

МУҲАММАД АЛ-ХОРАЗМИЙ НОМИДАГИ
ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ
ҲУЗУРИДАГИ ПЕДАГОГ КАДРЛАРНИ ҚАЙТА ТАЙЁРЛАШ ВА
УЛАРНИНГ МАЛАКАСИНИ ОШИРИШ ТАРМОҚ МАРКАЗИ

2019



ЎҚУВ-УСЛУБИЙ
МАЖМУА

МОБИЛ ИЛОВАЛАР ЯРАТИШ

“Информатика ва ахборот технологиялари” йўналиши

**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ
ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ**

**ОЛИЙ ТАЪЛИМ ТИЗИМИ ПЕДАГОГ ВА РАЎБАР КАДРЛАРИНИ
ҚАЙТА ТАЙЁРЛАШ ВА УЛАРНИНГ МАЛАКАСИНИ ОШИРИШНИ
ТАШКИЛ ЭТИШ БОШ ИЛМИЙ - МЕТОДИК МАРКАЗИ**

**МУЎАММАД АЛ-ХОРАЗМИЙ НОМИДАГИ
ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ
ЎУЗУРИДАГИ ПЕДАГОГ КАДРЛАРНИ ҚАЙТА ТАЙЁРЛАШ ВА
УЛАРНИНГ МАЛАКАСИНИ ОШИРИШ ТАРМОҚ МАРКАЗИ**

«Информатика ва ахборот технологиялари» йўналиши

**“МОБИЛ ИЛОВАЛАР ЯРАТИШ”
МОДУЛИ БЎЙИЧА
Ў Қ У В – У С Л У Б И Й М А Ж М У А**

Тошкент - 2019

Модулнинг ўқув-услубий мажмуаси Олий ва ўрта махсус, касб-хунар таълими ўқув-методик бирлашмалари фаолиятини Мувофиқлаштирувчи кенгашининг 2019 йил 18 октябрдаги 5 – сонли баённомаси билан маъқулланган ўқув дастури ва ўқув режасига мувофиқ ишлаб чиқилган.

Тузувчилар: Қўчқоров Т.А. - ТАТУ, «Компьютер тизимлари» кафедраси доценти, PhD

Такризчилар: С.Медетов – Нант политехника университети (Франция), Электрон ва рақамли технологиялар кафедраси профессори
М.Якубов - ТАТУ “Ахборот технологиялари” кафедраси профессори, ф-м.ф.д.

Модулнинг ўқув-услубий мажмуаси Муҳаммад ал-Хоразмий номидаги Тошкент ахборот технологиялари университети Кенгашининг 2019 йил 29 августдаги 1 (694) – сонли баённомаси билан тавсия қилинган

МУНДАРИЖА

| | |
|--|-----|
| I. Ишчи дастур | 5 |
| II. Модулни ўқитишда фойдаланиладиган интерфаол таълим методлари . | 9 |
| III. Назарий материаллар | 19 |
| IV. Амалий машғулот материаллари..... | 101 |
| V. Кейслар банки..... | 155 |
| VI. Глоссарий | 162 |
| VII. Адабиётлар рўйхати..... | 168 |

І БЎЛІМ

ИШЧИ ДАСТУР

Кириш

Дастур Ўзбекистон Республикаси Президентининг 2015 йил 12 июндаги “Олий таълим муассасаларининг раҳбар ва педагог кадрларини қайта тайёрлаш ва малакасини ошириш тизимини янада такомиллаштириш чора-тадбирлари тўғрисида”ги ПФ-4732-сонли, 2017 йил 7 февралдаги “Ўзбекистон Республикасини янада ривожлантириш бўйича Ҳаракатлар стратегияси тўғрисида”ги ПФ-4947-сонли, 2019 йил 27 августдаги “Олий таълим муассасалари раҳбар ва педагог кадрларининг узлуксиз малакасини ошириш тизимини жорий этиш тўғрисида”ги ПФ-5789-сонли Фармонлари, шунингдек 2017 йил 20 апрелдаги “Олий таълим тизимини янада ривожлантириш чора-тадбирлари тўғрисида”ги ПҚ–2909-сонли қарорида белгиланган устувор вазифалар мазмунидан келиб чиққан ҳолда тузилган бўлиб, у олий таълим муассасалари педагог кадрларининг касб маҳорати ҳамда инновацион компетентлигини ривожлантириш, соҳага оид илғор хорижий тажрибалар, янги билим ва малакаларни ўзлаштириш, шунингдек амалиётга жорий этиш кўникмаларини такомиллаштиришни мақсад қилади.

Дастур доирасида берилаётган мавзулар таълим соҳаси бўйича педагог кадрларни қайта тайёрлаш ва малакасини ошириш мазмуни, сифати ва уларнинг тайёргарлигига қўйиладиган умумий малака талаблари ва ўқув режалари асосида шакллантирилган бўлиб, унинг мазмуни Ўзбекистоннинг миллий тикланишдан миллий юксалиш босқичида олий таълим вазифалари, таълим-тарбия жараёнларини ташкил этишнинг норматив-ҳуқуқий ҳужжатлари, илғор таълим технологиялари ва педагогик маҳорат, таълим жараёнларида ахборот-коммуникация технологияларини қўллаш, амалий хорижий тил, тизимли таҳлил ва қарор қабул қилиш асослари, махсус фанлар негизида илмий ва амалий тадқиқотлар, ўқув жараёнини ташкил этишнинг замонавий услублари бўйича сўнгги ютуқлар, педагогнинг креатив компетентлигини ривожлантириш, глобал Интернет тармоғи, мультимедиа тизимларидан фойдаланиш ва масофавий ўқитишнинг замонавий шакллари қўллаш бўйича тегишли билим, кўникма, малака ва компетенцияларни ривожлантиришга йўналтирилган.

Қайта тайёрлаш ва малака ошириш йўналишининг ўзига хос хусусиятлари ҳамда долзарб масалаларидан келиб чиққан ҳолда дастурда тингловчиларнинг махсус фанлар доирасидаги билим, кўникма, малака ҳамда компетенцияларига қўйиладиган талаблар такомиллаштирилиши мумкин.

Ушбу дастурда мобил қурилмалар учун операцион тизимлар, иловалар структураси, андроид тизими учун Java дастурлаш тили, андроид фойдаланувчи интерфейсини яратиш, иловаларда ходисалар ва жараёнлар, менюларни бошқариш, мобил иловаларда маълумотлар базаси билан ишлаш, GPS хизмати, тармоқли дастурлаш, илованинг сервер қисми билан ишлаш ва JSON хизматидан фойдаланиш муаммолари баён этилган.

Модулнинг мақсади ва вазифалари

“Мобил иловалар яратиш” модулининг мақсад ва вазифалари:

- мобил қурилмалар учун операцион тизимлар, иловалар структураси, андроид тизими учун Java дастурлаш тили ва объектга йўналтирилган дастурлаш тамойилларидан фойдаланиш, андроид фойдаланувчи интерфейсини яратиш, иловаларда ходисалар ва жараёнлар билан ишлаш, менюларни бошқариш, мобил иловаларда маълумотлар базаси билан ишлаш, GPS хизмати, илованинг сервер

қисми билан ишлаш ва JSON хизматидан фойдаланиш ва уларни амалиётга қўллаш малакавий кўникмаларини шакллантириш;

Модул бўйича тингловчиларнинг билими, кўникмаси, малакаси ва компетенцияларига қўйиладиган талаблар

“Мобил иловалар яратиш” курсини ўзлаштириш жараёнида амалга ошириладиган масалалар доирасида:

Тингловчи:

- мобил қурилмалар учун операцион тизимлар ва уларнинг таснифлари;
- андроид операцион тизими ва унинг архитектураси;
- Java дастурлаш тили ва объектга йўналтирилган дастурлаш тамойиллари ҳақида **билимларга** эга бўлиши лозим.

Тингловчи:

- Андроид тизимида фойдаланувчи интерфейсини ярата олиш;
- иловаларда ҳодисалар ва жараёнлар билан ишлаш
- иловаларда маълумотлар базаси ва контент провайдерлардан фойдаланиш;
- JSON хизматидан фойдаланиш **кўникма ва малакаларини** эгаллаши зарур.

Тингловчи:

- андроид платформасида фойдаланувчи интерфейсини яратиш;
- сифатли ва қулай интерфейсга эга илова яратиш **компетенцияларни** эгаллаши лозим.

Модулни ташкил этиш ва ўтказиш бўйича тавсиялар

“Мобил иловалар яратиш” курси маъруза ва амалий машғулотлар шаклида олиб борилади.

Курсни ўқитиш жараёнида таълимнинг замонавий методлари, ахборот-коммуникация технологиялари қўлланилиши назарда тутилган:

- маъруза дарсларида замонавий компьютер технологиялари ёрдамида презентацион ва электрон-дидактик технологиялардан;

- ўтказиладиган амалий машғулотларда техник воситалардан, экспресс-сўровлар, тест сўровлари, ақлий ҳужум, гуруҳли фикрлаш ва бошқа интерактив таълим усуллари қўллаш назарда тутилади.

Модулнинг ўқув режадаги бошқа модуллар билан боғлиқлиги ва узвийлиги

“Мобил иловалар яратиш” модули мазмуни ўқув режадаги “Веб дастурлаш” ва “Электрон таълим муҳитини ривожлантириш” ўқув модуллари билан узвий боғланган ҳолда педагогларнинг мобил иловалар яратиш бўйича касбий педагогик тайёргарлик даражасини оширишга хизмат қилади.

Модул бўйича соатлар тақсимоти

| № | Модуль мавзулари | Аудитория укув юкламаси | | | |
|----|---|-------------------------|-----------|----------------|-----------------|
| | | Жами | жумладан | | |
| | | | Назарий | Амаий машғулот | Кўчма машғулоти |
| 1. | Мобил қурилмалар учун операцион тизимлар. Андроид архитектураси ва иловалар структураси таснифи. | 2 | 2 | | |
| 2 | Андроид муҳитини ўрнатиш ва созлаш. Дастлабки андроид иловасини яратиш ва мослаш. | 2 | | 2 | |
| 3 | Объектга йўналтирилган дастурлаш тамойиллари. Андроид тизими учун Java дастурлаш тили. | 2 | 2 | | |
| 4 | Java дастурлаш тилида синф ва объектлардан фойдаланиш. ОЙД принциплари. Абстракт синфлардан фойдаланиш. Java дастурлаш тилида ворислик ва полиморфизмдан фойдаланиб дастурлаш. | 2 | | 2 | |
| 5 | Андроид фойдаланувчи интерфейсини яратиш. Иловаларда ҳодисалар ва жараёнлар. | 2 | 2 | | |
| 6 | Андроид тизимида Layout ва View лардан фойдаланиш. Жараёнлар ва ҳодислар билан ишлаш. Андроид тизимида фойдаланувчи интерфейсини яратиш расм ва менюларни жойлаштириш. Объект анимацияси. | 2 | | 2 | |
| 7 | Мобил иловаларда маълумотлар базаси ва контент провайдерлардан фойдаланиш. GPS хизмати. | 4 | 2 | | |
| 8 | Иловалар яратишда маълумотлар базасидан фойдаланиш. SQLite. Сўровлар яратиш. | 2 | | 2 | |
| 9 | Google maps хизматидан фойдаланиш. Фойдаланувчи жойлашган ўрнини аниқлаш. | 2 | | 2 | |
| 10 | Мобил иловалар яратишда тармоқли дастурлаш. Илованинг сервер қисми билан ишлаш. JSON хизматидан фойдаланиш. | 2 | 2 | | |
| 11 | Мобил иловалар яратишда тармоқли дастурлаш. Илованинг сервер қисми билан ишлаш. JSON хизматидан фойдаланиш. | 2 | | 2 | |
| | Жами: | 22 | 10 | 12 | |

НАЗАРИЙ МАШҒУЛОТЛАР МАЗМУНИ

1 - мавзу: Мобил қурилмалар учун операцион тизимлар. Андроид архитектураси ва иловалар структураси таснифи.

(2 соат)

Мобил қурилмаларда операцион тизим тушунчаси. Мобил қурилмаларнинг операцион тизимлар платформаси. Андроид операцион тизими ва унинг архитектураси. Андроид операцион тизими хизматлари. Андроид операцион тизими ривожланиши ва унинг бошқа операцион тизимлардан фарқи. Андроид тизимда иловалар яратиш учун керакли инструментал воситалар. Иловалар структураси. Иловаларнинг ташкил этувчилари ва уларнинг таснифи. AVD ни ўрнатиш.

2 - мавзу: Объектга йўналтирилган дастурлаш тамойиллари. Андроид тизими учун Java дастурлаш тили.

(2 соат)

Андроид тизими учун Java дастурлаш муҳити. Java дастурлаш тилининг асосий конструкциялари. Махсус синф ва функциялар. Объектга йўналтирилган дастурлаш асосий хусусиятлари (ворислик, инкапсуляция, полиморфизм). Абстракт синфлар.

3 - мавзу: Андроид фойдаланувчи интерфейсини яратиш. Иловаларда ходисалар ва жараёнлар. Андроид тизимида Layout ва View лардан фойдаланиш. Анимация.

(2 соат)

Андроид иловаларида ходисалар ва жараёнлар. Activity яшаш цикли, dialog ойналарда activity ларни ҳосил қилиш. Intent лар ва улардан фойдаланиш. Жараёнларни бир – бири билан боғлаш, Fragment лардан фойдаланиш. GUI компоненталари. Views ва ViewGroup лардан фойдаланиш. View лардан фойдаланиш. Рўйхатлар ва улардан фойдаланиш. Махсус fragment лардан фойдаланиш. Gallery, ImageSwitcher, GridView, ва ImageView компоненталаридан фойдаланиш. Расмларни жойлаштириш. Илова менюларини бошқариш. Қўшимча view лардан фойдаланиш. AnalogClock ва DigitalClock view. WebView компонентаси ва ундан фойдаланиш.

4 - мавзу: Мобил иловаларда маълумотлар базаси ва контент провайдерлардан фойдаланиш. GPS хизмати.

(2 соат)

SharedPreferences объекти орқали илова маълумотларини сақлаш. Файлларга маълумот ёзиш ва ўқиш (internal ва external сақловчилар. SQLite асослари. SQLite да маълумотлар базасини яратиш. Paketлараро маълумот алмашиш. Content provider лардан фойдаланиш. Content provider яратиш. Google map лойиҳа яратиш. Map API калитини ўрнатиш. Map ни иловада ҳосил қилиш. Марказларда ZOOM бошқаруви ва маркерлар қўшиш. Жойлашув ўрнини аниқлаш ва у ҳақида маълумот олиш.

**5 - мавзу: Мобил иловалар яратишда тармоқли дастурлаш. Илованинг сервер қисми билан ишлаш. JSON хизматидан фойдаланиш .
(2 соат)**

Мобил иловаларда тармоқли дастурлаш, тармоққа уланиш ва уни текшириш. HTTP орқали серверга уланиш. Қурилманинг IP манзилени бошқариш. Сервердаги маълумотларни юклаб олиш, JSON хизматидан фойдаланиш.

АМАЛИЙ МАШҒУЛОТЛАР МАЗМУНИ

1-амалий машғулот.

Андроид муҳитини ўрнатиш ва сошлаш. Дастлабки андроид иловасини яратиш ва мослаш (2 соат)

Мобил қурилмалар учун иловалар яратишда фойдаланадиган дастурий воситалар муҳитини ўрнатиш ва сошлаш. Иловаларни яратиш ва дастлабки мослаш. Android операцион тизими учун иловалар яратишда керакли бўлган инструментал дастурий воситаларни ўрнатиш ва дастлабки мобил иловаларни (Android тизими учун) яратиш

2-амалий машғулот.

Java дастурлаш тилида синф ва объектлардан фойдаланиш. ОИД принциплари. Абстракт синфлардан фойдаланиш. Java дастурлаш тилида ворислик ва полиморфизмдан фойдаланиб дастурлаш (2 соат)

Иловаларни яратиш учун дастурлаш тилларидан фойдаланиш кўникмаларига эга бўлиш. Java дастурлаш тили асосий конструкцияларидан фойдаланиш бўйича асосий кўникмаларни такрорлаш. Синфлар ва объектлардан фойдаланиш. Объектга йўналтирилган дастурлаш асосий принциплари (ворислик, инкапсуляция, полиморфизм, абстракт ва аноним синфлар ва бошқалар) дан фойдаланиш. Java тилида синфлараро ворислик ваобъектга йўналтирилган дастурлаш асосий принципларидан бири бўлган полиморфизмдан фойдаланиш. Супер синф ва ворис синф орасида муносабатлар. Методларни қайта ёзиш ва методларни қайта юклаш хусусиятларидан фойдаланиш.

3-амалий машғулот.

Андроид тизимида Layout ва View лардан фойдаланиш. Жараёнлар ва ходислар билан ишлаш. Андроид тизимида фойдаланувчи интерфейсини яратиш расм ва менюларни жойлаштириш. Объект анимацияси (2 соат)

Андроид иловаларини яратишда activity лардан фойдаланиш. Асосий Java (activity) файллар ва layout (xml) лар структураси ва уларни ўзгартириш кўникмаларига эга бўлиш, бир ва бир нечта activity ва layout ларни бир – бирига боғлаш. Мобил қурилмага мос фойдаланувчи интерфейсини яратиш. Android тизимида фойдаланувчи интерфейсини яратиш. Gallery, ImageView ва ImageSwitcher компоненталаридан фойдаланиб расмларни намоиш қиладиган илова яратиш. Объектлар учун анимациядан фойдаланиш ва уларни ҳаракатлантириш.

4 – амалий машғулот:

Иловалар яратишда маълумотлар базасидан фойдаланиш. SQLite. Сўровлар яратиш (2 соат)

Андроид тизимида фойдаланувчи интерфейсини яратишда маълумотлар базасидан фойдаланиш, лойихага мос маълумотлар базаси, жадваллар ва сўровлар

яратиш. Илова маълумотларини XML файлга сақлашда SharedPreferences объектидан фойдаланиш, файлларга маълумот ёзиш ва ўқиш, Sqlite маълумотлар базасини бошқариш тизимидан фойдаланиш

5 – амалий машғулот.

Google maps хизматидан фойдаланиш (2 соат)

Андроид тизимида Google map хизматидан фойдаланиш ва фойдаланувчи жойлашган ўрнини аниқлаш. Google map хизматидан фойдаланишда керакли синфлардан фойдаланиш. Илова учун Google map калитини генерация қилиш.

6 – амалий машғулот.

Мобил иловаларда тармоқли дастурлаш. Сервер билан ишлаш. JSON хизмати (2 соат)

Андроид тизимида фойдаланувчи интерфейсини яратиш ва иловага мос маълумотларни сервердан ўқиб олиш, JSON хизматидан фойдаланиш

ЎҚИТИШ ШАКЛЛАРИ

Мазкур модуль бўйича қуйидаги ўқитиш шаклларида фойдаланилади:

- маърузалар, амалий машғулотлар (маълумотлар ва технологияларни англаб олиш, ақлий қизиқишни ривожлантириш, назарий билимларни мустаҳкамлаш);
- давра суҳбатлари (қўрилаётган лойиҳа ечимлари бўйича таклиф бериш қобилиятини ошириш, эшитиш, идрок қилиш ва мантиқий хулосалар чиқариш);
- баҳс ва мунозаралар (лойиҳалар ечими бўйича далиллар ва асосли аргументларни тақдим қилиш, эшитиш ва муаммолар ечимини топиш қобилиятини ривожлантириш).

II БЎЛИМ

МОДУЛНИ ЎҚИТИШДА
ФОЙДАЛАНИЛАДИГАН
ИНТЕРФАОЛ ТАЪЛИМ
МЕТОДЛАРИ

II. МОДУЛНИ ЎҚИТИШДА ФЙДАЛАНИЛАДИГАН ИНТЕРФАОЛ ТАЪЛИМ МЕТОДЛАРИ

«Блум кубиги» методи

Методнинг мақсади: Мазкур метод тингловчиларда янги ахборотлар тизимини қабул қилиш ва билимларни ўзлаштирилишини енгиллаштириш мақсадида қўлланилади, шунингдек, бу метод тингловчилар учун “Очиқ” саволлар тузиш ва уларга жавоб топиш машқи вазифасини белгилайди.

Методни амалга ошириш тартиби:

1. Ушбу методни қўллаш учун, оддий куб керак бўлади. Кубнинг ҳар бир томонида кўйидаги сўзлар ёзилади:
 - **Санаб беринг, таъриф беринг (оддий савол)**
 - **Нима учун (сабаб-оқибатни аниқлаштирировчи савол)**
 - **Тушинтириб беринг (муаммони ҳар томонлама қараш саволи)**
 - **Таклиф беринг (амалиёт билан боғлиқ савол)**
 - **Мисол келтиринг (ижодкорликни ривожлантирировчи савол)**
 - **Фикр беринг (таҳлил қилиш ва баҳолаш саволи)**
2. Ўқитувчи мавзуни белгилаб беради.
3. Ўқитувчи кубикни столга ташайди. Қайси сўз чиқса, унга тегишли саволни беради.

“KWLH” методи

Методнинг мақсади: Мазкур метод тингловчиларда янги ахборотлар тизимини қабул қилиш ва билимларни тизимлаштириш мақсадида қўлланилади, шунингдек, бу метод тингловчилар учун мавзу бўйича кўйидаги жадвалда берилган саволларга жавоб топиш машқи вазифасини белгилайди.

Изоҳ. KWLH:

Know – нималарни биламан?

Want – нимани билишни хоҳлайман?

How - қандай билиб олсам бўлади?

Learn - нимани ўрганиб олдим?.

| “KWL” методи | |
|--|--|
| 1. Нималарни биламан: - | 2. Нималарни билишни хоҳлайман, нималарни билишим керак: - |
| 3. Қандай қилиб билиб ва топиб оламан: - | 4. Нималарни билиб олдим: - |

“W1H” методи

Методнинг мақсади: Мазкур метод тингловчиларда янги ахборотлар тизимини қабул қилиш ва билимларни тизимлаштириш мақсадида қўлланилади, шунингдек, бу метод тингловчилар учун мавзу бўйича қўйидаги жадвалда берилган олти саволларга жавоб топиш машқи вазифасини белгилайди.

| | | |
|------------|---|--|
| What? | Нима? (таърифи, мазмуни, нима учун ишлатилади) | |
| Where? | Қаерда (жойлашган, қаердан олиш мукин)? | |
| What kind? | Қандай? (параметрлари, турлари мавжуд) | |
| When? | Қачон? (ишлатилади) | |
| Why? | Нима учун? (ишлатилади) | |
| How? | Қандай қилиб? (яратилади, сақланади, тўлдирилади, таҳрирлаш мумкин) | |

“SWOT-таҳлил” методи.

Методнинг мақсади: мавжуд назарий билимлар ва амалий тажрибаларни таҳлил қилиш, таққослаш орқали муаммони ҳал этиш йўллари топишга, билимларни мустаҳкамлаш, такрорлаш, баҳолашга, мустақил, танқидий фикрлашни, ностандарт тафаккурни шакллантиришга хизмат қилади.

| | |
|--------------------------|--------------------------|
| S – (strength) | • кучли томонлари |
| W – (weakness) | • заиф, кучсиз томонлари |
| O – (opportunity) | • имкониятлари |
| T – (threat) | • хавфлар |

“БЕЕР” методи

Методнинг мақсади: Бу метод мураккаб, кўптармоқли, мумкин қадар, муаммоли характеридаги мавзуларни ўрганишга қаратилган. Методнинг моҳияти шундан иборатки, бунда мавзунинг турли тармоқлари бўйича бир хил ахборот берилади ва айти пайтда, уларнинг ҳар бири алоҳида аспектларда муҳокама этилади. Масалан, муаммо ижобий ва салбий томонлари, афзаллик, фазилат ва камчиликлари, фойда ва зарарлари бўйича ўрганилади. Бу интерфаол метод танқидий, таҳлилий, аниқ мантикий фикрлашни муваффақиятли ривожлантиришга ҳамда ўқувчиларнинг мустақил ғоялари, фикрларини ёзма ва оғзаки шаклда тизимли баён этиш, ҳимоя қилишга имконият яратади. “Беер” методидан маъруза машғулотларида индивидуал ва жуфтликлардаги иш шаклида, амалий ва семинар машғулотларида кичик гуруҳлардаги иш шаклида мавзу юзасидан билимларни мустаҳкамлаш, таҳлили қилиш ва таққослаш мақсадида фойдаланиш мумкин.

Методни амалга ошириш тартиби:



тренер-ўқитувчи иштирокчиларни 5-6 кишидан иборат кичик гуруҳларга ажратади;



тренинг мақсади, шартлари ва тартиби билан иштирокчиларни таништиргач, ҳар бир гуруҳга умумий муаммони таҳлил қилиниши зарур бўлган қисмлари туширилган тарқатма



ҳар бир гуруҳ ўзига берилган муаммони атрофлича таҳлил қилиб, ўз мулоҳазаларини тавсия этилаётган схема бўйича тарқатмага ёзма баён қилади;



навбатдаги босқичда барча гуруҳлар ўз тақдимотларини ўтказадилар. Шундан сўнг, тренер томонидан таҳлиллар умумлаштирилади, зарурий ахборотлар билан тўлдирилади ва мавзу яқунланади.

| Муаммоли савол | | | | | |
|----------------|-----------|-----------|-----------|-----------|-----------|
| 1-усул | | 2-усул | | 3-усул | |
| афзаллиги | камчилиги | афзаллиги | камчилиги | афзаллиги | камчилиги |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| Хулоса: | | | | | |

“Кейс-стади” методи

«Кейс-стади» - инглизча сўз бўлиб, («case» – аниқ вазият, ҳодиса, «stadi» – ўрганмоқ, таҳлил қилмоқ) аниқ вазиятларни ўрганиш, таҳлил қилиш асосида ўқитишни амалга оширишга қаратилган метод ҳисобланади. Мазкур метод дастлаб 1921 йил Гарвард университетида амалий вазиятлардан иқтисодий бошқарув фанларини ўрганишда фойдаланиш тартибида қўлланилган. Кейсда очик ахборотлардан ёки аниқ воқеа-ҳодисадан вазият сифатида таҳлил учун фойдаланиш мумкин.

“Кейс методи” ни амалга ошириш босқичлари

| Иш босқичлари | Фаолият шакли ва мазмуни |
|---|--|
| 1-босқич: Кейс ва унинг ахборот таъминоти билан таништириш | <ul style="list-style-type: none"> ✓ якка тартибдаги аудио-визуал иш; ✓ кейс билан танишиш(матнли, аудио ёки медиа шаклда); ✓ ахборотни умумлаштириш; ✓ ахборот таҳлили; ✓ муаммоларни аниқлаш |
| 2-босқич: Кейсни аниқлаштириш ва ўқув топшириғни белгилаш | <ul style="list-style-type: none"> ✓ индивидуал ва гуруҳда ишлаш; ✓ муаммоларни долзарблик иерархиясини аниқлаш; ✓ асосий муаммоли вазиятни белгилаш |
| 3-босқич: Кейсдаги асосий муаммони таҳлил этиш орқали ўқув топшириғининг ечимини излаш, ҳал этиш йўллари ишлаб чиқиш | <ul style="list-style-type: none"> ✓ индивидуал ва гуруҳда ишлаш; ✓ муқобил ечим йўллари ишлаб чиқиш; ✓ ҳар бир ечимнинг имкониятлари ва тўсиқларни таҳлил қилиш; ✓ муқобил ечимларни танлаш |
| 4-босқич: Кейс ечимини ечимини шакллантириш ва асослаш, тақдимот. | <ul style="list-style-type: none"> ✓ якка ва гуруҳда ишлаш; ✓ муқобил вариантларни амалда қўллаш имкониятларини асослаш; ✓ ижодий-лойиҳа тақдимотини тайёрлаш; ✓ якуний хулоса ва вазият ечимининг амалий аспектиларини ёритиш |

“Ассесмент” методи

Методнинг мақсади: мазкур метод таълим олувчиларнинг билим даражасини баҳолаш, назорат қилиш, ўзлаштириш кўрсаткичи ва амалий кўникмаларини текширишга йўналтирилган. Мазкур техника орқали таълим олувчиларнинг билиш фаолияти турли йўналишлар (тест, амалий кўникмалар, муаммоли вазиятлар машқи, қиёсий таҳлил, симптомларни аниқлаш) бўйича ташҳис қилинади ва баҳоланади.

Методни амалга ошириш тартиби:

“Ассесмент”лардан маъруза машғулотларида талабаларнинг ёки катнашчиларнинг мавжуд билим даражасини ўрганишда, янги маълумотларни баён қилишда, семинар, амалий машғулотларда эса мавзу ёки маълумотларни ўзлаштириш даражасини баҳолаш, шунингдек, ўз-ўзини баҳолаш мақсадида индивидуал шаклда фойдаланиш тавсия этилади. Шунингдек, ўқитувчининг ижодий ёндашуви ҳамда ўқув мақсадларидан келиб чиқиб, ассесментга кўшимча топшириқларни киритиш мумкин.

Ҳар бир катакдаги тўғри жавоб 5 балл ёки 1-5 балгача баҳоланиши

мумкин.



Тест

Муаммоли вазият

**Тушунча таҳлили
(симптом)**

Амалий вазифа

“Инсерт” методи

Методни амалга ошириш тартиби:

- ўқитувчи машғулотга қадар мавзунинг асосий тушунчалари мазмуни ёритилган матнни тарқатма ёки тақдимот кўринишида тайёрлайди;
- янги мавзу моҳиятини ёритувчи матн таълим олувчиларга тарқатилади ёки тақдимот кўринишида намойиш этилади;
- таълим олувчилар индивидуал тарзда матн билан танишиб чиқиб, ўз шахсий қарашларини махсус белгилар орқали ифодалайдилар. Матн билан ишлашда талабалар ёки қатнашчиларга қуйидаги махсус белгилардан фойдаланиш тавсия этилади:

| Белгилар | Матн |
|---|------|
| “V” – таниш маълумот. | |
| “?” – мазкур маълумотни тушунмадим, изоҳ керак. | |
| “+” бу маълумот мен учун янгилик. | |
| “– ” бу фикр ёки мазкур маълумотга қаршиман? | |

Белгиланган вақт якунлангач, таълим олувчилар учун нотаниш ва тушунарсиз бўлган маълумотлар ўқитувчи томонидан таҳлил қилиниб, изоҳланади, уларнинг моҳияти тўлиқ ёритилади. Саволларга жавоб берилади ва машғулот якунланади.

Ш БЎЛИМ

НАЗАРИЙ
МАТЕРИАЛЛАР

III. НАЗАРИЙ МАТЕРИАЛЛАР

1 - мавзу: Мобил қурилмалар учун операцион тизимлар. Андроид архитектураси ва иловалар структураси таснифи. (2 соат)

Режа:

1. Мобил қурилмаларда операцион тизим тушунчаси ва платформалари.
2. Андроид операцион тизими ва унинг архитектураси.
3. Андроид операцион тизими ривожланиши ва унинг бошқа операцион тизимлардан фарқи.
4. Андроид тизими учун керакли инструментал воситалар ва уларни ўрнатиш. Виртуал қурилмадан фойдаланиш.

Таянч иборалар: *Android, Cellular, infrared blaster, GPS (Global Positioning Satellite) mobile navigation, PDA (Personal digital assistant), JDK (Java Development Kit), SDK (Software Development Kit), AVD (Android Virtual Device), ADT (Android Development Tools) plugin*

1.1. Мобил қурилмаларда операцион тизим тушунчаси

Мобил операцион тизимлар – бу мобил қурилмалар (смартфонлар, таблетлар, PDA ва бошқа мобил қурилмалар) ни бошқариш учун мўлжалланган тизимлар ҳисобланади. Мобил операцион тизимлар иккита асосий платформа яъни “user-facing” дастурий таъминот платформаси ва қурилмаларни бошқариш учун мўлжалланган қуйи сатҳдаги реал-вақт операцион тизидан иборат бўлади. Мобил операцион тизимлар асосан 1979 йилда пайдо бўлган ва бунда жараёнларни бошқариш учун (таксимланган тизим) http://en.wikipedia.org/wiki/Embedded_system тизимлардан фойдаланилган. Қуйидаги жадвалда (1-жадвал) мобил операцион тизимларнинг ривожланиш босқичларини кўришимиз мумкин.

1-жадвал. Мобил ОТ ларнинг ривожланиш босқичлари

| Яратилган йили | Қурилма ва ОТ номи | Таснифи |
|----------------|--|--|
| 1979–1992 | Мобил қурилмалар | Жараёнларни бошқариш учун embedded systems тизимлардан фойдаланилган |
| 1993 | Биринчи смартфонлар | Touchscreen, e-mail ва PDA хусусиятлари мавжуд |
| 1996-2002 | Palm OS, Windows CE, Nokia S40 | Touchscreen, e-mail, интернет хизматлари ва PDA хусусиятлари мавжуд |
| 2002 | BlackBerry смартфонлари | Хавфсизлик протоколлари, |
| 2005 | Нокиа, биринчи интернет таблет, Maemo OS | e-mail, интернет хизматлари |
| 2007 | Apple iPhone, iOS | "mobile phone" ва "internet communicator |

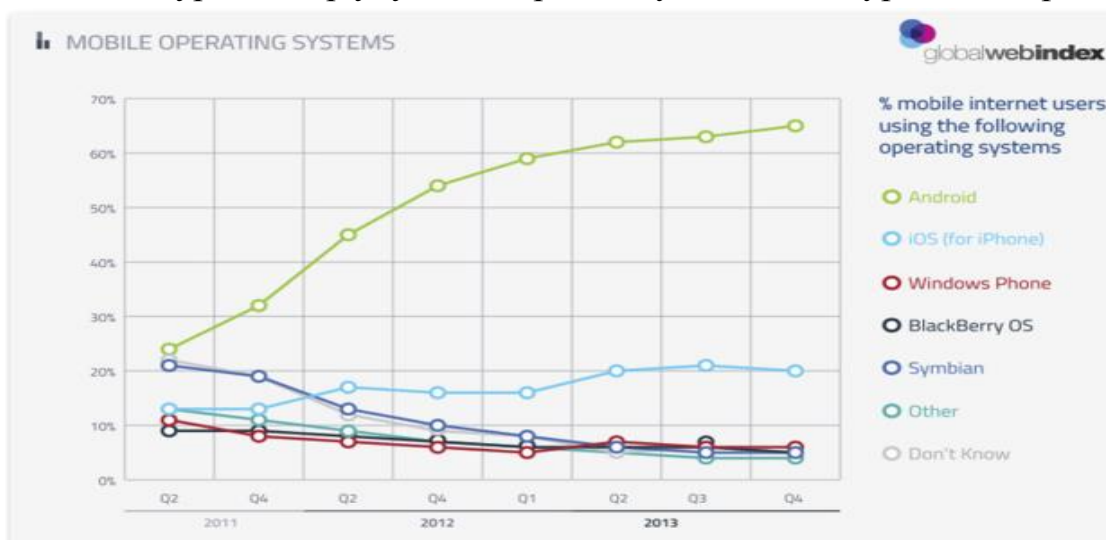
| Яратилган йили | Қурилма ва ОТ номи | Таснифи |
|----------------|----------------------------------|--|
| 2007-2008 | Android, ОНА | Биринчи андроид смартфонлари |
| 2010 | Windows Phone OS | Биринчи windows смартфонлари |
| 2011 | MeeGo биринчи Linux мобил тизими | Nokia , Intel ва Linux Foundation асосида Nokia N9 да синалган |

Ҳар бир операцион тизим платформаси учун дастурлаш тиллари ва дастурий восита ишлаб чиқиш муҳитлари мавжуд бўлиб, мобил қурилмаларга дастурий воситалар яратиш учун платформалар қуйидаги 2-жадвалда келтирилган.

2-жадвал. Платформалар учун дастурлаш тиллари

| Платформа номи | Дастурлаш тили |
|--------------------|-------------------------------------|
| Android | Java, қисман C/C++ |
| BlackBerry | Java |
| iOS SDK | Objective – C, Objective Pascal |
| Java ME | Java |
| Windows Phone (WP) | Visual C++, Visual C#, Visual Basic |
| Marmalade | C/C++ |
| Python | Python |
| Firefox OS | HTML5, CSS, JavaScript |

Мобил қурилмалар учун ОТ ларнинг қўлланилиш кўрсаткичлари



1.1-расм. Мобил қурилмалар учун ОТ ларнинг қўлланилиш кўрсаткичлари

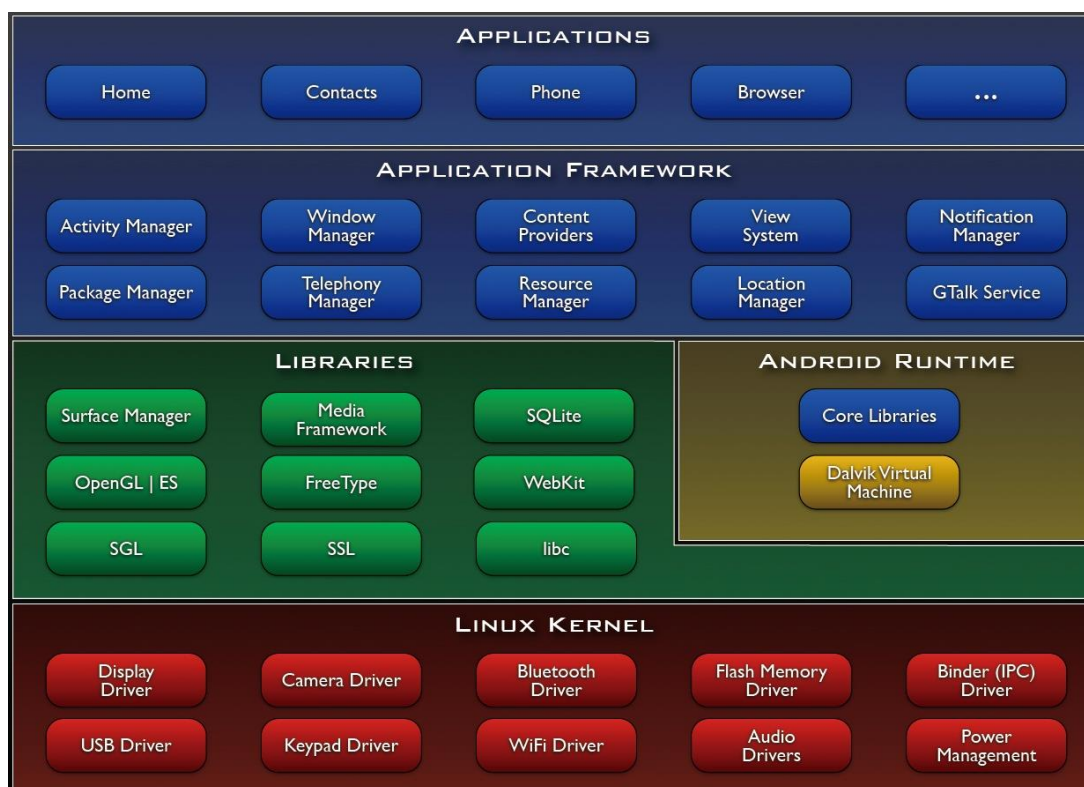
Замонавий мобил қурилмалар учун операцион тизимларнинг **асосий функциялари**

- Touchscreen

- Cellular
- Bluetooth
- Wi-Fi
- GPS mobile navigation
- camera, video camera
- speech recognition, voice recorder
- music player
- near field communication (NFC)
- infrared blaster (IR blaster).

1.2. Андроид операцион тизими ва унинг архитектураси

Android – бу мобил қурилмалар учун мўлжалланган турли хил иловалар ва хизматларни ўз ичига олган операцион тизим. **Android** архитектураси Java асосида қурилган бўлиб, андроид тизим учун барча илова ва дастурлар Java дастурлаш тили ёрдамида яратилади¹.



1.2-расм. Андроид ОТ архитектураси

Linux kernel – бу Android тизимининг ядроси ҳисобланади ва **115 patch** дан иборат. Бу асосан тизим функционалини яъни, процесслар бошқаруви, хотирани бошқариш, қурилмаларни (дисплей, камера, клавиатура ва ҳ.к. лар)

¹ “Beginning Android 4 Application Development”, chapter-1, Getting Started with Android Programming. Architecture of Android, pages 32-33.

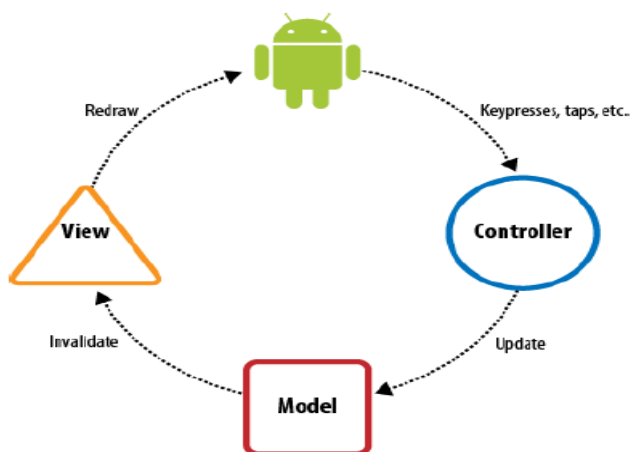
бошқариш учун хизмат қилади. Бундан ташқари тармоқ компонентларини, курилмалар драйверларини бошқаради.

Libraries – Иловаларни бошқариш учун зарур бўлган кутубхоналардан иборат. Бунга **Web browser engine WebKit**, маълумотларни сақлаш ва иловалараро маълумотларни алмашиш учун **SQLite database** кутубхонаси, **audio ва video** маълумотларни бошқариш учун **Media Framework**, интернет хавфсизлиги учун **SSL** кутубхонаси, графикани бошқариш учун **OpenGL**, touchscreen хизмати учун **Surface Manager** ва ҳ.к. лар. **Android Runtime** – **Dalvik Virtual Machine** компонентини бошқариш учун ишлатиладиган кутубхона ҳисобланади. Бунда ҳар бир илованинг ўз процессида ишлаши таъминланади. Ўз навбатида бу кутубхоналар иловалар яратишда Java дастурлаш тилини ишлатиш учун ҳам ёрдамберади.

Application Framework – юқори сатҳдаги Java синф ва пакетлар жамланмасидир. Бунда тайёр хизматлардан фойдаланиш мумкин (**Broadcast Receivers, Content Providers, Camera Control** ва ҳ.к. лар)

Applications – бу фойдаланувчи учун мўлжалланган иловалар сатҳи ҳисобланади (**Contacts Books, Browser, Games**)¹.

Андроид иловаларида фойдаланувчи интерфейсини яратишда MVC (**Model View Controller**) технологиясидан фойдаланилади. Бунда модел илова учун керакли маълумотларни маълумотлар базаси ёки бошқа манбадан олиб беради ва контроллер орқали илова маълумотлари фойдаланувчи интерфейсига юборилади.



1.3-расм. Android GUI архитектураси

¹ “Beginning Android 4 Application Development”, chapter-1, Getting Started with Android Programming. Architecture of Android, pages 32-33.

1.3. Андроид операцион тизими ривожланиши ва унинг бошқа операцион тизимлардан фарқи

3-жадвал. Android ва iOS OT лар фарқи

| Хусусияти | Android | iOS |
|-----------------------|--|--|
| Company/Developer | Google | Apple Inc. |
| OS family | Linux | OS X, UNIX |
| Initial release | September 23, 2008 | July 29, 2007 |
| Programmed in | C, C++, java | C, C++, Objective-C |
| Source model | Open source | Closed, with open source components. |
| Internet browsing | Google Chrome (or Android Browser on older versions; other browsers are available) | Mobile Safari (Other browsers are available) |
| Interface | Touch screen, Smartwatch | Touch screen |
| Messaging | Google Hangouts | iMessage |
| Maps | Google Maps | Apple Maps |
| App store | Google Play – 1,000,000+ Apps. | Apple app store – 1,000,000+ Apps |
| Market share | 81% of smartphones | 12.9% of smartphones |
| Available language(s) | 32 Languages | 34 Languages |
| Latest stable release | Android 6.0 Marshmallow (October 5, 2015) | 9.3.1 (March 31, 2016) |

Ҳозирги кунда Android қурилмаларнинг жаҳон бозоридаги турлари

- Smartphones
- Tablets
- E-reader devices
- Netbooks
- MP4 players
- Internet TVs

1.4. Андроид тизими учун керакли инструментал воситалар ва уларни ўрнатиш. Виртуал қурилмадан фойдаланиш

Андроид тизимда иловалар яратиш учун керакли инструментал воситалар қуйидагилардир:

- Java JDK5 ёки JDK6
- Android SDK

- Eclipse IDE for Java Developers (optional)
- Android Development Tools (ADT) Eclipse Plugin (optional)
- Android Virtual Device (AVD)
- Android Developers Tools (ADT) (Android SDK ва ADT Plugin билан биргаликда)

Андроид тизимда иловалар яратиш учун керакли инструментал воситаларни ўрнатиш куйидаги қадамлар орқали амалга оширилади¹:

1-қадам: Java development kit (JDK) ни ўрнатиш (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)

2- қадам: Android SDK (Software Development Kit) ни ўрнатиш (<http://developer.android.com/sdk/index.html>)

3- қадам: Eclipse IDE ни ўрнатиш (<http://www.eclipse.org/downloads/>)

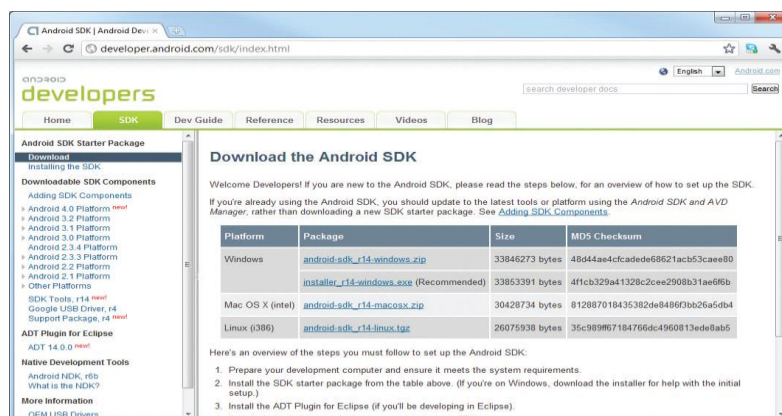
4- қадам: Android development tools (ADT) eclipse учун plugin ни ўрнатиш (<https://dl-ssl.google.com/android/eclipse/>)

5- қадам: Android virtual машинаси (AVD) ни ўрнатиш

Android SDK ўз ичига куйидагиларни олади

- Debugger
- Libraries
- an emulator
- Documentation
- sample code
- tutorials.

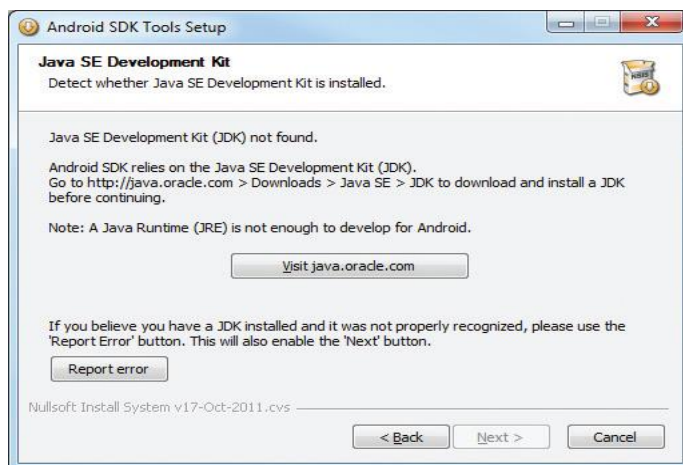
<http://developer.android.com/sdk/index.html> дан Android SDK юклаб олинади ва дастур ўрнатилгандан кейин *Android SDK Manager* ишга туширилади. Бунинг учун куйидаги кетма-кетликни амалга ошириш лозим: **All Programs > Android SDK Tools > SDK Manager**



1.4-расм. Android SDKни юклаб олиш ойнаси

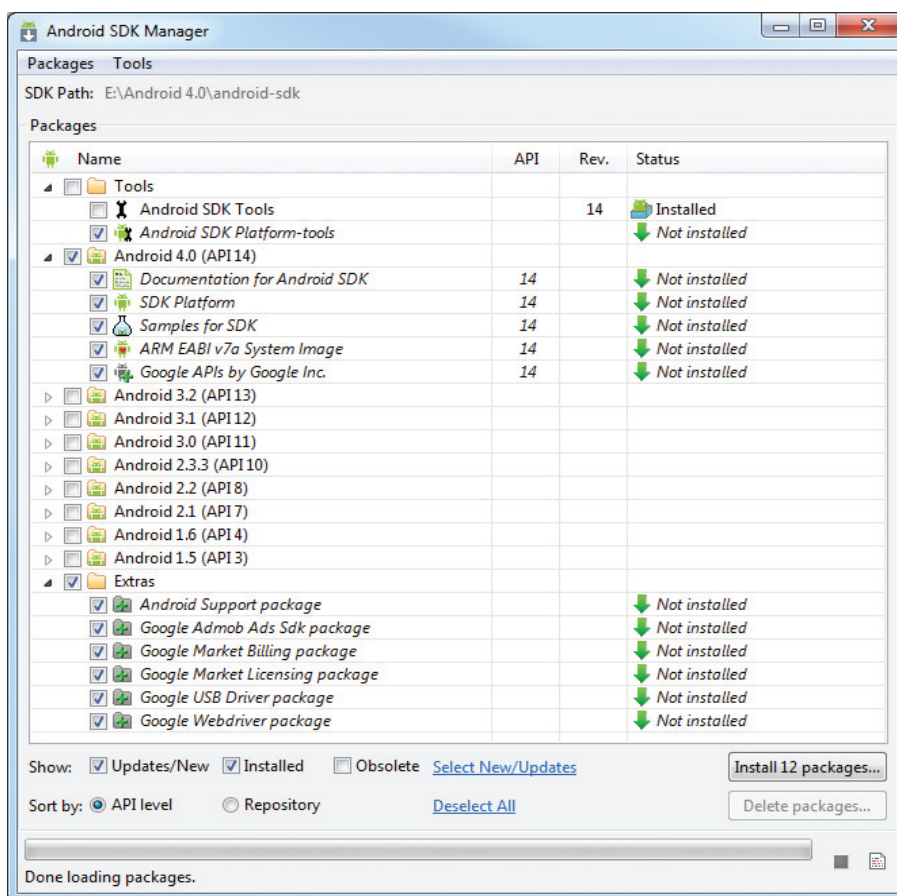
¹ “Beginning Android 4 Application Development”, chapter-1, Getting Started with Android Programming. Obtaining the required tools, pages 38-47.

Агар жорий компьютерда Java (JDK) ўрнатилмаган бўлса, у ҳолда Android SDK Manager ишга туширилганда қуйидагича хатолик чиқади. (1.5-расм)

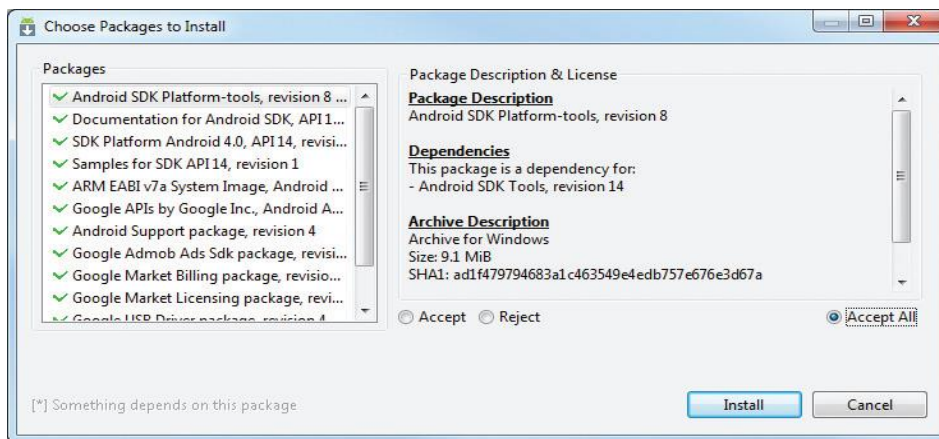


1.5-расм. Android SDK Manager ишга тушуришдаги хатолик

Агар компьютерда Java (JDK) ўрнатилган бўлса, у ҳолда Android SDK Manager мувоффақиятли ишга тушади ва қуйидагича Android SDK Manager ни созлаш ойнаси пайдо бўлади (1.6-расм).



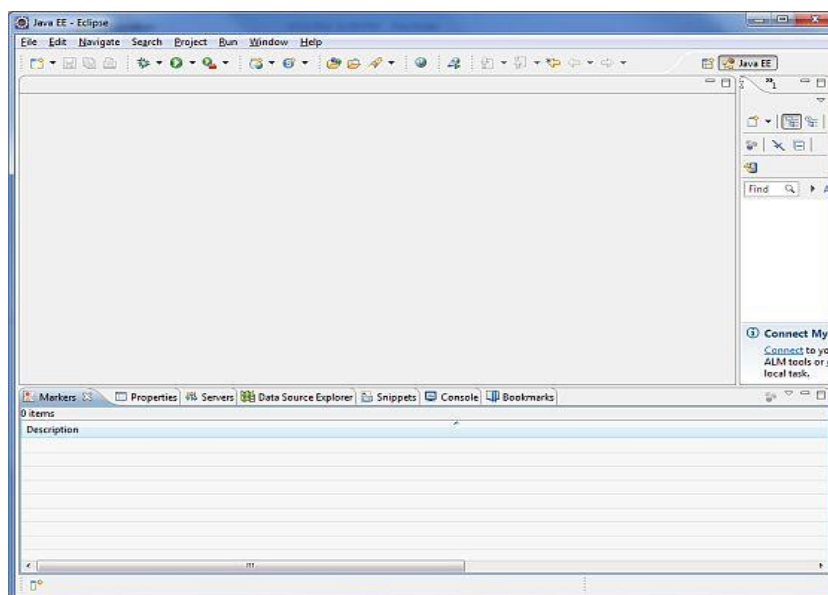
1.6-расм. Android SDK Manager ни созлаш ойнаси



Ҳар бир API level учун асосан иккита платформа мавжуд. *SDK Platform, Google APIs by Google Inc.* (*Google maps* учун мўлжалланган иловалар яратишда фойдаланилади)

Иловани яратиш учун керакли кутубхоналар ўрнатилгандан кейин илова кодини ёзиш ва уни ишга тушириб кўриш учун инструментал дастурий воситалар керак бўлади. Ушбу дастурдан биттаси Eclipse бўлиб, андроид иловалар яратишда қулай интерфейси билан ажралиб туради. Бундан ташқари Netbeans, IntelliJ IDEA, Android Studio каби дастурлар мавжуд бўлиб, булар ҳам иловалар ишлаб чиқишда самарали фойдаланилади. Берилган ссилка (<http://www.eclipse.org/downloads/>) орқали биз Eclipse дастурининг керакли версиясини юклаб олишимиз мумкин. Юклаб олинган дастурни архивдан чиқариб олиб ишга туширамиз.

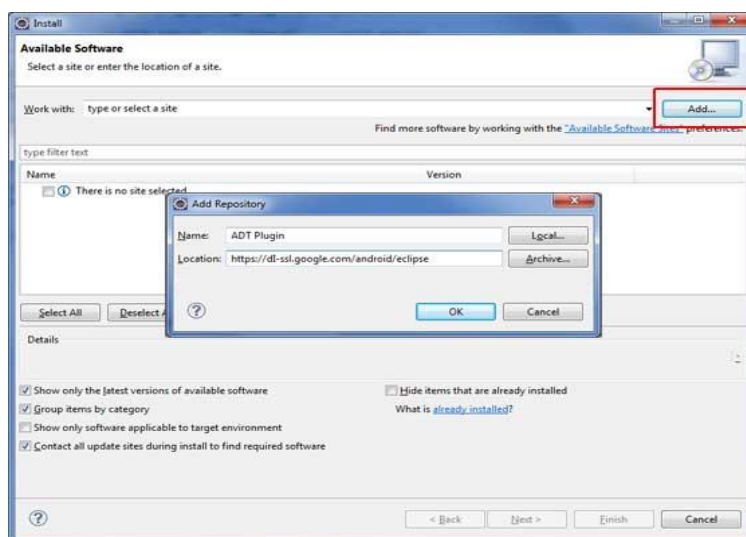
Мисол учун (C:\eclipse\eclipse.exe):



1.7-расм. Eclipse дастури дастлабки ишчи ойнаси

Android Development Tools (ADT) Plugin ни ўрнатиш

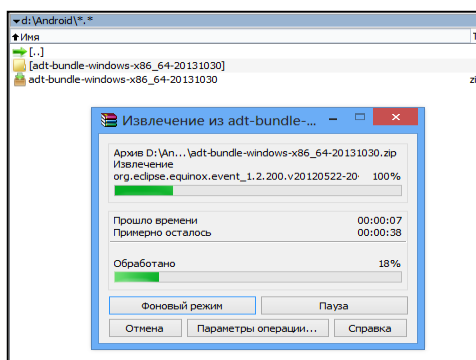
- Eclipse учун Android Development Tool plugin ни созлаш куйидагича
- Eclipse ишга туширилади
- **Help > Software Updates > Install New Software.**
- *Name: ADT Plugin*
- *Location: <https://dl-ssl.google.com/android/eclipse/>*
- *Press OK ->Next->Finish*



1.8-расм. ADT Plugin ни ўрнатиш

Android дастлабки илова яратиш. “hello world” мисолида. Android тизимида илова яратиш жараёнида тўғридан – тўғри ADT дан фойдаланиш қулайлик яратади. Яъни ушбу восита android муҳитида илова яратиш учун керакли бўлган барча инструментал дастурий воситаларни ўз ичига олади. Албатта AVD (Android virtual device) алоҳида ўрнатилади.

- “<http://developer.android.com/sdk/index.html?hl=sk#download>” link орқали **ADT bundle** ни юклаб оламиз.
- Юклаб олинган архив файлини архивдан чиқарамиз

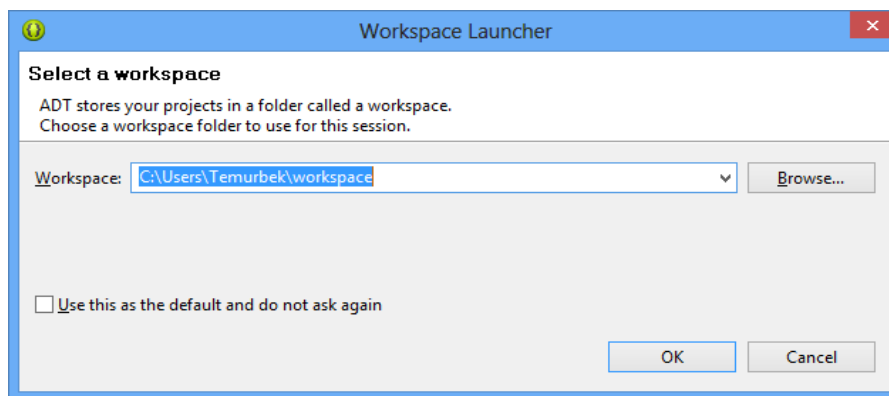


adt-bundle-windows-x86_64 папкаси асосан қуйидагилардан иборат

- *eclipse*
- *sdk*

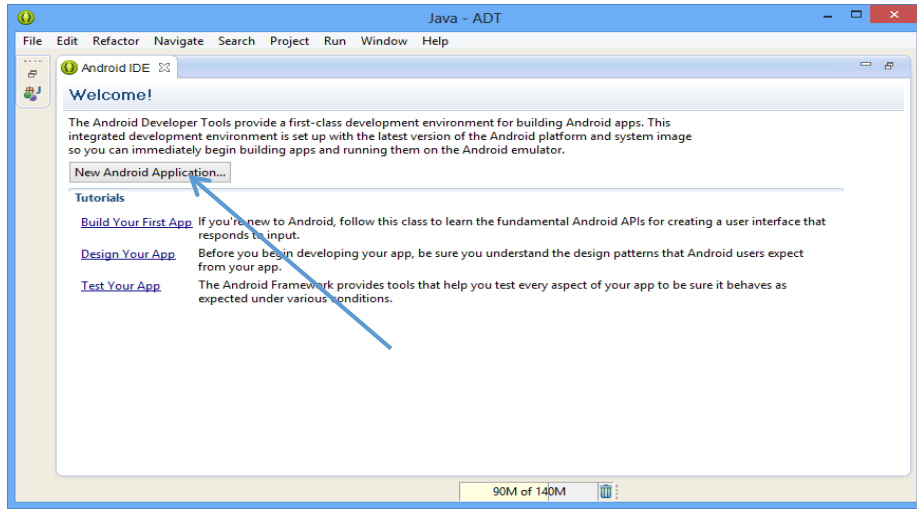
Асосий дастур ишчи ойнасини ишга тушириш учун *eclipse* папкасидан *eclipse.exe* файлини ишга туширамыз.

| Имя | Тип | Размер | Дата |
|-----------------|------|---------|------------------|
| [..] | | <Папка> | 22.04.2014 21:52 |
| [configuration] | | <Папка> | 22.04.2014 21:52 |
| [dropins] | | <Папка> | 24.10.2013 13:22 |
| [features] | | <Папка> | 24.10.2013 13:22 |
| [p2] | | <Папка> | 24.10.2013 13:21 |
| [plugins] | | <Папка> | 24.10.2013 13:22 |
| [readme] | | <Папка> | 24.10.2013 13:22 |
| .eclipseproduct | | 60 | 04.02.2013 04:25 |
| artifacts | xml | 81 195 | 24.10.2013 13:22 |
| eclipse | exe | 312 320 | 04.02.2013 05:05 |
| eclipse | ini | 484 | 24.10.2013 13:24 |
| eclipsec | exe | 17 920 | 04.02.2013 05:05 |
| epl-v 10 | html | 16 536 | 04.02.2013 04:28 |
| notice | html | 9 051 | 04.02.2013 04:28 |

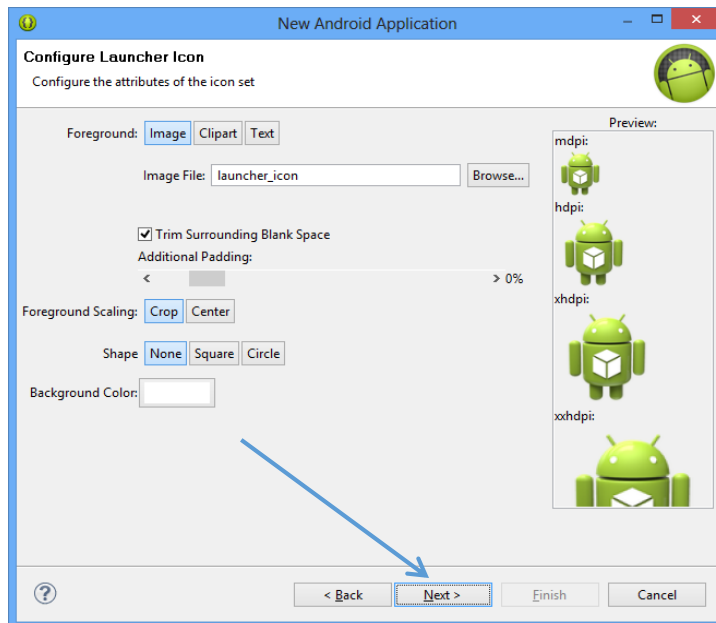
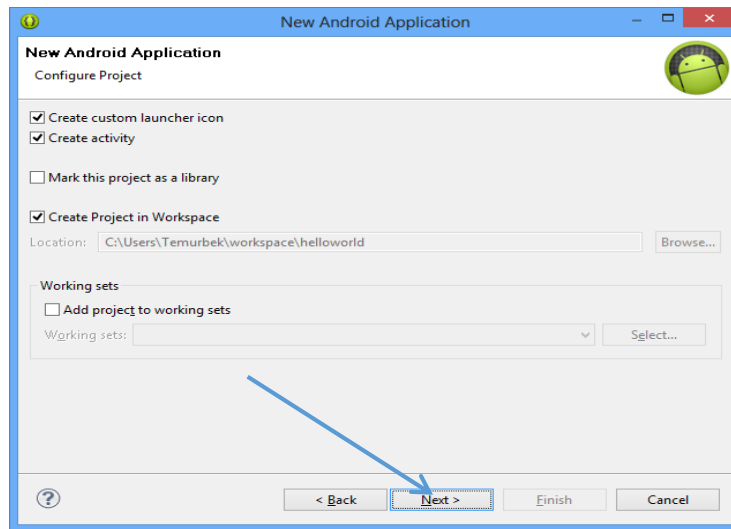


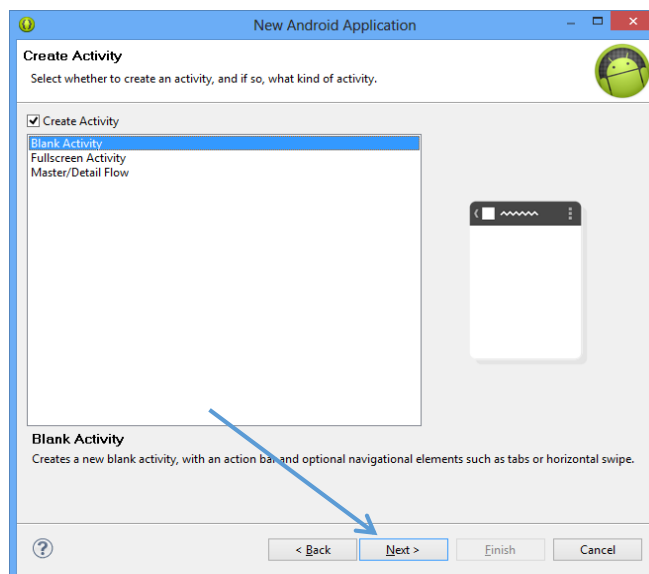
1.9-расм. Eclipse (ADT) дастурини ишга тушуриш жараёни

ADT нинг дастлабки ойнаси. Ушбу ойнадан “New Android Application” тугмаси босилади ва кейинга расмларда кўрсатилганидек кетма-кет амаллар бажарилади.

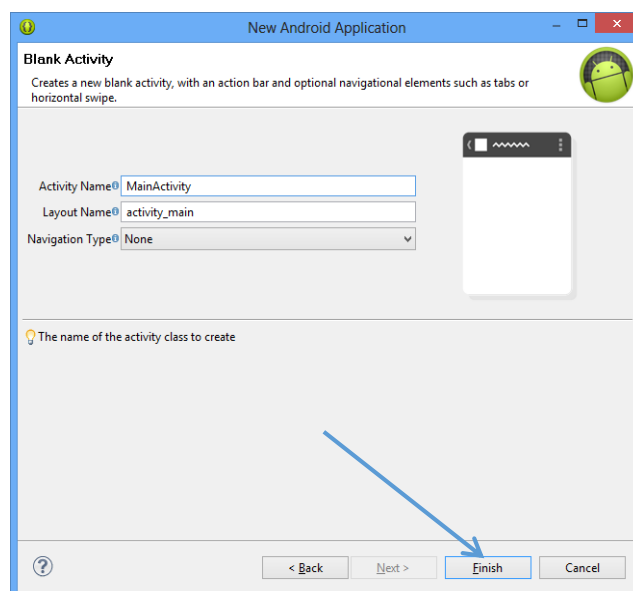


1.10-рәс. ADT-ның дәфтәрки өйкәш





1.10-а) б) с) – расм. Янги лойиҳа яратиш ойнаси



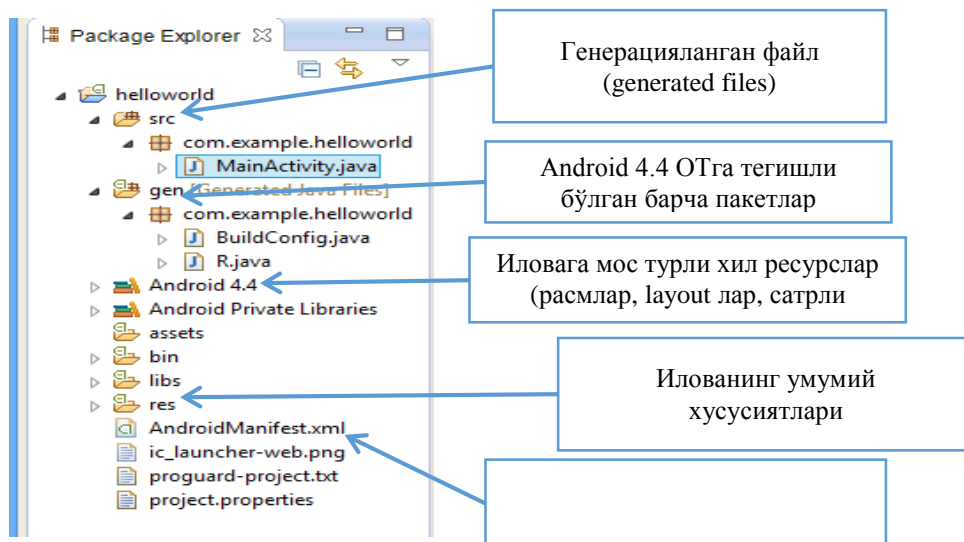
Лойиҳанинг
ташкил этувчи
файлларини (package)

Файл кодлар
қисми

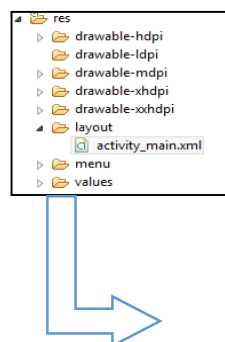
Лойиҳа
хусусиятлари

Лойиҳа
юқланишидаги
emulator ва
узелларнинг ҳолати

1.10-д) г) – расм. Янги лойиҳа яратиш ойнаси



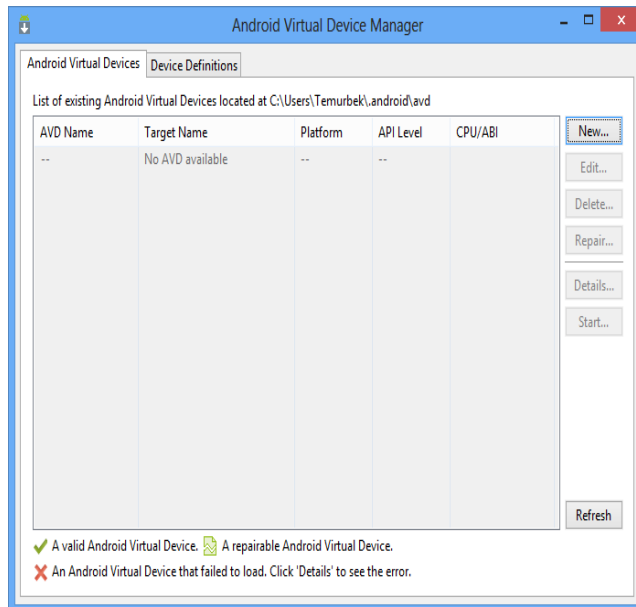
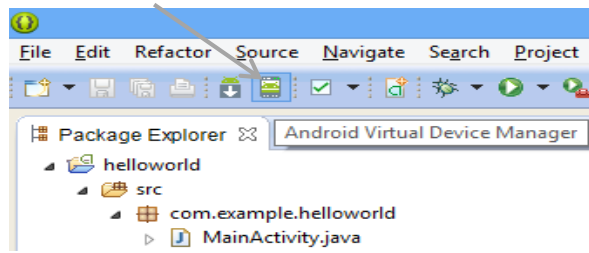
1.11-расм. Яратилган илованинг мавжуд папка ва файллар таснифи



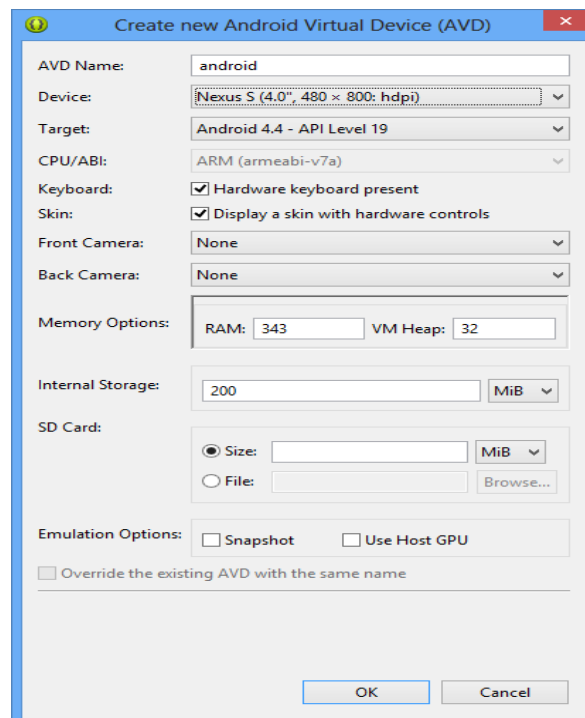
1.12-расм. Илованинг дизайн кўриниши (activity_main.xml)

Виртуал қурилмадан фойдаланиш. AVD ни яратиш

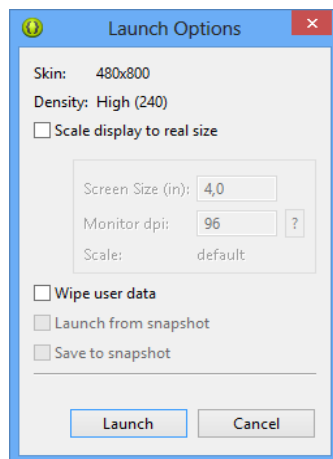
Бунинг учун қуйидаги расмда кўрсатилган “*Android Virtual Device Manager*” тугмасини босамиз



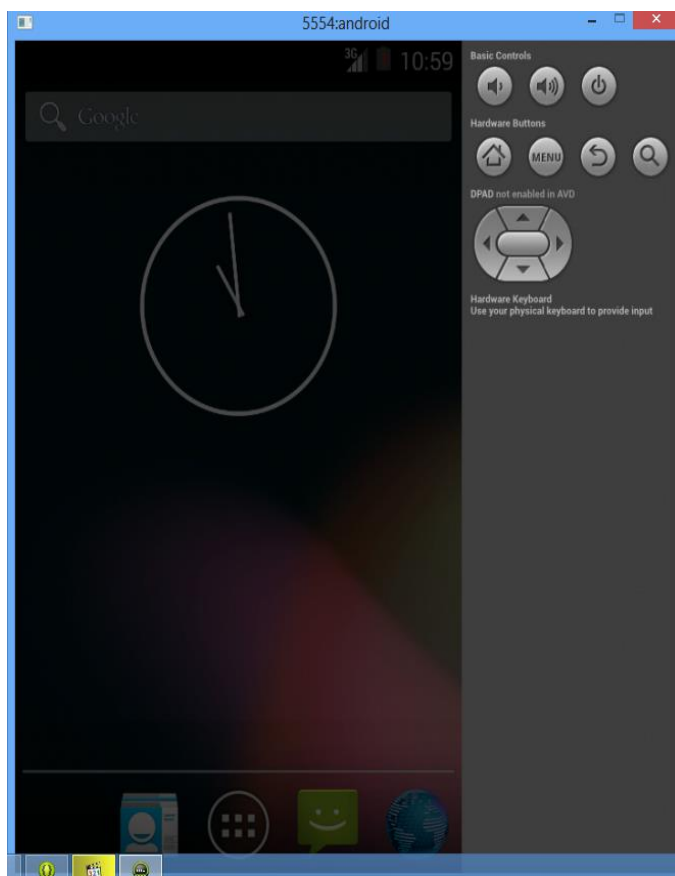
1.13-расм. “Android Virtual Device Manager” ойнаси



1.13а-расм. “Android Virtual Device Manager” ойнаси



1.14-расм. Виртуал қурилмани (AVD) ишга тушириш жараёни



1.15-расм. Виртуал қурилмани (AVD) ойнаси

Android да тезкор emulator. Genymotion virtual машинасида фойдаланиш. Android да кўпчилик фойдаланувчиларимиз компиляцияни AVD (Android Virtual Device) emulator орқали амалга оширади ва бунда натижа чиқишини кутиш керак бўлади. Ушбу камчиликни сезиларли даражада бартараф этиш учун **Genymotion** virtual машинасида фойдаланишимиз мумкин.

Назорат саволлари:

1. Мобил операцион тизим нима?
2. Мобил операцион тизимлар нечта платформадан иборат ва уларнинг номларини айтинг?
3. Мобил қурилмалар учун мўлжалланган қандай платформалар (ОТ) мавжуд (камида 8 тасини сананг)?
4. Замонавий мобил қурилмалар учун мўлжалланган операцион тизимларнинг асосий функциялари (10 тасини сананг)?
5. Андроид нима?
6. Андроид қачон ишлаб чиқарилган ва унинг охириги версияси қайси?
7. Андроиднинг Jelly Bean номли тури қайси қайси версиялар ва API level ни ўз ичига олади?
8. Андроид архитектураси қандай бўлимлар асосида тушунтирилади?
9. Андроид архитектурасига биноан “Linux Kernel” сатҳи қандай вазифаларни бажаради?
10. Андроид архитектурасига биноан “Libraries” сатҳи қандай вазифаларни бажаради?
11. Андроид архитектурасига биноан “Android runtime” сатҳи қандай вазифаларни бажаради?
12. Андроид архитектурасига биноан “Application Framework” сатҳи қандай вазифаларни бажаради?
13. Андроид операцион тизимининг асосий хусусиятлари қайсилар (камида 8 тасини сананг)?
14. Андроидда фойдаланувчи график интерфейсини яратиш қайси технология (концепция) асосида амалага оширилади?
15. Андроид ва iOS нинг фарқлари (камида 8 тасини сананг)?
16. JDK нима ва у қандай вазифани бажаради?
17. SDK нима ва у қандай вазифани бажаради?
18. ADT да янги Андроид лойиҳа яратилди. Ушбу яратилган лойиҳа ичида автоматик ҳосил бўлган AndroidManifest.xml файлини изоҳланг?
19. ADT да янги Андроид лойиҳа яратилди, ва ушбу лойиҳа ишга туширилгандан кейин яратиладиган .apk файли қайерда (лоийҳанинг қайси папкасида) жойлашади?
20. AVD нинг ўрнига қандай виртуал машиналардан фойдаланиш мумкин?
21. Андроид тиимида иловалар яратиш учун керакли бўладиган асосий компоненталар қайсилар?
22. AVD нима ва у қандай вазифани бажаради?

Фойдаланилган адабиётлар:

1. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-1. Getting started with Android programming
2. “Android™ Application Development Cookbook”, by Wei-Meng Lee, printed at Sharda Offset Press, Delhi in 2015, pages – 385.
3. http://en.wikipedia.org/wiki/Mobile_operating_system
4. http://en.wikipedia.org/wiki/Comparison_of_mobile_operating_systems
5. [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
6. http://www.tutorialspoint.com/android/android_overview.htm
7. http://www.diffen.com/difference/Android_vs_iOS

2 - мавзу: Объектга йўналтирилган дастурлаш тамойиллари. Андроид тизими учун Java дастурлаш тили. (2 соат)

Режа:

1. Андроид тизими учун Java дастурлаш муҳити. Java дастурлаш тилининг асосий конструкциялари.
2. Махсус синф ва функциялар. Объектга йўналтирилган дастурлаш хусусиятлари.
3. Абстракт синфлар.

Таянч иборалар: *Variables, Instance variables, Declaration, Instantiation, Java, Applet, Package, Import, Polymorphism, Inheritance, Encapsulation, Abstraction, Runtime (Dynamic) Polymorphism, Compile time (Static) Polymorphism*

2.1. Андроид тизими учун Java дастурлаш муҳити. Java дастурлаш тилининг асосий конструкциялари

Java - Объектга йўналтирилган дастурлаш тили бўлиб, 1991 йилда Sun компанияси томонидан ишлаб чиқарилган ва унинг биринчи версияси 1995 йилда фойдаланувчиларга тақдим қилинган. Ҳозирги кунда 3 та edition бор:

- Java 2 Platform Standard Edition (J2SE)
- Java 2 Platform Enterprise Edition (J2EE)
- Java 2 Platform Micro Edition (J2ME)

Java application - Дастурий иловалар асосан Java тили ва унинг кутубхоналари орқали яратилади, Java applet лар эса HTML билан

биргаликда ишлайдиган кичик ҳажмдаги дастурлар ҳисобланади. Бунда java файллари ва html файллари биргаликжа мавжуд бўлади(~~.java and html files (~~.html)

Java дастурлаш тили асослари

- Java тилидаги маълумотлар типлари
- Ўзгарувчиларни эълон қилиш ва уларнинг типлари
- Java modifiers (package, public, private, protected, *static*, *final*, *abstract*, *synchronized*)
- Асосий операторлар
- Цикл (while, do...while, for) операторлари
- Шарт (if, if...else, nested if, nested if...else, switch) операторлари
- Java да сонлар
- Символлар ва сатрлар билан ишлаш
- Массивлар
- Методлар
- Файллари билан ишлаш
- Истисноларни бошқариш
- Regular Expression

Юқорида айтиб ўтилганидек, Java тили тўлиқ объектга йўналтирилган дастурлаш тили бўлиб, бунда қуйидаги тушунчалар асосий ҳисобланади.

- **Object** – барча объектлар ҳолат (**states**) ва ҳаракат (**behaviors**) дан ташкил топади. Масалан машина объект бўлса унинг ҳолати – *ранги, тури, тезлиги* бўлса, ҳаракати – *тезлашиши, секинлашиши, тўхташи, ҳаракатни бошлаши* ва ҳ.к. лар ҳисобланади;
- **Class** – объект учун керакли барча ҳолатларни ўзида мужассамлаштирган **шаблон**;
- **Methods** – объектнинг ҳаракатини (**behaviors**) таъминловчи кодлар блокидир. Синф бир нечта методдан иборат бўлиши мумкин. Ҳар бир метод орқали объект ҳаракати учун керакли бўлган барча жараёнлар мантиқий кетма-кетлиги (маълумотларни юклаш, ўқиш, ёзиш, базага ёзиш, дастур интерфейсини бошқариш ва ҳ.к. лар) амалга оширилади;
- **Instance Variables** – объект мурожаат қилувчи ўзгарувчилар.

Java да биринчи дастур

```
public class MyFirstJavaProgram {  
    // This is my first java program. This will print 'Hello World' as the output
```

```

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

```

Java дастурининг умумий структураси

```

package // defines which class library this class belongs
import // defines external class/ class library to include here
class class_name {
    variables
    methods() {
    }
}

```

2.2. Махсус синф ва функциялар. Объектга йўналтирилган дастурлаш хусусиятлари.

Java тилида синфлар:

Синфлар қуйидагича ўзгарувчилардан ташкил топиши мумкин:

- **Local variables** – бу ўзгарувчилар фақатгина метод ёки конструктор ичида эълрн қилинади, бажарилади ва шу блокни ичида иш фаолиятини тўхтатади;
- **Instance variables** – бу синф ичидаги лекин методлардан ташқарида эълон қилинадиган ўзгарувчилардир. Ушбу ўзгарувчиларга синфнинг методлари, конструкторлари томонидан мурожаат қилиш мумкин;
- **Class variables** - бу синф ичида ва методлардан ташқарида эълон қилинадиган **static** ўзгарувчилардир.

Мисол:

```

public class Dog{
    String breed;
    int age;
    String color;
    void barking(){
    }
    void hungry(){
    }
    void sleeping(){
    }
}

```

Синф конструктори

Конструктор - бу Синф чакирилганда яъни жорий синфнинг объекти яратилганда ишга тушади, номи синф номи билан бир хил бўлади, синф методи ҳисобланади ва синфда конструкторлар бир нечта бўлиши мумкин яъни конструкторни overload қилса бўлади.

Мисол:

```
public class Puppy{
    public Puppy(){
    }
    public Puppy(String name){
        // This constructor has one parameter, name.
    }
}
```

Объект яратиш. Синфнинг объектини яратишда куйидаги кетма-кетлик амалга оширилади ва шу амалларни кетма-кет бажариш орқали объект яратилади:

- **Declaration** – объектни эълон қилиш;
- **Instantiation** – new калит сўзидан фойдаланиш;
- **Initialization** – объектга дастлабки қиймат бериш (new калит сўзидан кейин конструкторни чақириш).

Мисол:

```
public class Puppy{
    public Puppy(String name){
        // This constructor has one parameter, name.
        System.out.println("Passed Name is :" + name );
    }
    public static void main(String []args){
        // Following statement would create an object myPuppy
        Puppy myPuppy = new Puppy( "tommy" );
    }
}
```

Натижа: **Passed Name is :tommy**

Ўзгарувчи ва методларга мурожаат

/ Объект яратиш */*

```
ObjectReference = new Constructor();
```

/ объект орқали ўзгарувчини чақирши */*

ObjectReference.variableName;

/ объект орқали методни чақирши */*

ObjectReference.MethodName();

Мисол:

Мисол:

```
public class Puppy{
    int puppyAge;
    public Puppy(String name){
        System.out.println("Passed Name is :"+ name );
    }
    public void setAge( int age ){
        puppyAge = age;
    }
    public int getAge( ){
        System.out.println("Puppy's age is :"+ puppyAge );
        return puppyAge;
    }
    public static void main(String []args){
        Puppy myPuppy = new Puppy( "tommy" );
        myPuppy.setAge( 2 );
        myPuppy.getAge( );
        System.out.println("Variable Value :"+ myPuppy.puppyAge );
    }
}
```

Натижа:

Passed Name is :tommy

Puppy's age is :2

Variable Value :2

Encapsulation – бу синф ўзгарувчиларининг (майдонлари) ҳимояланганлиги


```

class Encapsulation {
    private int summa = 500;
    protected int smax = 45500;
}

public int getSumma() {
    return this.summa;
}

public class EncapsulationDemo {
    public static void main(String []argc) {
        Encapsulation d = new Encapsulation();
        System.out.println(d.getSumma());
    }
}

```

Encapsulation

Ворислик (inheritance) - бошқа бир синфдан ворис (насл) олиш. Java да насл олиш **extends** калит сўзидан фойдаланилади.

Мисол:

```

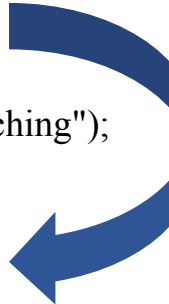
class Android extends MobilePhone { ... }

class MobilePhone {
    public void switch() {
        System.out.println("Phone is switching");
    }
}

class Android extends MobilePhone {
    public void switchOS() {
        System.out.println("Switching Android 4.4");
    }
}

public class Demo {
    public static void main(String []args) {
        Android kitkat = new Android();
        kitkat.switch();
        kitkat.switchOS();
    }
}

```



Polymorphism – бу грекча сўздан олинган бўлиб, “**poly**” – кўп, “**morphos**” – формалар, “кўп формалилик” ёки дастурлаш тилида “бир хил метод (объект, ўзгарувчи) турли хил интерфейс (масала)” деган маънони билдиради. Бу дастурчи учун “*flexible*” дастур тузиш имкониятини яратиб

беради¹. Polymorphism икки хил бўлади:

- Runtime (Dynamic) Polymorphism
- Compile time (Static) Polymorphism

Method overloading ва method overriding бир-биридан фарқи

| Method overloading | Method overriding |
|--|---|
| Бир номли лекин турли хил параметрли функциялар ва уларга мурожаат қилиш | Бир номли ва бир хил параметрли функциялар ва уларга мурожаат қилиш |
| Method overloading битта синф ичида бажарилади | Method overriding супер синф ва ворис синф асосида бажарилади |
| Метод қайтариш типи бир хил ёки ҳар хил бўлиши мумкин | Метод қайтариш типи ҳар доим бир хил бўлади |
| JVM функция параметрига қараб қайси функция чақирилатганини аниқлайди | JVM объект типига қараб (яъни қайси синфга тегишли) қайси функция чақирилатганини аниқлайди |

2.3 Абстракт синфлар.

Абстракт синфлар - Камида битта абстракт методга эга бўлган синф – абстракт синф дейилади. “abstract” калит сўзидан фойдаланилади ва абстракт синфдан объект яратиб бўлмайди. Эълон қилиниши қуйидаги мисолда кўрсатилган:

```
abstract class MyClass{
    abstract void calculate(double x); // метод
}
....
MyClass m; // хато!!! //compilation error!!!
```

Назорат саволлари

1. Java қачон қайси ташкилоттомонидан ишлаб чиқилган?
2. Синф ва объектнинг бир-биридан фарқи?
3. Java да маълумотлар типлари (ҳар бир тип хтирадан қанча жой эгаллашини ҳам кўрсатинг)?
4. Объектга йўналтирилган дастурлашнинг асосий хусусиятлари қайсилар?
5. Java да синфлар қандай ўзгарувчиларни ўз ичига олади?
6. Конструктор нима ва унинг оддий методдан фарқи?

¹ http://www.tutorialspoint.com/java/java_polymorphism.htm

7. Объект яратиш (тълик объект яратилганда) жараёнида қандай кетма – кетлик амалга оширилади?
8. Объект қандай қилиб инициализация қилинади (Мисол келтиринг)?
9. Объектнинг ўзгарувчиси ва методига қандай мурожаат қилинади (Мисол келтиринг)?
10. Java да ворис олиш учун қандай калит сўздан фойдаланилади ва синфдан ворис олишга мисол келтиринг?
11. Қандай турдаги полиморфизмлар мавжуд?
12. Method overloading нима ва унга мисол келтиринг?
13. Method overriding нима ва унга мисол келтиринг?
14. Бир номли лекин турли хил параметрли функциялар ва уларга мурожаат қилиш нима дейилади?
15. Бир номли ва бир хил параметрли функциялар ва уларга мурожаат қилиш нима дейилади?
16. Method overriding битта синф ичида амалга оширилиши мумкинми. Мумкин ёки мумкин эмаслигини исботланг?
17. Абстракт синф нима ва у қандай хусусиятларга эга?
18. Абстракт синф қандай яратилади?

Адабиётлар ва интернет ресурслар

1. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-1. Getting started with Android programming
2. "Java The Complete Reference Eighth Edition" by Herbert Schildt, 2014, Глава-6,7,8,9,10. стр.145-259
3. Core Java “An Integrated Approach”, Include all versions up to Java7 by Dr. R. Nageswara Rao, 205, 646 pages
4. http://www.tutorialspoint.com/java/java_polymorphism.htm
5. <http://beginnersbook.com/2013/03/polymorphism-in-java/>
6. http://www.tutorialspoint.com/java/java_abstraction.htm
7. http://www.tutorialspoint.com/java/java_object_classes.htm

3 - мавзу: Андроид фойдаланувчи интерфейсини яратиш. Иловаларда ходисалар ва жараёнлар. Андроид тизимида Layout ва View лардан фойдаланиш. Анимация.

(2 соат)

Режа

1. Андроид иловаларида ходисалар ва жараёнлар. Activity яшаш цикли.
2. GUI компоненталари. Views ва ViewGroup лардан фойдаланиш. Layout лар ва улардан фойдаланиш.
3. *Gallery*, *ImageSwitcher*, *GridView*, ва *ImageView* компоненталаридан фойдаланиш. Расмларни жойлаштириш.
4. Илова менюларини бошқариш.
5. Иловаларда анимациядан фойдаланиш.

Таянч иборалар: *Activity*, *fragment*, *intents*, *ImageView*, *GridView*

3.1 Андроид иловаларида ходисалар ва жараёнлар. Activity яшаш цикли

Фойдаланувчи интерфейсини бошқарувчи асосий .java кенгайтмали файл бўлиб, Activity синфидан ворис олган ҳолда яратилади. Мисол учун:

- email иловаси битта activity почтага келган янги хатларни кўрсатиб туради;
- бошқа бир activity жўнатилган хатларни кўрсатади;
- бошқаси эса келган хатларни ўқиш учун ишлатилади.

Агар яратилган илова бир ечта activity дан иборат бўлса, у ҳолда битта activityни (файлни) асосий деб кўрсатиб қўйилиши лозим ва бу activity илова ишга туширилганда чақирилади¹.

Мисол-1: MainActivity.java

```
package com.example.helloworld;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class MainActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

¹ Beginning Android 4 Application Development, Chapter-2. Activities, fragments and intents, pages 64-65.

```

super.onCreate(savedInstanceState);
setContentView (R.layout.activity_main);
}
}

```

Одатда activity дастур интерфейсини ўзида мужассамлаштирган XML файлини res/layout папкасидан чақириб олади. Бизнигн мисолимизда **activity_main.xml** файли қуйидагича чақириляпти:

setContentView (R.layout.activity_main);

Иловада ишлатилаётган ҳар бир activity AndroidManifest.xml файлида қуйидагича эълон қилиб қўйилиши лозим:

```

<?xml version="1.0" encoding="utf-8">
<manifest xmlns:android=http://schemas.android.com/apk/res/android
.....
<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >
    <activity
        android:label="@string/app_name"
        android:name=".MainActivity" >
        <intent-filter >
            <action android:name="android.intent.action.MAIN" />
            <category
                android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
</manifest>

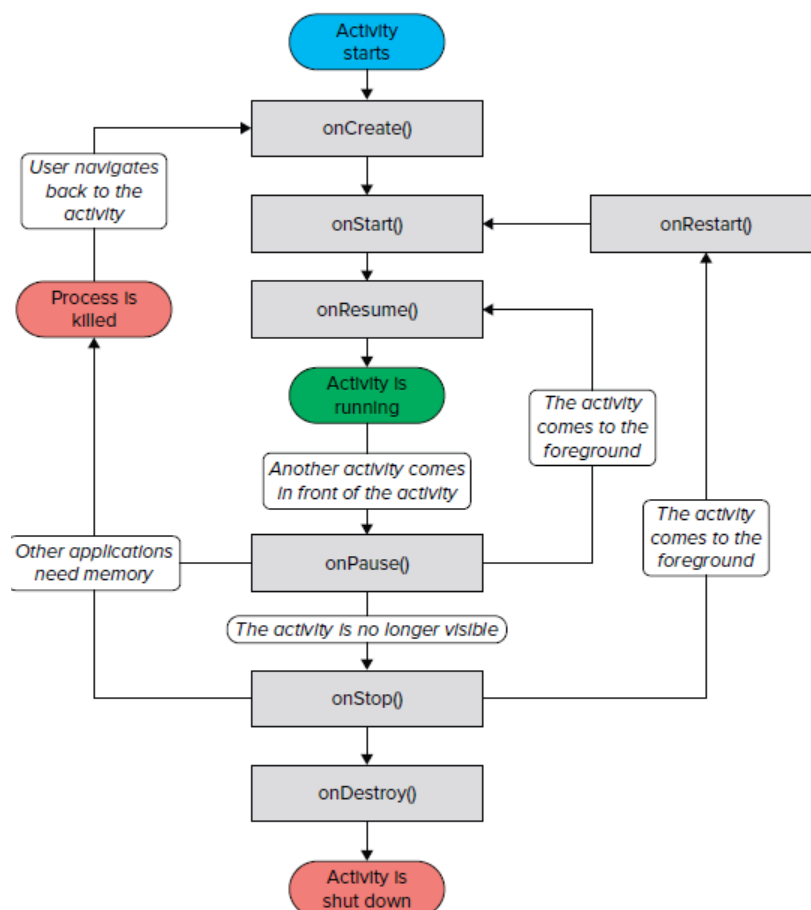
```

Activity яшаш цикли¹

- **onCreate()** — activity ишга тушганда биринчи марта чақирилади;
- **onStart()** — activity натижаси фойдаланувчига кўринганда чақирилади;
- **onResume()** — activity фойдаланувчи томонидан ишлатилаётганда чақирилади;

¹ Beginning Android 4 Application Development, Chapter-2. Activities, fragments and intents, pages 64-68.

- **onPause()** — жорий activity фаолияти вақтинчалик тўхтатилиб бошқа activity фаоллаштирилганда чақирилади;
- **onStop()** — activity фойдаланувчига кўринмай қолганида чақирилади;
- **onDestroy()** — activity тўхтатилган вақтда чақирилади;
- **onRestart()** — activity иш фаолияти тўхтатаилиб қайта ишга туширилганида чиқарилади.



3.1-расм. Activity яшаш цикли визуал кўринишда

Мисол-2:

```

package com.example.helloworld;
import android.os.Bundle;
import android.app.Activity;
import android.util.Log;
public class MainActivity extends Activity {
    String tag = "Lifecycle";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

```

```

public void onStart() {
    super.onStart();
    Log.d(tag, "In the onStart() event");
}
public void onRestart() {
    super.onRestart();
    Log.d(tag, "In the onRestart() event");
}
public void onResume() {
    super.onResume();
    Log.d(tag, "In the onResume() event");
}
public void onPause() {
    super.onPause();
    Log.d(tag, "In the onPause() event");
}
public void onStop(){
    super.onStop();
    Log.d(tag, "In the onStop() event");
}
public void onDestroy(){
    super.onDestroy();
    Log.d(tag, "In the onDestroy() event");
}
}

```

Дастур натижаси. LogCat console

Declaration Console LogCat

Search for messages. Accepts Java regexes. Prefix with pid; app; tag; or text: to limit scope.

| L... | Time | PID | TID | Application | Tag | Text |
|------|------------------|------|------|-------------|-----------|--------------------------|
| D | 09-22 05:52:4... | 1385 | 1385 | | Lifecycle | In the onStart() event |
| D | 09-22 05:52:4... | 1385 | 1385 | | Lifecycle | In the onResume() event |
| D | 09-22 05:53:2... | 1385 | 1385 | | Lifecycle | In the onPause() event |
| D | 09-22 05:53:2... | 1385 | 1385 | | Lifecycle | In the onStop() event |
| D | 09-22 05:53:2... | 1385 | 1385 | | Lifecycle | In the onDestroy() event |
| D | 09-22 05:53:3... | 1385 | 1385 | | Lifecycle | In the onStart() event |
| D | 09-22 05:53:3... | 1385 | 1385 | | Lifecycle | In the onResume() event |
| D | 09-22 05:53:3... | 1385 | 1385 | | Lifecycle | In the onPause() event |
| D | 09-22 05:53:3... | 1385 | 1385 | | Lifecycle | In the onStop() event |
| D | 09-22 05:53:4... | 1385 | 1385 | | Lifecycle | In the onRestart() event |
| D | 09-22 05:53:4... | 1385 | 1385 | | Lifecycle | In the onStart() event |
| D | 09-22 05:53:4... | 1385 | 1385 | | Lifecycle | In the onResume() event |

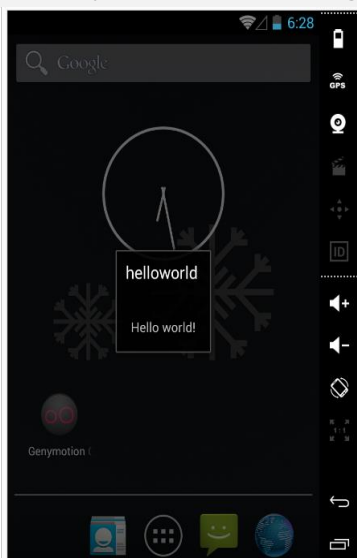
3.2-пачм. LogCat console

activity лар натижаси dialog ойнаси кўринишида ишлатиш мумкин. Асосан activity лар илованинг тўлиқ ойнасида ҳосил бўлади. Лекин activity натижаларини dialog ойнада ҳосил қилиш мумкин. Мисол учун “warning message” ларни экранга чиқариш. Бунинг учун **AndroidManifest.xml** файлида жойлашган **<Activity>** элементининг **android:theme** атрибути ўзгартирилади¹:

```
android:theme="@style/AppTheme" >
```

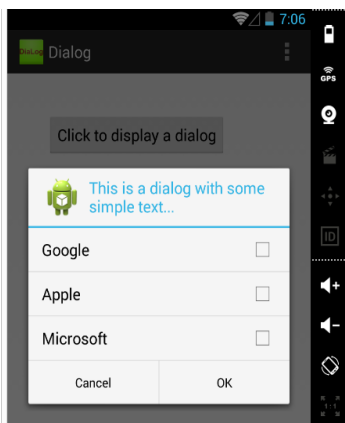
ўрнига, қуйидагича ёзилади

```
android:theme="@android:style/Theme.Dialog">
```



3.3-расм. Theme.Dialog стилида илова кўриниши

Activity орқали Dialog Window ҳосил қилиш



3.2 GUI компоненталари. Views ва ViewGroup лардан фойдаланиш. Layout лар ва улардан фойдаланиш

Андроид тизимида фойдаланувчи интерфейси асосан widget лар яъни buttons, labels, textboxes ва ҳ.к. лар ва буларнинг жойлашувига мос layout лардан иборат бўлади. Ўз навбатида бу компоненталар XML файл шаклида

¹ Beginning Android 4 Application Development, Chapter-2. Activities, fragments and intents, pages 69-70.

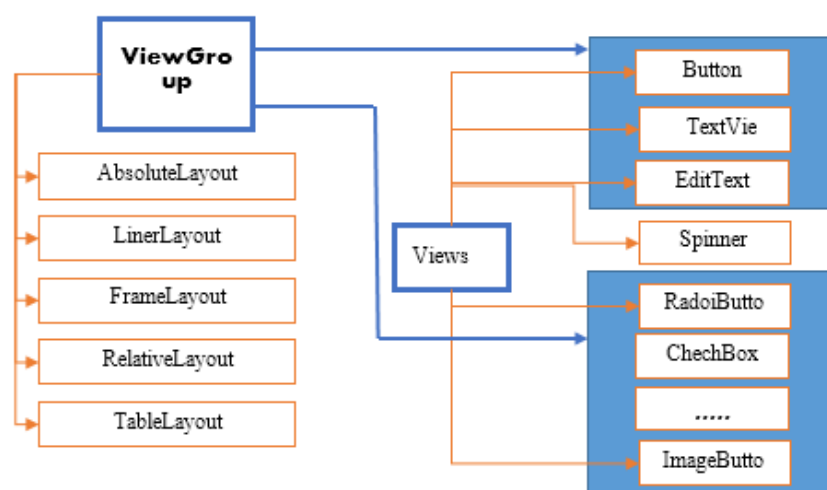
сақланади ва activity асосида бошқарилади¹. XML файлига мисол:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
</LinearLayout>
```

Activity синфининг setContentView() методи орқали керакли XML файл яъни дастур интерфейси қуйидагича чақирилади:

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
}
```

Демак дастур компиляцияси жараёнида XML файлда жойлашган ҳар бир элемент ва унинг атрибутига мос Android GUI class яратилади. Ушбу операциядан кейингина activity ишга туширилганда илова интерфейси чақирилади. Фойдаланувчи интерфейси асосан график компоненталар view ва viewgroup лардан иборат бўлади.



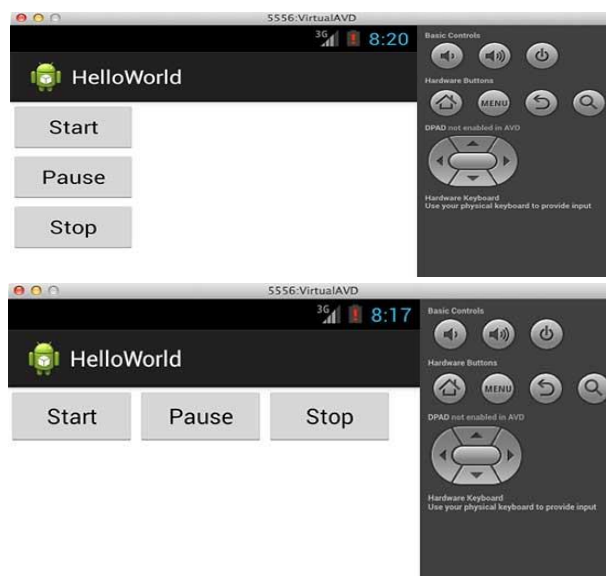
3.4-расм. Компоненталар гуруҳи ва элементлари

¹ Beginning Android 4 Application Development, Chapter-3. Getting to Know the Android User Interface, pages 133-134.

ViewGroup – бу компоненталар яъни бир нечта *views* лар бирлашмаси ва бу компоненталар *layout* ларда жойлашади. **Views** – бу компоненталар, яъни интерфейс учун керакли бўлган ҳар бир компонента (button, checkbox, textview, ... ва ҳ.к.лар).

LinearLayout - бу компоненталарни битта қаторга (горизонтал) ёки устун (вертикал)га жойлаштириш учун ишлатилади¹.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <Button android:id="@+id/btnStartService"
        android:layout_width="150px"
        android:layout_height="wrap_content"
        android:text="@string/start_service"/>
    <Button android:id="@+id/btnPauseService"
        android:layout_width="150px"
        android:layout_height="wrap_content"
        android:text="@string/pause_service"/>
    <Button android:id="@+id/btnStopService"
        android:layout_width="150px"
        android:layout_height="wrap_content"
        android:text="@string/stop_service"/>
</LinearLayout>
```



¹ Beginning Android 4 Application Development, Chapter-3. Getting to Know the Android User Interface, pages 135-136.

AbsoluteLayout – бу компоненталарни маълум ўлчамлар ва оралиқлар асосида жойлаштириш имкониятини яратади¹.

<AbsoluteLayout

```
android:layout_width="fill_parent"
android:layout_height="fill_parent"
xmlns:android="http://schemas.android.com/apk/res/android" >
<Button
    android:layout_width="188dp"
    android:layout_height="wrap_content"
    android:text="Button"
    android:layout_x="126px"
    android:layout_y="361px" />
<Button
    android:layout_width="113dp"
    android:layout_height="wrap_content"
    android:text="Button"
    android:layout_x="12px"
    android:layout_y="361px" />
```

</AbsoluteLayout>

Мисолда кўринадик, берилган Button компоненталари **android_layout_x** ва **android_layout_y** атрибутлари қиймати асосида жойлашади. Лекин битта муаммо бор. AbsoluteLayout асосида яратилган фойдаланувчи интерфейслари Андроид ОТ нинг кейинги версиялари ва экран турига қараб хатоликлар келиб чиқиши мумкин.



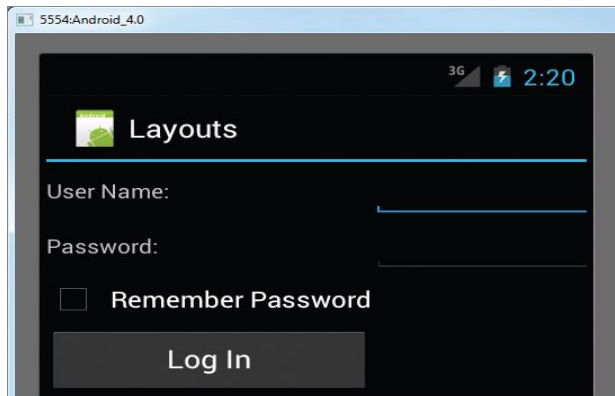
¹ Beginning Android 4 Application Development, Chapter-3. Getting to Know the Android User Interface, page 143.

TableLayout - компоненталарнинг жадвал асосида жойлашувидир. Бунда <TableRow> элементи жадвал қаторини белгилаб беради. Ҳар бир қатор бир ёки бир нечта компонентани ўз ичига олади. Қатор узунлиги мос ҳолда компоненталар узунликлари йиғиндисидан иборат бўлади¹.

```
<TableLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_height="fill_parent"
  android:layout_width="fill_parent" >
  <TableRow>
    <TextView
      android:text="User Name:"
      android:width="120dp"
    />
    <EditText
      android:id="@+id/txtUserName"
      android:width="200dp" />
  </TableRow>
  <TableRow>
    <TextView
      android:text="Password:"
    />
    <EditText
      android:id="@+id/txtPassword"
      android:password="true"
    />
  </TableRow>
  <TableRow>
    <TextView />
    <CheckBox android:id="@+id/chkRememberPassword"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="Remember Password"
    />
  </TableRow>
</TableLayout>
```

¹ Beginning Android 4 Application Development, Chapter-3. Getting to Know the Android User Interface, pages 144-145.

```
<Button
    android:id="@+id/buttonSignIn"
    android:text="Log In" />
</TableRow>
</TableLayout>
```



RelativeLayout – бу компоненталарни бир – бирига мос (боғлиқ) ҳолда жойлаштириш учун ишлатилади. Яъни бир компонентани ўнг тарафидан (чап тарафидан, юқори қисмидан, пастки қисмидан) иккинчи компонентани жойлаштириш мумкин бўлади. RelativeLayout да компоненталар куйидаги атрибутлар асосида бир – бири билан боғлиқ ҳолда жойлаштирилади¹.

Android тизимида икки ҳил кўринишдаги ориентация мавжуд: portrait ва landscape. Бу ориентациялар икки ҳил усул ёрдамида бошқарилади ва улар куйидагилардир.

- **Anchoring** — экраннинг тўртта бурчаги ҳисобига компоненталарни бошқариш;
- **Resizing and repositioning** — ориентация ҳолатига қараб турли хил layout (xml file) ларга мувожаат қилиш орқали компоненталарни экранга чиқариш.

Андроид иловаларини ишлаб чиқишда ишлатиладиган компоненталар (ускуналар) View лар деб аталади ва уларнинг куйидаги турлари мавжуд².

- **Basic views** — Энг кўп фойдаланиладиган оддий view лар бўлиб, булар TextView, EditText, Button ва бошқалардир
- **Picker views** — TimePicker ва DatePicker яъни соат ва вақтни бошқариш учун ишлатиладиган компонента ҳисобланади.

¹ Beginning Android 4 Application Development, Chapter-3. Getting to Know the Android User Interface, pages 145-146.

² Beginning Android 4 Application Development, Chapter-4. Designing Your User Interface With Views pages 187-194.

- **List views** — Дастур интерфейсида рўйхатларни ҳосил қилиш ва улардан фойдаланиш учун ишлатиладиган `ListView` ва `SpinnerView` компоненталардир.
- **Specialized fragments** — махсус функцияларни бажариш учун мўлжалланган **fragment** лар

Basic view лардан фойдаланиш

- **Button** — тугма ҳисобланиб, ушбу тугмани босиш орқали маълум операция бажарилади
- **ImageButton** — `button` га ўхшаш `widget` бўлиб, бунда расми тугма сифатида ишлатиш мумкин
- **EditText** — `TextView` view нинг қўшимча имкониятли тури ҳисобланиб, фойдаланувчи ушбу компонента орқали маълумот киритиши мумкин
- **CheckBox** — `button` нинг махсус тури ҳисобланиб, иккита ҳолатда бўлиши мумкин: `checked` ёки `unchecked`
- **RadioGroup** ва **RadioButton** — `RadioButton` да ҳам икки хил ҳолат мавжуд: `checked` ёки `unchecked`. `RadioGroup` бу бир нечта `RadioButton` дан иборат бўлиб бунда фақатгина битта `RadioButton` `checked` ҳолатида бўлиши мумкин.
- **ToggleButton** — `checked/unchecked` ҳолатининг `indicator` ли кўриниши

Мисол:

```

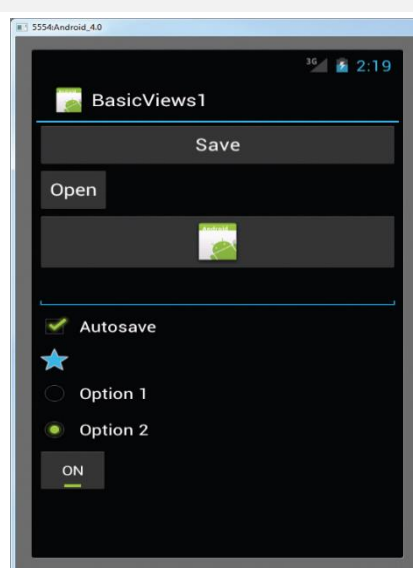
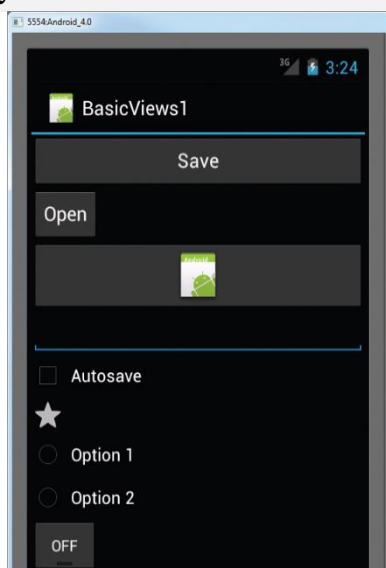
LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <Button android:id="@+id/btnSave"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="save" />
    <Button android:id="@+id/btnOpen"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Open" />
    <ImageButton android:id="@+id/btnImg1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_launcher" />
    <EditText android:id="@+id/txtName"

```

```

        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
<CheckBox android:id="@+id/chkAutosave"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Autosave" />
<CheckBox android:id="@+id/star"
    style="?android:attr/starStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<RadioGroup android:id="@+id/rdbGp1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >
<RadioButton android:id="@+id/rdb1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Option 1" />
<RadioButton android:id="@+id/rdb2"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Option 2" />
</RadioGroup>
<ToggleButton android:id="@+id/toggle1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>

```



3.5-расм. Фойдаланувчи интерфейсида компоненталар кўриниши

View ларда **Event** лар орқали бошқариш

Демак бу қандай амалга оширилади, яъни button ни босганда қандай action ҳосил бўлади, checkbox босилганида қандай белгиланида ва ҳ.к. лар. Бу ҳолат илова компоненталарида ҳодисаларни бошқариш деб аталади ва қуйидаги мисол билан кўриб чиқамиз. Юқорида келтирилган BasicViews1 лойиҳасининг асосий файллари яъни BasicViews1Activity.java қуйидагича ўзгартирилади¹.

```
public class BasicViews1Activity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
//---Button view---
        Button btnOpen = (Button) findViewById(R.id.btnOpen);
        btnOpen.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                DisplayToast("You have clicked the Open button");
            }
        });
//---Button view---
        Button btnSave = (Button) findViewById(R.id.btnSave);
        btnSave.setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View v) {
                DisplayToast("You have clicked the Save button");
            }
        });
//---CheckBox---
        CheckBox checkBox = (CheckBox) findViewById(R.id.chkAutosave);
        checkBox.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                if (((CheckBox)v).isChecked())
                    DisplayToast("CheckBox is checked");
                else
```

¹ Beginning Android 4 Application Development, Chapter-4. Designing Your User Interface With Views pages 196-199.


```

        DisplayToast("CheckBox is unchecked");
    }
});
//---RadioButton---
    RadioGroup radioGroup = (RadioGroup) findViewById(R.id.rdbGp1);
    radioGroup.setOnCheckedChangeListener(new
OnCheckedChangeListener() {
        public void onCheckedChanged(RadioGroup group, int checkedId) {
            RadioButton rb1 = (RadioButton) findViewById(R.id.rdb1);
            if (rb1.isChecked()) {
                DisplayToast("Option 1 checked!");
            } else {
                DisplayToast("Option 2 checked!");
            }
        }
    });
//---ToggleButton---
    ToggleButton toggleButton = (ToggleButton) findViewById(R.id.toggle1);
    toggleButton.setOnClickListener(new View.OnClickListener(){
        public void onClick(View v) {
            if (((ToggleButton)v).isChecked())
                DisplayToast("Toggle button is On");
            else
                DisplayToast("Toggle button is Off");
        }
    });
// closed onCreate() method
    private void DisplayToast(String msg)
    {
        Toast.makeText(getApplicationContext(), msg,
        Toast.LENGTH_SHORT).show();
    }
} // closed BasicViews1Activity class

```

ProgressBar view компонентаси дастур томонидан қандайдир операция бажарилиши давомда кутиш ҳолатини тасвирлашда ишлатилади. Мисол учун интернетдан бирон бир дастурни юклаб олишда, бирон ахборот тизимига кириш жараёнида ва бошқа ҳолатларда фойдаланувчи маълум вақт кутишига тўғри келади. Андроид фойдаланувчи интерфрейсида вақтни ва

санани кўрсатиб туриш учун қуйидаги view лардан фойдаланилади. **TimePicker** view – вақтни кўрсатиш компоненти, **DatePicker** view – санани кўрсатиш компоненти ҳисобланади.

ListView – рўйхат ҳосил қилиш учун ишлатиладиган компонента ҳисобланади. Android да рўйхат ҳосил қилиш учун 2 та компонентадан, ListView ва SpinnerView дан фойдаланади. ListView асосан рўйхатларни вертикал кўринишдаги давоми мавжуд бўлган (scrolling list) интерфейсида ҳосил қилади.

3.3 Gallery, ImageSwitcher, GridView, ва ImageView компоненталаридан фойдаланиш. Расмларни жойлаштириш

Андроид иловаларида расмларни жойлаштириш. Андроид иловаларида расмларни жойлаштириш ImageView, Gallery, ImageSwitcher ва GridView компоненталаридан фойдаланилади. Gallery – бу расмлар жамланмасини экранда кўрсатиш учун ишлатиладиган компонента бўлиб, бунда расмлар горизонтал list кўринишида жойлашиб, жорий расм актив ҳолатда бўлади. Gallery view да расмлар бир тарафдан иккинчи тарафга ҳаракатланади ва мос ҳолда расмлар активлиги ўзгариб туради¹.



ImageAdapter дан фойдаланиш. Бу мисолда ImageAdapter синфи (BaseAdapter синфининг вориси) Gallery ва ImageView компоненталарни бир – бирига боғлаш учун ишлатилади. Одатда BaseAdapter синфи AdapterView ва маълумотлар (база, рўйхат, массив) орасида боғловчи бўлиб хизмат қилади. AdapterView компоненталарига қуйидагилар қиради:

- ListView
- GridView
- Spinner
- Gallery

BaseAdapter синфининг ворис синфлари қуйидагилар:

- ListAdapter
- ArrayAdapter
- CursorAdapter
- SpinnerAdapter

¹ Beginning Android 4 Application Development, Chapter-5. Displaying pictures and menus with View, pages 247-249.

Демак, Gallery view ва ImageView компоненталари расмларни (тасвир) экранда навбати билан кўрсатиш учун ишлатилади ва Галереядан белгиланган расм ImageView соҳасида ҳосил бўлади. Баъзи ҳолларда расмлар алмашиш жараёнида анимацион ҳаракатлар узатилади. Буни амалга ошириш учун Андроид тизимида Gallery view билан биргаликда *ImageSwitcher* компонентаси.



3.6-расм. Андроид иловаларида Галереядан фойдаланиш GridView компонентаси расмларни (бирор ресурсни) жадвал кўринишида жойлаштириш учун ишлатилади. GridView компонентаси ImageView билан биргаликда расмларни экранга икки ўлчовли грид кўринишида ҳосил қилиш учун ишлатилади.

3.4 Илова менюларини бошқариш

Барча иловаларда менюлар ҳар доим ишлатилади ва булар фойдаланувчи интерфейсини (илова интерфейсини) бошқаришда қулайликлар яратади. Android тизимида икки хил турдаги меню мавжуд¹:

Options menu — Илованинг қўшимча хусусиятларини очиб берувчи меню ҳисобланади ва бажариладиган операциялар жорий activityга боғлиқ бўлади. Android иловаларида ушбу менюни ҳосил қилиш учун MENU тугмаси босилади.

Context menu — Жорий activity да бажариладиган операциялар учун керакли командаларни очиб берувчи меню ҳисобланиб, Android тизимида иловага тегишли бўлган context менюни очиш учун жорий тугма ушлаб турилади.

Context menu бизга айнан менюни аниқ бир view га улаш имконини беради. Бунда қуйидаги методлардан фойдаланилади

– **onCreateContextMenu**

¹ Beginning Android 4 Application Development, Chapter-5. Displaying pictures and menus with View, pages 262-263.

– **onContextItemSelected**

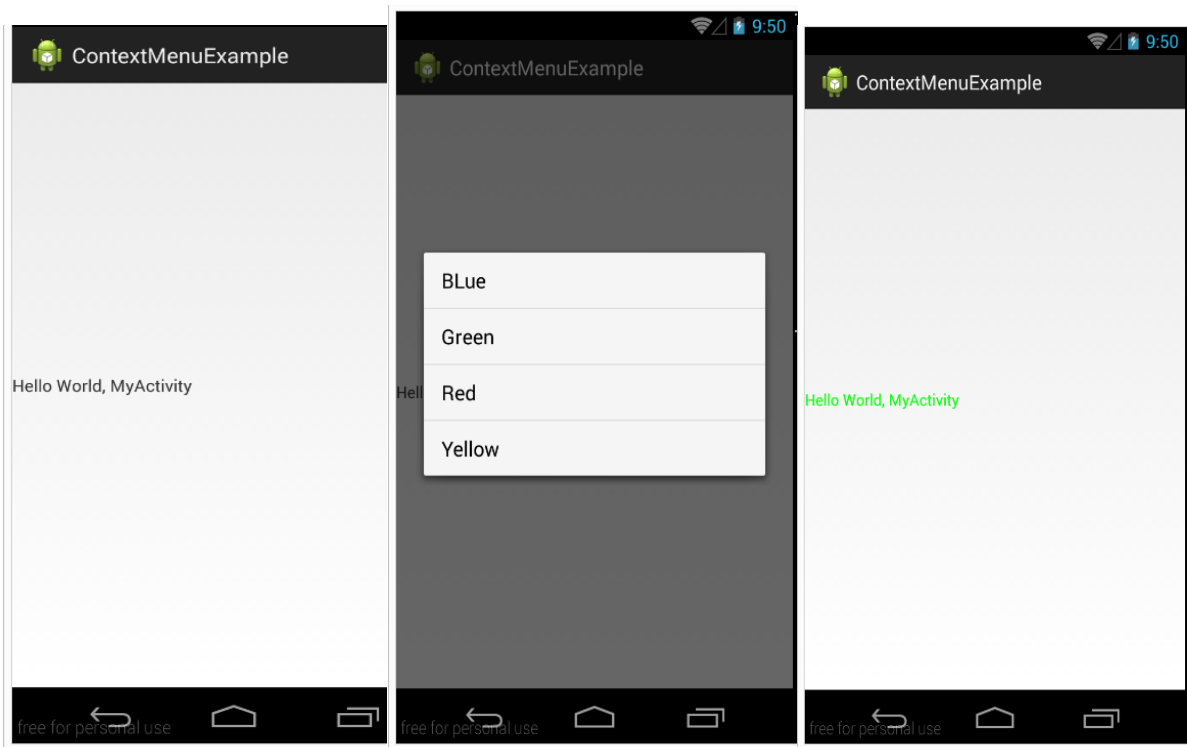
onCreateContextMenu() – методи қуйидагича бўлади

```
public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
    menu.add(0, 1, 0, "blue");
    menu.add(0, 2, 0, "green");
    menu.add(0, 3, 0, "red");
    menu.add(0, 4, 0, "yellow");
}
```

onContextItemSelected() – методи қуйидагича бўлади

@Override

```
public boolean onContextItemSelected(MenuItem item) {
    switch (item.getItemId()){
        case 1:
            textView.setTextColor(Color.BLUE);break;
        case 2:
            textView.setTextColor(Color.GREEN); break;
        case 3:
            textView.setTextColor(Color.RED); break;
        case 4:
            textView.setTextColor(Color.YELLOW);break;
    }
    return super.onContextItemSelected(item);}
}
```



Option Menu

Бу menu андроид иловасининг кўшимча имконияталарини ҳосил қилишда (бир менюдан иккинчи менюга ўтишда) ишлатилади.

Ишлатилиши худди олдинги слайдларда кўриб ўтган ContextMenu билан бир хил фақат бунда, **onCreateContextMenu()** ўрнига **onCreateOptionsMenu()** методи, **onContextItemSelected()** ўрнига **onOptionsItemSelected()** методи ишлатилади.

3.5 Иловаларда анимациядан фойдаланиш

Андроид иловаларида анимация турлари

- **Alpha** – объектнинг шаффофлик даражаси ўзгариши
- **Scale** – объектнинг ўлчами ўзгарган ҳолда бор ёки йўқ бўлиши
- **Translate** – объектни бир жойдан иккинчи жойга ҳаракатлантириш
- **Rotate** – объектни буриш

Alpha

- **AlphaAnimation** - конструкторига fromAlpha ва toAlpha параметрлар берилади.
- **fromalpha** – бу объектни анимация бошланмасидан олдинги шаффофлик даражаси
- **toAlpha** - бу объектни анимация охиридаги шаффофлик даражаси
- **setDuration** – бу анимацияни давом этиш вақти
- **setRepeatCount** – анимацияни давом этиш сони

- **setRepeatMode** – анимация бошлангандан view ни позицияси
- **reverse** – объект орқага қайтади
- **restart** – ҳаракат объектнинг бошида турган жойидан бошланади.

Scale

- **fromXScale** — x нинг бошланғич масштаби. 1 дан 0 гача бўлган Float типдаги қийматларни қабул қилади.
- **toXScale** — x нинг охириги масштаби. 1 дан 0 гача бўлган Float типдаги қийматларни қабул қилади.
- **fromYScale** у нинг бошланғич масштаби. 1 дан 0 гача бўлган Float типдаги қийматларни қабул қилади.
- **toYScale** у нинг охириги масштаби. 1 дан 0 гача бўлган Float типдаги қийматларни қабул қилади.
- **pivotX** — x ўқи бўйича объектни экранда туриш координатаси ва унинг қийматлари % ларда олинади
- **pivotY** — у ўқи бўйича объектни экранда туриш координатаси ва унинг қийматлари % ларда олинади

Translate

- **fromXDelta** — x ўқи бўйича бошланғич координатаси
- **toXDelta** — x ўқи бўйича охириги координатаси
- **fromYDelta** — у ўқи бўйича бошланғич координатаси
- **toYDelta** — у ўқи бўйича охириги координатаси

Rotate

- **fromDegrees** — анимация бошланмасдан олдинги бошланғич бурчак
- **toDegrees** - анимация охириги тугаш бурчак
- **pivotX** —x ўқи бўйича объект координатаси;
- **pivotY** — у ўқи бўйича объект координатаси;

Назорат саволлари:

1. Activity нима?
2. Activity ларни бир – бирига боғлаш учун қандай синфдан фойдаланилади?
3. onCreate () методи нима учун ишлатилади?
4. Activity (дастур ойнаси) натижаси фойдаланувчига кўринганда қайси функция чақирилади?
5. Activity яшаш цикли нима?
6. Activity яратилганда то иш фаолиятини тўхтатгунча тизим томонидан

- қандай методлар ишга туширилади?
7. Берилган код қисми нима вазифани бажаради?

```
public void onClick(View view) {
    startActivity(new Intent("com.example.SecondActivity"));
}
```
 8. Бир activity дан иккинчи activity га қиймат жўнатиш учун қайси методдан фойдаланилади?
 9. Берилган код қисми нима вазифани бажаради?

```
Toast.makeText(this, "Hello World", Toast.LENGTH_SHORT).show();
```
 10. Андроид тизимида фойдаланувчи интерфейсини яратишда қандай layout (қолип) лардан фойдаланилади
 11. Ориентацияни “**Anchoring**” бошқариш усули ва унинг асосий хусусиятини айтинг?
 12. Ориентацияни “**Resizing and repositioning**” бошқариш усули ва унинг асосий хусусиятини айтинг?
 13. List view ларга қайсилар киради ва уларнинг асосий вазифаси нима?
 14. CheckBox қиймати true ёки false эканлигини текшириш учун CheckBox синфининг қандай методдан фойдаланилади?
 15. Gallery нима учун ишлатилади ва Gallery синфининг қандай методлари мавжуд?
 16. Gallery ва ImageView компоненталарини қандай фарқлаш мумкин?
 17. BaseAdapter синфининг қандай абстракт методлари мавжуд?
 18. ListView ва GridView виджетларининг бир-биридан фарқи?
 19. Context меню яратиш учун қандай методлардан фойдаланилади?
 20. Option меню яратиш учун қандай методлардан фойдаланилади?

Фойдаланилган адабиётлар:

1. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-2. Activities, fragments and Intents;
2. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-3. Android User Interface;
3. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-5. Displaying Pictures and Menus with Views.

**4 - мавзу: Мобил иловаларда маълумотлар базаси ва контент
провайдерлардан фойдаланиш. GPS хизмати.
(2 соат)**

Режа:

1. SharedPreferences объекти орқали илова маълумотларини сақлаш.
2. Файлларга маълумот ёзиш ва ўқиш (internal ва external сақловчилар
3. SQLite асослари.
4. Content provider лардан фойдаланиш.
5. Google map лойиҳа яратиш.

Таянч иборалар: *OpenGL ES, Source code, Universal app, GUI, IDE, OS (Operating System), Resources, Service, Surface, SurfaceView, View, Viewgroup Widget*

4.1 SharedPreferences объекти орқали илова маълумотларини сақлаш

Иловалардан фойдаланиш жараёнида иловага тегишли бирор маълумотни вақтинчалик қайердадир сақлашга ва унга қайта мурожаат қилишга тўғри келади. Android тизимида 3 хил кўринишда маълумотларни сақлаш мумкин¹:

- Кичик ўлчамдаги ва тез-тез мурожаат қилиш учун ишлатиладиган *sharedpreferences* объектидан фойдаланиш
- Файллар тизимидан фойдаланиш
- Релацион маълумотлар базасини бошқариш тизими SQLite маълумотлар базасидан фойдаланиш

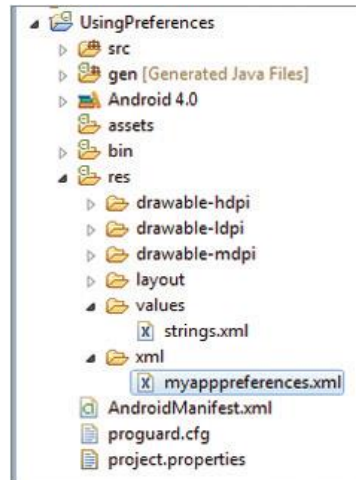
Демак Андроид тизимида иловаларнинг оддий, вақтинчалик маълумотларини сақлаш учун ишлатиладиган SharedPreferences объектидан қандай фойдаланамиз. Масалан, сизнинг иловангизда матн маълумотлари мавжуд ва ушбу матн шрифт ўлчамини жорий ҳолат учун ўзгартирмоқчисиз. Бунда “Settings” менюсига кирамиз ва у ердан “changeTextSize” тугмасини босамиз ва матн шрифт ўлчамини ўзгартириб сақлаш “save” тугмасини босамиз. Бунинг учун SharedPreferences объектининг name/value жуфтлиги орқали илова маълумотлари автоматик тарзда XML файлида сақланди. Илова маълумотларини SharedPreferences объектида сақлаш: UsingPreferences иловаси.

1. Янги Android лойиҳа яратамизг ва уни **UsingPreferences** деб **номлаймиз**.
2. /res папкасида **xml** номли янги папка яратамиз ва унинг ичида

¹ “Beginning Android™ 4 Application Development”. Chapter-6. Data Persistence, pages 279-283.

myapppreferences.xml номли xml файл яратамиз (расмда кўрсатилгандек)

3. myapppreferences.xml файли куйидагича бўлади:



Илованинг **myapppreferences.xml** файли

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen
  xmlns:android="http://schemas.android.com/apk/res/android">
  <PreferenceCategory android:title="Category 1">
    <CheckBoxPreference
      android:title="Checkbox"
      android:defaultValue="false"
      android:summary="True or False"
      android:key="checkboxPref" />
  </PreferenceCategory>
  <PreferenceCategory android:title="Category 2">
    <EditTextPreference
      android:summary="Enter a string"
      android:defaultValue="[Enter a string here]"
      android:title="Edit Text"
      android:key="editTextPref" />
    <RingtonePreference
      android:summary="Select a ringtone"
      android:title="Ringtones"
      android:key="ringtonePref" />
  </PreferenceCategory>
</PreferenceScreen>
  android:title="Second Preference Screen"
  android:summary="Click here Preference Screen"
```

```

        android:key="secondPrefScreenPref" >
    <EditTextPreference
        android:summary="Enter a string"
        android:title="Edit Text (second Screen)"
        android:key="secondEditTextPref" />
    </PreferenceScreen>
</PreferenceCategory>
</PreferenceScreen>

```

4. Package ичида, **AppPreferenceActivity** номли class файл яратилади.

5. **AppPreferenceActivity.java** файли қуйидагича бўлади:

```

package com.example.UsingPreferences;
import android.os.Bundle;
import android.preference.PreferenceActivity;

    public class AppPreferenceActivity extends PreferenceActivity {
        @Override
        public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            //to change preference file name add this code
            addPreferencesFromResource(R.xml.myapppreferences);
        }
    }

```

6. **AndroidManifest.xml** файлида **AppPreferenceActivity** class ни **<activity>** элементи орқали чақириб қўйилади:

```

.....
<activity android:name=".AppPreferenceActivity"
    android:label="@string/app_name">
    <intent-filter>
        <action
            android:name="com.example.AppPreferenceActivity" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
.....

```

7. **main.xml** асосий лайоут файли эса қуйидагича бўлади (изоҳ: ушбу компоненталар

<LinearLayout> ичида жойлаштирилади):

<Button

```
android:id="@+id/btnPreferences"
android:text="Load Preferences Screen"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:onClick="onClickLoad"/>
```

<Button

```
android:id="@+id/btnDisplayValues"
android:text="Display Preferences Values"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:onClick="onClickDisplay"/>
```

<EditText

```
android:id="@+id/txtString"
android:layout_width="fill_parent"
android:layout_height="wrap_content" />
```

<Button

```
android:id="@+id/btnModifyValues"
android:text="Modify Preferences Values"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:onClick="onClickModify"/>
```

8. UsingPreferencesActivity.java файли куйидагича бўлади:

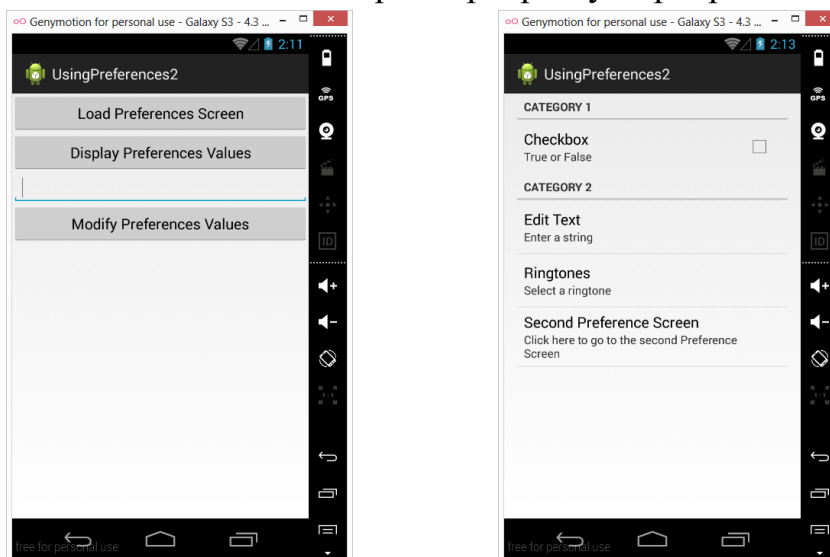
```
package com.example.UsingPreferences;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
public class UsingPreferencesActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
    public void onClickLoad(View view) {
        Intent i = new Intent("com.example.AppPreferenceActivity");
```

```

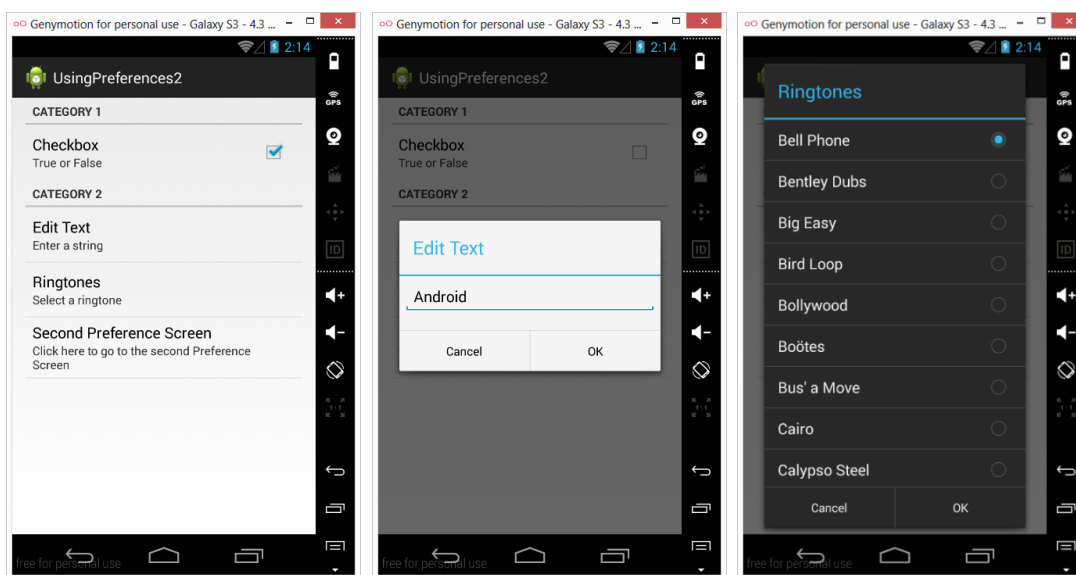
        startActivity(i);
    }
}

```

9. Дастур натижсини Android emulator да кўрамиз. **Load Preferences Screen** тугмасини босамиз ва баъзи параметрларни ўзгартирамиз

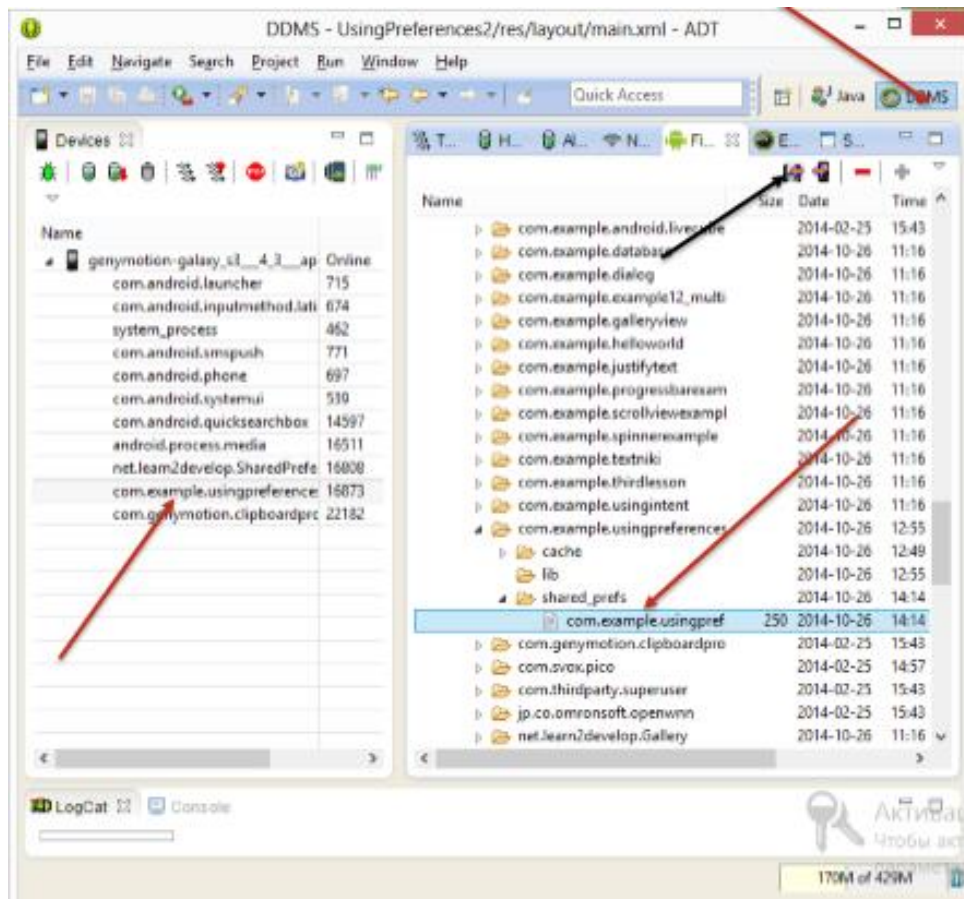


4.1-а-расм. UsingPreferences илова натижаси



4.1-б-расм. UsingPreferences илова натижаси

Демак ушбу ўзгартиришлардан кейин маълумотлар қайерда сақланади.
 DDMS -> Device (genymotion-galaxy-s3)-com.example.usingpreferences
 data -> data -> com.example.usingpreferences->shared_prefs->
 com.example.usingpref.xml

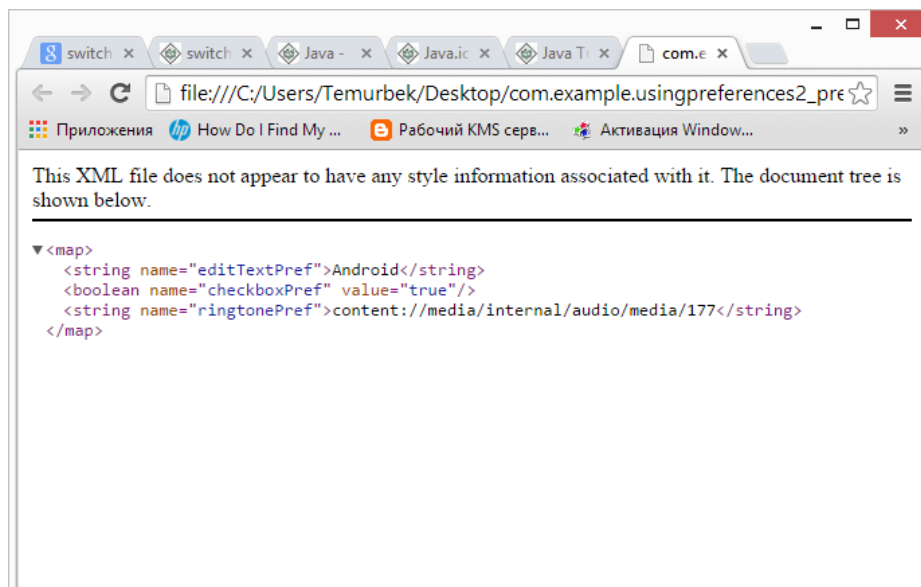


4.2-расм. UsingPreferences илованинг “DDMS” ойнасидаги натижаси

Ушбу *com.example.usingpref.xml* файлни текшириб кўрамиз. Бунинг учун 4.2-расмда кўрсатилган қуйтдаги тугмалардан биринчисини танлаймиз ва илова ҳақидаги сақланган файлни экспорт қиламиз ва браузер орқали натижани текшириб кўрамиз.



Демак кўрсатилган тугма орқали компьютер хотирасига сақланган файлни бирон бир браузер орқали очиб кўрамиз.



4.3-расм. Экспорт қилинган файл натижаси

4.2 Файлларга маълумот ёзиш ва ўқиш (internal ва external сақловчилар)

Файлга маълумот ёзиш учун `FileOutputStream` ва `OutputStreamWriter` синфларидан фойдаланилади ва ўз навбатида файлдан маълумотларни ўқиш учун эса `FileInputStream` ва `InputStreamReader` синфларидан фойдаланилади Илова маълумотларини файлга ёзиш¹

1. Янги **Files** номли Андроид илова яратамиз.
2. Илованинг **main.xml** лойхат файли қуйидагича бўлади:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Please enter some text" />
<EditText
    android:id="@+id/txtText1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
<Button
```

¹ “Beginning Android™ 4 Application Development”. Chapter-6. Data Persistence, Persisting data to Files pages 291-294.

```

    android:id="@+id/btnSave"
    android:text="Save"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:onClick="onClickSave" />
<Button
    android:id="@+id/btnLoad"
    android:text="Load"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:onClick="onClickLoad" />
</LinearLayout>

```

3. **FilesActivity.java** файли қуйидагича бўлади:

```

public class FilesActivity extends Activity {
    EditText textBox;
    static final int READ_BLOCK_SIZE = 100;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        textBox = (EditText) findViewById(R.id.txtText1);
    }
    public void onClickSave(View view) { // файлга ёзиш методи
        String str = textBox.getText().toString();
        try
        {
            FileOutputStream fOut = openFileOutput("textfile.txt",
MODE_WORLD_READABLE);
            OutputStreamWriter osw = new OutputStreamWriter(fOut);
            ///---маълумотларни файлга ёзиш---
            osw.write(str);
            osw.flush();
            osw.close();
            ///---файлга ёзиган маълумотларни экранга чиқариш---
            Toast.makeText(getApplicationContext(), "File saved successfully!",
Toast.LENGTH_SHORT).show();

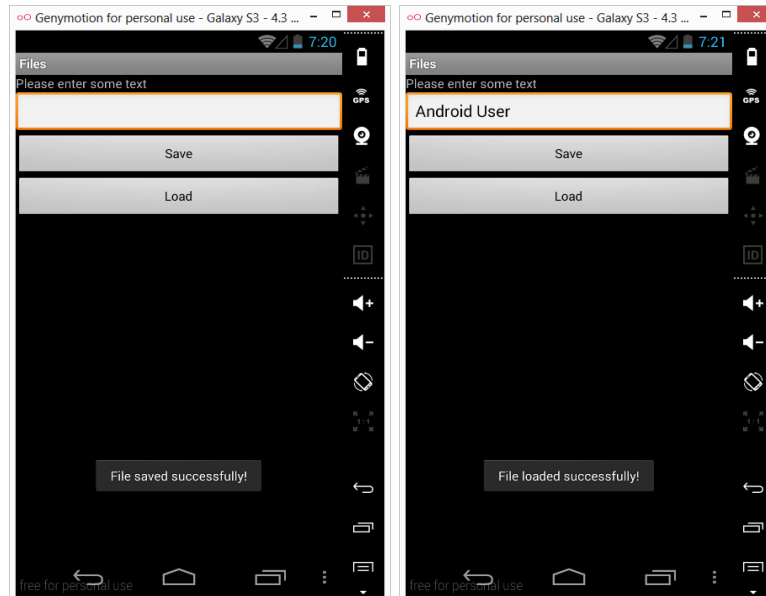
```

```

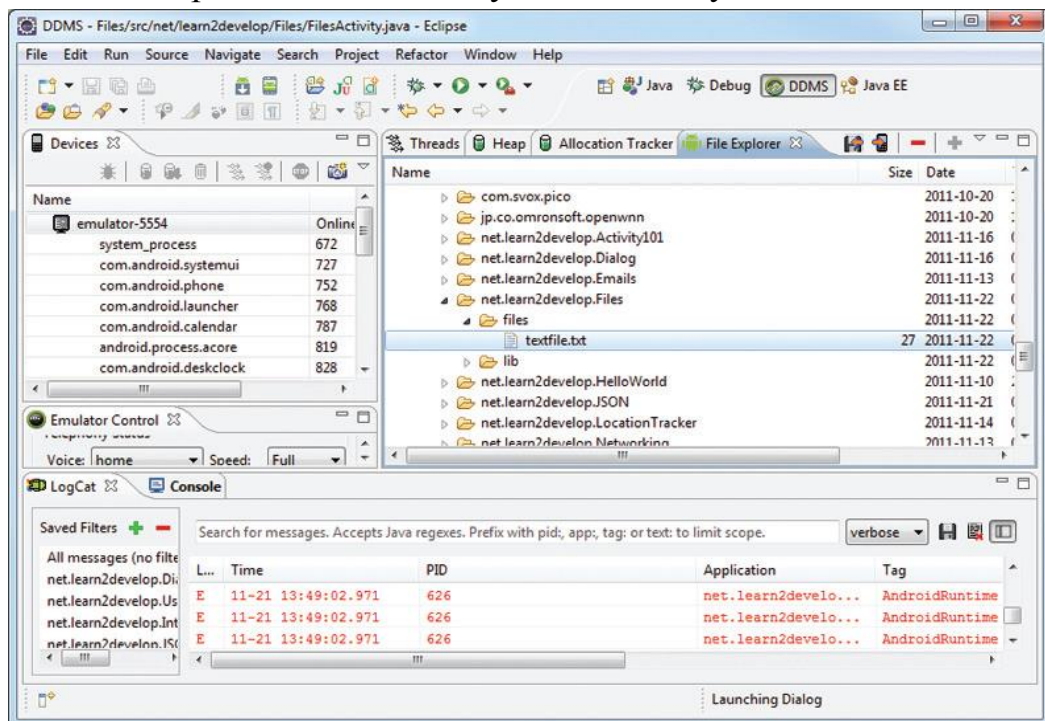
    //---EditText ичидаги ёзувни олиб ташилаш---
    textBox.setText("");
}
catch (IOException ioe)
{
    ioe.printStackTrace();
}
}
public void onClickLoad(View view) { //файлдан ўқиш
    try{
        FileInputStream fIn = openFileInput("textfile.txt");
        InputStreamReader isr = new InputStreamReader(fIn);
        char[] inputBuffer = new char[READ_BLOCK_SIZE];
        String s = "";
        int charRead;
        while ((charRead = isr.read(inputBuffer))>0)
        {
            //---convert the chars to a String---
            String readString = String.copyValueOf(inputBuffer, 0, charRead);
            s += readString;
            inputBuffer = new char[READ_BLOCK_SIZE];
        }
        textBox.setText(s);
        Toast.makeText(getApplicationContext(), "File loaded successfully!",
Toast.LENGTH_SHORT).show();
    }
    catch (IOException ioe) {
        ioe.printStackTrace();
    }
}
}

```

4. Дастур натижаси



4.4-расм. Файлга маълумот ёзиш ва ўқиш иловаси



4.5-расм. Илова маълумотлар файлининг “DDMS” ойнасидаги кўриниши

4.3 Маълумотлар базаси. SQLite.

DBAdapter ёрдамчи синфи

- Андроид тизимида маълумотлар базаси қуйидагича сақланади¹
 - /data/data/<package_name>/databases
- Демак, Андроид иловаларида маълумотлар базаси билан ишлаш учун DBAdapter ёрдамчи синфни яратишимиз лозим
- Ушбу синф SQLite маълумотлар базасини яратиш, очиш, ёпиш ва у билан ишлаш имкониятини яратиб беради

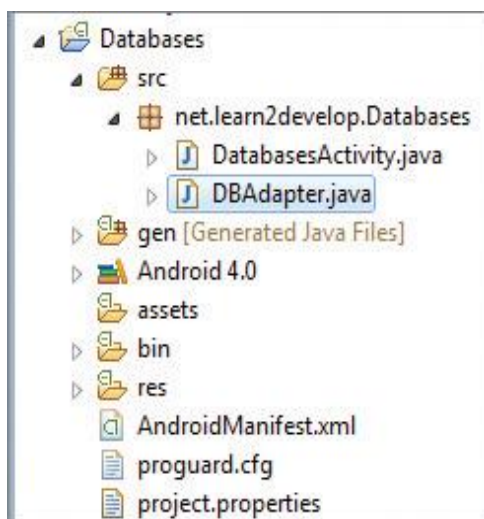
¹ “Beginning Android™ 4 Application Development”. Chapter-6. Data Persistence, pages 301-309.

- Қуйидаги мисолда *MyDB* номли маълумотлар базаси ва *contacts* номли жадвал яратилади
- Жадвал устунлари: *_id*, *name*, ва *email*

| _id | name | email |
|------------|-------------|--------------|
| | | |
| | | |
| | | |

Маълумотлар базаси билан ишлаш

1. Янги “Databases” номли лойиҳа яратамиз
2. Лойиҳага DBAdapter.java номли янги синф қўшамиз ва бу синф илова учун маълумотлар базасини бошқариш учун хизмат қилади
3. Ўз навбатида DBAdapter.java файли қуйидагича ўзгартирилади



DBAdapter.java

```
public class DBAdapter {
    static final String KEY_ROWID = "_id";
    static final String KEY_NAME = "name";
    static final String KEY_EMAIL = "email";
    static final String TAG = "DBAdapter";
    static final String DATABASE_NAME = "MyDB";
    static final String DATABASE_TABLE = "contacts";
    static final int DATABASE_VERSION = 1;
    static final String DATABASE_CREATE =
        "create table contacts (_id integer primary key autoincrement, "
        + "name text not null, email text not null);";
}
```

```

final Context context;
DatabaseHelper DBHelper;
SQLiteDatabase db;
public DBAdapter(Context ctx) {
    this.context = ctx;
    DBHelper = new DatabaseHelper(context);
}
private static class DatabaseHelper extends SQLiteOpenHelper
    DatabaseHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
}
@Override
public void onCreate(SQLiteDatabase db) {
try {
    db.execSQL(DATABASE_CREATE);
 } catch (SQLException e) {
    e.printStackTrace();
 }
 }
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
    db.execSQL("DROP TABLE IF EXISTS contacts");
    onCreate(db);
 }
public DBAdapter open() throws SQLException
 {
    db = DBHelper.getWritableDatabase();
    return this;
 }
public void close()
 {
    DBHelper.close();
 }
public long insertContact(String name, String email)
 {
    ContentValues initialValues = new ContentValues();
    initialValues.put(KEY_NAME, name);
    initialValues.put(KEY_EMAIL, email);

```

```

    return db.insert(DATABASE_TABLE, null, initialValues);
}
public boolean deleteContact(long rowId)
{
return db.delete(DATABASE_TABLE, KEY_ROWID + "=" + rowId, null) > 0;
}
public Cursor getAllContacts() {
    return db.query(DATABASE_TABLE, new String[] {KEY_ROWID,
KEY_NAME, KEY_EMAIL}, null, null, null, null, null);
}
public Cursor getContact(long rowId) throws SQLException {
    Cursor mCursor = db.query(true, DATABASE_TABLE, new String[]
{KEY_ROWID, KEY_NAME, KEY_EMAIL}, KEY_ROWID + "=" + rowId, null,
    null, null, null, null);
    if (mCursor != null) {
        mCursor.moveToFirst();
    }
    return mCursor;
}
public boolean updateContact(long rowId, String name, String email) {
    ContentValues args = new ContentValues();
    args.put(KEY_NAME, name);
    args.put(KEY_EMAIL, email);
    return db.update(DATABASE_TABLE, args, KEY_ROWID + "=" +
rowId, null) > 0;
}
}

```

MainActivity.java

```

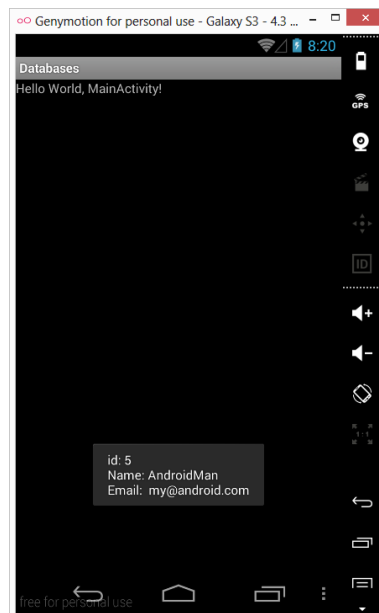
public class MainActivity extends Activity {
    public void onCreate{
        DBAdapter db = new DBAdapter(this);
        db.open();
        long id = db.insertContact("AndroidMan", "my@android.com");
        id = db.insertContact("iOSMan", "my@ios.com");
        db.close();
        db.open();
        Cursor c = db.getAllContacts();
        if (c.moveToFirst()) {
            do {

```

```

        DisplayContact(c);
    } while (c.moveToNext());
}
db.close();
}
public void DisplayContact(Cursor c){
    Toast.makeText(this,
        "id: " + c.getString(0) + "\n" +
        "Name: " + c.getString(1) + "\n" +
        "Email: " + c.getString(2),
        Toast.LENGTH_LONG).show();
}
}
}

```



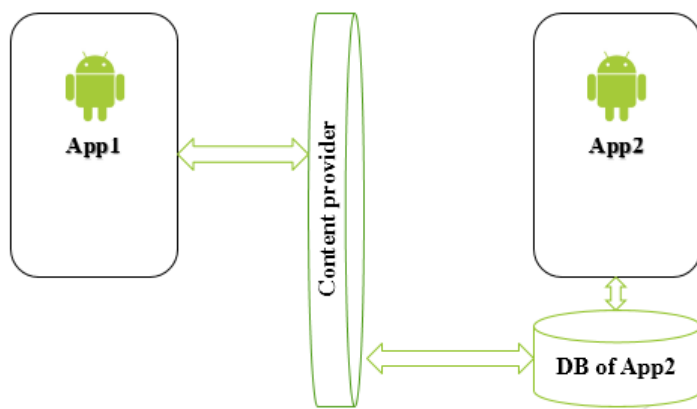
4.6-расм. Илова маълумотларини базадан ўқиб олиш

4.4. Content provider лардан фойдаланиш

Content providers – бу пакетлараро маълумот алмашаш учун ишлатиладиган провайдер ҳисобланади. Android тизимида, пакетлараро маълумот алмашиш учун content provider лардан фойдаланиш қулай ҳисоланади. Content provider лар оддий маълумотлар базасига ўхшаш бўлиб, контент учун сўрав яратиш, контент маълумотларини ўзгартириш, ва янги контент қўйиш ёки уни ўчириш мумкин¹. Лекин, оддий маълумотлар базасидан фарқли равишда, контент маълумотлари МБ да (жадвал асосида),

¹ “Beginning Android™ 4 Application Development”. Chapter-7. Content Providers, pages 321-329.

файлда, ёки тармоқда сақланиши мумкин. Content provider дан фойдаланиш структураси куйидаги расмда кўрсатилган (4.6-расм).



4.6-расм. Иловаларда Content provider дан фойдаланиш структураси

Android тизимида пакетлараро маълумотлар алмашиш. Android тизимида мавжуд content provider ларга куйидагиларни мисол қилиш мумкин:

- **Browser** — маълумотларни browser bookmark, browser history, ва ҳ.к. ларда сақлаш
- **CallLog** — маълумотларни “missed calls”, “call details” ва ҳ.к. лар сифатида сақлаш
- **Contacts** — contact маълумотларини сақлаш
- **MediaStore** — media файлларни audio, video ва image сифатидасақлаш
- **Settings** — қурилма маълумотларини settings ва preferences ларда сақлаш

Булардан ташқари, content provider ларни фойдаланувчи томонидан яратиш ҳам мумкин.

Мавжуд базадан content provider орқали маълумот ўқиш

- Content provider орқали маълумот ўқиш сўрови (query URI) куйидагича ва бу сўров URI деб аталади:
 - `<standard_prefix>://<authority>/<data_path>/<id>`
- **standard prefix** - content provider учун бу хусусият content:// билан бошланади.
- **authority** - content provider номи ҳисобланиб, одатда илова пакети номи билан бир хил бўлади. Мисол учун: com.wroх.provider ёки net.learn2develop.provider.
- **data path** – чақирилаётган маълумот жойлашган жойи. Масалан Contacts content provider орқали барча контактга тегишли бўлган маълумотларни олиш учун ушбу маълумот жойлашган ўрни (URI)

қуйидагича бўлиши мумкин:

- **content://contacts/people.**
- **id** - бу контент провайдердан сўралаётган маълумот (record) id си.
Мисол учун contact дан фойдаланувчи id = 2 га тенг:
 - **content://contacts/people/2.**

Content provider лар учун сўровларга мисоллар

| QUERY STRING | Таснифи |
|---------------------------------|--|
| content://media/internal/images | Қурилмадаги локал жойлашган расмлар (internal images) рўйхатини қайтаради |
| content://media/external/images | Қурилмадаги глобал жойлашган расмлар (external storage, SD card) рўйхатини қайтаради |
| content://call_log/calls | Call Log даги қўнғироқлар рўйхатини қайтаради (calls registered) |
| content://browser/bookmarks | Browser даги bookmarks рўйхатини қайтаради |

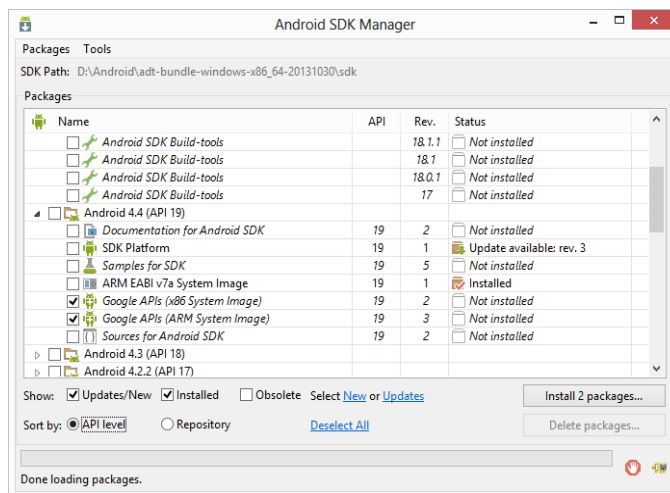
4.5. Google map лойиҳа яратиш

Android иловаларида location – based services яъни фойдаланувчи жойлашган жойига қараб хизмат кўрсатиш имконияти мавжуд бўлиб, бу хизмат LBS деб аталади. LBS хизмати сизнинг Android иловангизда map (ер картаси) ни кўрсатиб бериш имкониятини яратади. Бу хизматни амалга ошириш учун Android SDK да махсус LocationManager синфи мавжуд ва у географик жойлашув бўйича ахборотларни олиш учун имконият яратади¹.

Андроид тизимида ер картасидан фойдаланувчи илова яратишдан олдин мавжуд Андроид OT да Google APIs ўрнатилганми ёки йўқми шуни текшириб кўриш лозим бўлади.

Бунинг учун Android SDK Manager ишга туширилади ва ушбу расмда кўрсатилган ойнадан Андроид талқини бўйича Google APIs ўрнатилганлигини текшириб кўрамиз, агар ўрнатилмаган бўлса, уни белгилаб “Install package” тугмасини босамиз.

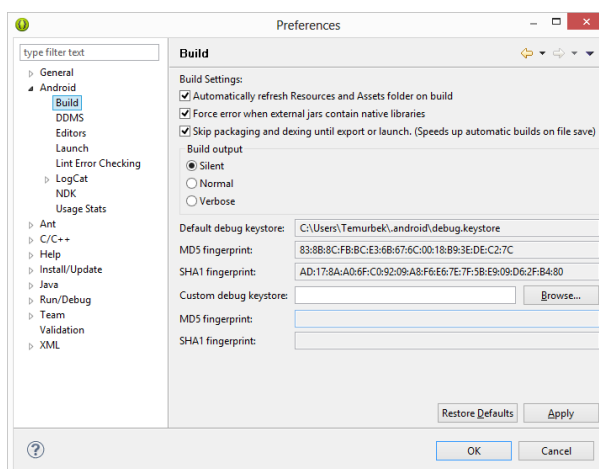
¹ “Beginning Android™ 4 Application Development”. Chapter-9. Location-Based Services, pages 379-385.



4.7-расм. Android SDK Manager ойнаси

LBS хизматидан фойдаланиш учун лойиха яратиш

1. Лойиха номи киритилгандан кейин “Compile With” бўлимидан Google APIs X86 яъни жорий Андроид OT га мос Google API ни танлаймиз
2. Лойиха яратилгандан кейин “explorer project” бўйича Google APIs папкасида махсус JAR файл (maps.jar) мавжудлигини текшириб кўрамиз
3. Лойиха яратилгандан кейин бепул Google Maps API калитини жорий иловага ўрнатиш лозим.
4. Албатта калит ўрнатилишида “Google’s terms of use” билан танишиб чиқилади ва зарур бўлган ҳолда қабул қилиш тугмаси босилади
5. Андроид иловаларининг калитлари ва улар ҳақидаги маълумотлар **debug.keystore** файлида сақланади ва у қуйидаги кўрсатилган расм бўйича жойлашади
6. Eclipse дастурининг Window ⇔ Preferences ойнасига кирилади



4.8-расм. Google map хизматидан фойдаланиш учун илова калитлари MD5 ва SHA1 fingerprint коддини генерация қилиш

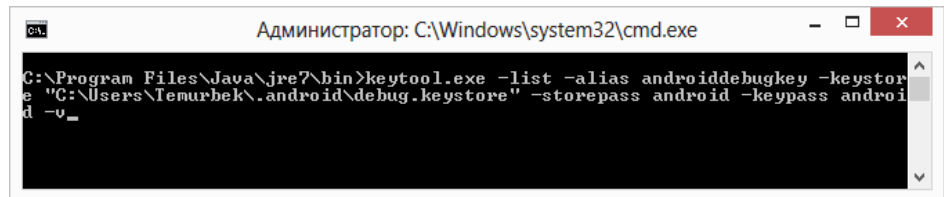
MD5 ва SHA1 fingerprint кодини генерация қилиш учун **Keytool.exe** дан фойдаланилади ва қуйидагича жойлашади

```
c:\Program Files\Java\jre7\bin\
```

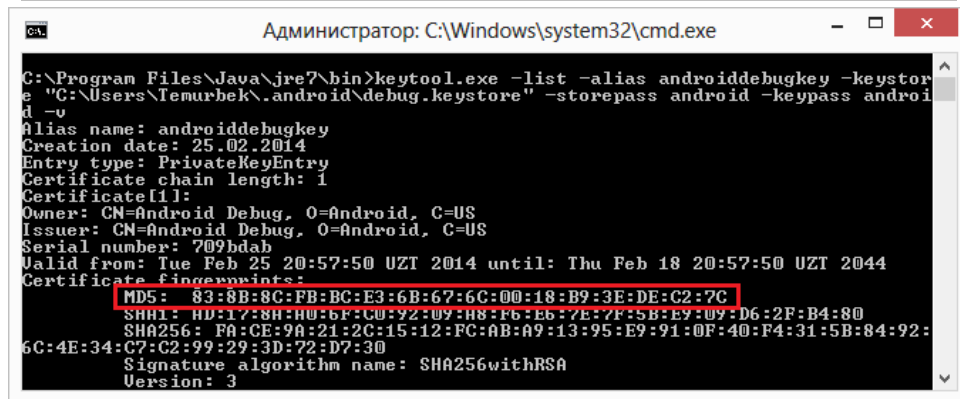
Демак қуйидаги буйруқни киритиш орқали биз юқоридаги кодларни генерация қилиб олишимиз мумкин бўлади

```
keytool.exe -list -alias androiddebugkey -keystore
```

```
"C:\Users\<username>\.android\debug.keystore" -storepass android -  
keypass android -v
```



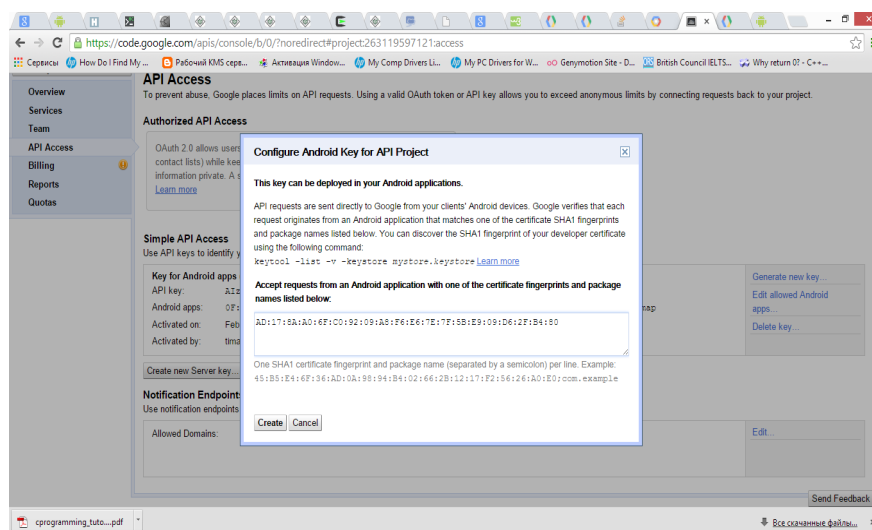
```
Администратор: C:\Windows\system32\cmd.exe
C:\Program Files\Java\jre7\bin>keytool.exe -list -alias androiddebugkey -keystore
"C:\Users\Temurbek\.android\debug.keystore" -storepass android -keypass androi
d -v
```



```
Администратор: C:\Windows\system32\cmd.exe
C:\Program Files\Java\jre7\bin>keytool.exe -list -alias androiddebugkey -keystore
"C:\Users\Temurbek\.android\debug.keystore" -storepass android -keypass androi
d -v
Alias name: androiddebugkey
Creation date: 25.02.2014
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 709bdab
Valid from: Tue Feb 25 20:57:50 UZT 2014 until: Thu Feb 18 20:57:50 UZT 2044
Certificate fingerprints:
MD5: 83:8B:8C:FB:BC:E3:6B:67:6C:00:18:B9:3E:DE:C2:7C
SHA1: HD:17:88:AD:6F:C0:72:09:A8:F6:E6:7E:7F:5B:E9:09:D6:2F:B4:80
SHA256: FA:CE:9A:21:2C:15:12:FC:AB:A9:13:95:E9:91:0F:40:F4:31:5B:84:92:
6C:4E:34:C7:C2:99:29:3D:72:D7:30
Signature algorithm name: SHA256withRSA
Version: 3
```

4.9-расм. MD5 ва SHA1 fingerprint кодини генерация қилиш

Лойиҳа учун Google API калитини олиш учун қуйидаги ссилкага кирамиз <https://code.google.com/apis/console>



4.10-расм. Лойиҳа учун Google API калитини олиш

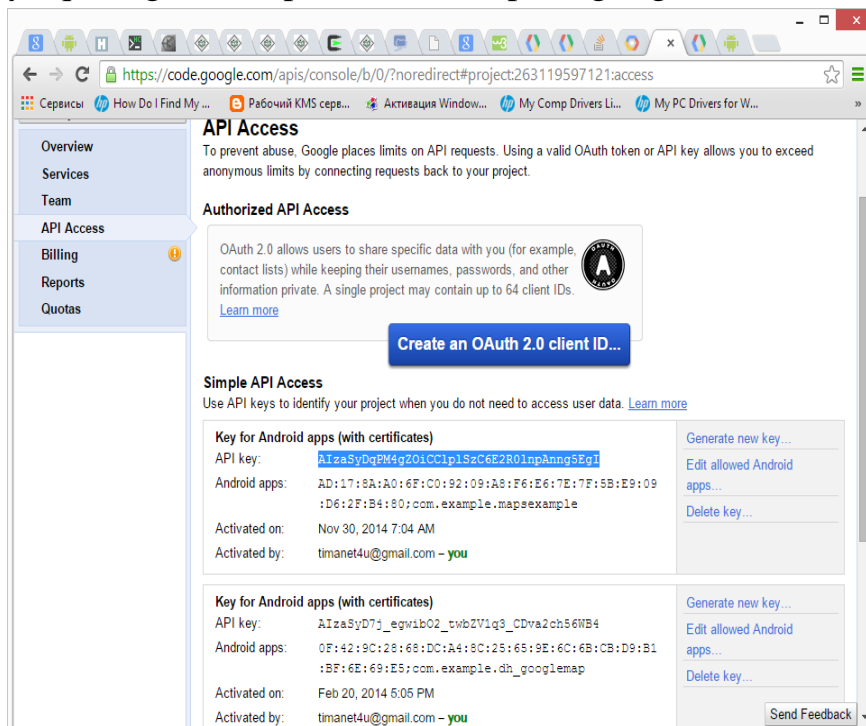
SHA1 fingerprint коди ва “;” дан кейин яратилган лойҳанинг package номи киритилади ва “Create” тугмаси босилади

Мисол учун менинг SHA1 кодим қуйидагича

AD:17:8A:A0:6F:C0:92:09:A8:F6:E6:7E:7F:5B:E9:09:D6:2F:B4:80

Жорий лойиҳа учун Google API калити эса қуйидагича

AIzaSyDqPM4gZOiCC1p1SzC6E2R0lnpAnng5EgI



4.11-расм. Лойиҳа учун Google API калити

Лойиҳада Map ни ҳосил қилиш

- Янги LBS номли Андроид лойиҳа яратамиз
- Яратилган лойиҳанинг AndroidManifest.xml файлига интернет учун рухсат берамиз
- `<uses-permission android:name="android.permission.INTERNET"/>`
- **main.xml** файлидаги TextView элементини қуйидаги код қисми билан алмаштираемиз.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
<com.google.android.maps.MapView
    android:id="@+id/mapView"
    android:layout_width="fill_parent"
```

```
    android:layout_height="fill_parent"
    android:enabled="true"
    android:clickable="true"
    android:apiKey="AIzaSyDqPM4gZOiCClplSzC6E2R0lnpAnng5EgI" />
</LinearLayout>
```

Лойиханинг AndroidManifest.xml файли қуйидагича бўлади:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.learn2develop.LBS"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="14" />
    <uses-permission android:name="android.permission.INTERNET"/>
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <uses-library android:name="com.google.android.maps" />
        <activity
            android:label="@string/app_name"
            android:name=".LBSActivity" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

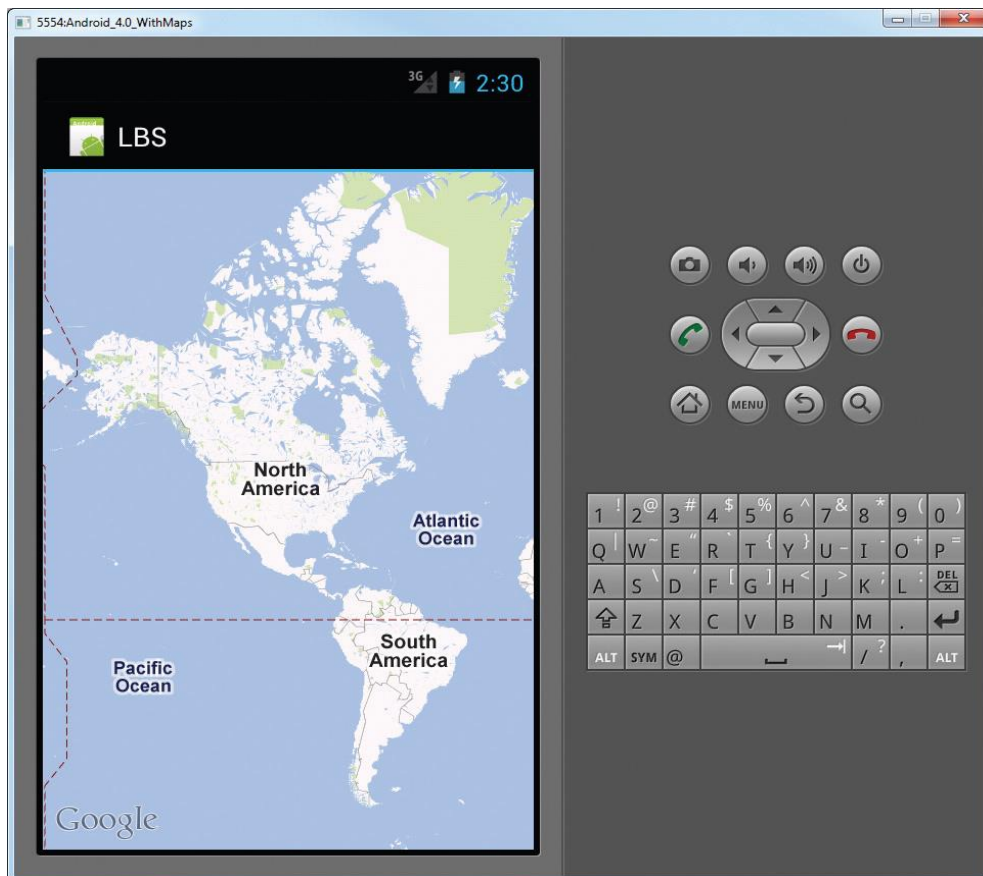
LBSActivity.java файли эса қуйидагичабўлади. **LBSActivity** синфи **MapActivity** синфидан ворис олиб яратилади.

```
package net.learn2develop.LBS;
import com.google.android.maps.MapActivity;
import android.os.Bundle;
public class LBSActivity extends MapActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.main);
}
@Override
protected boolean isRouteDisplayed() {
return false;
}
}

```



4.12-расм. Иловада Google map дан фойдаланиш

Назорат саволлари:

1. Андроид иловалариа маълумотлар базасини ташкил қилишнинг қандай турлари мавжуд?
2. SQLite нима?
3. Sharedpreferences объекти орқали сақланган маълумотлар қандай кўринишда сақланади?
4. Sharedpreferences объекти орқали илованинг қандай маълумотлари сақланади?

5. Илова маълумотларини файлга ёзишда қандай оқимли синфлардан фойдаланилади?
6. Илова маълумотларини файлдан ўқиш олишда қандай оқимли синфлардан фойдаланилади?
7. Андроид тизимида internal ва external сақловчилар деганда нимани тушунасиз?
8. SQLiteOpenHelper синфининг асосий вазифаси?
9. SQLiteDatabase синфининг асосий вазифаси?
10. SQLiteDatabase синфининг қандай методлари мавжуд?
11. *Content providers* нима ва у қачон ишлатилади?
12. Android тизимида мавжуд content provider ларга мисол келтиринг?
13. Content provider орқали маълумот ўқиш сўрови стандарт кўриниши қандай?
14. Content provider лар учун сўровларга мисоллар келтиринг?

Фойдаланилган адабиётлар:

1. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-6. Data Persistence
2. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-7. Content Providers
3. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-9. Location based services
4. <http://www.tutorialspoint.com/sqlite/>
5. http://www.tutorialspoint.com/sqlite/sqlite_installation.htm
6. http://www.tutorialspoint.com/sqlite/sqlite_create_database.htm
7. http://www.tutorialspoint.com/android/android_internal_storage.htm

**5 - мавзу: Мобил иловалар яратишда тармоқли дастурлаш. Илованинг сервер қисми билан ишлаш. JSON хизматидан фойдаланиш .
(2 соат)**

Режа:

1. Мобил иловаларда тармоқли дастурлаш.
2. HTTP орқали серверга уланиш.
3. JSON хизматидан фойдаланиш.

Таянч сўзлар: *destinationAddress, scAddress, sentIntent, deliveryIntent, Dialog, Drawable, Broadcast Receiver, Layout Resource, Manifest File*

5.1 Мобил иловаларда тармоқли дастурлаш

Ҳозирги кунда интернет тармоғи барча соҳаларда ривожланиб улгурган. Демак биз мобил курилмалардан фойдаланиш жараёнида фақат **SMS/MMS** ёки электрон почта хабарлари эмас балки интернет хизматларидан фойдаланишимиз мумкин. Бунинг учун иловаларни яратишда тармоқда дастурлаш, яъни, махсус протоколлар (**HTTP**), махсус методлар (**GET, POST**), маълумотларни **юклаб олиш** ва бошқа **хизматлар (JSON)** дан фойдаланиш кўникмаларига эга бўлишимиз лозим¹.

Демак тармоқ билан ишлаш жараёнида биринчи бўлиб, тармоқ ҳолатини яъни уланганлик ҳолатини текшириб кўрамиз. Андроид тизимида бунинг учун **ConnectivityManager** синфидан фойдаланилади. Ушбу синф орқали яратилган объект **getSystemService()** методини чақириш орқали инициализация қилинади. Ушбу ҳолат бўйича қуйида дастур коди келтирилган:

```
ConnectivityManager check = (ConnectivityManager)  
this.context.getSystemService(Context.CONNECTIVITY_SERVICE);
```

Бундан кейин биз тармоқ ҳақидаги маълумотни тўлиқ олишимиз керак бўлади. Бунинг учун **getAllNetworkInfo** методидан фойдаланилади. Ушбу методи **NetworkInfo** типдаги массив қайтаради ва қуйидагича фойдаланилади:

```
NetworkInfo[] info = check.getAllNetworkInfo();
```

Энди ушбу **info** номли массивга келган қиймат орқали илова тармоққа уланганлигини текшириш учун **getState** методидан фойдаланилади:

¹ “Beginning Android™ 4 Application Development”. Chapter-10. Networking, pages 421-423.

```

for (int i = 0; i<info.length; i++){
    if (info[i].getState() == NetworkInfo.State.CONNECTED){
        Toast.makeText(context, "Internet is connected
        Toast.LENGTH_SHORT).show();
    }
}

```

Бундан ташқари тармоқ ҳолатини текшириш учун қуйидаги констатнталардан фойдаланилади:

| № | State |
|---|---------------|
| 1 | CONNECTING |
| 2 | DISCONNECTED |
| 3 | DISCONNECTING |
| 4 | SUSPENDED |
| 5 | UNKNOWN |

5.2 HTTP орқали серверга уланиш

Интернет борлигини текширганимиздан кейин ушбу тармоқ орқали турли хил операциялар амалга оширишимиз мумкин. Мисол учун интернет орқали матн ёки binary (мисол учун расм) маълумотларни юклаб олиш ва ҳ.к.лар.

Android тизимида ушбу операцияларни амалга ошириш учун **HttpURLConnection** ва **URL** синфларидан фойдаланилади. Қуйидаг код қисми орқали кўрсатилган интернет манзил учун url объекти яратилади:

```

String link = "http://www.google.com";
URL url = new URL(link);

```

url яратилгандан кейин HttpURLConnection синфига жўнатилади ва HttpURLConnection объекти кўрсатилган интернет манзили билан боғланади. Бу боғланишни эса HttpURLConnection синфининг **connect** методидан фойдаланиб қуйидагича амалга оширилади.

```

HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.connect();

```

Демак ушбу боғланиш орқали мазкур саҳифа HTML ҳужжати коддини ўқиб оламиз. Маълумотларни ўқиб олиш учун эса **InputStream** ва **BufferedReader** синфларидан фойдаланилади. Қуйида мисол келтирилган:

```

InputStream is = conn.getInputStream();

```

```
BufferedReader reader =new BufferedReader(new InputStreamReader(is, "UTF-8"));
String webPage = "",data="";
while ((data = reader.readLine()) != null){
    webPage += data + "\n";
}
```

URLConnection синфининг куйидагича методлари мавжуд. Куйидаги жадвалда connect() методидан бошқа методлар ҳам келтирилган:

| № | Метод |
|---|--|
| 1 | disconnect() – интернет боғланишни тугатиш учун ишлатилади |
| 2 | getRequestMethod() – ушбу метод жорий ишлаб турган сервернинг HTTP методини қайтаради |
| 3 | getResponseCode() – масофадаги HTTP сервер “response code” қайтаради |
| 4 | setRequestMethod (String method) - HTTP сервер томонидан сўралган методни ўрнатади |
| 5 | usingProxy() - проху server мавжуд ёки йўқлигини қайтаради |

Интернет боғланишни текшириш

1. Янги Android лойиҳа яратамиз ва уни **Networking** деб номлаймиз.
2. Илованинг **AndroidManifest.xml** файлига куйидаги “permission” қўшиб кўйилади:

```
<uses-permission
    android:name="android.permission.INTERNET"/>
```

3. **MainAcivity.java** файли мос ҳолда ўзгартирилади
4. **activity_main.xml** файли ҳам мос ҳолда ўзгартирилади

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
<Button
    android:text="checkConnection"
```



```
    android:id="@+id/Button01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>
```

MainActivity.java

```
//Package name here
import android.app.Activity;
import android.os.Bundle;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;
import android.util.Log;
public class MainActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void checkInternetConnection(){
        ConnectivityManager check = (ConnectivityManager) this.context.
        getSystemService(Context.CONNECTIVITY_SERVICE);
        if (check != null)
        {
            NetworkInfo[] info = check.getAllNetworkInfo();
            if (info != null)
                for (int i = 0; i <info.length; i++)
                    if (info[i].getState() == NetworkInfo.State.CONNECTED)
                    {
                        Toast.makeText(context, "Internet is connected",
                        Toast.LENGTH_SHORT).show();
                    }
        }
    }
    else{
        Toast.makeText(context, "not conencted to internet",
        Toast.LENGTH_SHORT).show();
    }
}
```

```
}  
}  
}
```

Binary маълумотларни юклаб олиш¹

1. Янги DownloadData номли Андроид лойиҳа яратамиз
2. Лойиҳанинг main.xml файлини қуйидагича ўзгартирамиз

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:orientation="vertical" >  
<ImageView  
    android:id="@+id/img"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center" />  
</LinearLayout>
```

3. **AndroidManifest.xml** файлига қуйидаги permission қўшиб қўйилади
<uses-permission android:name="android.permission.INTERNET"/>

4. MainActivity.java

```
// Package номи лойиҳангизга мос қолдириш  
// Керакли синфларни import қилиш  
public class MainActivity extends Activity {  
    ImageView img;  
    private InputStream OpenHttpConnection(String urlString) throws  
IOException{  
    InputStream in = null; int response = -1;  
    URL url = new URL(urlString);  
    URLConnection conn = url.openConnection();  
    if (!(conn instanceof HttpURLConnection))  
        throw new IOException("Not an HTTP connection");  
    try{  
        HttpURLConnection httpConn = (HttpURLConnection) conn;  
        httpConn.setAllowUserInteraction(false);  
        httpConn.setInstanceFollowRedirects(true);
```

¹ "Beginning Android™ 4 Application Development". Chapter-10. Networking, pages 424-426.

```

        httpConn.setRequestMethod("GET");
        httpConn.connect();
        response = httpConn.getResponseCode();
        if (response == HttpURLConnection.HTTP_OK) {
            in = httpConn.getInputStream();
        }
    }
    catch (Exception ex){
        Log.d("Networking", ex.getLocalizedMessage());
        throw new IOException("Error connecting");}
    return in;
}
private Bitmap DownloadImage(String URL)
{
    Bitmap bitmap = null;
    InputStream in = null;
    try {
        in = OpenHttpConnection(URL);
        bitmap = BitmapFactory.decodeStream(in);
        in.close();
    } catch (IOException e1) {
        Log.d("NetworkingActivity", e1.getLocalizedMessage());
    }
    return bitmap;
}
private class DownloadImageTask extends AsyncTask<String, Void, Bitmap>
{
    protected Bitmap doInBackground(String... urls) {
        return DownloadImage(urls[0]);
    }
    protected void onPostExecute(Bitmap result) {
        ImageView img = (ImageView) findViewById(R.id.img);
        img.setImageBitmap(result);
    }
}
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
}

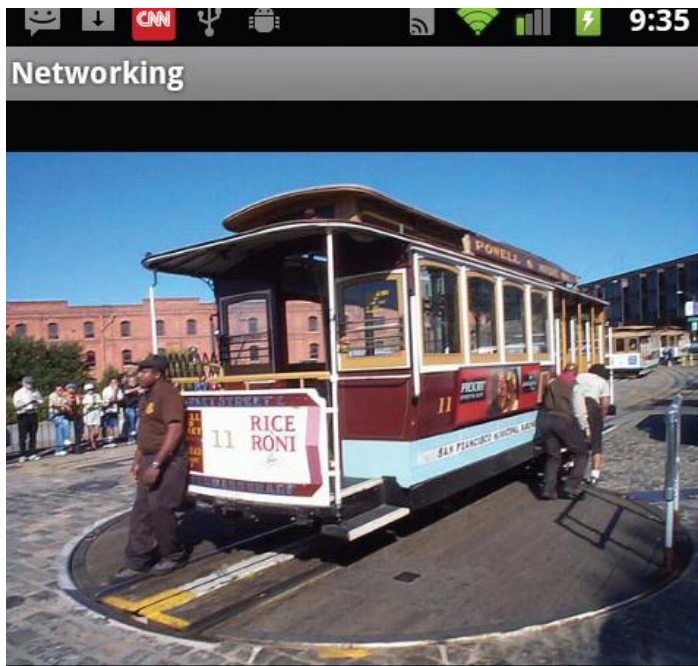
```

```

new DownloadImageTask().execute(
    "http://www.mayoff.com/5-01cablecarDCP01934.jpg");
}
} // MainActivity синфи

```

Изоҳ: Дастур натижаси сифатида кўрсатилган манзил бўйича расм юклар олинадиган ва иловани <ImageView> га расм юкланиши



5.1-расм. Расм интернет орқали юклар олиш иловаси

5.3. JSON хизматидан фойдаланиш

JSON – инглизча **JavaScript Object Notation** сўзларининг бош ҳарфларидан олинган бўлиб, тармоқ орқали маълумот ўқиш ва ёзиш учун қулай ва тез хизмат ҳисобланади. **JSON** хизматининг қуйидаги синфлари мавжуд¹

- **JSONArray**
- **JSONObject**
- **JSONStringer**
- **JSONTokenizer**

JSON компонентлари

- **Array()** - JSON data массиви (объектлар бирлашмаси)
- **Objects()** - JSON объекти
- **Key** – объектнинг калити

¹ http://www.tutorialspoint.com/android/android_json_parser.htm
 “Beginning Android™ 4 Application Development”. Chapter-10. Networking, pages 437-443.

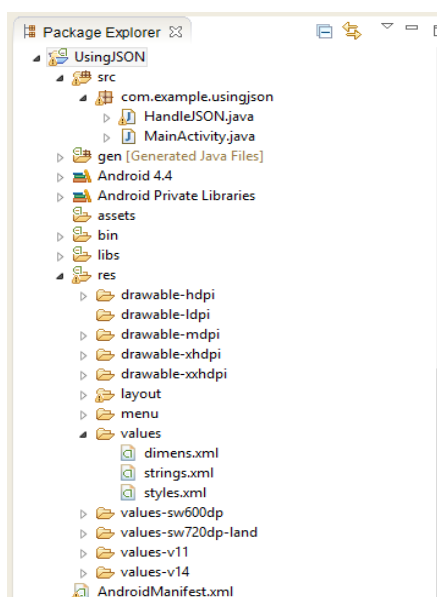
- **Value** – объект қиймати



5.2-расм. JSON маълумот кўриниши

JSON дан фойдаланиш

1. Янги Андроид лойиҳа яратамиз ва уни UsingJSON деб номлаймиз
2. Лойиҳа қуйидаги файллардан иборат ва уларни қуйидаги слайдларда кўрсатилганидек ўзгартирамиз
3. res/layout/activity_main.xml
4. res/values/string.xml
5. AndroidManifest.xml
6. src/package/MainActivity.java
7. src/package/HandleJSON.java



res/layout/activity_main.xml

```
RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="15dp"
        android:text="@string/location"
        android:textAppearance="?android:attr/textAppearanceMedium" />
    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/textView1"
        android:layout_alignParentRight="true"
        android:ems="10" />
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView1"
        android:layout_below="@+id/textView1"
        android:layout_marginTop="68dp"
        android:text="@string/country"
        android:textAppearance="?android:attr/textAppearanceSmall" />
    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:layout_below="@+id/textView2"
        android:layout_marginTop="19dp"
        android:text="@string/temperature"
        android:textAppearance="?android:attr/textAppearanceSmall" />
<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView3"
    android:layout_below="@+id/textView3"
    android:layout_marginTop="32dp"
    android:text="@string/humidity"
    android:textAppearance="?android:attr/textAppearanceSmall" />
<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView4"
    android:layout_below="@+id/textView4"
    android:layout_marginTop="21dp"
    android:text="@string/pressure"
    android:textAppearance="?android:attr/textAppearanceSmall" />
<EditText
    android:id="@+id/editText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/textView3"
    android:layout_toRightOf="@+id/textView3"
    android:ems="10" >
    <requestFocus />
</EditText>
<EditText
    android:id="@+id/editText3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/textView3"
    android:layout_alignBottom="@+id/textView3"
    android:layout_alignLeft="@+id/editText2"

```

```
android:ems="10" />
```

```
<EditText
```

```
android:id="@+id/editText4"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:layout_above="@+id/textView5"
```

```
android:layout_alignLeft="@+id/editText1"
```

```
android:ems="10" />
```

```
<EditText
```

```
android:id="@+id/editText5"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:layout_alignBaseline="@+id/textView5"
```

```
android:layout_alignBottom="@+id/textView5"
```

```
android:layout_alignRight="@+id/editText4"
```

```
android:ems="10" />
```

```
<Button
```

```
android:id="@+id/button1"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

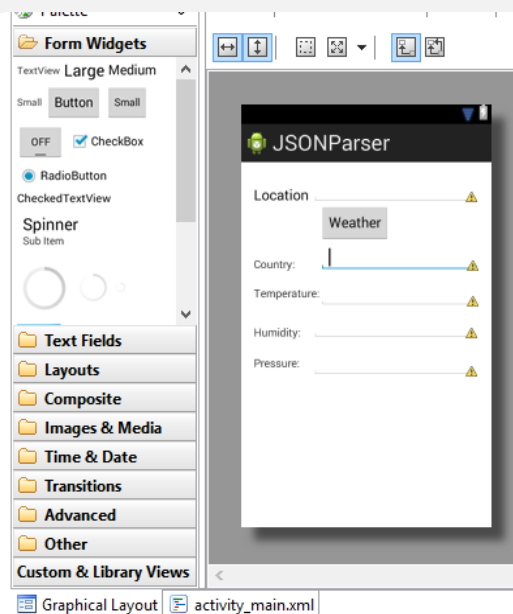
```
android:layout_alignLeft="@+id/editText2"
```

```
android:layout_below="@+id/editText1"
```

```
android:onClick="open"
```

```
android:text="@string/weather" />
```

```
</RelativeLayout>
```



res/values/string.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">JSONParser</string>
  <string name="action_settings">Settings</string>
  <string name="hello_world">Hello world!</string>
  <string name="location">Location</string>
  <string name="country">Country:</string>
  <string name="temperature">Temperature:</string>
  <string name="humidity">Humidity:</string>
  <string name="pressure">Pressure:</string>
  <string name="weather">Weather</string>
</resources>
```

src/package/MainActivity.java

```
public class MainActivity extends Activity {
  private String url1 =
  "http://api.openweathermap.org/data/2.5/weather?q=";
  private EditText location,country,temperature,humidity,pressure;
  private HandleJSON obj;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    location = (EditText)findViewById(R.id.editText1);
    country = (EditText)findViewById(R.id.editText2);
    temperature = (EditText)findViewById(R.id.editText3);
    humidity = (EditText)findViewById(R.id.editText4);
    pressure = (EditText)findViewById(R.id.editText5);
  }
  public void open(View view){
    String url = location.getText().toString();
    String finalUrl = url1 + url;
    country.setText(finalUrl);
    obj = new HandleJSON(finalUrl);
    obj.fetchJSON();
    while(obj.parsingComplete);
  }
}
```

```

country.setText(obj.getCountry());
temperature.setText(obj.getTemperature());
humidity.setText(obj.getHumidity());
pressure.setText(obj.getPressure());
}
}

```

src/package/HandleJSON.java

```

public class HandleJSON {
    private String country = "country";
    private String temperature = "temperature";
    private String humidity = "humidity";
    private String pressure = "pressure";
    private String urlString = null;
    public volatile boolean parsingComplete = true;
    public HandleJSON(String url){
        this.urlString = url;
    }
    public String getCountry(){
        return country;
    }
    public String getTemperature(){
        return temperature;
    }
    public String getHumidity(){
        return humidity;
    }
    public String getPressure(){
        return pressure;
    }
    @SuppressWarnings("NewApi")
    public void readAndParseJSON(String in) {
        try {
            JSONObject reader = new JSONObject(in);
            JSONObject sys = reader.getJSONObject("sys");
            country = sys.getString("country");
            JSONObject main = reader.getJSONObject("main");
            temperature = main.getString("temp");

```

```

    pressure = main.getString("pressure");
    humidity = main.getString("humidity");
    parsingComplete = false;
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}

public void fetchJSON(){
    Thread thread = new Thread(new Runnable(){
        @Override
        public void run() {
            try {
                URL url = new URL(urlString);
                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
                conn.setReadTimeout(10000 /* milliseconds */);
                conn.setConnectTimeout(15000 /* milliseconds */);
                conn.setRequestMethod("GET");
                conn.setDoInput(true);
                // Starts the query
                conn.connect();

                InputStream stream = conn.getInputStream();
                String data = convertStreamToString(stream);
                readAndParseJSON(data);
                stream.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
    thread.start();
}

static String convertStreamToString(java.io.InputStream is) {
    java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");
    return s.hasNext() ? s.next() : "";
}
}

```

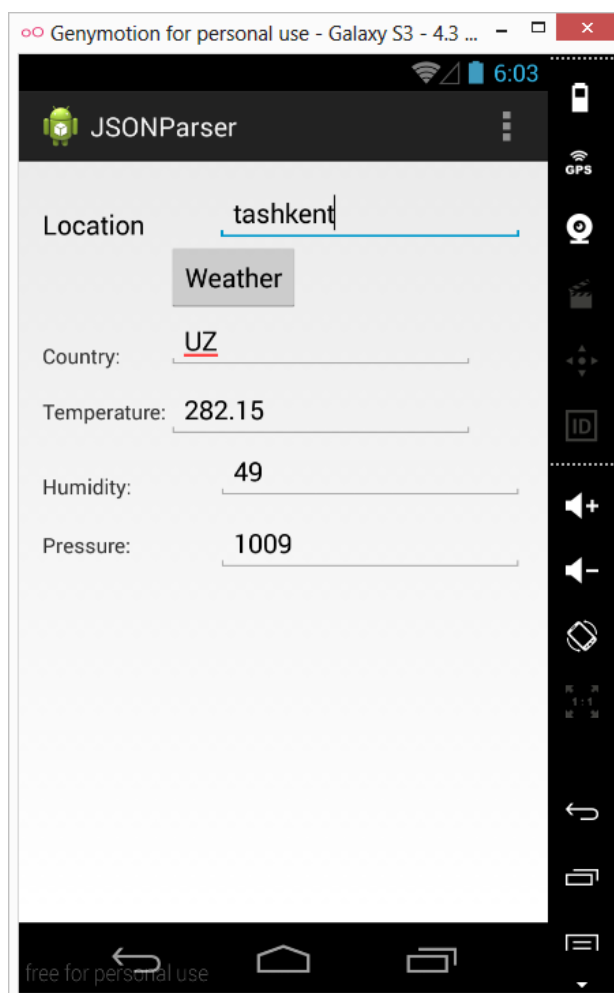
AndroidManifest.xml файлига permission қуйидагича қўшилади

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.usingjson"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="18" />
        <uses-permission android:name="android.permission.INTERNET"/>
    <application>..... </application>
</manifest>

```

Дастур натижаси



5.3-расм. JSON орқали маълумотларни сервердан юклаб олиш

Назорат саволлари:

1. Мобил иловаларда тармоқли дастурлаш?
2. ConnectivityManager синфи нима учун ишлатилади ва у қандай инициализация қилинади?
3. Андроид қурилмасининг тармоққа уланганлик ҳолатини текшириш учун қандай констатнталардан фойдаланилади?
4. HttpURLConnection синфи таснифини келтиринг?
5. HttpURLConnection синфининг қандай методлари мавжуд?
6. Илова интернетга боғланиши учун манифест файлида қандай рухсат эълон қилиниши зарур?
7. JSON нима вақачон ишлатилади?
8. JSON хизматининг асосий синфларини айтинг ва уларга изоҳ беринг?
9. JSON объекти қандай компонентлардан иборат бўлади?
10. JSON кўринишдаги маълумотга мисол кетиринг ва бўлакларга бўлиш қандай амалга оширилади?

Фойдаланилган адабиётлар:

1. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-10. Networking
2. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-10. Consuming JSON services
3. <http://www.tutorialspoint.com/json/>
4. http://www.tutorialspoint.com/android/android_json_parser.htm
5. <http://beginnersbook.com/2015/04/json-tutorial/>

IV БЎЛИМ

АМАЛИЙ МАШҒУЛОТ
МАТЕРИАЛЛАРИ

IV. АМАЛИЙ МАШҒУЛОТ МАТЕРИАЛЛАРИ

АМАЛИЙ МАШҒУЛОТЛАР МАЗМУНИ

1-амалий машғулот. Андроид муҳитини ўрнатиш ва созлаш. Дастлабки андроид иловасини яратиш ва мослаш. (2 соат)

Ишдан мақсад: Android операцион тизими учун иловалар яратишда керакли бўлган инструментал дастурий воситаларни ўрнатиш ва дастлабки мобил иловаларни (Android тизими учун) яратиш кўникмаларига эга бўлиш.

Ушбу амалий иш давомида қуйидагиларни **бажариш лозим:**

- Java development kit (JDK) ни ўрнатиш
- Android SDK (Software Development Kit) ни ўрнатиш
- Eclipse IDE ни ўрнатиш
- Android development tools (ADT) eclipse учун plugin ни ўрнатиш
- Android virtual машинасини (AVD) ўрнатиш

Ишни бажариш учун намуна

Androidда илова ишлаб чиқиш учун қуйидаги дастурлар ўрнатилган бўлиши керак.

1. Java Development Kit (JDK)
2. Android Software Development Kit (SDK)
3. Eclipse (компилятор)
4. Eclipse ва SDK нинг биргаликдаги кўриниши яъни **ADT (Android development tool)** ни ўрнатиш ҳам мумкин.
5. Androidда иловаларни ишга тушириш учун Android Virtual Device (AVD) виртуал муҳити бўлиши

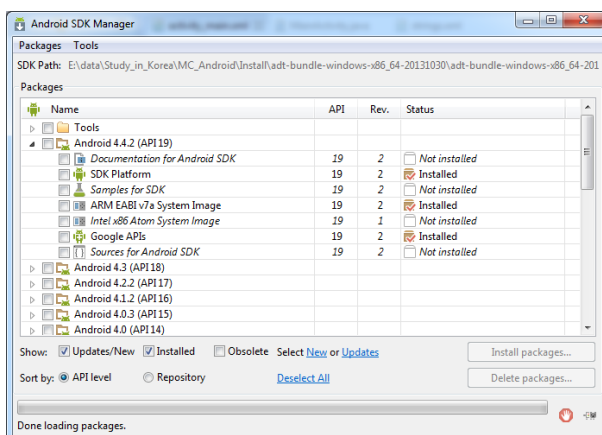
Керакли бўлган дастурий воситаларни ўрнатиш ва созлаш қуйидаги қадамлар билан кўрсатилган:

- 1-қадам: Java development kit (JDK) ни ўрнатиш (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
- 2- қадам: Android SDK (Software Development Kit) ни ўрнатиш (<http://developer.android.com/sdk/index.html>)

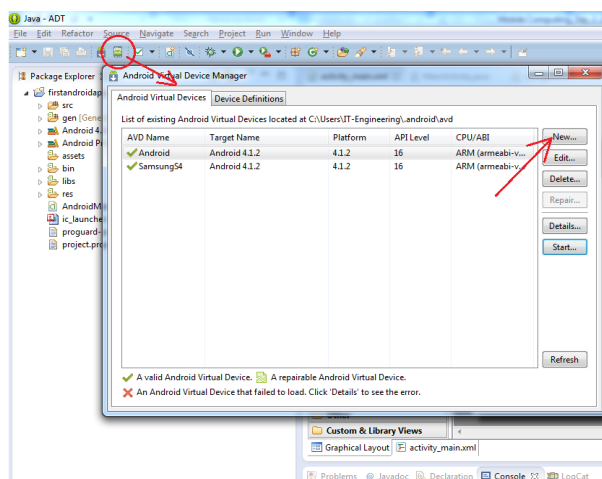
- 3- қадам: Eclipse IDE ни ўрнатиш (<http://www.eclipse.org/downloads/>)
- 4- қадам: Android development tools (ADT) eclipse учун plugin ни ўрнатиш (<https://dl-ssl.google.com/android/eclipse/>)
- 5- қадам: Android virtual машинасини (AVD) ўрнатиш

Қуйида ADT (Android development toolkit) ва Android virtual машинасини (AVD) ўрнатиш кетма-кетлиги келтирилган.

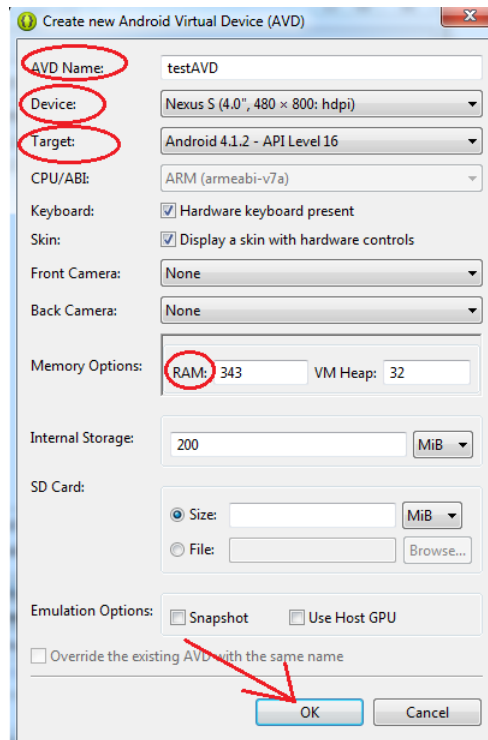
1. JDK (Java Development Kit, offer latest version)
2. ADT – Android Development Toolkit (Android SDK – Software Development Kit)
http://developer.android.com/sdk/index.html?utm_source=weibolife#download
3. Юқорида берилган интернет ссилкадан ADT кўчириб олинади ва архивдан чиқарилиб “eclipse” папкасидан eclipse.exe файли ишга туширилади
4. Қўшимча сошлаш (SDK ни ўрнатиш ва AVD яратиш)
 - SDK Manager (SDK нинг керакли бўлган атрибутларини ўрнатиш: **SDK Platform, ARM**)



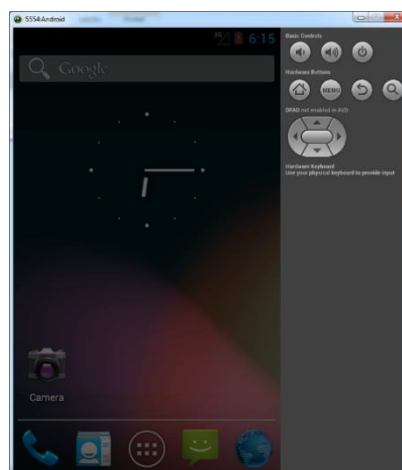
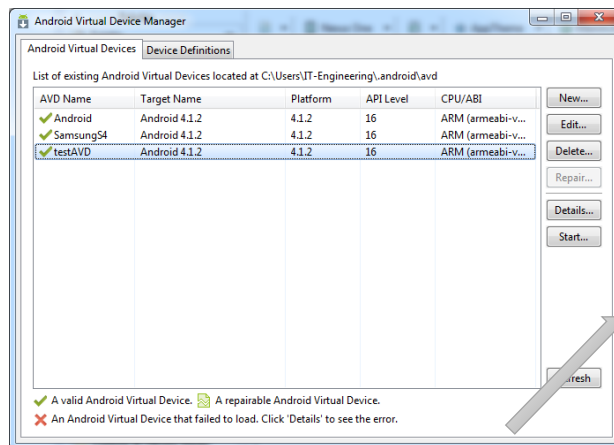
- AVD (Android Virtual Device) ни яратиш



- “Create new AVD window” ойнасидаги кўрсатилган майдонлар тўлдирилади



- Яратилган AVD ни ишга тушириш



Назорат саволлари

1. Андроид тизимда иловалар яратиш учун керакли инструментал воситаларни айтинг?
2. ADT нима ва ундан қандай фойдаланилади?
3. ADT да янги Андроид лойиҳа яратилди. Ушбу яратилган лойиҳа ичида автоматик ҳосил бўлган src папкасида нималар жойлашади?
4. ADT да янги Андроид лойиҳа яратилди. Ушбу яратилган лойиҳа ичида автоматик ҳосил бўлган gen папкасида нималар жойлашади?
5. ADT да янги Андроид лойиҳа яратилди. Ушбу яратилган лойиҳа ичида автоматик ҳосил бўлган res папкасида нималар жойлашади?
6. ADT да янги Андроид лойиҳа яратилди. Ушбу яратилган лойиҳа ичида автоматик ҳосил бўлган AndroidManifest.xml файлини изоҳланг?
7. ADT да янги Андроид лойиҳа яратилди, ва ушбу лойиҳа ишга туширилгандан кейин яратиладиган .apk файли қайерда (лойиҳанинг қайси папкасида) жойлашади?
8. AVD нинг ўрнига қандай виртуал машиналардан фойдаланиш мумкин?
9. Андроид тизимида иловалар яратиш учун керакли бўладиган асосий компоненталар қайсилар?

Фойдаланилган адабиётлар

1. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-1. Getting started with Android programming
2. “Android™ Application Development Cookbook”, by Wei-Meng Lee, printed at Sharda Offset Press, Delhi in 2015, pages – 385.
3. http://en.wikipedia.org/wiki/Mobile_operating_system
4. http://en.wikipedia.org/wiki/Comparison_of_mobile_operating_systems
5. [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
6. http://www.tutorialspoint.com/android/android_overview.htm

2-амалий машғулот. Java дастурлаш тилида синф ва объектлардан фойдаланиш. ОЙД принциплари. Абстракт синфлардан фойдаланиш. Java дастурлаш тилида ворислик ва полиморфизмдан фойдаланиб дастурлаш. (2 соат)

Ишдан мақсад: Java дастурлаш тили асосий конструкцияларидан фойдаланиш бўйича асосий кўникмаларни такрорлаш. Синфлар ва объектлардан фойдаланиш. Объектга йўналтирилган дастурлаш асосий принциплари (ворислик, инкапсуляция, полиморфизм, абстракт ва аноним синфлар ва бошқалар) дан фойдаланиш кўникмаларига эга бўлиш. Java тилида синфлараро ворислик ваобъектга йўналтирилган дастурлаш асосий принципларидан бири бўлган полиморфизмдан фойдаланиш кўникмаларига эга бўлиш

Масаланинг кўйилиши: Тингловчи вариантда келтирилган топшириқларни Java дастурлаш тили асосида ечиши ва натижа олиши лозим. Тингловчи вариант бўйича берилган мисолларни Java тилидаворислик ва полиморфизм хусусиятидан фойдаланиб дастур тузиш ва натижа олиш.

Ишни бажариш учун намуна

Мазкур амалий иш давомида Java дастурлаш тили асосий конструкцияларини ўзлаштириш учун 1-топшириқ берилган бўлиб, бунга кўра Java тилида киритиш ва чиқариш стандарт операторлардан фойдаланиш бўйича кўникма ҳосил қилинади. Мисол учун қуйидаги мисолда содда дастур тузилган бўлиб, бунда Scanner синфи терминал орқали ўзгарувчига қиймат ўқиш, System.out.println() эса натижани терминалга чиқариш учун ишлатилган.

```
package pkg2.pkg1.amaliy;

import static java.lang.Math.pow;
import java.util.Scanner;

public class Amaliy {
    public static void main(String[] args) {
        int n = 0,a;
        double S = 0;
        Scanner var=new Scanner(System.in);
        n=var.nextInt();
```

```

a=var.nextInt();
for(int i=0;i<n;i++)
    S=S+1/pow(a,pow(2,i));
    if(a==0){
        System.out.println("Dasturda xatolik!");
    }
    else{
        System.out.println(S);
    }
}
}

```

```

package pkg2.pkg1.amaliy;

import static java.lang.Math.pow;
import java.util.Scanner;

public class Amaliy {
    public static void main(String[] args) {
        int n = 0,a;
        double S = 0;
        Scanner var=new Scanner(System.in);
        n=var.nextInt();
        a=var.nextInt();
        for(int i=0;i<n;i++)
            S=S+1/pow(a,pow(2,i));
            if(a==0){
                System.out.println("Dasturda xatolik!");
            }
            else{
                System.out.println(S);
            }
        }
    }
}

```

Ўз навбатида тингловчи навбатдаги топшириқда синфлар ва объектлардан фойдаланиш, синфларда ворислик ва абстракт синфлардан фойдаланиш бўйича кўникмаларни ҳосил қилади. Бу бўйича қуйида мисол келтирилган.

Самалёт - Лайнер:

```

package samolyot1;
class samolyot{
    protected String nomi;
    protected int ogir;
    protected String yunalish;

    public samolyot(String nomi,int ogir,String yunalish)
    {

```

```

    this. nomi=nomi;
    this. ogir=ogir;
    this. yunalish=yunalish;
};
public void show_samolyot()
{
    System.out.println( "Samolyot nomi:=: " +nomi);
    System.out.println("Samoltot og'irligi:=: " +ogir + " tonna ");
    System.out.println( "Samolyot yunalishi:=: " +yunalish);
}
}
class odam_tashuvchi_layner extends samolyot {

    private int urindiq;
    private int tezligi;

public odam_tashuvchi_layner(String nomi, int ogir, String yunalish,
int urindiq, int tezligi )
{
    super(nomi,ogir,yunalish);
    this. urindiq=urindiq;
    this. tezligi = tezligi;
}
public void show_card()
{
    show_samolyot() ;
    System.out.println("Samolyot o'rindiqlar soni:= : " +urindiq);
    System.out.println("Samolyotni tezligi: " +tezligi+" km/soat");
}
}
public class Samolyot1 {
    public static void main(String[] args) {
        odam_tashuvchi_layner card=new odam_tashuvchi_layner(" PJ 200 ",180, "
Toshkent && Navoiy ",200,1200);
        card.show_card();
    } }

```

Спортчи - Хоккейчи;

```

package oyinchi;
class oyinchi1 {
    protected String komanda;
    protected String akusti;
public oyinchi1(String komanda,String akusti)
{
    this.komanda=komanda;
    this.akusti=akusti;
};
public void show_oyinchi()
{
    System.out.println( "Komanda nomi:= " +komanda);
    System.out.println( "Akusti yunalishi:= " +akusti);
}}
class hokeist extends oyinchi1 {
    private int buyi;
    private int tezligi;
    private int ogir;
public hokeist(String komanda, String akusti,
int buyi, int tezligi, int ogir )
{
    super(komanda,akusti);
    this.buyi=buyi;
    this.tezligi = tezligi;
    this.ogir = ogir;
}
public void show_card()
{
    show_oyinchi() ;
    System.out.println("O'yinchini bo'yi:= : " +buyi + " sm ");
    System.out.println("O'yinchini tezligi:= " +tezligi+" km/soat");
    System.out.println("O'yinchini og'irligi:= " +ogir+" kg");
}}
public class Oyinchi {
    public static void main(String[] args) {
        hokeist card=new hokeist(" OQ AYIQ ", " Hujumchi ",182,25,99);
        card.show_card();
    }
}

```

```

29     super(komanda,akusti);
30     this.buyi=buyi;
31     this.tezligi = tezligi;
32     this.ogir = ogir;
33 }
34 public void show_card()
35 {
36     show_oyinchi() ;
37     System.out.println("O'yinchini bo'yi:= : " +buyi + " sm ");
38     System.out.println("O'yinchini tezligi:= " +tezligi+" km/soat");
39     System.out.println("O'yinchini og'irligi:= " +ogir+" kg");
40 }
41 }
42
43 public class Oyinchi {
44     public static void main(String[] args) {
45
46         hokeist card=new hokeist(" OQ AYIQ "," Hujumchi ",182,25,99);
47         card.show_card();
48     }
49 }
50 }

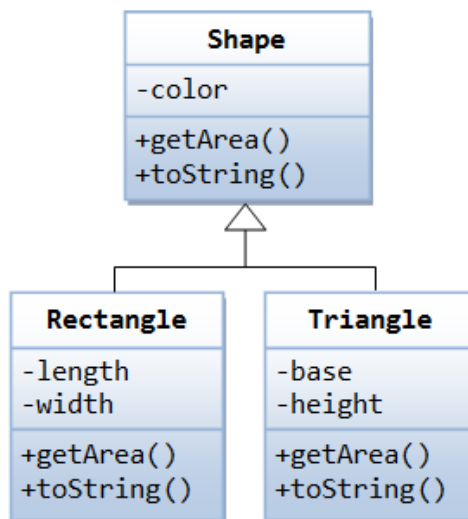
```

```

ывод - oyinchi (run) x
run:
Komanda nomi:=: OQ AYIQ
Akusti yunalishi:=: Hujumchi
O'yinchini bo'yi:= : 182 sm
O'yinchini tezligi:= 25 km/soat
O'yinchini og'irligi:= 99 kg
СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)

```

Қуйидаги расмда кўрсатилгандек, shakl (shape) –бу супер синф бўлиб, тўртбурчак (Rectangle), учбурчаклар (Triangle) эса ворис синф ҳисобланади. Демак булар орасида қандай боғлиқлик бор ва бунда полиморфизм қандай вазифани бажаради. Қуйида дастур ва унинг натижаси келтирилган.



Shape.java

```

// Define superclass Shape
public class Shape {
// Private member variable
private String color;

// Constructor

```

```

public Shape (String color) {
this.color = color;
}

@Override
public String toString() {
return "Shape of color=\"" + color + "\"";
}

// All shapes must has a method called getArea()
public double getArea() {
System.err.println("Shape unknown! Cannot compute area!");
return 0; // Need a return to compile the program
}
}

```

Rectangle.java

```

// Define Rectangle, subclass of Shape
public class Rectangle extends Shape {
// Private member variables
private int length;
private int width;

// Constructor
public Rectangle(String color, int length, int width) {
super(color);
this.length = length;
this.width = width;
}

@Override
public String toString() {
return "Rectangle of length=" + length + " and width=" + width + ", subclass of " + super.toString();
}

@Override
public double getArea() {
return length*width;
}
}

```

Triangle.java

```

// Define Triangle, subclass of Shape
public class Triangle extends Shape {
// Private member variables
private int base;
private int height;

// Constructor
public Triangle(String color, int base, int height) {
super(color);
}
}

```



```

this.base = base;
this.height = height;
}

@Override
public String toString() {
return "Triangle of base=" + base + " and height=" + height + ", subclass of " + super.toString();
}

@Override
public double getArea() {
return 0.5*base*height;
}
}

```

TestShape.java

```

// A test driver program for Shape and its subclasses
public class TestShape {
public static void main(String[] args) {
    Shape s1 = new Rectangle("red", 4, 5);
    System.out.println(s1);
    System.out.println("Area is " + s1.getArea());

    Shape s2 = new Triangle("blue", 4, 5);
    System.out.println(s2);
    System.out.println("Area is " + s2.getArea());

    Shape s3 = new Shape("green");
    System.out.println(s3);
    System.out.println("Area is " + s3.getArea());
}
}

```

Дастур натижаси:

```

Problems @ Javadoc Declaration Console
<terminated> TestShape [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (16.09.2014 14:44:39)
Rectangle of length=4 and width=5, subclass of Shape of color="red"
Area is 20.0
Triangle of base=4 and height=5, subclass of Shape of color="blue"
Area is 10.0
Shape of color="green"
Area is 0.0
Shape unknown! Cannot compute area!

```

Назорат саволлари:

1. Синф ва объектнинг бир-биридан фарқи?
2. Java да маълумотлар типлари (хар бир тип хтирадан қанча жой эгаллашини ҳам кўрсатинг)?
3. Java да синфлар қандай ўзгарувчиларни ўз ичига олади?
4. Объект яратиш (тўлиқ объект яратилганда) жараёнида қандай кетма – кетлик амалга оширилади?
5. Объект қандай қилиб инициализация қилинади (Мисол келтиринг)?
6. Объектнинг ўзгарувчиси ва методига қандай мурожаат қилинади (Мисол келтиринг)?
7. Java да ворис олиш учун қандай калит сўздан фойдаланилади ва синфдан ворис олишга мисол келтиринг?
8. Абстракт синф нима ва у қандайхусусиятларга эга?
9. Абстракт синф қандай яратилади?
10. Java да маълумотлар типлари (хар бир тип хтирадан қанча жой эгаллашини ҳам кўрсатинг)?
11. Объектга йўналтирилган дастурлашнинг асосий хусусиятлари қайсилар?
12. Java да синфлар қандай ўзгарувчиларни ўз ичига олади?
13. Объект яратиш (тўлиқ объект яратилганда) жараёнида қандай кетма – кетлик амалга оширилади?
14. Объект қандай қилиб инициализация қилинади (Мисол келтиринг)?
15. Объектнинг ўзгарувчиси ва методига қандай мурожаат қилинади (Мисол келтиринг)?
16. Java да ворис олиш учун қандай калит сўздан фойдаланилади ва синфдан ворис олишга мисол келтиринг?
17. Қандай турдаги полиморфизмлар мавжуд?
18. Method overloading нима ва унга мисол келтиринг?
19. Method overriding нима ва унга мисол келтиринг?
20. Бир номли лекин турли хил параметрли функциялар ва уларга мурожаат қилиш нима дейилади?
21. Бир номли ва бир хил параметрли функциялар ва уларга мурожаат қилиш нима дейилади?
22. Method overriding битта синф ичида амалга оширилиши мумкинми. Мумкин ёки мумкин эмаслигини исботланг?

Фойдаланилган адабиётлар:

1. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-1. Getting started with Android programming
2. "Java The Complete Reference Eighth Edition" by Herbert Schildt, 2014, Глава-6,7,8,9,10. стр.145-259
3. Core Java “An Integrated Approach”, Include all versions up to Java7 by Dr. R. Nageswara Rao, 2015, 646 pages
4. http://www.tutorialspoint.com/java/java_polymorphism.htm
5. <http://beginnersbook.com/2013/03/polymorphism-in-java/>
6. http://www.tutorialspoint.com/java/java_abstraction.htm
7. http://www.tutorialspoint.com/java/java_object_classes.htm

3-амалий машғулот.

**Андроид тизимида Layout ва View лардан фойдаланиш. Жараёнлар ва ходислар билан ишлаш. Андроид тизимида фойдаланувчи интерфейсини яратиш расм ва менюларни жойлаштириш. Объект анимацияси.
(2 соат)**

Ишдан мақсад: Андроид иловаларини яратишда activity лардан фойдаланиш. Асосий Java (activity) файллар ва layout (xml) лар структураси ва уларни ўзгартириш кўникмаларига эга бўлиш, бир ва бир нечта activity ва layout ларни бир – бирига боғлаш.

Android тизимида фойдаланувчи интерфейсини яратиш. Gallery, ImageView ва ImageSwitcher компоненталаридан фойдаланиб расмларни намойиш қиладиган илова яратиш кўникмаларига эга бўлиш.

Масаланинг қўйилиши: Андроид тизимида битта ойнадан иккинчи ойнага ўтиш ва орқага (олдинги ойнага) қайтувчи илова яратилсин ва ҳар бир ойнада мос ҳолда хабарлар ҳосил қилинсин. Мос ҳолда xml файллар таснифи келтирилсин.

Gallery, ImageView ва ImageSwitcher компоненталаридан фойдаланиб Галерея дастурини яратиш.

Ишни бажариш учун намуна

Андроидда янги лойиҳа яратилади ва яратилган лойиҳа қуйидаги кўрсатилган файллар асосида тахрирланади¹ (*SubActivity.java*, *activity_sub.xml*)

¹ http://www.tutorialspoint.com/android/android_user_interface_layouts.htm

файллари янги яратилади)

MainActivity.java

```
import android.os.Bundle;
import android.app.Activity;
import android.view.*; // Menu;
import android.content.Intent;
import android.widget.TextView;

public class MainActivity extends Activity {
    TextView mText;
    final static int ACT_EDIT = 0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mText = (TextView)findViewById(R.id.textView1);
    }
    public void mOnClick(View v) {

        Intent intent = new Intent(this, SubActivity.class);
        startActivity(intent);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >
```

```

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="this is main activity" />

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView1"
    android:layout_marginTop="36dp"
    android:onClick="mOnClick"
    android:text="Call" />

```

</RelativeLayout>

SubActivity.java

```

package com.example.example12_multiple_activities;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
public class SubActivity extends Activity {
    EditText mEdit;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sub);
    }
    public void mOnClick(View v) {
        finish();
    }
}

```

activity_sub.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView2"

```

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="this is sub activity" />
```

<EditText

```
android:id="@+id/editText1"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:ems="10" />
```

<Button

```
android:id="@+id/button2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:onClick="mOnClick"  
android:text="OK" />
```

<Button

```
android:id="@+id/button3"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:onClick="mOnClick"  
android:text="Cancel" />
```

</LinearLayout>

Manifest.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
package="com.example.example12_multiple_activities"  
android:versionCode="1"  
android:versionName="1.0" >  
  
<uses-sdk  
android:minSdkVersion="8"  
android:targetSdkVersion="18" />  
  
<application  
android:allowBackup="true"  
android:icon="@drawable/ic_launcher"  
android:label="@string/app_name"  
android:theme="@style/AppTheme" >  
<activity  
android:name="com.example.example12_multiple_activities.MainActivity"
```

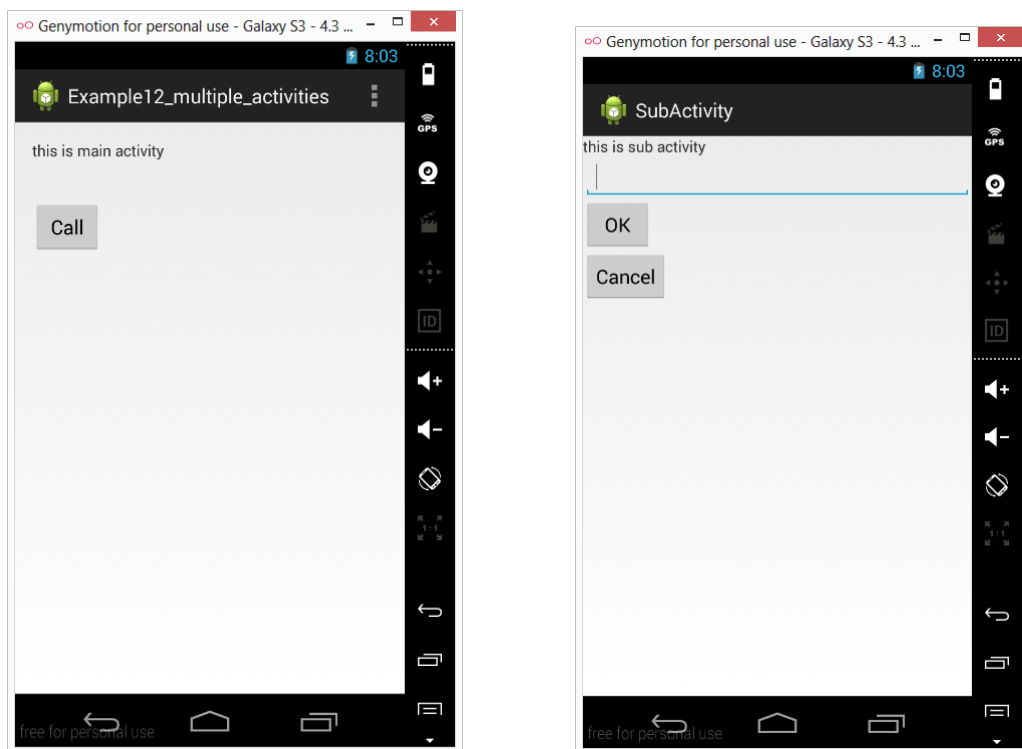
```

android:label="@string/app_name" >
<intent-filter>
  <action android:name="android.intent.action.MAIN" />

  <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<activity
  android:name="com.example.example12_multiple_activities.SubActivity"
  android:label="SubActivity" />
</application>
</manifest>

```

Дастур натижаси

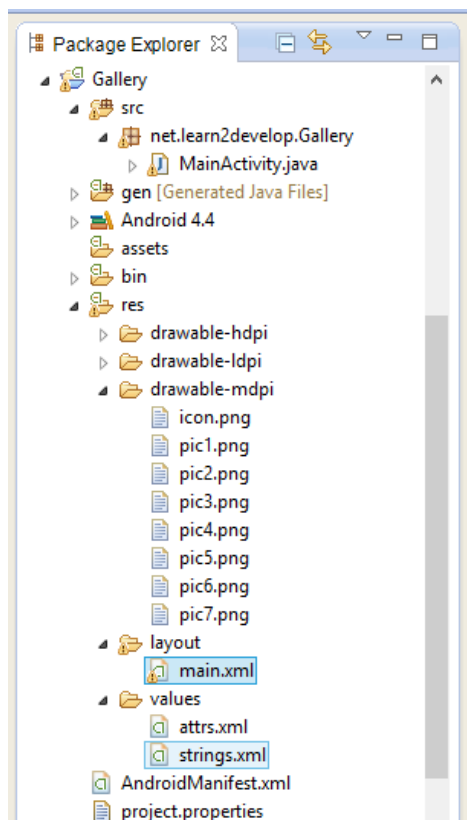


Ишни бажариш учун намуна¹

1. Eclipse (ADT) да “Gallery” номли лойиҳа яратинг

¹ “Beginning Android™ 4 Application Development”. Chapter-5. Displaying Pictures and Menus with Views pages 248-254.

2. Иловада керакли бўлган расмларни `res/drawable-mdpi` папкасига кўчиринг (мисол учун `pic1.png`, `pic2.png` ва ҳ.к.)
3. Лойиҳа файлларини мос ҳолда ўзгартиринг
 - a. `MainActivity.java` (`src` папкасида жойлашган асосий активитий файл)
 - b. `main.xml` (`res/layout` папкасида жойлашган лайаут файл)
4. Илованинг `res/values` папкасига `attrs.xml` номли файлни қўшинг
5. Натижада илованинг папкалар структураси қуйидагича бўлади (1-расм)



1-расм. Мазкур илованинг файллар структураси

6. Мос ҳолда файлларнинг кодлари қуйида келтирилган

MainActivity.java

`package package номи киритилсин (com.example.gallery)`

```
import android.app.Activity;
```

```
import android.os.Bundle;
```

```
import android.content.Context;
```

```
import android.content.res.TypedArray;
```

```
import android.view.View;
```

```
import android.view.ViewGroup;
```

```
import android.widget.AdapterView;
```

```
import android.widget.AdapterView.OnItemClickListener;
```

```
import android.widget.BaseAdapter;
```

```
import android.widget.Gallery;
```

```
import android.widget.ImageView;
```



```

import android.widget.Toast;

public class MainActivity extends Activity {
    //---the images to display---
    Integer[] imageIDs = {
        R.drawable.pic1,
        R.drawable.pic2,
        R.drawable.pic3,
        R.drawable.pic4,
        R.drawable.pic5,
        R.drawable.pic6,
        R.drawable.pic7
    };

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Gallery gallery = (Gallery) findViewById(R.id.gallery1);

        gallery.setAdapter(new ImageAdapter(this));
        gallery.setOnItemClickListener(new OnItemClickListener()
        {
            public void onItemClick(AdapterView<?> parent, View v, int position, long id)
            {
                Toast.makeText(getApplicationContext(),
                    "pic" + (position + 1) + " selected",
                    Toast.LENGTH_SHORT).show();

                //---display the images selected---
                ImageView imageView = (ImageView) findViewById(R.id.image1);
                imageView.setImageResource(imageIDs[position]);
            }
        });
    }

    public class ImageAdapter extends BaseAdapter
    {
        private Context context;
        private int itemBackground;

        public ImageAdapter(Context c)
        {
            context = c;
            //---setting the style---
            TypedArray a = obtainStyledAttributes(R.styleable.Gallery1);
            itemBackground = a.getResourceId(
                R.styleable.Gallery1_android_galleryItemBackground, 0);
            a.recycle();
        }

        //---returns the number of images---
        public int getCount() {

```

```

    return imageIDs.length;
}

//---returns the ID of an item---
public Object getItem(int position) {
    return position;
}

//---returns the ID of an item---
public long getItemId(int position) {
    return position;
}

//---returns an ImageView view---
public View getView(int position, View convertView, ViewGroup parent) {
    ImageView imageView = new ImageView(context);
    imageView.setImageResource(imageIDs[position]);
    imageView.setScaleType(ImageView.ScaleType.FIT_XY);
    imageView.setLayoutParams(new Gallery.LayoutParams(150, 120));
    imageView.setBackgroundResource(itemBackground);
    return imageView;
}
}
}

```

main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Images of San Francisco" />

<Gallery
    android:id="@+id/gallery1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />

<ImageView
    android:id="@+id/image1"
    android:layout_width="320px"
    android:layout_height="250px"
    android:scaleType="fitXY" />

</LinearLayout>

```

attrs.xml

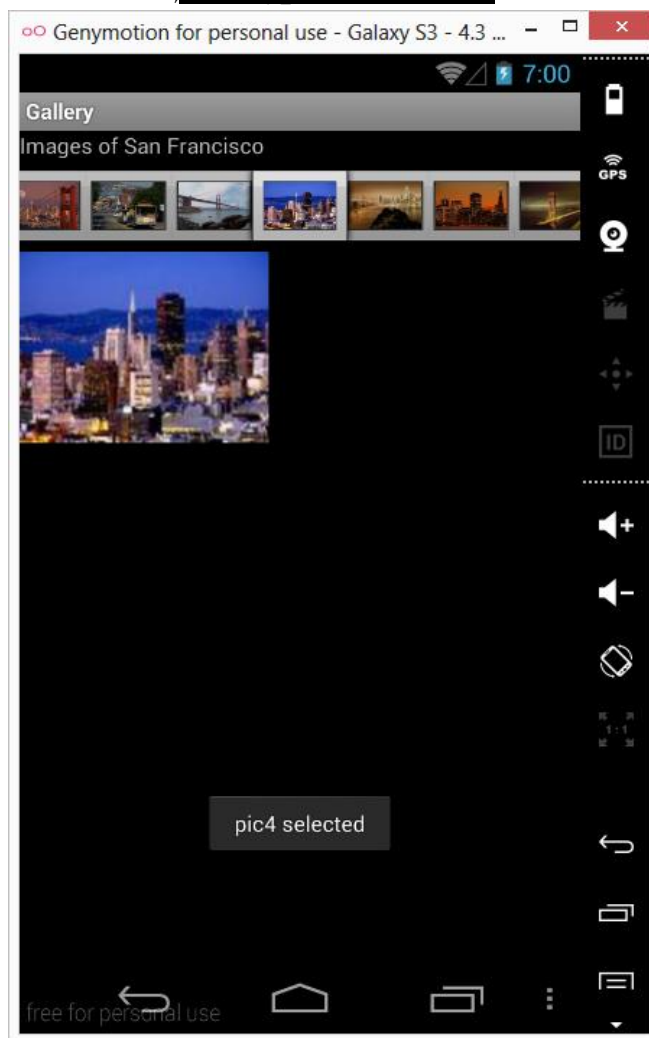
```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <declare-styleable name="Gallery1">
        <attr name="android:galleryItemBackground" />
    </declare-styleable>

```

1

Дастур натижаси



Назорат саволлари:

1. Activity нима?
2. Activity ларни бир – бирига боғлаш учун қандай синфдан фойдаланилади?
3. onCreate () методи нима учун ишлатилади?
4. Activity (дастур ойнаси) натижаси фойдаланувчига кўринганда қайси функция чақирилади?

5. Activity яратилганда то иш фаолиятини тўхтатгунча тизим томонидан қандай методлар ишга туширилади?
6. Берилган код қисми нима вазифани бажаради?

```
public void onClick(View view) {
    startActivity(new Intent("com.example.SecondActivity"));
}
```
7. Бир activity дан иккинчи activity га қиймат жўнатиш учун қайси методдан фойдаланилади?
8. Берилган код қисми нима вазифани бажаради?

```
Toast.makeText(this, "Hello World", Toast.LENGTH_SHORT).show();
```
9. Андроид тизимида фойдаланувчи интерфейсини яратишда қандай layout (қолип) лардан фойдаланилади
10. Андроид тизимида фойдаланувчи интерфейсини яратишда қандай layout (қолип) лардан фойдаланилади
11. Ориентацияни “**Anchoring**” бошқариш усули ва унинг асосий хусусиятини айтинг?
12. Ориентацияни “**Resizingandrepositioning**” бошқариш усули ва унинг асосий хусусиятини айтинг?
13. List view ларга қайсилар киради ва уларнинг асосий вазифаси нима?
14. ImageButton ва Button нинг қандай фарқи мавжуд?
15. EditText компонентаси ҳолатини фақат махфийсўз (парол) киритишга мўлажаллаш учун EditText элементига қандай атрибут қўшиш лозим?
16. CheckBox қиймати trueёкиfalse эканлигини текшириш учун CheckBox синфининг қандай методдан фойдаланилади?
17. Gallery нима учун ишлатилади ва Gallery синфининг қандай методлари мавжуд?
18. Gallery ва ImageView компоненталарини қандай фарқлаш мумкин?
19. BaseAdapter синфи нима учун ишлатилади?
20. BaseAdapter синфининг қандай абстракт методлари мавжуд?
21. ListView ва GridView виджетларининг бир-биридан фарқи?
22. Context меню яратиш учун қандай методлардан фойдаланилади?
23. Option меню яратиш учун қандай методлардан фойдаланилади?

Фойдаланилган адабиётлар:

1. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-2. Activities, fragments and Intents
2. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-3. Android User Interface
3. “Android™ Application Development Cookbook”, by Wei-Meng Lee, printed at Sharda Offset Press, Delhi in 2015, pages – 385. Chapter-1. Android Fundamentals
4. http://www.tutorialspoint.com/android/android_user_interface_layouts.htm
5. <http://developer.android.com/intl/ru/guide/topics/ui/declaring-layout.html>

4 – амалий машғулот:

Иловалар яратишда маълумотлар базасидан фойдаланиш. SQLite. Сўровлар яратиш. (2 соат)

Ишдан мақсад: Андроид тизимида фойдаланувчи интерфейсини яратишда маълумотлар базасидан фойдаланиш, лойихага мос маълумотлар базаси, жадваллар ва сўровлар яратиш кўникмаларига эга бўлиш

Масаланинг кўйилиши: Тингловчи вариант бўйича берилган лойихани Андроид тизимида ишлаб чиқиш ва илова учун керакли бўлган маълумотларни маълумотлар базасидан ўқиб олиши ва emulator орқали натижа олиши лозим.

Ишни бажариш учун намуна

Мисол: Андроид тизимида AddressBook дастури тузилсин. Дастурда базага янги маълумот кўшиши, киритилган маълумотни ўзгартириш ва ўчириш функциялари мавжуд бўлсин¹.

Масаланинг ечилиш:

Илова учун керакли бўлган файллар қуйидагилар



¹ <http://www.tutorialspoint.com/sqlite/>

Бу ерда:

DBHelper.java – маълумотлар базасини бошқариш учун

DisplayContact.java – ҳар бир контактга тегишли маълумотни кўрсатиб беради

MainActivity.java – маълумотлар базасидаги мавжуд контактлар рўйхатини кўрсатади

activity_display_contact.xml – ҳар бир контактга тегишли маълумотни кўрсатувчи лайоут

activity_main.xml – мавжуд контактлар рўйхатини кўрсатувчи лайоут

display_contact.xml – меню хусусиятларини (delete, update) ҳосил қилувчи лайоут

mainmenu.xml – янги контакт қўшувчи меню лайоути

string.xml – илова элементларидаги матнлар рўйхати

Ҳар бир файл коди қуйида алоҳида-алоҳида кўрсатилган

Дастур дизайни ва менюлар учун xml файллар хусусияти қуйида кўрсатилган.

activity_display_contact.xml

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/scrollView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    tools:context=".DisplayContact" >
```

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="370dp"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    >
```

```
<EditText
    android:id="@+id/editTextName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_marginTop="5dp"
```

```

    android:layout_marginLeft="82dp"
    android:ems="10"
    android:inputType="text" >
</EditText>

<EditText
    android:id="@+id/editTextEmail"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/editTextStreet"
    android:layout_below="@+id/editTextStreet"
    android:layout_marginTop="22dp"
    android:ems="10"
    android:inputType="textEmailAddress" />

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/editTextName"
    android:layout_alignParentLeft="true"
    android:text="@string/name"
    android:textAppearance="?android:attr/textAppearanceMedium" />

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/editTextCity"
    android:layout_alignParentBottom="true"
    android:layout_marginBottom="28dp"
    android:onClick="run"
    android:text="@string/save" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/editTextEmail"
    android:layout_alignLeft="@+id/textView1"
    android:text="@string/email"
    android:textAppearance="?android:attr/textAppearanceMedium" />

<TextView
    android:id="@+id/textView5"

```

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignBottom="@+id/editTextPhone"  
android:layout_alignLeft="@+id/textView1"  
android:text="@string/phone"  
android:textAppearance="?android:attr/textAppearanceMedium" />
```

```
<TextView  
    android:id="@+id/textView4"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_above="@+id/editTextEmail"  
    android:layout_alignLeft="@+id/textView5"  
    android:text="@string/street"  
    android:textAppearance="?android:attr/textAppearanceMedium" />
```

```
<EditText  
    android:id="@+id/editTextCity"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignRight="@+id/editTextName"  
    android:layout_below="@+id/editTextEmail"  
    android:layout_marginTop="30dp"  
    android:ems="10"  
    android:inputType="text" />
```

```
<TextView  
    android:id="@+id/textView3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignBaseline="@+id/editTextCity"  
    android:layout_alignBottom="@+id/editTextCity"  
    android:layout_alignParentLeft="true"  
    android:layout_toLeftOf="@+id/editTextEmail"  
    android:text="@string/country"  
    android:textAppearance="?android:attr/textAppearanceMedium" />
```

```
<EditText  
    android:id="@+id/editTextStreet"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignLeft="@+id/editTextName"  
    android:layout_below="@+id/editTextPhone"  
    android:ems="10"  
    android:inputType="text" >
```



```

    <requestFocus />
</EditText>

<EditText
    android:id="@+id/editTextPhone"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/editTextStreet"
    android:layout_below="@+id/editTextName"
    android:ems="10"
    android:inputType="phone|text" />

</RelativeLayout>
</ScrollView>

```

activity_main.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <ListView
        android:id="@+id/listView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true" >
    </ListView>

</RelativeLayout>

```

display_contact.xml

```

<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/Edit_Contact"
        android:orderInCategory="100"
        android:title="@string/edit"/>
    <item
        android:id="@+id/Delete_Contact"
        android:orderInCategory="100"
        android:title="@string/delete"/>

```

```
</menu>
```

mainmenu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
  <item android:id="@+id/item1"
    android:icon="@drawable/ic_launcher"
    android:title="@string/Add_New"
    android:showAsAction="ifRoom|withText">
  </item>
</menu>
```

string.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

  <string name="app_name">Address Book</string>
  <string name="action_settings">Settings</string>
  <string name="hello_world">Hello world!</string>
  <string name="Add_New">Add New</string>
  <string name="edit">Edit Contact</string>
  <string name="delete">Delete Contact</string>
  <string name="title_activity_display_contact">DisplayContact</string>

  <string name="name">Name</string>
  <string name="phone">Phone</string>
  <string name="email">Email</string>
  <string name="street">Street</string>
  <string name="country">City/State/Zip</string>

  <string name="save">Save Contact</string>

  <string name="deleteContact">Are you sure, you want to delete it.</string>
  <string name="yes">Yes</string>
  <string name="no">No</string>

</resources>
```

Асосий activity файллар кўриниши қуйидагича

DBHelper.java

```
package com.example.addressbook;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Hashtable;

import android.content.ContentValues;
import android.content.Context;
```

```

import android.database.Cursor;
import android.database.DatabaseUtils;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteDatabase;

public class DBHelper extends SQLiteOpenHelper {

    public static final String DATABASE_NAME = "MyDBName.db";
    public static final String CONTACTS_TABLE_NAME = "contacts";
    public static final String CONTACTS_COLUMN_ID = "id";
    public static final String CONTACTS_COLUMN_NAME = "name";
    public static final String CONTACTS_COLUMN_EMAIL = "email";
    public static final String CONTACTS_COLUMN_STREET = "street";
    public static final String CONTACTS_COLUMN_CITY = "place";
    public static final String CONTACTS_COLUMN_PHONE = "phone";

    private HashMap hp;

    public DBHelper(Context context)
    {
        super(context, DATABASE_NAME , null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // TODO Auto-generated method stub
        db.execSQL(
            "create table contacts " +
            "(id integer primary key, name text,phone text,email text, street text,place text)"
        );
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // TODO Auto-generated method stub
        db.execSQL("DROP TABLE IF EXISTS contacts");
        onCreate(db);
    }

    public boolean insertContact (String name, String phone, String email, String street,String
    place)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();

```

```

contentValues.put("name", name);
contentValues.put("phone", phone);
contentValues.put("email", email);
contentValues.put("street", street);
contentValues.put("place", place);

db.insert("contacts", null, contentValues);
return true;
}
public Cursor getData(int id){
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor res = db.rawQuery( "select * from contacts where id="+id+"", null );
    return res;
}
public int numberOfRows(){
    SQLiteDatabase db = this.getReadableDatabase();
    int numRows = (int) DatabaseUtils.queryNumEntries(db, CONTACTS_TABLE_NAME);
    return numRows;
}
public boolean updateContact (Integer id, String name, String phone, String email, String
street,String place)
{
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put("name", name);
    contentValues.put("phone", phone);
    contentValues.put("email", email);
    contentValues.put("street", street);
    contentValues.put("place", place);
    db.update("contacts", contentValues, "id = ? ", new String[] { Integer.toString(id) } );
    return true;
}

public Integer deleteContact (Integer id)
{
    SQLiteDatabase db = this.getWritableDatabase();
    return db.delete("contacts",
    "id = ? ",
    new String[] { Integer.toString(id) });
}
public ArrayList getAllCotacts()
{
    ArrayList array_list = new ArrayList();
    //hp = new HashMap();
    SQLiteDatabase db = this.getReadableDatabase();

```

```

Cursor res = db.rawQuery( "select * from contacts", null );
res.moveToFirst();
while(res.isAfterLast() == false){
    array_list.add(res.getString(res.getColumnIndex(CONTACTS_COLUMN_NAME)));
    res.moveToNext();
}
return array_list;
}
}

```

DisplayContact.java

```

package com.example.addressbook;

import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class DisplayContact extends Activity {

    int from_Where_I_Am_Coming = 0;
    private DBHelper mydb ;
    TextView name ;
    TextView phone;
    TextView email;
    TextView street;
    TextView place;
    int id_To_Update = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_contact);
        name = (TextView) findViewById(R.id.editTextName);
        phone = (TextView) findViewById(R.id.editTextPhone);
    }
}

```

```

email = (TextView) findViewById(R.id.editTextStreet);
street = (TextView) findViewById(R.id.editTextEmail);
place = (TextView) findViewById(R.id.editTextCity);

mydb = new DBHelper(this);

Bundle extras = getIntent().getExtras();
if(extras !=null)
{
    int Value = extras.getInt("id");
    if(Value>0){
        //means this is the view part not the add contact part.
        Cursor rs = mydb.getData(Value);
        id_To_Update = Value;
        rs.moveToFirst();
        String nam =
rs.getString(rs.getColumnIndex(DBHelper.CONTACTS_COLUMN_NAME));
        String phon =
rs.getString(rs.getColumnIndex(DBHelper.CONTACTS_COLUMN_PHONE));
        String emai =
rs.getString(rs.getColumnIndex(DBHelper.CONTACTS_COLUMN_EMAIL));
        String stree =
rs.getString(rs.getColumnIndex(DBHelper.CONTACTS_COLUMN_STREET));
        String plac =
rs.getString(rs.getColumnIndex(DBHelper.CONTACTS_COLUMN_CITY));
        if (!rs.isClosed())
        {
            rs.close();
        }
        Button b = (Button)findViewById(R.id.button1);
        b.setVisibility(View.INVISIBLE);

        name.setText((CharSequence)nam);
        name.setFocusable(false);
        name.setClickable(false);

        phone.setText((CharSequence)phon);
        phone.setFocusable(false);
        phone.setClickable(false);

        email.setText((CharSequence)emai);
        email.setFocusable(false);
        email.setClickable(false);

        street.setText((CharSequence)stree);

```

```

        street.setFocusable(false);
        street.setClickable(false);

        place.setText((CharSequence)plac);
        place.setFocusable(false);
        place.setClickable(false);
    }
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    Bundle extras = getIntent().getExtras();
    if(extras !=null)
    {
        int Value = extras.getInt("id");
        if(Value>0){
            getMenuInflater().inflate(R.menu.display_contact, menu);
        }
        else{
            getMenuInflater().inflate(R.menu.main, menu);
        }
    }
    return true;
}

public boolean onOptionsItemSelected(MenuItem item)
{
    super.onOptionsItemSelected(item);
    switch(item.getItemId())
    {
        case R.id.Edit_Contact:
            Button b = (Button)findViewById(R.id.button1);
            b.setVisibility(View.VISIBLE);
            name.setEnabled(true);
            name.setFocusableInTouchMode(true);
            name.setClickable(true);

            phone.setEnabled(true);
            phone.setFocusableInTouchMode(true);
            phone.setClickable(true);

            email.setEnabled(true);
            email.setFocusableInTouchMode(true);
            email.setClickable(true);

```

```

street.setEnabled(true);
street.setFocusableInTouchMode(true);
street.setClickable(true);

place.setEnabled(true);
place.setFocusableInTouchMode(true);
place.setClickable(true);

return true;
case R.id.Delete_Contact:

AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setMessage(R.string.deleteContact)
.setPositiveButton(R.string.yes, new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        mydb.deleteContact(id_To_Update);
        Toast.makeText(getApplicationContext(), "Deleted Successfully",
Toast.LENGTH_SHORT).show();
        Intent intent = new
Intent(getApplicationContext(),com.example.addressbook.MainActivity.class);
        startActivity(intent);
    }
})
.setNegativeButton(R.string.no, new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        // User cancelled the dialog
    }
});
AlertDialog d = builder.create();
d.setTitle("Are you sure");
d.show();

return true;
default:
return super.onOptionsItemSelected(item);

}
}

public void run(View view)
{
    Bundle extras = getIntent().getExtras();
    if(extras !=null)
    {

```



```

        int Value = extras.getInt("id");
        if(Value>0){
            if(mydb.updateContact(id_To_Update,name.getText().toString(),
phone.getText().toString(), email.getText().toString(), street.getText().toString(),
place.getText().toString())){
                Toast.makeText(getApplicationContext(), "Updated",
Toast.LENGTH_SHORT).show();
                Intent intent = new
Intent(getApplicationContext(),com.example.addressbook.MainActivity.class);
                startActivity(intent);
            }
            else{
                Toast.makeText(getApplicationContext(), "not Updated",
Toast.LENGTH_SHORT).show();
            }
        }
        else{
            if(mydb.insertContact(name.getText().toString(), phone.getText().toString(),
email.getText().toString(), street.getText().toString(), place.getText().toString())){
                Toast.makeText(getApplicationContext(), "done", Toast.LENGTH_SHORT).show();
            }
            else{
                Toast.makeText(getApplicationContext(), "not done",
Toast.LENGTH_SHORT).show();
            }
            Intent intent = new
Intent(getApplicationContext(),com.example.addressbook.MainActivity.class);
            startActivity(intent);
        }
    }
}
}

```

MainActivity.java

```

package com.example.addressbook;

import java.util.ArrayList;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuItem;

```

```

import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class MainActivity extends Activity {
    public final static String EXTRA_MESSAGE = "com.example.AddressBook.MESSAGE";

    private ListView obj;
    DBHelper mydb;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mydb = new DBHelper(this);
        ArrayList array_list = mydb.getAllCotacts();

        ArrayAdapter arrayAdapter =
            new ArrayAdapter(this, android.R.layout.simple_list_item_1, array_list);

        //adding it to the list view.
        obj = (ListView)findViewById(R.id.listView1);
        obj.setAdapter(arrayAdapter);

        obj.setOnItemClickListener(new OnItemClickListener(){

            @Override
            public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
            long arg3) {
                // TODO Auto-generated method stub
                int id_To_Search = arg2 + 1;
                Bundle dataBundle = new Bundle();
                dataBundle.putInt("id", id_To_Search);
                Intent intent = new
                Intent(getApplicationContext(), com.example.addressbook.DisplayContact.class);
                intent.putExtras(dataBundle);
                startActivity(intent);
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {

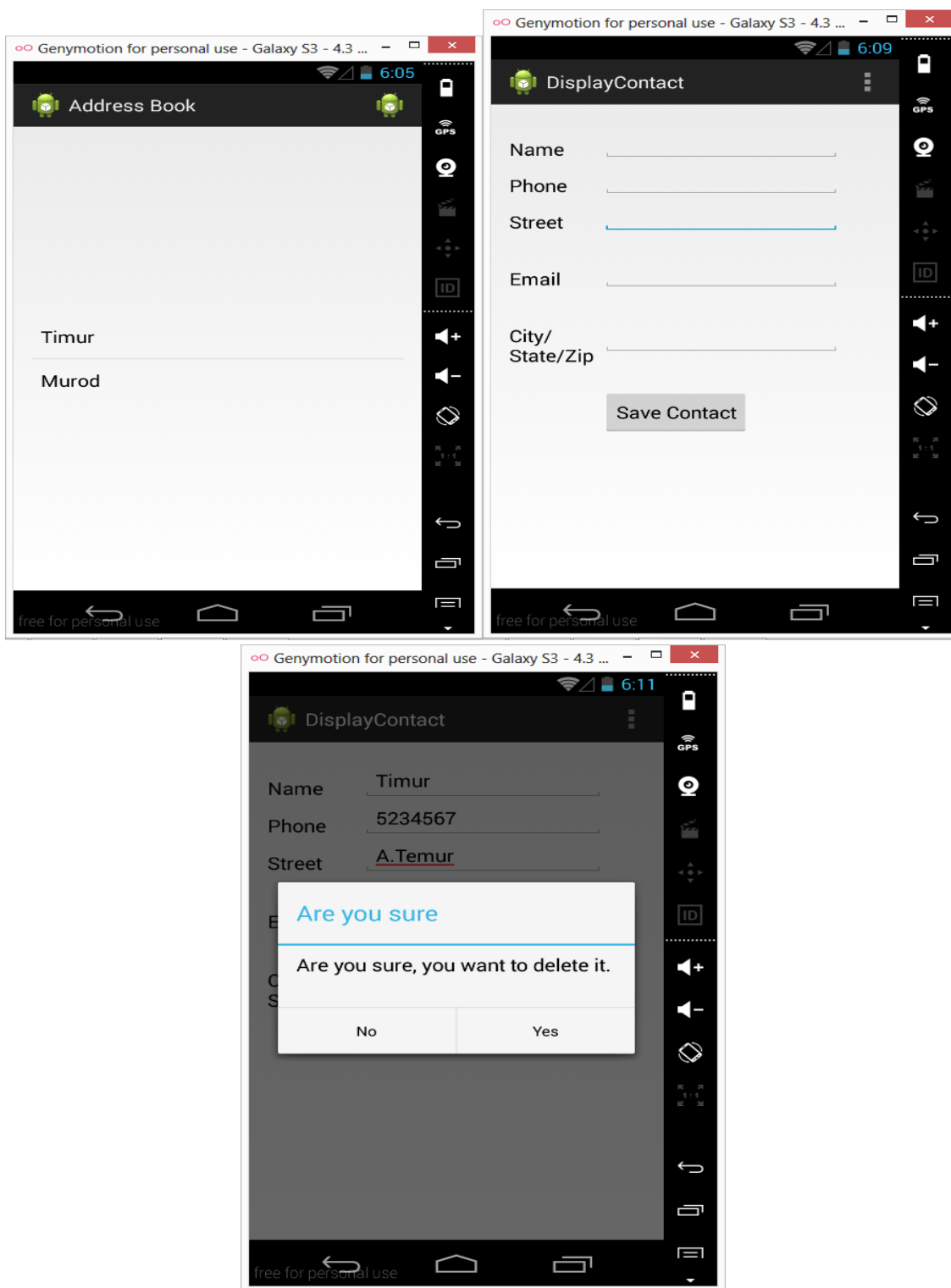
```

```

// Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.mainmenu, menu);
return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    super.onOptionsItemSelected(item);
    switch(item.getItemId())
    {
        case R.id.item1:
            Bundle dataBundle = new Bundle();
            dataBundle.putInt("id", 0);
            Intent intent = new
Intent(getApplicationContext(),com.example.addressbook.DisplayContact.class);
            intent.putExtras(dataBundle);
            startActivity(intent);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
public boolean onKeyDown(int keycode, KeyEvent event) {
    if (keycode == KeyEvent.KEYCODE_BACK) {
        moveTaskToBack(true);
    }
    return super.onKeyDown(keycode, event);
}
}

```

Дастур натижаси



Назорат саволлари:

1. Андроид иловалари маълумотлар базасини ташкил қилишнинг қандай турлари мавжуд?
2. Sharedpreferences объекти орқали сақланган маълумотлар қандай кўринишда сақланади?
3. Sharedpreferences объекти орқали илованинг қандай маълумотлари сақланади?
4. Илова маълумотларини файлга ёзишда қандай оқимли синфлардан фойдаланилади?
5. Илова маълумотларини файлдан ўқиш олишда қандай оқимли синфлардан фойдаланилади?
6. Андроид тизимида internal ва external сақловчилар деганда нимани тушунасиз?
7. SQLiteOpenHelper синфининг асосий вазифаси?
8. SQLiteDatabase синфининг асосий вазифаси?
9. SQLiteDatabase синфининг қандай методлари мавжуд?
10. *Content providers* нима ва у қачон ишлатилади?
11. Android тизимида мавжуд content provider ларга мисол келтиринг?
12. Content provider орқали маълумот ўқиш сўрови стандарт кўриниши қандай?
13. Content provider лар учун сўровларга мисоллар келтиринг?

Фойдаланилган адабиётлар:

1. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-6. Data Persistence
2. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-7. Content Providers
3. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-9. Location based services
4. <http://www.tutorialspoint.com/sqlite/>
5. http://www.tutorialspoint.com/sqlite/sqlite_installation.htm
6. http://www.tutorialspoint.com/sqlite/sqlite_create_database.htm
7. http://www.tutorialspoint.com/android/android_internal_storage.htm

5 – амалий машғулот.

Google maps хизматидан фойдаланиш. Фойдаланувчи жойлашган ўрнини аниқлаш. (2 соат)

Ишдан мақсад: Android тизимида Google maps хизматидан фойдаланиш ва фойдаланувчи жойлашган ўрнини аниқлаш ва уни хабар сифатида жўнатиш иловасини ишлаб чиқиш.

Масаланинг қўйилиши: Тингловчи яшаш жойини Google map хизматидан фойдаланиб аниқлаши ва киритилган **Longitude** ва **Latitude** кийматларини Google map орқали кўрсатиши лозим:

Ишни бажариш учун намуна¹

1. Янги Android лойиҳа яратамизг ва уни **LBS** деб номлаймиз.
2. /res.layout папкасида **main.xml** номли файлни қуйидагича ўзгартирамиз
3. package ичида **LBSActivity.java** номли синф қуйидагича ўзгартирилади
4. Лойиҳанинг **AndroidManifest.xml** файли қуйидагича ўзгартирилади

Изоҳ: Лойиҳа учун Google API калитини олиш учун қуйидаги ссилкага кирамиз

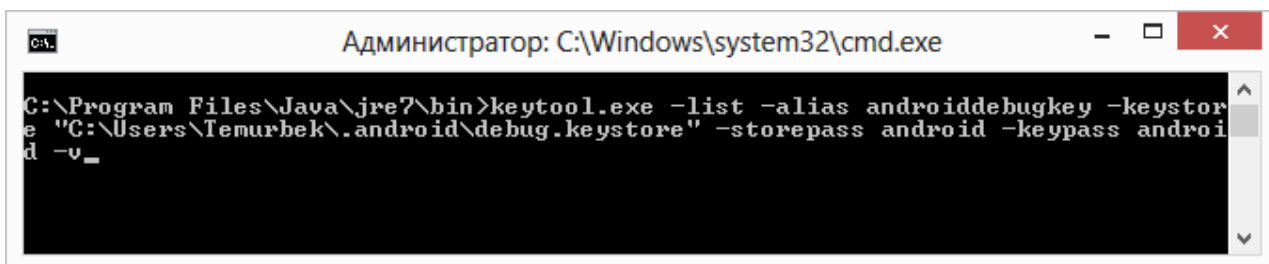
<https://code.google.com/apis/console>

Ушбу ссилкага кириб, SHA1 коди киритилади ва жорий лойиҳа учун Google API калитини генерация қилиб беради.

SHA1 коди қуйидагича аниқланади.

cmd.exe ойнасига кирилади ва c:\Program Files\Java\jre7\bin га кириб қуйидаги командани терамиз ва “Enter” тугмасини босамиз:

```
► keytool.exe -list -alias androiddebugkey -keystore  
"C:\Users\<username>\.android\debug.keystore" -storepass android -  
keypass android -v
```



```
Администратор: C:\Windows\system32\cmd.exe  
C:\Program Files\Java\jre7\bin>keytool.exe -list -alias androiddebugkey -keystore  
e "C:\Users\Temurbek\.android\debug.keystore" -storepass android -keypass androi  
d -v
```

¹ “Beginning Android™ 4 Application Development”. Chapter-9. Location-Based Services, pages 380-391.

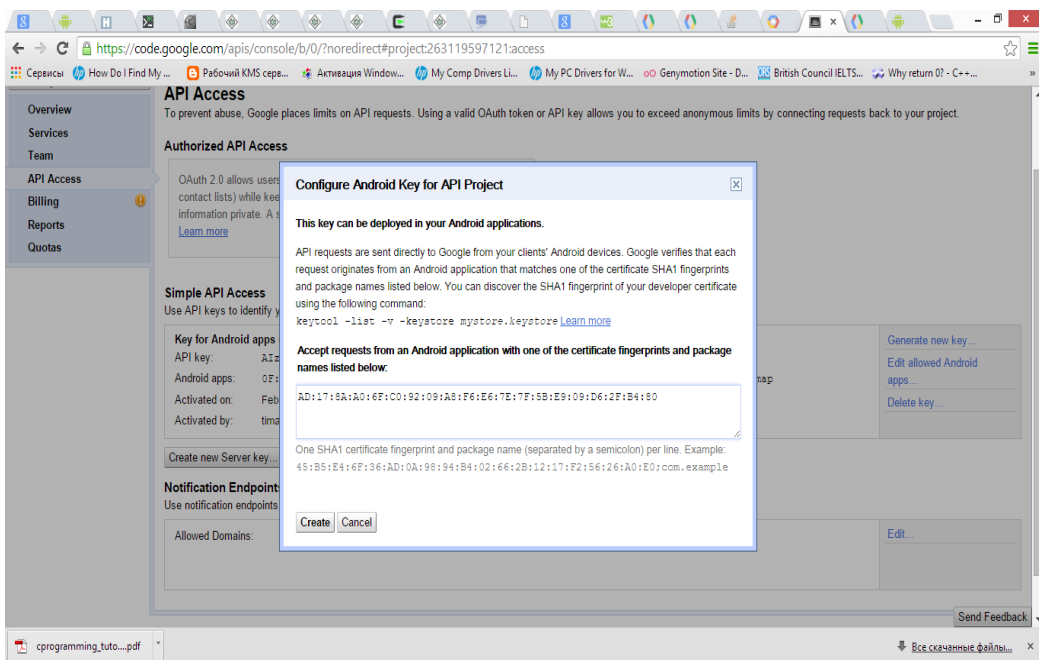
```

Администратор: C:\Windows\system32\cmd.exe

C:\Program Files\Java\jre7\bin>keytool.exe -list -alias androiddebugkey -keystore
"C:\Users\Temurbek\.android\debug.keystore" -storepass android -keypass androi
d -v
Alias name: androiddebugkey
Creation date: 25.02.2014
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 709bdab
Valid from: Tue Feb 25 20:57:50 UZT 2014 until: Thu Feb 18 20:57:50 UZT 2044
Certificate fingerprints:
MD5: 83:8B:8C:FB:BC:E3:6B:67:6C:00:18:B9:3E:DE:C2:7C
SHA1: AD:17:8A:AD:6F:00:92:09:A8:F6:E6:7E:7F:5B:E9:09:D6:2F:B4:80
SHA256: FA:CE:9A:21:2C:15:12:FC:AB:A9:13:95:E9:91:0F:40:F4:31:5B:84:92:
6C:4E:34:C7:C2:99:29:3D:72:D7:30
Signature algorithm name: SHA256withRSA
Version: 3

```

Натижада юқоридаги расмда кўрсатилганидек, MD5 ва SHA1 кодлари ҳосил бўлади. Бундан SHA1 коди ёзиб олинади ва юқорида кўрсатилганидек сайтга кириб жорий илова учун API калит генерация қилинади. Изоҳ: <https://code.google.com/apis/console> сайтга мурожаат қилганда Google Account яъни gmail.com доменида рўйхатдан ўтган бўлиши лозим



/res/layout/main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
<com.google.android.maps.MapView
    android:id="@+id/mapView"
    android:layout_width="fill_parent"

```

```
android:layout_height="fill_parent"
android:enabled="true"
android:clickable="true"
android:apiKey="AIzaSyDqPM4gZOiCClplSzC6E2R0lnpAnng5EgI" />
</LinearLayout>
```

LBSActivity.java

```
package номи
import com.google.android.maps.GeoPoint;
...// бошқа синфлар импорт қилинади
public class LBSActivity extends MapActivity {
    MapView mapView;
    MapController mc;
    GeoPoint p;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        mapView = (MapView) findViewById(R.id.mapView);
        mapView.setBuiltInZoomControls(true);
        mapView.setSatellite(true);
        mapView.setTraffic(true);
        mc = mapView.getController();
        String coordinates[] = {"1.352566007", "103.78921587"};
        double lat = Double.parseDouble(coordinates[0]);
        double lng = Double.parseDouble(coordinates[1]);
        p = new GeoPoint(
            (int) (lat * 1E6),
            (int) (lng * 1E6));
        mc.animateTo(p);
        mc.setZoom(13);
        mapView.invalidate();
    }
}
```

AndroidManifest.java

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="net.learn2develop.LBS"
android:versionCode="1"
android:versionName="1.0" >
<uses-sdk android:minSdkVersion="14" />
```



```

<uses-permission android:name="android.permission.INTERNET"/>
<application
android:icon="@drawable/ic_launcher"
android:label="@string/app_name" >
<uses-library android:name="com.google.android.maps" />
<activity
android:label="@string/app_name"
android:name=".LBSActivity" >
<intent-filter >
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
</manifest>

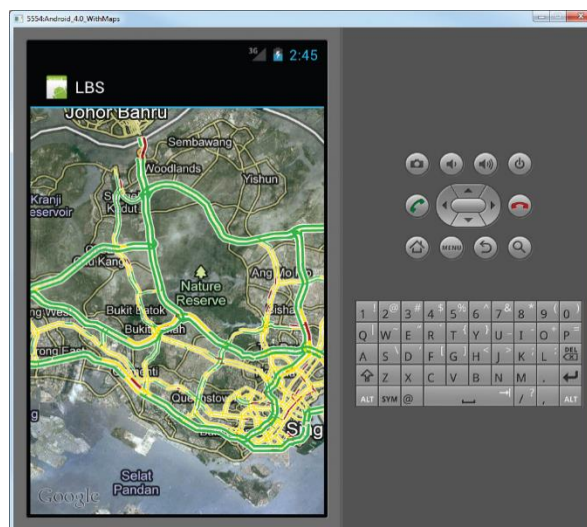
```

Дастур натижаси

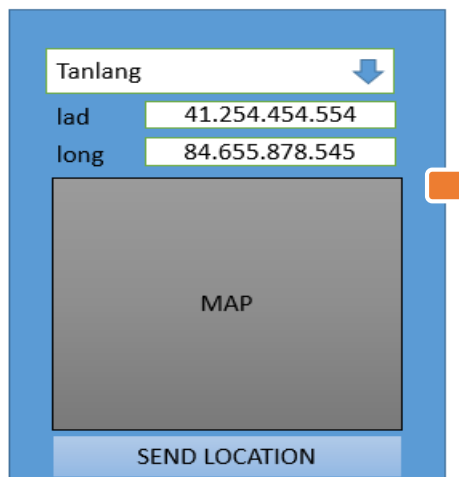
Илова ишга тушганда биринчи бўлиб очиладиган location

Latitude = 1.352566007

Longitude = 103.78921587



Топшириқ. Қуйидаги интерфейсга эга бўлган мобил илова ишлаб чиқинг. Дастур интерфейси учун намуна



Назорат саволлари:

1. Андроид иловаларида LBS хизмати нима?
2. Андроид ОТ учун API лардаги “Google API” билан “SDK Platform” нинг бир-биридан фарқи нимада?
3. Лойиҳанинг Map API калитини ўрнатиш кетма-кетлиги қандай?
4. Андроид иловаларининг калитлари қаерда сақланади ва улар қандай генерация қилинади?
5. Жорий лойиҳа учун олинган Google API калитидан қандай фойдаланилади?
6. MapController синфи ва унинг методларидан фойдаланиш? Мисол келтиринг?
7. Cell tower triangulation лар ва улардан фойдаланиш?
8. MapView синфи ва унинг методларидан фойдаланиш? Мисол келтиринг?

Фойдаланилган адабиётлар:

1. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-6. Data Persistence
2. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-7. Content Providers
3. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-9.
4. http://www.tutorialspoint.com/android/android_google_maps.htm
5. <https://developers.google.com/maps/documentation/android-api/start>

6 – амалий машғулот.

Мобил иловаларда тармоқли дастурлаш. Сервер билан ишлаш. JSON хизмати. (4 соат)

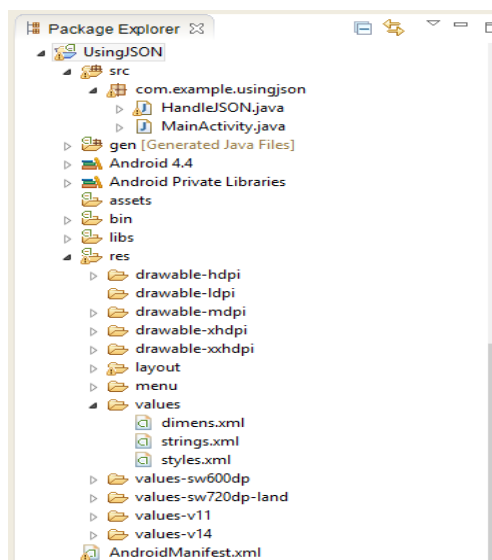
Ишдан мақсад: Андроид тизимида фойдаланувчи интерфейсини яратиш ва иловага мос маълумотларни сервердан ўқиб олиш, JSON хизматидан фойдаланиш кўникмаларига эга бўлиш

Масаланинг қўйилиши: Тингловчи вариант бўйича берилган лойиҳани Андроид тизимида ишлаб чиқиш ва илова учун керакли бўлган маълумотларни json хизмати орқали сервердан ўқиб олиши ва emulator орқали натижа олиши лозим.

Ишни бажариш учун наъмуна

JSON дан фойдаланиш¹

1. Янги Андроид лойиҳа яратамиз ва уни UsingJSON деб номлаймиз
2. Лойиҳа қуйидаги файллардан иборат ва уларни қуйидаги слайдларда кўрсатилганидек ўзгартирамиз
3. res/layout/activity_main.xml
4. res/values/string.xml
5. AndroidManifest.xml
6. src/package/MainActivity.java
7. src/package/HandleJSON.java



res/layout/activity_main.xml

```
RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
```

¹ http://www.tutorialspoint.com/android/android_json_parser.htm
“Beginning Android™ 4 Application Development”. Chapter-10. Networking, pages 437-443.

```

android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".MainActivity" >
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="15dp"
    android:text="@string/location"
    android:textAppearance="?android:attr/textAppearanceMedium" />
<EditText
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/textView1"
    android:layout_alignParentRight="true"
    android:ems="10" />
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView1"
    android:layout_below="@+id/textView1"
    android:layout_marginTop="68dp"
    android:text="@string/country"
    android:textAppearance="?android:attr/textAppearanceSmall" />
<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView2"
    android:layout_marginTop="19dp"
    android:text="@string/temperature"
    android:textAppearance="?android:attr/textAppearanceSmall" />
<TextView
    android:id="@+id/textView4"

```

```

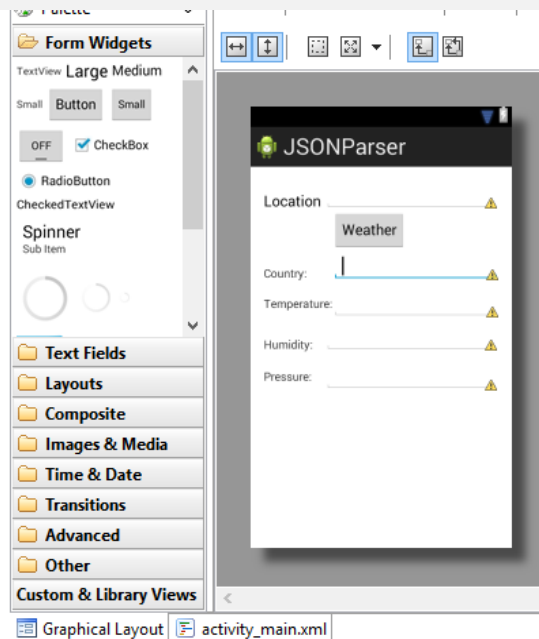
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView3"
    android:layout_below="@+id/textView3"
    android:layout_marginTop="32dp"
    android:text="@string/humidity"
    android:textAppearance="?android:attr/textAppearanceSmall" />
<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView4"
    android:layout_below="@+id/textView4"
    android:layout_marginTop="21dp"
    android:text="@string/pressure"
    android:textAppearance="?android:attr/textAppearanceSmall" />
<EditText
    android:id="@+id/editText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/textView3"
    android:layout_toRightOf="@+id/textView3"
    android:ems="10" >
    <requestFocus />
</EditText>
<EditText
    android:id="@+id/editText3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/textView3"
    android:layout_alignBottom="@+id/textView3"
    android:layout_alignLeft="@+id/editText2"
    android:ems="10" />
<EditText
    android:id="@+id/editText4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/textView5"
    android:layout_alignLeft="@+id/editText1"
    android:ems="10" />
<EditText
    android:id="@+id/editText5"

```

```

android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/textView5"
android:layout_alignBottom="@+id/textView5"
android:layout_alignRight="@+id/editText4"
android:ems="10" />
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/editText2"
    android:layout_below="@+id/editText1"
    android:onClick="open"
    android:text="@string/weather" />
</RelativeLayout>

```



res/values/string.xml

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">JSONParser</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>
    <string name="location">Location</string>
    <string name="country">Country:</string>
    <string name="temperature">Temperature:</string>
    <string name="humidity">Humidity:</string>
    <string name="pressure">Pressure:</string>

```

```
<string name="weather">Weather</string>
</resources>
```

src/package/MainActivity.java

```
public class MainActivity extends Activity {
    private String url1 = "http://api.openweathermap.org/data/2.5/weather?q=";
    private EditText location, country, temperature, humidity, pressure;
    private HandleJSON obj;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        location = (EditText)findViewById(R.id.editText1);
        country = (EditText)findViewById(R.id.editText2);
        temperature = (EditText)findViewById(R.id.editText3);
        humidity = (EditText)findViewById(R.id.editText4);
        pressure = (EditText)findViewById(R.id.editText5);
    }
    public void open(View view){
        String url = location.getText().toString();
        String finalUrl = url1 + url;
        country.setText(finalUrl);
        obj = new HandleJSON(finalUrl);
        obj.fetchJSON();
        while(obj.parsingComplete);
        country.setText(obj.getCountry());
        temperature.setText(obj.getTemperature());
        humidity.setText(obj.getHumidity());
        pressure.setText(obj.getPressure());
    }
}
```

src/package/HandleJSON.java

```
public class HandleJSON {
    private String country = "county";
    private String temperature = "temperature";
    private String humidity = "humidity";
    private String pressure = "pressure";
    private String urlString = null;
    public volatile boolean parsingComplete = true;
    public HandleJSON(String url){
```

```

    this.urlString = url;
}
public String getCountry(){
    return country;
}
public String getTemperature(){
    return temperature;
}
public String getHumidity(){
    return humidity;
}
public String getPressure(){
    return pressure;
}
@SuppressLint("NewApi")
public void readAndParseJSON(String in) {
    try {
        JSONObject reader = new JSONObject(in);
        JSONObject sys = reader.getJSONObject("sys");
        country = sys.getString("country");
        JSONObject main = reader.getJSONObject("main");
        temperature = main.getString("temp");
        pressure = main.getString("pressure");
        humidity = main.getString("humidity");
        parsingComplete = false;
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
public void fetchJSON(){
    Thread thread = new Thread(new Runnable(){
        @Override
        public void run() {
            try {
                URL url = new URL(urlString);
                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
                conn.setReadTimeout(10000 /* milliseconds */);
                conn.setConnectTimeout(15000 /* milliseconds */);
                conn.setRequestMethod("GET");
                conn.setDoInput(true);
                // Starts the query

```



```

        conn.connect();
        InputStream stream = conn.getInputStream();
        String data = convertStreamToString(stream);
        readAndParseJSON(data);
        stream.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
});
thread.start();
}
static String convertStreamToString(java.io.InputStream is) {
    java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");
    return s.hasNext() ? s.next() : "";
}
}

```

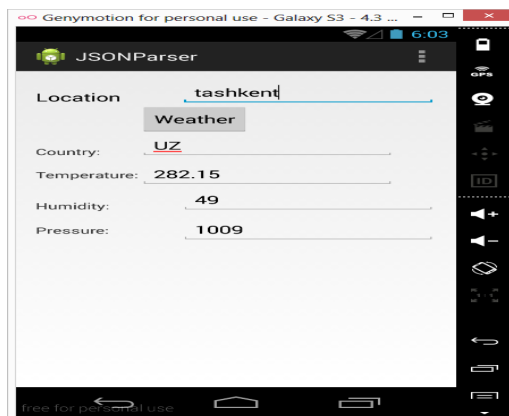
AndroidManifest.xml файлига permission қуйидагича қўшилади

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.usingjson"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="18" />
        <uses-permission android:name="android.permission.INTERNET"/>
    <application>..... </application>
</manifest>

```

Дастур натижаси



Назорат саволлари:

1. Мобил иловаларда тармоқли дастурлаш?
2. ConnectivityManager синфи нима учун ишлатилади ва у қандай инициализация қилинади?
3. Андроид қурилмасининг тармоққа уланганлик ҳолатини текшириш учун қандай констатнталардан фойдаланилади?
4. HttpURLConnection синфи таснифини келтиринг?
5. HttpURLConnection синфининг қандай методлари мавжуд?
6. Илова интернетга боғланиши учун манифест файлида қандай рухсат эълон қилиниши зарур?
7. JSON нима вақачон ишлатилади?
8. JSON хизматининг асосий синфларини айтинг ва уларга изоҳ беринг?
9. JSON объекти қандай компонентлардан иборат бўлади?
10. JSON кўринишдаги маълумотга мисол кетиринг ва бўлакларга бўлиш қандай амалга оширилади?

Фойдаланилган адабиётлар:

1. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-10. Networking
2. “Beginning Android™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533. Chapter-10. Consuming JSON services
3. <http://www.tutorialspoint.com/json/>
4. http://www.tutorialspoint.com/android/android_json_parser.htm
5. <http://beginnersbook.com/2015/04/json-tutorial/>

V БЎЛИМ

КЕЙСЛАР БАНКИ

V. КЕЙСЛАР БАНКИ

1-Кейс. Мобил қурилма учун Андроид опреацион тизимининг 5.0 (*API Level*: 21) версияси учун илова ишлаб чиқилди. Сизнинг телефонингиздаги Андроид опреацион тизимининг версияси 4.3 (*API Level*: 18). Мобил иловани телефонингизга ўрнатиб ишга туширмоқчи бўлганингизда хатолик келиб чиқди. Яъни илова ишламади.

Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Мобил иловани ишга тушириш учун бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

2-Кейс. Илова ишлаб чиқилди (Андроид опреацион тизимининг версияси 4.4 *API Level*: 19) ва у ишлаши жараёнида битта ойнадан иккинчи ойнага ўтишда хатолик келиб чиқди. Яъни иловада ойналарни бошқариш учун иккита *activity* мавжуд ва бир ойнадан иккинчи ойнага ўтишда *activity* чақирилганда хатолик бўлди.

Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабаблар ва ҳал этиш йўллари ядвал асосида изоҳланг (индивидуал ва кичик гуруҳда).

| Муаммо тури | Келиб чиқиш сабаблари | Ҳал этиш йўллари |
|-------------|-----------------------|------------------|
| | | |

3-Кейс. Илова ишлаб чиқилди (Андроид опреацион тизимининг версияси 4.4 *API Level*: 19) ва иловада Галлерея ва расмлардан фойдаланилган. Мазкур ишлаб чиқилган иловани турли кўринишдаги қурилмаларда (таблет, смартфон ва ҳ.к., изоҳ: қурилма экранларининг ўлчамлари ҳар хил) ишга туширганимизда расмлар ўлчами қурилма экранининг ўлчамига мос тушмайди. Мисол учун илова компоненталари

Samsung S3 қурилмасида яхши кўринади лекин, Tablet Nexus 10 қурилмасида эса кичкина ёки экраннинг бир бурчагига жойлашиб қолади. Яъни ишлаб чиқилган илова дизайни барча қурилмалар учун стандарт эмас.

Кейснi бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Яъни ишлаб чиқилган илова дизайни барча қурилмалар учун стандарт бўлишини таъминлашда бажариладиган ишлар кетма-кетлигини белгиланг (жуфтликлардаги иш)

4-Кейс. Мобил илова ишлаб чиқилди ва ушбу мобил илованинг маълумотлар базаси ҳам мавжуд. Шундай ҳолат юзага келдики, бунда сизнинг ишлаб чиққан иловангизни бошқа қурилмага ўтказиб ишга туширганда маълумотлар базасига уланиш бўйича хатолик келиб чиқди. Яъни иловага тегишли маълумотларни ўқиш учун маълумотлар базасида хатолик бўлди ва илова ишламади.

Кейснi бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабаблар ва ҳал этиш йўлларини жадвал асосида изоҳланг (индивидуал ва кичик гуруҳда).

| Муаммо тури | Келиб чиқиш сабаблари | Ҳал этиш йўллари |
|-------------|-----------------------|------------------|
| | | |

5-Кейс. Мобил илова ишлаб чиқилди ва ушбу мобил илова ўзига керакли бўлган маълумотларни тармоқ орқали JSON хизмати асосида қабул қилади. Илова шундай ишлаб чиқилиши керакки, бунда охириги марта иловага мурожаат қилганда интернетдан юклаб олинган маълумотлар сақлаб қолинсин ва интернетга уланмаган ҳолда ҳам илова охириги юклаб олинган маълумотларни кўрсатиб берсин.

Мисол учун об-ҳавони кўрсатиб бериш учун мўлжалланган илова мавжуд бўлиб, илова охириги марта 2 кун олдин интернетга уланган ва сўнги

маълумотни юклаб олган. Жорий ҳолатда эса интернет мавжуд эмас, лекин шунга қарамасдан 2 кун олдинги маълумотни яъни об-ҳаво маълумотини кўрсатиб берсин.

Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- Илова интернетга уланмаган ҳолда ҳам ўзига керакли бўлган охириги маълумотларни кўрсатиб беришини таъминланг ва ушбу муаммони ҳал қилиш учун ечимни белгиланг (жуфтликлардаги иш).

6-Кейс. Мобил илова ишлаб чиқилди ва ушбу мобил иловада расм учун анимация белгиланган. Бунда расм кўрсатилган траектория бўйича чексиз такрорланган ҳолда ҳаракатда бўлади. Лекин шундай зарур ҳолат бўлдики, жорий иловани орқа фонга ўтказиб туриш ва бошқа иловани ишга тушириш керак бўлди. Бу ҳолатда жорий анимацияли иловани орқа фонга ўтказганда ундаги объектлар анимациясини тўхтатиш лозим бўлади, бўлмаса илова курилма оператив хотирасидан ортикча фойдаланади. Яъни ушбу анимацияга эга илова орқа фонга ўтганда курилма оператив хотирасидан ортикча жой эгаллаб туради. Ушбу ҳолатда объектлардан анимацияни вақтинчалик тўхтатиб туриш талаб қилинади.

Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабаблар ва ҳал этиш йўлларини жадвал асосида изоҳланг (индивидуал ва кичик гуруҳда).

| Муаммо тури | Келиб чиқиш сабаблари | Ҳал этиш йўллари |
|-------------|-----------------------|------------------|
| | | |

7-Кейс. Ҳоҳлаган муҳитда мобил илова яратилганда унда камида битта асосий бўлган activity автоматик яратилади. Мисол учун Android Development Tools Eclipse IDE муҳитида *MainActivity.java* номли activity автоматик яратилади ва буда onCreate() номли метод “@Override” белгиси асосида мавжуд бўлади. Ушбу сўзнинг маъноси ва объектга йўналтирилган дастурлашнинг асосий тамойилларидан бири полиморфизмни ифодалаб бериш лозим бўлади.

Кейсни бажариш босқчилари ва топшириқлар:

- onCreate() номли методдаги “@Override” калит сўзини изоҳланг (индивидуал ҳолда).
- Объектга йўналтирилган дастурлашнинг полиморфизм тамойилидан фойдаланишга мисол келтиринг (индивидуал ҳолда).

8-Кейс. Илова ишлаб чиқилди ва унда маълумотларни сақлаш турларидан (SharedPreferences, File I/O, SQLite) бирдан фойдаланилган. Иловага тегишли маълумотларни ҳақиқатдан ҳам базага ёзилганлигини текшириш лозим бўлади ва баъзи ҳолатларда маълумотларни хато ёзиб қўйиш ҳолатлари кузатилади. Ушбу ҳолат учун мониторинг олиб бориш лозим бўлади.

Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммони келтириб чиқарган асосий сабабларни белгиланг (индивидуал ва кичик гуруҳда).
- DDMS мониторинг имкониятидан фойдаланиш ва иловага тегишли бўлган маълумотларни мавжуд ёки мавжуд эмаслигини текширинг (индивидуал ҳолда).

9-Кейс. Барча дастурлаш муҳитида дастур ишлаб чиқишда компиляция жараёнида хатоликларни кузатиш мумкин вабусиз дастур ишлаб чиқиш иложи йўқ. Лекин мобил илова иўлаб чиқиш вақтида илова натижаси виртуал машина орқали ишга туширилади ва иловада хатолик бўлса одатда “unfortunately does't work application.....” кўринишадаги хатолик келиб чиқади ва хатоликни айнан кўриш имконияти виртуал курилмада мавжуд эмас. Ушбу ҳолатда компиляция жараёнини мониторинг қилиш йўллари аниқлаш лозим бўлади.

Кейсни бажариш босқчилари ва топшириқлар:

- Кейсдаги муаммога мос ҳолда мобил илова ишлаб чиқишда компиляция жараёнини мониторинг қилиш йўллари аниқланг (индивидуал).
- Иловадаги хатоликларни филтрлаш ва уларни бошқариш имкониятларини кўрсатиб беринг (индивидуал ҳолда).

Амалий топшириқлар

I. Java тилида берилган мисолларни ечинг?

1. Комплекс сонлар устида амаллар бажаринг (Мисол учун: $2+3i, 3+4i \Rightarrow 5+7i$)?
2. Иккита вақт оралиғидаги фарқни топувчи дастур тузинг?
3. Берилган натурал сон квадратини топувчи дастурни қуйидаги қонуният бўйича тузинг:
 $1^2 = 1$
 $2^2 = 1 + 3$
 $3^2 = 1 + 3 + 5$
 $4^2 = 1 + 3 + 5 + 7$
.....
 $N^2 = 1 + 3 + 5 + 7 + \dots + (2N-1)$
4. Қуйидаги ифодани ҳисобланг
$$\frac{\cos 1}{\sin 1} * \frac{\cos 1 + \cos 2}{\sin 1 + \sin 2} * \dots * \frac{\cos 1 + \cos 2 + \dots + \cos n}{\sin 1 + \sin 2 + \dots + \sin n};$$

II. Синфлар ва объектлардан фойдаланиб дастурлаш. ОЙД принципларидан (ворислик, полиморфизм) фойдаланинг

1. Шакл, (айлана, тўртбурчак, учбурчак)
2. Мавжудот, (одам, ҳайвон, ҳашорат)
3. Транспорт (енгил авто, оғир авто, сувтранспорти, ҳавотранспорти)
4. Спортчи (футболчи, боксчи, теннисчи)

III. Қуйидаги берилган компоненталардан фойдаланиб содда Андроид иловаси учун интерфейс яратинг

1. Биринчи ойнада 3 button ва иккинчи ойнада 2 та textview, 2 та button мавжуд илова интерфейсини яратинг. Дастурнинг xml файллари таснифи келтирилсин
2. 3 та textview, 2 та radiobutton ва 1 та progressbar қўйилган илова интерфейсини яратинг. Дастурнинг xml файли таснифи келтирилсин.
3. onCreate(), onStart() ва onResume() методларини ишлатган ҳолда ҳар бир метод чақирилганда натижа (бирон бир хабар берувчи) чиқувчи дастур тузинг
4. Биринчи ойнада 2 та autoCompleteTextView, 2 та editText ва 2 та button қўйилган илова интерфейсини яратинг. Дастурнинг xml файли таснифи келтирилсин.

IV. Қуйидаги берилган вариантлар бўйича Галлерия иловасини яратинг

1. Students Gallery иловасини ишлаб чиқиш
2. Computers Gallery иловасини ишлаб чиқиш
3. Books Gallery иловасини ишлаб чиқиш
4. Football Stars Gallery иловасини ишлаб чиқиш
5. Cars Gallery иловасини ишлаб чиқиш
6. iPhones Gallery иловасини ишлаб чиқиш
7. Cities Gallery иловасини ишлаб чиқиш
8. Airplanes Gallery иловасини ишлаб чиқиш

VI БЎЛИМ

ГЛОССАРИЙ

VII. ГЛОССАРИЙ

| Термин | Ўзбек тилидаги шарҳи | Инглиз тилидаги шарҳи |
|----------------------------|---|--|
| .apk file | андроид илованинг ўрнатиладиган файли | Android application package file. Each Android application is compiled and packaged in a single file that includes all of the application's code (.dex files), resources, assets, and manifest file. |
| Action | иловада Intent орқали жўнатилувчи хабар | A description of something that an Intent sender wants done. An action is a string value assigned to an Intent. |
| Intent | иловада ойналар/activity бўйлаб жўнатилувчи хабар объекти | An message object that you can use to launch or communicate with other applications/activities asynchronously. |
| Intent Filter | манифест файлида эълон қилинадиган интент объектини филтрловчи объект | A filter object that an application declares in its manifest file, to tell the system what types of Intents each of its components is willing to accept and with what criteria. |
| Activity | илованинг биронта ойнаси (интерфейс) бошқарувчи Java файл | A single screen in an application, with supporting Java code, derived from the Activity class. |
| adb (Android Debug Bridge) | SDK орқали иловани ишлатувчи дастур (debugger) | Android Debug Bridge, a command-line debugging application included with the SDK. |

| | | |
|-------------------------------------|--|--|
| SDK (Software Development Kit) | андроид учун кутубхона | A set of tools that can be used to develop software applications targeting a specific platform such as Facebook or Chartboost. For example, the Facebook SDK allows developers to add Facebook features into an app. |
| JDK (Java Development Kit) | Java дастурлаш тили учун кутубхона | The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets. |
| Application | бу ишлаб чиқилган илова | From a component perspective, an Android application consists of one or more activities, services, listeners, and intent receivers. |
| Canvas | иловада чизилган ёки юклаб олинган тасвир | A drawing surface that handles compositing of the actual bits against a Bitmap or Surface object. |
| Dalvik | андроид платформасининг виртуал машинаси | The Android platform's virtual machine. |
| DDMS (Dalvik Debug Monitor Service) | SDK орқали илова ҳолатини график интерфейси орқали бошқариш учун дастур (debugger) | Dalvik Debug Monitor Service, a GUI debugging application included with the SDK. |
| Dialog | фойдаланувчи интерфейс учун мулоқот ойнаси | A floating window that acts as a lightweight form. |
| Drawable | иллованинг визуал ресурслари сақланувчи | A compiled visual resource that can be used as a |

| | | |
|--------------------|---|---|
| | папка | background, title, or other part of the screen. |
| Broadcast Receiver | иловага кириб келаётган ёки ундан жўнатилаётган интент объектини бошқарувчи илова синфи | An application class that listens for Intents that are broadcast, rather than being sent to a single target application/activity. |
| Layout Resource | илова ойналарининг кўринишларини сақловчи XML файл | An XML file that describes the layout of an Activity screen. |
| Manifest File | илова учун керакли барча маълумотларни сақловчи XML файл (мисол учун, илова номи, интент филтрлар, интернетга боғланиш) | An XML file that each application must define, to describe the application's package name, version, components (activities, intent filters, services), imported libraries, and describes the various activities, and so on. |
| Resources | илова учун керакли бўлган ресурслар (расм, аудио, видео ва бошқа файллар) | Nonprogrammatic application components that are external to the compiled application code, but which can be loaded from application code using a well-known reference format. |
| Service | иловаорти хизматлар яратиш учун синф | An object of class Service that runs in the background (without any UI presence) to perform various persistent actions, such as playing music or monitoring network activity. |
| Surface | Canvas объектини бошқариш учун объект | An object of type Surface representing a block of memory that gets |

| | | |
|-------------|---|--|
| | типи | composited to the screen. |
| SurfaceView | Canvas объектини бошқариш учун динамик объект | A View object that wraps a Surface for drawing, and exposes methods to specify its size and format dynamically. |
| View | илова интерфейсида фойдаланиладиган компоненталар (button, checkbox, listview, gallery ва бошқалар) | An object that draws to a rectangular area on the screen and handles click, keystroke, and other interaction events. |
| Viewgroup | илова интерфейсида фойдаланиладиган компоненталар гуруҳи ёки жамланмаси | A container object that groups a set of child Views. |
| Widget | илова интерфейсида фойдаланиладиган компоненталардан фойдаланиб ишлаб чиқилган график компонент | One of a set of fully implemented View subclasses that render form elements and other UI components, such as a text box or popup menu. |
| Window | иловани интерфейсларини яратиш учун абстракт синф | In an Android application, an object derived from the abstract class Window that specifies the elements of a generic window, such as the look and feel (title bar text, location and content of menus, and so on). |
| OpenGL ES | OpenGL ES кутубхонасидан мураккаб 3D расмларни тузишда фойдаланиш мумкин | Android provides OpenGL ES libraries that you can use for fast, complex 3D images. |
| .dex file | Компиляция қилинган Android дастури коди файли | Compiled Android application code file. |

| | | |
|-----------------------|---|--|
| Source code | Бу бошқа ишлаб чиқарувчилардан сотиб олиш мумкин бўлган framework ҳисобланади | It's a framework that you can buy from other developers and on marketplaces. |
| Universal app | Телефон қурилмаларида ҳам планшет қурилмаларида ҳам ишлайдиган илова | A single app that can run on both phone and tablet devices. |
| GUI | Фойдаланувчи график интерфейси | Graphic User Interface. |
| IDE | Қурилган ривожланиш муҳити | Integrated Development Environment. |
| OS (Operating System) | Операцион тизим. Қурилмадаги энг муҳим дастур | Operating System. The most important program on a device. |

VII БЎЛИМ

АДАБИЁТЛАР
РЎЙХАТИ

АДАБИЁТЛАР РЎЙХАТИ

I. Ўзбекистон Республикаси Президентининг асарлари

1. Каримов И.А. Ўзбекистон мустақилликка эришиш оstonасида. - Т.:“Ўзбекистон”, 2011.
2. Мирзиёев Ш.М. Буюк келажакимизни мард ва олижаноб халқимиз билан бирга қураимиз. – Т.: “Ўзбекистон”. 2017. – 488 б.
3. Мирзиёев Ш.М. Миллий тараққиёт йўлимизни қатъият билан давом эттириб, янги босқичга кўтарамиз – Т.: “Ўзбекистон”. 2017. – 592 б.

II. Норматив-ҳуқуқий ҳужжатлар

4. Ўзбекистон Республикасининг Конституцияси. – Т.: Ўзбекистон, 2019.
5. Ўзбекистон Республикасининг “Таълим тўғрисида”ги Қонуни.
6. Ўзбекистон Республикасининг “Коррупцияга қарши курашиш тўғрисида”ги Қонуни.
7. Ўзбекистон Республикаси Президентининг 2015 йил 12 июндаги “Олий таълим муассасаларининг раҳбар ва педагог кадрларини қайта тайёрлаш ва малакасини ошириш тизимини янада такомиллаштириш чора-тадбирлари тўғрисида”ги ПФ-4732-сонли Фармони.
8. Ўзбекистон Республикаси Президентининг 2017 йил 7 февралдаги “Ўзбекистон Республикасини янада ривожлантириш бўйича Ҳаракатлар стратегияси тўғрисида”ги 4947-сонли Фармони.
9. Ўзбекистон Республикаси Президентининг 2018 йил 3 февралдаги “Хотин-қизларни қўллаб-қувватлаш ва оила институтини мустаҳкамлаш соҳасидаги фаолиятни тубдан такомиллаштириш чора-тадбирлари тўғрисида”ги ПФ-5325-сонли Фармони.
10. Ўзбекистон Республикаси Президентининг 2019 йил 17 июндаги “2019-2023 йилларда Мирзо Улуғбек номидаги Ўзбекистон Миллий университетида талаб юқори бўлган малакали кадрлар тайёрлаш тизимини тубдан такомиллаштириш ва илмий салоҳиятини ривожлантириш чора-тадбирлари тўғрисида”ги ПҚ-4358-сонли Қарори.
11. Ўзбекистон Республикаси Президентининг 2019 йил 11 июлдаги «Олий ва ўрта махсус таълим тизимида бошқарувнинг янги тамойилларини жорий этиш чора-тадбирлари тўғрисида»ги ПҚ-4391- сонли Қарори.
12. Ўзбекистон Республикаси Президентининг 2019 йил 11 июлдаги «Олий ва ўрта махсус таълим соҳасида бошқарувни ислоҳ қилиш чора-тадбирлари тўғрисида»ги ПФ-5763-сон фармони.
13. Ўзбекистон Республикаси Президентининг 2019 йил 27 августдаги “Олий таълим муассасалари раҳбар ва педагог кадрларининг узлуксиз малакасини ошириш тизимини жорий этиш тўғрисида”ги ПФ-5789-сонли фармони.
14. Ўзбекистон Республикаси Президентининг “2019-2021 йилларда Ўзбекистон Республикасини инновацион ривожлантириш стратегиясини

тасдиқлаш тўғрисида”ги 2018 йил 21 сентябрдаги ПФ-5544-сонли Фармони.

15. Ўзбекистон Республикаси Президентининг 2019 йил 27 майдаги “Ўзбекистон Республикасида коррупцияга қарши курашиш тизимини янада такомиллаштириш чора-тадбирлари тўғрисида”ги ПФ-5729-сон Фармони.

16. Ўзбекистон Республикаси Президентининг 2017 йил 2 февралдаги “Коррупцияга қарши курашиш тўғрисида”ги Ўзбекистон Республикаси Қонунининг қоидаларини амалга ошириш чора-тадбирлари тўғрисида”ги ПҚ-2752-сонли қарори.

17. Ўзбекистон Республикаси Президентининг “Олий таълим тизимини янада ривожлантириш чора-тадбирлари тўғрисида”ги 2017 йил 20 апрелдаги ПҚ-2909-сонли қарори.

18. Ўзбекистон Республикаси Президентининг “Олий маълумотли мутахассислар тайёрлаш сифатини оширишда иқтисодиёт соҳалари ва тармоқларининг иштирокини янада кенгайтириш чора-тадбирлари тўғрисида”ги 2017 йил 27 июлдаги ПҚ-3151-сонли қарори.

19. Ўзбекистон Республикаси Президентининг “Нодавлат таълим хизматлари кўрсатиш фаолиятини янада ривожлантириш чора-тадбирлари тўғрисида”ги 2017 йил 15 сентябрдаги ПҚ-3276-сонли қарори.

20. Ўзбекистон Республикаси Президентининг “Олий таълим муассасаларида таълим сифатини ошириш ва уларнинг мамлакатда амалга оширилаётган кенг қамровли ислохотларда фаол иштирокини таъминлаш бўйича кўшимча чора-тадбирлар тўғрисида”ги 2018 йил 5 июндаги ПҚ-3775-сонли қарори.

21. Ўзбекистон Республикаси Вазирлар Маҳкамасининг 2012 йил 26 сентябрдаги “Олий таълим муассасалари педагог кадрларини қайта тайёрлаш ва уларнинг малакасини ошириш тизимини янада такомиллаштириш чора-тадбирлари тўғрисида”ги 278-сонли Қарори.

Ш. Махсус адабиётлар

1. “Beginning Андроид™ 4 Application Development”, by Wei-Meng Lee, printed at Radha Offset, Delhi in 2015, pages – 533.
2. “Android™ Application Development Cookbook”, by Wei-Meng Lee, printed at Sharda Offset Press, Delhi in 2015, pages – 385.
3. “Programming Андроид second edition” by Zigurd Mednieks, Laird Dornin, G. Blake Meike, and Masumi Nakamura, September 2014, pages – 564.
4. "Java The Complete Reference Eighth Edition" by Herbert Schildt, 2014, pages - 1048
5. Core Java “An Integrated Approach”, Include all versions up to Java7 by Dr. R. Nageswara Rao, 2015, 646 pages
6. “Introduction to Андроид Application Development Fourth edition” Developer’s library by Joseph Anuzzi, Jr. Lauren Darcey Shane Conder, November 2015.

7. “Андроид Wireless Application Development Volume I: Андроид Essentials, 3rd Edition”, Published Feb 23, 2014 by Addison-Wesley Professional, pages-449.
8. “Professional Андроид Application Development”, Reto Meier – 2014 Published by John Wiley & Sons, pages-787

Интернет сайтлар

1. <http://www.tutorialspoint.com/Android/index.htm>
2. <http://www.tutorialspoint.com/java/index.htm>
3. Андроид app course laboratory (<https://sites.google.com/site/Androidappcourse/labs/lab-7>)
4. <http://www.amazon.com/Introduction-Android-Application-Development-Essentials/dp/0321940261>
5. <http://www.amazon.com/Android-Application-Development-Rick-Rogers/dp/8184047339>
6. [http://vlab.wikispaces.asu.edu/file/view/\[Android _____\].Android.Application.Development.-.Programming.With.The.Google.Sdk.\(O'reilly,.2009,.0596521472\).pdf](http://vlab.wikispaces.asu.edu/file/view/[Android_____].Android.Application.Development.-.Programming.With.The.Google.Sdk.(O'reilly,.2009,.0596521472).pdf)
7. http://dl.e-book-free.com/2013/07/Android_application_development_cookbook.pdf