{RI} кедергісі терінің жағдайына байланысты 10 нан 100 кОм дейін өзгеріп тұруы мүмкін. Зақымдалмаған құрғақ терінің кедергісі 100 кОм да көп болуы мүмкін. Тері зақымдалғанда, ластанғанда және ылғалданғанда оның кедергісі күрт төмендеп он килоОмға дейін жетуі мүмкін. C_1 – адаммен сымның арасындағы сыйымдылық. Конденсатор диэлектригі тері қабаты және адам денесімен сымның арасындағы ауа болып табылады. Егер жанасу кернеуі үлкен болса, онда диэлектрик тесіледі және тізбек тек ғана r_{BH} кедергісі арқылы тұйықталады.

r_{BH} кедергісі – бұлшық ет және дене ішкі органдарының кедергісі. Ол үлкен емес - 0,4 және 1 кОм аралығында болады. Өйткені адамның бұлшық еттерінде органикалық заттар бар, олар электр тоғын жақсы өткізеді.

Әдетте, электр тораптарындағы қауіпсіздікті талдауда және есептеуде адам денесінің кедергісі активті және 1 кОм деп қабылданады.

Электр тоғынан зақымдалудың сипаттамасы уақытқа байланысты болады. Тоқтың көп уақыт әсер етуінде тері қызады және тері терлеп ылғалданады, соның нәтижесінде оның кедергісі төмендеп адам арқылы ток өтетін тоқтың шамасы күрт өседі.

Электр тоғынан зақымдалудың сипаттамасы адамның жеке физиологиялық ерекшеліктеріне де байланысты. Егер адамның денсаулығы жақсы болса, онда электр зақымдалуы аз болады. Егер адамның жүрек қантамырлары жүйесі, терісі, жүйке жүйесі ауыратын болса және мас күйінде болса, онда электр тоғынан алған зақымдалуы аз мәнді тоқтың өзінде күрделі болады.

Тоқтың жүріп өту уақытына байланысты жанасу кернеуі және тоқтың рұқсат етілетін мәндері 4.2 - кестесінде келтірілген. Олар қорғаныстық құрылғысын есептеуде қолданылады.

4.2 кестесі. Адам арқылы жүріп өтуге рұқсат етілетін жанасу кернеуі және ток

Қондырғы	Нормалық шамасы	Тоқтың әсер ету уақыты, с					
		,1	,2	,5	,7	,0	,0
Айнымалы тоқта 1000 В дейінгі кернеуде, жиілігі 50 Гц 400 Гц жиілікте	U_{PP}, B	00	50	00	5	0	6
	I_h, mA	00	50	00	5	0	6
	U_{PP}, B		00	00	40	00	
	I_h, mA		00	00	40	00	
Тұрақты тоқта	U_{PP}, B						

	I_h, mA	00	00	50	00	50	00
		00	00	50	00	50	0

4.2 Жұмыс орындағы микроклимат

Электр қондырғылары деп электр энергиясын өндіретін, түрлендіретін, тарататын және тұтынатын қондырғыларды айтады. Оларға генераторлар, электр қозғалтқыштары, трансформаторлар, түзеткіштер, радио және телевизиялық байланыс аппаратуралары және т. б. жатады.

Электр қондырғыларында жұмыс жасау қауіпсіздігі электр сұлбаларына және электр қондырғыларының параметрлеріне, номиналды кернеуге, қоршаған ортаға және пайдалану жағдайына байланысты болады.

Қауіпсіздікті қамтамасыз ету үшін барлық электр қондырғылары (ПУЭ) электр қондырғыларын орнату ережелеріне (ЭОЕ) сәйкес 1000 В дейінгі және 1000 В жоғары болып бөлінеді.

1000 В жоғары қондырғылар өте қауіпті болғандықтан, олардың қорғаныс қауіпсіздік шараларына қатаң талаптар қойылады.

Электр қауіпсіздігі деңгейі бойынша жұмыс жағдайлары 3 категорияға бөлінеді: адамдардың электр тоғына түсіп қалу қаупі өте жоғары, өте қауіпті және жоғары қауіп жоқ.

Өте жоғары қауіпті жағдайда төмендегі белгілердің біреуі болуы керек:

- ылғалдылық (салыстырмалы ылғалдылық 75% тен көп);
- температура (ұзақ уақыт бойы +35° С астам болатын), т.б.

Бұл жағдайларда қауіпсіздікті азайту үшін кіші кернеуді (42 В) пайдаланған дұрыс.

Өте қауіпті жағдайда төмендегі белгілердің біреуі болуы керек:

- ерекше ылғалдылық (салыстырмалы ылғалдылық 100% ке дейін);
- электр құрылғысының тоқ жүретін бөліктерін және оқшауламаларды бұзатын химиялық активті орта.

Бұл жағдайларда қауіпсіздікті азайту үшін кіші кернеуді (12 В) пайдаланған дұрыс.

Жоғары қауіпті жоқ жағдайда жоғарыда көрсетілген белгілер болмайды.

4.3 Электр тораптарындағы қауіпті талдау

Адам арқылы өтетін тоқ электр қондырғысындағы кернеуге, адамның торапқа жанасу сұлбасына, тораптағы бейтарап режиміне, сонымен бірге электр торабындағы жерге салыстырғандағы сыйымдылық және кедергіге байланысты болады.

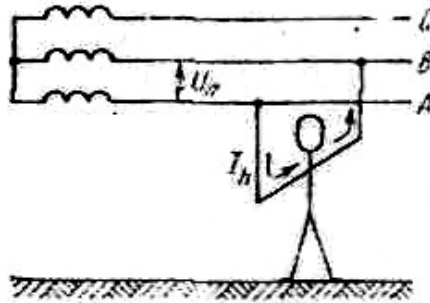
Кернеу астындағы электр қондырғының бөліктеріне адамның бір фазалы (бір полюсті) және екі фазалы (екі полюсті) жанасуы болуы мүмкін. Екі фазалы жанасу дегеніміз, кернеу астындағы электр қондырғысына адамның бір уақытта екі фазасына жанасуы. Бір фазалы жанасу дегеніміз,

адамның кернеу астындағы электр қондырғысының бір фазасына жанасуын айтамыз. Адамның электр торабындағы екі фазасына жанасуы өте қауіпті, өйткені, бұл жағдайда жанасу кернеуі тораптағы кернеудің шамасына тең болады

4.3 - суретінде адамның электр торабындағы екі фазасына жанасуы келтірілген. Мұнда адам арқылы өтетін тоқты төмендегі теңдеумен анықтауға болады

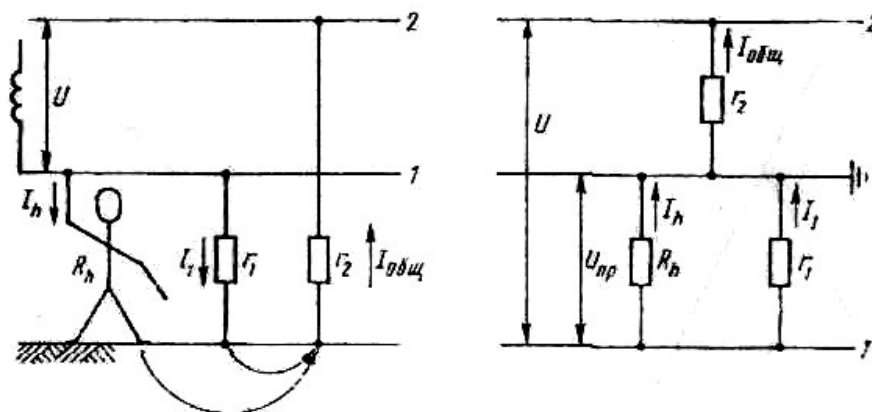
$$I_h = \frac{U_{\text{ЛП}}}{R_h} = \frac{U_{\text{Л}}}{R_h} = \frac{\sqrt{3} \cdot U_{\phi}}{R_h} \quad (4.3)$$

Бұл теңдеудегі I_h - адам арқылы өтетін ток, А; $U_{\text{жс}}$ - жанасу кернеуі, В; $U_{\text{Л}}$ - айнымалы тоқтағы үш фазалы тораптар үшін сызықтық кернеу, В; U_{ϕ} - фазалық кернеу, В; R_h - адам денесінің кедергісі, Ом.



4.3 –суреті. Адамның электр торабындағы екі фазасына жанасуы

Адамның бір фазаға жанасуындағы эквиваленті орынбасу сұлбасына 4.3-суретінде көрсетілген. Есептеуді оңайлату үшін сымның жерге салыстырғандағы сыйымдылығын ескермейміз.



4.3 – сурет. Адамның электр торабына бір фазалы жанасуы

Адам арқылы өтетін тоқ төмендегі теңдеумен анықталады

$$I_h = \frac{U_{PP}}{R_h} \quad (4.2)$$

Орынбасу сұлбасын қолдана отара, жанасу кернеуін былай анықтаймыз

$$U_{жан} = \frac{R_h \cdot r_1}{R_h + r_1} \cdot I_{Общ} \quad (4.3)$$

Орынбасу сұлбасынан жалпы тоқты табамыз

$$I_{жалпы} = \frac{U}{r_2 + R_h \cdot r_1 / (R_h + r_1)} \quad (4.4)$$

Жалпы тоқ теңдеуінен жанасу кернеуін былай анықтаймыз:

$$U_{жан} = U \cdot \frac{r_1 \cdot R_h}{r_1 \cdot r_2 + r_1 \cdot R_h + r_2 R_h} \quad (4.5)$$

(4.5) теңдеуін қолдана отара адам арқылы өтетін тоқ төмендегі түрге келеді

$$I_h = \frac{U_{жан}}{R_h} = U \cdot \frac{r_1}{r_1 \cdot r_2 + r_1 \cdot R_h + r_2 R_h} \quad (4.6)$$

Егер $r_1 = r_2 = r_{ИЗ}$ ($r_{ИЗ}$ – оқшаулама кедергісі) болса, онда (4.5) және (4.6) теңдеулерінен

$$U_{жан} = U \cdot \frac{R_h}{2R_h + r_{ИЗ}} \quad (4.7)$$

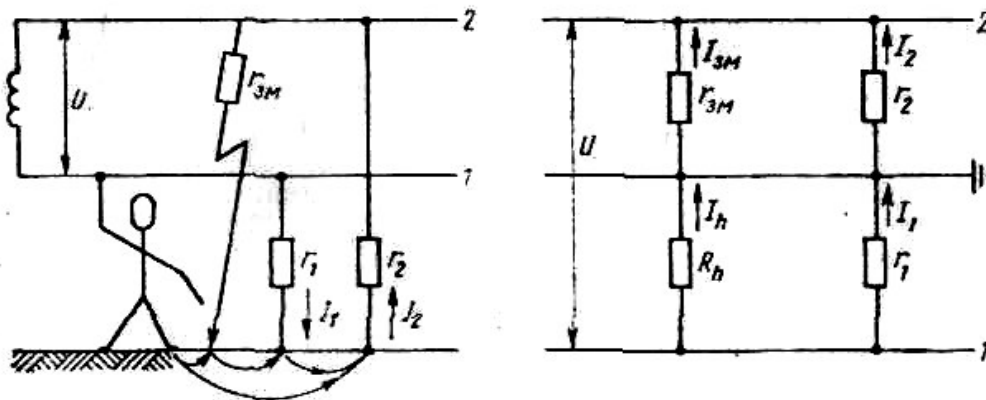
$$I_h = U \cdot \frac{1}{2R_h + r_{ИЗ}} \quad (4.8)$$

(4.8) теңдеуіне талдау жасап отыра, торап оқшауламасының кедергісі өскен сайын, адам арқылы өтетін тоқ азаятындығын байқаймыз.

Егер адам оқшауланған еденде аяқ киімі мен тұрған болса, онда еденнің r_{II} және аяқ киімнің $r_{Об}$ кедергілерін ескеру керек, сонда (4.8) теңдеуі төмендегі түрде болады

$$I_h = U \cdot \frac{1}{2 \cdot (R_h + r_{II} + r_{Об}) + r_{ИЗ}} \quad (4.9)$$

Бір фаза жерге тұйықталған апатты режимде эквивалентті сұлбаны төмендегі сұлбаға ауыстыру керек.



4.3 – сурет. Апатты режимде адамның электр торабына жанасу сұлбасы

Жерге тұйықталу кедергісі r_{3M} , әдетте, оқшауламалар кедергісінен r_1 , r_2 және адам кедергісінен R_h аз болады, сондықтан, (4.5) және (4.6) теңдеулеріндегі r_2 кедергісін эквивалентті кедергіге $r_{\text{э}} = r_2 \cdot r_{13} / (r_2 + r_{13})$ ауыстыра отыра төмендегілерді аламыз

$$U_{\text{жан}} = U \cdot \frac{r_1 \cdot R_h}{r_1 \cdot r_{\text{э}} + r_1 \cdot R_h + r_{\text{э}} R_h}, \quad (4.10)$$

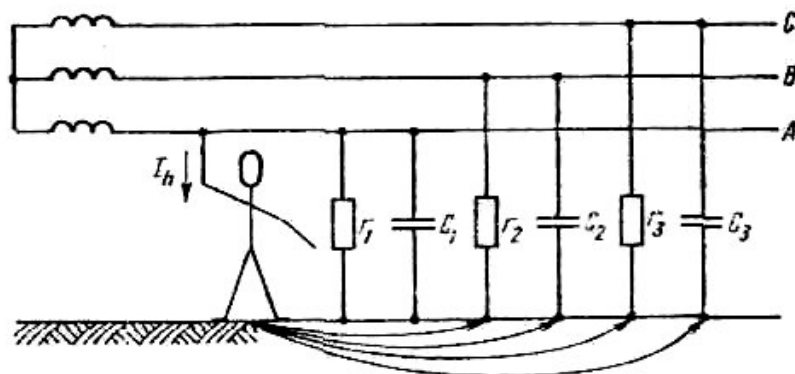
$$I_h = U \cdot \frac{r_1}{r_1 \cdot r_{\text{э}} + r_1 \cdot R_h + r_{\text{э}} R_h}. \quad (4.11)$$

r_{3M} кедергісі нөлге жақын болғандықтан, (4.10) және (4.11) теңдеулері төмендегі түрге енеді:

$$U_{\text{жан}} = U; \quad I_h = \frac{U}{R_h}. \quad (4.12)$$

Апаттық режимде жанасу кернеуі тораптың кернеуіне тең болады.

Құрылғыны қоректендіру үшін көп жағдайда бейтарабы жерге қосылған және оқшауланған үш фазалы тораптар қолданылады. Бейтарабы оқшауланған тораптың бір фазасына адам жанасқанда дене арқылы өтетін тоқ оқшаулама кедергісіне және жерге салыстырғандағы фаза сыйымдылығына байланысты болады (4.4 - сурет).



4.4 – сурет. Бейтарабы оқшауланған үш фазалы тораптарға бір фазалық жанасу

Сымдар оқшауламасындағы кедергілер $r_1 = r_2 = r_3 = r_{ИЗ}$ және сыйымдылықтар $C_1 = C_2 = C_3 = C$ бір - біріне тең болса, онда

$$I_h = \frac{U_\phi}{R_h} \cdot \frac{1}{\sqrt{1 + r_{ИЗ} \cdot (r_{ИЗ} + 6 \cdot R_h) / 9 \cdot R_h^2 \cdot (1 + r_{ИЗ}^2 \cdot \omega^2 \cdot C^2)}} \quad (4.13)$$

Егер фазалардағы жерге салыстырғандағы сыйымдылықтарды ескермесек, онда (3.8) теңдеуі төмендегі қарапайым түрге келеді

$$I_h = \frac{U_\phi}{R_h + (r_{ИЗ} / 3)} \quad (4.14)$$

Егер оқшаулама кедергісі шексіз көп болса $r_1 = r_2 = r_3 = \infty$, ал фазалар сыйымдылығы бір-біріне тең болса (кабелдік желілер үшін) $C_1 = C_2 = C_3 = C$, онда адам арқылы өтетін тоқты есептеу теңдеуі төмендегідей

$$I_h = \frac{U_\phi}{R_h^2 + (x_c / 3)^2} \quad (4.15)$$

мұнда $x_c = 1 / \omega \cdot C$ – сыйымдылықты кедергі, Ом.

(4.15) теңдеуінен тораптағы сыйымдылық қаншалықты көп болса, адам арқылы өтетін тоқ сонша үлкен болатындығын көреміз.

Бейтарабы оқшауланған тораптың апатты жұмыс режимінде бір фазаның жерге тұйықталуы пайда болады. Егер бұл жағдайда жерге тұйықталу кедергісі аз болса, онда зақымдалмаған фазаға адам жанасқан кезіндегі дене арқылы өтетін тоқ төмендегі теңдеумен анықталады

$$I_h = \frac{U_\phi \cdot \sqrt{3}}{R_h + r_{ЗМ}} \quad (4.16)$$

Тәжірибеде $r_{ЗМ}$ кедергісі R_h кедергісінен бір шама аз болады, сондықтан жанасу кернеуін былай анықтаймыз

$$U_{ИП} = I_h \cdot R_h = U_\phi \cdot \sqrt{3} = U_{Л} \quad (4.17)$$

Сонымен, адам желілік кернеу астында болады және оқшаулама кедергісі қорғанысқа ие емес. Сондықтан, бұл жағдайдағы жанасу қалыпты жұмыс торабына жанасуға қарағанда өте қауіпті болады.

4.4 Электр қондырғыларындағы қорғаныс шаралары

Травматизмді талдау жасаған кезде, электр травмаларының жартысынан көбі құрылғының тоқ жүретін бөліктеріне жанасқан кезде болатындығын көреміз.

Электр қондырғының тоқ жүретін бөліктері деп жұмыс режимінде тоқ жүретін бөліктерін айтамыз. Оларға мысал ретінде: сымдар, аппаратура элементтерінің түйіспелері және т.б. айтуға болады.

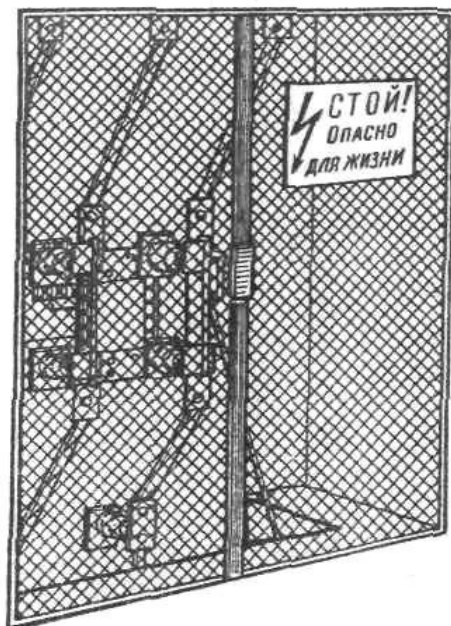
Электр қондырғыларының тоқ жүретін бөліктеріне жанасудан қорғануда қоршаулар, блокировка, оқшаулау, және тоқ жүретін бөліктерді қол жетпейтін биіктікте орналасу қолданылады.

Қоршаулар сыртқы қабатты, шкаф түрінде және т.б. болуы мүмкін. Олар құрылғының бір бөлігі ретінде немесе тасымалды түрде болады. Қоршаулар бір тегіс және тор түрінде орындалады. Оларды ашу және жабу тек ғана арнайы құрылғылар немесе құралдар көмегімен орындалады (4.1-сурет).

Блокировка өте қауіпті жұмыс жағдайларында қолданылады. Олар радио және телевизиялық аппаратураларда, сынақ жасау стендтерінде және жоғары кернеуде оқшауламаны сынауға арналған қондырғыларда қолданылады. Жұмыс істеу принциптері бойынша олар электрлік және механикалық түрлерге бөлінеді.

Оқшаулау. Қазіргі уақытта пайдалану жағдайларына байланысты жұмыстық және қайталама оқшаулау пайдаланылады. Адамдардың тоққа түсіп қалмауын қамтамасыз етуде оқшауламаларды бақылау жүргізіледі. (1 кВ дейінгі электр тораптарында оқшаулама кедергісі 0,5 мОм).

Адамдарға төнетін қауіпті ескерту үшін ескеру плакаттары ілініп қойылады: М: “Жоғары кернеу”, “Адам өміріне қауіпті”, “Тоқта” “Қоспа, адамдар жұмыс істеп жатыр”, “Мына жерде жұмыс істе”, “Жерге қосылған” және т.б. қолданылады.



4.1-сурет. Электр қондырғысын тормен қоршау

4.2 Электр қондырғыларының тоқ жүрмейтін бөліктеріне жанасудан қорғану

Жоғарыда атап өткен сияқты, тоқ жүрмейтін бөліктері деп аталатын электр қондырғысының металдық корпусы кернеу астында болуы мүмкін.

Адамдарды электр қондырғыларының тоқ жүрмейтін бөліктеріне жанасудан қорғауда қорғаныстық жерге қосу және нөлдеу қолданылады.

Қорғаныстық жерге қосу дегеніміз, кернеу астында қалуы мүмкін болған электр қондырғысының металды бөліктерін жермен арнайы қосуды айтамыз. Қорғаныстық жерге қосуды машина корпусының, аспаптардың, электр инструменттерінің, каркас, щит, пульт және шкафтардың металды корпустарына, сонымен бірге кабелдік муфталар, электр сымның болаттан жасалған трубаларының металды бөліктеріне жүргізеді.

Қорғаныстық жерге қосудың мақсаты, ол корпус пен жер арасындағы кернеуді, яғни жанасу кернеуін және де соған байланысты адам денесі арқылы жүріп өтетін тоқты қауіпсіз шамаға дейін азайту болып табылады.

Жерге қосу құрылғысы жерге қосқыштан және жерге қосатын сымдардан тұрады (4.2 –сурет).

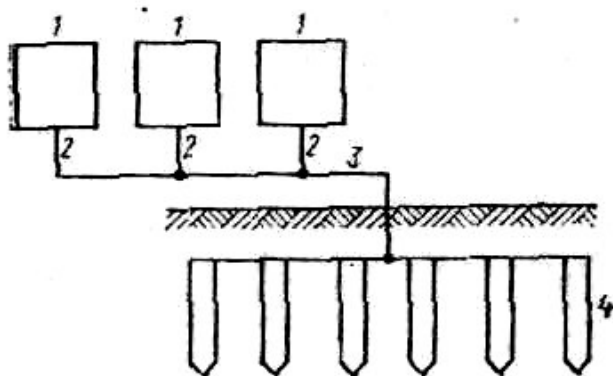
Жерге қосқыштар табиғи және жасанды болуы мүмкін.

Табиғи жерге қосқыш ретінде ғимараттардың жерге көмілген тоқ өткізетін бөліктерін, су және басқа құбырларын, кабелдің қорғасын қабатын пайдалануға болады. Бірақ та табиғи жерге қосқыш ретінде газ және басқа жарылғыш заттар жүретін құбырларды пайдалануға болмайды.

Жасанды жерге қосқыш ретінде болат, мыстан жасалған құбырларды (уголки) және басқа металдарды қолданады. Олар траншеяға топырақ қатпайтын тереңдікте көміледі. (М: құбыр диаметрі 5-6 см, ал қалыңдығы 3.5 мм аз болмауы керек). Жерге қосқыштар бір-бірімен балқытылып (сварка) бекітіледі.

Жерге қосатын сымдар оқшауланған және оқшауланбаған болуы мүмкін. Егер жерге қосатын сым ретінде мыс сымды пайдалансақ, оның қимасы 4 мм^2 аз болмауы керек, ал алюминийде -6 мм^2 .

Жерге қосқыштар және жерге қосатын сымдардың жалғанған түйіспелері сенімді болуы керек.



4.2- сурет. Қорғаныстық жерге қосу құрылғысының сұлбасы

Қорғаныстық жерге қосу құрылғысының техникалық жағдайын анықтау үшін оларды мезгілімен тексеріп және кедергісін өлшеп тұру керек. Кәсіпорындарда қорғаныстық жерге қосу құрылғысының кедергісін өлшеу жылына 2 рет жүргізіледі: жазда (топырақ кепкен кезде), қыста (топырақ қатқан кезде). Электр станцияларында тексеру әр жыл сайын болса, ал әуе

және кабелдік желілерінде найзағай болу мерзімінің алдында (сәуір-мамыр айлары) жүргізіледі.

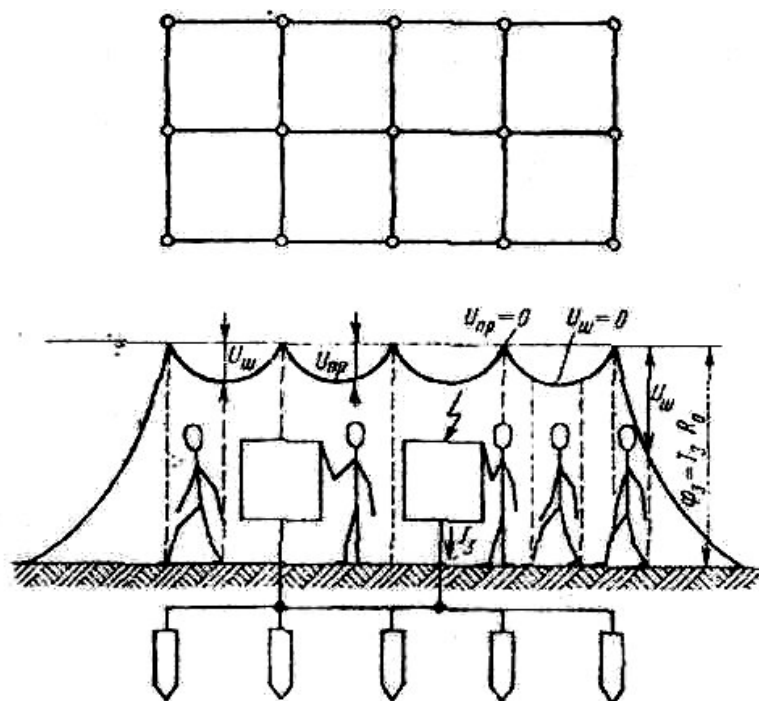
Өлшеуді арнайы М-416 өлшегісімен немесе амперметр-вольтметр әдісімен жүргізуге болады.

Қорғаныстық жерге қосу өзінің құрылғысы бойынша сыртқа шығарылған және контурлық болуы мүмкін.

Соңғы жылдары контурлық жерге қосу көп қолданысқа ие. Ол аудандардағы потенциалдардың теңесуіне және қадам мен жанасу кернеулерінің азаюына алып келеді. 4.3 – суретінде контурлық жерге қосу сұлбасы және аудандағы потенциалдардың таралу графигі келтірілген. Суреттен, егер адам екі жерге қосқыштар арасында тұрса, онда жанасу кернеуі үлкен, ал егер адам жерге қосқыштың үстінде тұрса, онда жанасу кернеуі кіші екендігін көреміз. Жерге қосқыштан қашықтаған сайын жанасу кернеуі азаяды және контурлық жерге қосқыш шетінде күрт өседі, мұнда потенциалдың күрт құлауы байқалады.

Нөлдеу кернеуі 1000 В дейінгі бейтарабы терең жерге қосылған төрт сымды үш фазалы тораптарда қолданылады.

Нөлдеу дегеніміз, кернеу астында қалуы мүмкін болған электр қондырғысының металды бөліктерін қорғаныстық нөлдік сыммен қосуды айтамыз.



4.3 – сурет. Контурлық қорғаныстық жерге қосу сұлбасы

Нөлдеудің мақсаты – бір фазалы қысқа тұйықталуда корпусқа өтіп кетуінің алдын - ала отыра, қорғанысты іске қосып және қоректену торабынан электр қондырғысын минимум уақыт ішінде ажырату.

Қорғау құралы ретінде балқымалы сақтандырғыштар және автоматты ажыратқыштар қолданылады. Үлкен тоқ пайда болған кезде (ҚТ тоғы)

балқымалы сақтандырғыштар күйеді немесе автоматта электромагнитті тізбек ашылады, тізбек ажыратылады және электр қондырғысы тораптан ажыратылады.

Нөлдеу құрылғысында қоректену көзінің бейтарабы жерге міндетті түрде қосылуы керек. Ол, нөлдік сымдағы, сонымен бірге фазалардың жерге кездейсоқ тұйықталуында электр қондырғысының корпусындағы кернеуді азайту үшін жүргізіледі.

Қорғаныстық нөлдік сым белгілі қашықтықта қайталап жерге міндетті түрде қосылуы керек. (М: ӘЖ үшін 250 м ден).

Нөлдік сымды қайталап қосу нөлдік сым үзіліп қалу жағдайында немесе фазаның корпусқа тұйықталуында электр тоғының әсер ету қауіпін азайту мақсатында жүргізіледі.

Қондырғыны пайдалану алдында, сонымен бірге мерзімді түрде нөлденуді тексеріп тұру қажет.

4.3 Қорғаныстық ажырату құрылғысы

Қорғаныстық ажырату дегеніміз, электр қондырғысына тоққа түсіп қалу қауіпі туған кезде, оны автоматты түрде ажыратуды қамтамасыз ететін тез әрекет етуші қорғаныс.

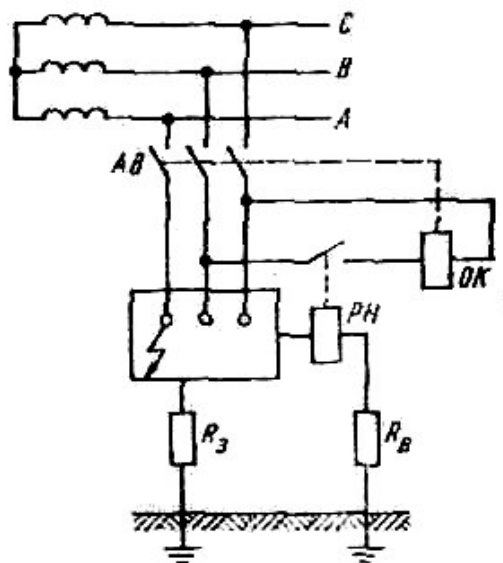
Тоққа түсіп қалу қауіпі туған кезде, сол тізбекті тез ажырату мақсатында қорғаныстық ажырату құрылғысы (УЗО) қолданылады.

Олар жерге қосу немесе нөлдеу сияқты қорғаныс шараларынан бөлек түрде де және оларға қосымша түрде де қолданылады. (Көп жағдайларда 1кВ электр қондырғыларында қолданысқа ие: жылжымалы электр құрылғылары, электр инструменттері, жерге қосу және нөлдеу мүмкін болмаған стационарлы электр қондырғыларында). Қорғаныстық ажырату құрылғысының (УЗО) сұлбасы 4.4 – суретінде көрсетілген.

УЗО сұлбасы корпус пен көмекші жерге қосқыш аралығына қосылған кернеу релесінен КР (РН) тұрады, ол корпусындағы кернеудің мәні шекті рұқсат етілетін мәнінен асып кеткенде іске қосылады және ол автоматты ажыратқыштың (АВ) айыру катушкасымен (ОК) тізбекті тұйықтайды.

Жетістігі – қарапайым және сенімді.

Кемшілігі – селективтілігі және өзін өзі бақылауының жоқтылығы сонымен бірге көмекші жерге қосқышты пайдалануы. Электр қондырғыларында барлық жұмыстар административтік, техникалық, кезекші және операторлы – жөндегіш адамдармен жүргізіледі. Бұл кезде ескерілетін жайт, зақымдану деңгейі жұмыс жасау шарттарына байланысты. Электр құрылғыларындағы барлық жұмыстар наряд бойынша, бұйрық бойынша және келесі пайдалануға рет-ретімен өту бойынша орындалады. Наряд - қалыптасқан форманың бланкісінде безендірілген жұмыс орындауға арналған жазбаша есеп. Нарядта бригада құрамы, жұмыс орны, басталу уақыты және жұмыстың аяқталуы; оның қауіпсіз орындалу шарттары, қауіпсіздікке жауап беретін адамдар көрсетілген. Жұмыстың қауіпсіздігіне жауап беретіндер деп наряд шығаратын, жұмыс басқарушы, бригаданың мүшесі болатын адамды айтамыз.



4.4 – сурет. Қорғаныстық ажырату құрылғысының (УЗО) сұлбасы.

Қорытынды

Қазіргі заман талаптарына сай ұялы телефондардың дамуына, олардың технологиялық өсуіне қарай, музыкалық плеерлер жеке құрылғылар ретінде әр түрлі қосымшалар өте танымал. Мобильді құрылғылар жадына музыка гигабайттарын жүктеуге және оны жақсы сапада тыңдауға болады. Бұл қосымша болашақта стильді әр түрлі түсті тақырыптарды таңдау мүмкіндігімен қарастырылатын, сондай-ақ қарапайым және сол уақытта өте стильді көрініс тауып ерекшеленетін қосымшалардың бірін қарастырып кеттік. Форматтардың бір бөлігі бойынша қосымша MP3 және WAV, OJJ, MIDI және FLAC ойнату мүмкіндіктері өте көп. Дыбысты реттеу үшін түрлі баспақтар мен бас реттегіші бар кіріктірілген эквалайзер де бар. Ыңғайлы свайппен басқарылатын, папкалар бойынша тректерді ойнату мүмкіндігі және орындаушыларды, жанрлар және басқа да талаптар бойынша жылдам іздеу мүмкіндігі болашақта қарастырылатын жаңа плеерді – Z Reader Music Player қосымшасын ұсындым.

Android платформалары мобильдік және компьютерлік құрылғыларға арналған қосымшаның құрылуын, онымен қалай жұмыс істеу принциптерінің барлығын атап өттік.

Z Reader Music Player қосымшасы қолдануға оңтайлы, өзіндік қолдану нұсқаулығы бар дербес қосымша. Z Reader Music Player қосымшасымен интернет желісі болған және болмаған жағдайларда қолдану жағынан өте оңтайлы қосымша.

Z Reader Music Player қосымшасының жұмыс істеу принципі: дерекқор файлы, түзету, бөліп алу, көрініс, транспорт, тректер, құру, эффекттер, анализ, құралдар, анықтамалық қосымшалары арқылы жүзеге асырылып отырады. Қосымшаның жұмыс істеу принципі берілген қосымшалардың алгоритмі бойынша ғана жұмыс жасайды.

Z Reader Music Player қосымшасының бірден бір ерекшеліктері көп екенін дәлелдермен дәйектедік. Мысалға алсақ, қарапайым бағдарлау сілтемелерін пайдалана отырып, тұлғаның әуесқой әуенін де жазуға болады. Мүмкін жеке бір тұлға битбокспен (beatbox) айналысатын болса, оның ойындағы әуенге жетіспей тұрған элементтер мен эффектілерді қосып, толыққанды су жаңа өнім алуға болады. Немесе әуесқой әнсүйер қауым әуеннің жаңадан жазылған немесе көшірілген екенін де тауып біле алады, яғни әуеннің плагиаттық пайыздарын шығаруға болатынын жеткізбекпін. Әрине Z Reader Music Player қосымшасы жаңадан құрастырылып шығарылған болғандықтан әлі де бірқатар түзетулер мен қосымша материалдардың енгізілуі қажеттігі бар. Алайда артықшылықтары басқа қосымшаларға қарағанда көп екендігін ескере отырып, жасап шыққан Android платформасына арналған Z Reader Music Player қосымшасы пайдаланылуға толық жарамды болып саналады.

Әдебиетер тізімі

- 1 Barry, Burd Android Application Development All-in-One For Dummies® / Barry Burd. - Москва: Машиностроение, 2011. - 816 с.
- 2 Нотон Java. Справочное руководство. Все, что необходимо для программирования на Java / Нотон, Патрик. - М.: Бином, 2015. - 448 с.
- 3 Роджерс, Рик Android. Разработка приложений / Рик Роджерс и др. - М.: ЭКОМ Паблшерз, 2010. - 400 с.

А қосымшасы (міндеті)

Техникалық тапсырма

А.1 Талаптарды талдау

Бірінші кезекте қосымшаны қанағаттандыруы тиіс талаптар анықталды:

- қолданба ұялы құрылғы ағысты(ақпаратты) оқу керек;
- әр оқылған файл смартфонның жедел жадына жүктеледі;
- файл музыкалық плеер арқылы орындалады.

А.2 Бағдарламаны әзірлеу

Бағдарламаны әзірлеу үшін кітапханалар жүктеліп, графикалық интерфейс құрылды.

Файл оқытылып дайын болғаннан кейін біз оны android-studio қосымшасы арқылы android платформасына арналған бағдарлама жасау үшін қоладанамыз.

А.3 Android-studio қосымшасын іске асыру

Бағдарлама интерфейсі келесідей болады.

Объектілер тізімі. Бұл компонентте қолданба анықтай алатын барлық объектілер тізімі толықтай көрсетіледі.

Бұл мәзір қолданбаны толық жұмыс істеу алаңы. Яғни бағдарламаның жұмыс істеуі.

Бағдарлама туралы. Қолданба жайлы барлық ақпаратты көру мүмкіндігі.

А.4 Қолданбаның жұмыс істеу процесі

Төмендегі сурттерде қолданбаның жұмыс істеу процесі көрсетілген.

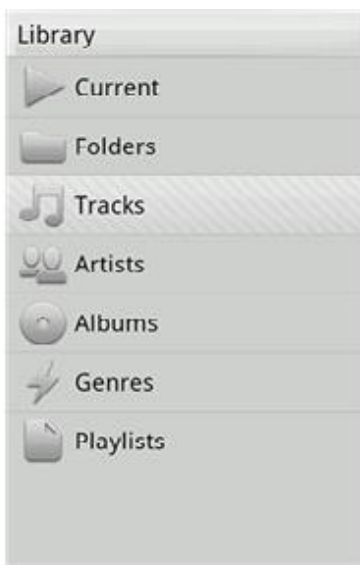
– плеердің визуализациясының 6 түрі, теңшелетін эквалайзер және көптеген басқа параметрлері болуы;

– музыканы жүктеу және тыңдау мүмкіндігі болу керек.

Ерекшеліктері:

- сәнді дизайн;
- топтамалық композициялар;
- ойнату тізімдерін басқару;
- 4 жұмыс үстелі виджеті және 2 құлыптау экраны;
- барлық белгілі дыбыстық файл пішімдерін қолдау мүмкіндіктері болуы қажет.

А қосымшасының жалғасы



А.1-сурет – Қолданбаның жұмыс істеу процесі

Ә қосымшасы (міндеті)

Бағдарламаның листингі

```
package com.example.macbookpro.sosmedical;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends ActionBarActivity implements View.OnClickListener{
    UserLocalStore userLocalStore;
    EditText etName, etAge, etUsername;
    Button bLogout, bOnas, bKontakty, bUslugi;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        etUsername = (EditText) findViewById(R.id.etUsername);
        etName = (EditText) findViewById(R.id.etName);
        etAge = (EditText) findViewById(R.id.etAge);
        bLogout = (Button) findViewById(R.id.bLogout);
        bOnas = (Button) findViewById(R.id.bOnas);
        bKontakty = (Button) findViewById(R.id.bKontakty);
        bUslugi = (Button) findViewById(R.id.bUslgi);
        bLogout.setOnClickListener(this);
        bOnas.setOnClickListener(this);
        bKontakty.setOnClickListener(this);
        bUslugi.setOnClickListener(this);
        userLocalStore = new UserLocalStore(this);
    }
    @Override
    public void onClick(View v) {
        switch(v.getId()){
            case R.id.bLogout:
                userLocalStore.getLoggedInUser();
                userLocalStore.setUserLoggedIn(false);
                Intent loginIntent = new Intent(this, Login.class);
                startActivity(loginIntent);
                break;
            case R.id.bOnas:
                Intent oNasIntent = new Intent(this, ONas.class);
                startActivity(oNasIntent);
                break;
            case R.id.bKontakty:
                Intent konIntent = new Intent(this, Kontakty.class);
                startActivity(konIntent);
                break;
            case R.id.bUslgi:
```


Ә қосымшасының жалғасы

```
        Intent usIntent = new Intent(this, Uslugi.class);
        startActivity(usIntent);
        break;
    }
}
@Override
protected void onStart() {
    super.onStart();
    if (authenticate() == true) {
        //displayUserDetails();
    }
}
private boolean authenticate() {
    if (userLocalStore.getLoggedInUser() == null) {
        Intent intent = new Intent(this, Login.class);
        startActivity(intent);
        return false;
    }
    return true;
}
}
package com.example.macbookpro.sosmedical;
import android.app.AlertDialog;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
public class Login extends ActionBarActivity implements View.OnClickListener {
    Button bLogin;
    TextView registerLink;
    EditText etUsername, etPassword;
    UserLocalStore userLocalStore;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        bLogin = (Button) findViewById(R.id.bLogin);
        etUsername = (EditText) findViewById(R.id.etUsername);
        etPassword = (EditText) findViewById(R.id.etPassword);
        registerLink = (TextView) findViewById(R.id.tvRegisterLink);
        bLogin.setOnClickListener(this);
        registerLink.setOnClickListener(this);
        userLocalStore = new UserLocalStore(this);
    }
    @Override
    public void onClick(View view) {
```

```
switch (view.getId()) {
    case R.id.bLogin:
        String username = etUsername.getText().toString();
        String password = etPassword.getText().toString();
        User user = new User(username, password);
        authenticate(user);
        break;
    case R.id.tvRegisterLink:
        Intent registerIntent = new Intent(Login.this, Register.class);
        startActivity(registerIntent);
        break;
}
}
private void authenticate(User user) {
    ServerRequest serverRequest = new ServerRequest(this);
    serverRequest.fetchUserDataAsyncTask(user, new GetUserCallback() {
        @Override
        public void done(User returnedUser) {
            if (returnedUser == null) {
                showErrorMessage();
            } else {
                logUserIn(returnedUser);
            }
        }
    });
}
private void showErrorMessage() {
    AlertDialog.Builder dialogBuilder = new AlertDialog.Builder(Login.this);
    dialogBuilder.setMessage("Incorrect user details");
    dialogBuilder.setPositiveButton("Ok", null);
    dialogBuilder.show();
}
private void logUserIn(User returnedUser) {
    userLocalStorage.storeUserData(returnedUser);
    userLocalStorage.setUserLoggedIn(true);
    startActivity(new Intent(this, MainActivity.class));
}
}
package com.example.macbookpro.sosmedical;
public class Lists {
    String d_time, doctor, user;
    public Lists(String d_time, String doctor, String user) {
        this.d_time = d_time;
        this.doctor = doctor;
        this.user = user;
    }
    public Lists(String d_time, String doctor) {
        this.d_time = d_time;
        this.doctor = doctor;
    }
}
```

Ә қосымшасының жалғасы

```
        this.user = "";
    }
}
package com.example.macbookpro.sosmedical;
import android.content.Context;
import android.content.SharedPreferences;

/**
 * Created by tundealao on 29/03/15.
 */
public class ListLocalStore {
    public static final String SP_NAME = "ListDetails";
    SharedPreferences listLocalDatabase;
    public ListLocalStore(Context context) {
        listLocalDatabase = context.getSharedPreferences(SP_NAME, 0);
    }
    public void storeListData(Lists lists) {
        SharedPreferences.Editor userLocalDatabaseEditor = listLocalDatabase.edit();
        userLocalDatabaseEditor.putString("d_time", lists.d_time);
        userLocalDatabaseEditor.putString("doctor", lists.doctor);
        userLocalDatabaseEditor.putString("user", lists.user);
        userLocalDatabaseEditor.commit();
    }
    public void setListLoggedIn(boolean loggedIn) {
        SharedPreferences.Editor userLocalDatabaseEditor = listLocalDatabase.edit();
        userLocalDatabaseEditor.putBoolean("loggedIn", loggedIn);
        userLocalDatabaseEditor.commit();
    }
    public void clearListData() {
        SharedPreferences.Editor userLocalDatabaseEditor = listLocalDatabase.edit();
        userLocalDatabaseEditor.clear();
        userLocalDatabaseEditor.commit();
    }
    public Lists getLoggedInlist() {
        if (listLocalDatabase.getBoolean("loggedIn", false) == false) {
            return null;
        }
        String d_time = listLocalDatabase.getString("d_ime", "");
        String doctor = listLocalDatabase.getString("doctor", "");
        String user = listLocalDatabase.getString("user", "");
        Lists list = new Lists(d_time, doctor, user);
        return list;
    }
}
package com.example.macbookpro.sosmedical;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
```

Ә қосымшасының жалғасы

```
import android.support.v7.app.ActionBarActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
public class Kontakty extends ActionBarActivity implements View.OnClickListener{
    Button bBack;
    TextView tvTelefon, tvCall;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kontakty);
        tvTelefon = (TextView) findViewById(R.id.tvTelefon);
        tvCall = (TextView) findViewById(R.id.tvCall);
        bBack = (Button) findViewById(R.id.bBack);
        tvCall.setOnClickListener(this);
        tvTelefon.setOnClickListener(this);
        bBack.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.bBack:
                Intent mainIntent = new Intent(this, MainActivity.class);
                startActivity(mainIntent);
                break;
            case R.id.tvTelefon:
                Intent callIntent = new Intent(Intent.ACTION_CALL);
                callIntent.setData(Uri.parse("tel:+" + tvTelefon.getText().toString().trim()));
                startActivity(callIntent);
                break;
            case R.id.tvCall:
                Intent caIntent = new Intent(Intent.ACTION_CALL);
                caIntent.setData(Uri.parse("tel:+" + tvCall.getText().toString().trim()));
                startActivity(caIntent);
                break;
        }
    }
}
package com.example.macbookpro.sosmedical;
import android.content.Intent;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
public class Kardiolog extends ActionBarActivity implements View.OnClickListener{
```

```
TextView tvDMR;
Button bBack;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_kardiolog);
    tvDMR = (TextView) findViewById(R.id.tvDMR);
    bBack = (Button) findViewById(R.id.bBack);
tvDMR.setOnClickListener(this);
    bBack.setOnClickListener(this);
}
@Override
public void onClick(View v) {
    Intent itdIntent = new Intent(this, Zapis.class);
    String s = "";
    switch (v.getId()){
        case R.id.tvDMR:
            s = "Джанайдар Мадина Рифовна";
            itdIntent.putExtra("myExtra", s);
            startActivity(itdIntent);
            break;
        case R.id.bBack:
            Intent backIntent = new Intent(this, Uslugi.class);
            startActivity(backIntent);
            break;
    }
}
}
package com.example.macbookpro.sosmedical;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
public class Nevrolog extends ActionBarActivity implements View.OnClickListener{
    TextView tvANA, tvHJA;
    Button bBack;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_nevrolog);
        tvANA = (TextView) findViewById(R.id.tvANA);
        tvHJA = (TextView) findViewById(R.id.tvHJA);
        bBack = (Button) findViewById(R.id.bBack);
        tvANA.setOnClickListener(this);
        tvHJA.setOnClickListener(this);
        bBack.setOnClickListener(this);
    }
}
```

```
}
@Override
public void onClick(View v) {
    Intent itdIntent = new Intent(this, Zapis.class);
    String s = "";
    switch (v.getId()){
import android.support.v7.app.ActionBarActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class Register extends ActionBarActivity implements View.OnClickListener{
    EditText etName, etAge, etUsername, etPassword;
    Button bRegister, bCheckUser;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        etName = (EditText) findViewById(R.id.etName);
        etAge = (EditText) findViewById(R.id.etAge);
        etUsername = (EditText) findViewById(R.id.etUsername);
        etPassword = (EditText) findViewById(R.id.etPassword);
        bRegister = (Button) findViewById(R.id.bRegister);
        bCheckUser = (Button) findViewById(R.id.bCheckUser);
        bCheckUser.setOnClickListener(this);
        bRegister.setOnClickListener(this);
        bRegister.setEnabled(false);
    }
    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.bCheckUser:
                String namec = etName.getText().toString();
                String usernamec = etUsername.getText().toString();
                String passwordc = etPassword.getText().toString();
                int agec = Integer.parseInt(etAge.getText().toString());
                User userc = new User(namec, agec, usernamec, passwordc);
                checkuser(userc);
                break;
            case R.id.bRegister:
                String name = etName.getText().toString();
                String username = etUsername.getText().toString();
                String password = etPassword.getText().toString();
                if (password.length() == 0){
                    AlertDialog.Builder dialogBuilder = new AlertDialog.Builder(Register.this);
                    dialogBuilder.setMessage("Please enter the password");
                    dialogBuilder.setPositiveButton("Ok", null);
                    dialogBuilder.show();
                }
            }
        }
    }
}
```

Ә қосымшасының жалғасы

```
    }else {
        int age = Integer.parseInt(etAge.getText().toString());

        User user = new User(name, age, username, password);
        registerUser(user);
    }
    break;

        GetUserCallback userCallBack) {
    progressDialog.show();
    new StoreUserDataAsyncTask(user, userCallBack).execute();
}
public void storeListDataInBackground(Lists lists,
        GetListBack listBack) {
    progressDialog.show();
    new StoreListDataAsyncTask(lists, listBack).execute();
}

public void fetchListDataAsyncTask(Lists lists, GetListBack listBack) {
    progressDialog.show();
    new fetchListDataAsyncTask(lists, listBack).execute();
}

public void fetchCUserDataAsyncTask(User user, GetUserCallback userCallBack) {
    progressDialog.show();
    new fetchCUserDataAsyncTask(user, userCallBack).execute();
}
public void fetchUserDataAsyncTask(User user, GetUserCallback userCallBack) {
    progressDialog.show();
    new fetchUserDataAsyncTask(user, userCallBack).execute();
}

/**
 * parameter sent to task upon execution progress published during
 * background computation result of the background computation
 */
public class StoreUserDataAsyncTask extends AsyncTask<Void, Void, Void> {
    User user;
    GetUserCallback userCallBack;

    public StoreUserDataAsyncTask(User user, GetUserCallback userCallBack) {
        this.user = user;
        this.userCallBack = userCallBack;
    }
    @Override
    protected Void doInBackground(Void... params) {
        ArrayList<NameValuePair> dataToSend = new ArrayList<NameValuePair>();
        dataToSend.add(new BasicNameValuePair("name", user.name));
        dataToSend.add(new BasicNameValuePair("username", user.username));
    }
}
```

Ә қосымшасының жалғасы

```
dataToSend.add(new BasicNameValuePair("password", user.password));
dataToSend.add(new BasicNameValuePair("age", user.age + ""));
HttpParams httpRequestParams = getHttpRequestParams();
HttpClient client = new DefaultHttpClient(httpRequestParams);
HttpPost post = new HttpPost(SERVER_ADDRESS
    + "Register.php");
try {
    post.setEntity(new UrlEncodedFormEntity(dataToSend));
android:padding="10dp">
<TextView
    android:id="@+id/tvAJO"
    android:layout_width="wrap_content"
    android:text="Ахметжанова Жамиля Оспановна"
    android:textSize="18dp"
    android:layout_gravity="center_horizontal"
    android:textStyle="bold"
    android:layout_height="wrap_content" />
<TextView
    android:id="@+id/tvKGV"
    android:layout_width="wrap_content"
    android:text="Клановец Галина Валерьевна"
    android:textSize="18dp"
    android:layout_gravity="center_horizontal"
    android:textStyle="bold"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/bBack"
    android:text="Назад"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
</LinearLayout>
</ScrollView>
<ScrollView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/background"
    xmlns:android="http://schemas.android.com/apk/res/android">
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:padding="10dp">
    <Button
    android:id="@+id/bUzi"
    android:text="УЗИ"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```



```
<Button
  android:id="@+id/bRentg"
  android:text="Рентген"
  android:layout_width="match_parent"
  android:layout_height="wrap_content" />
<Button
  android:id="@+id/bKard"
  android:text="Кардиолог"
  android:layout_width="match_parent"
  android:layout_height="wrap_content" />
<Button
  android:id="@+id/bNevro"
  android:text="Невролог"
  android:layout_width="match_parent"
  android:layout_height="wrap_content" />
<Button
  android:id="@+id/bTera"
  android:text="Терапевт"
  android:layout_width="match_parent"
  android:layout_height="wrap_content" />
<Button
  android:id="@+id/bBack"
  android:text="В главное меню"
  android:layout_width="match_parent"
  android:layout_height="wrap_content" />
</LinearLayout>
</ScrollView>
<ScrollView
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:background="@drawable/background"
  xmlns:android="http://schemas.android.com/apk/res/android">
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:orientation="vertical"
  android:layout_height="match_parent"
  android:layout_width="match_parent"
  android:padding="10dp">
  <TextView
    android:layout_width="wrap_content"
    android:text="Здравствуйте"
    android:layout_height="wrap_content" />
  <TextView
    android:id="@+id/tvUser"
    android:layout_width="wrap_content"
    android:text=""
  android:layout_height="wrap_content" />
  <TextView
    android:layout_width="wrap_content"
```

Ә қосымшасының жалғасы

```
        android:text="Ваш врач:"
        android:layout_height="wrap_content" />
<TextView
    android:id="@+id/tvDoc"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<TextView
    android:layout_width="wrap_content"
    android:text="Введите дату и время:"
    android:layout_height="wrap_content" />
<TextView
    android:layout_width="wrap_content"
    android:text="Пример: 06/05/15 8:00"
    android:layout_height="wrap_content"
    android:textColor="#ffff0116"
    android:textSize="10dp" />
<EditText
    android:id="@+id/etDate"
    android:layout_width="match_parent"
    android:layout_marginBottom="10dp"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/bCheck"
    android:text="Проверить время"
    android:layout_width="match_parent"
    android:layout_marginBottom="10dp"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/bGo"
    android:text="Записаться"
    android:layout_width="match_parent"
    android:layout_marginBottom="10dp"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/bBack"
    android:text="Назад"
    android:layout_width="match_parent"
    android:layout_marginBottom="10dp"
    android:layout_height="wrap_content" />
</LinearLayout>
</ScrollView>
```