

№ 1 Установка и настройка среды

Изучение структуры проекта, работе в среде Android Studio

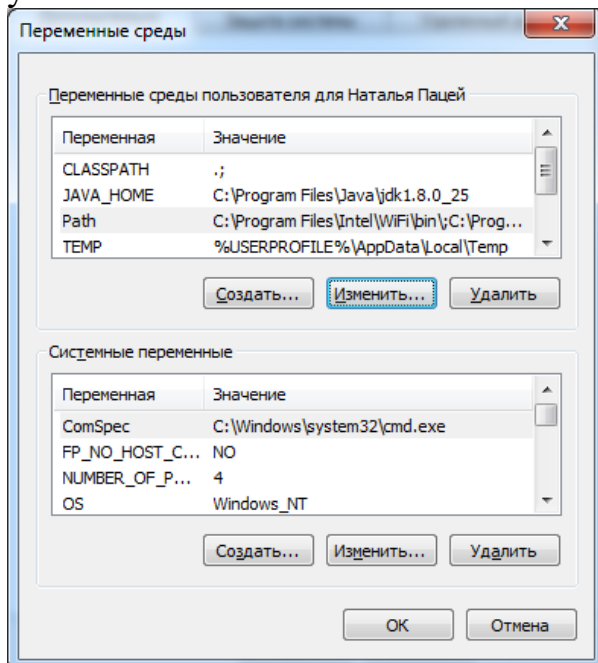
Задание

1 часть

- 1) Установите подходящую версию JDK.

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

- 2) Задайте пути к выполняемым файлам - системные переменные PATH, CLASSPATH и JAVA_HOME . **Панель управления -> Система и безопасность->Система**, выберете **Дополнительные параметры системы**. В появившемся окне нажмите кнопку **Переменные среды**. В открывшемся окне в блоке **Системные переменные** нужно проверить существует ли переменная **PATH**, если переменная не существует, то ее нужно создать нажав на кнопку **Создать**, если переменная PATH уже существует, то ее нужно **Изменить**. Для переменной PATH нужно установить **Значение переменной** — путь к каталогу, в который была установлена Java



Если переменной PATH уже присвоены другие значения, то новое значение добавляется через точку с запятой «;» в самом конце строки.

Также необходимо создать переменную **CLASSPATH**, если она еще не создана. В качестве ее значения необходимо указать «.;»

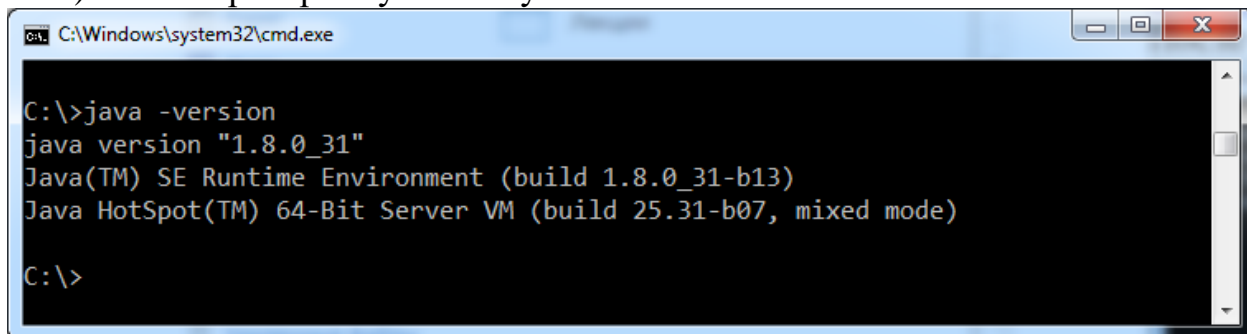
Подобным образом создается/изменяется переменная **JAVA_HOME**.

Значение этой переменной — путь к каталогу, в который была установлена Java (как для PATH), но уже без поддиректории bin. Например:

Значение в PATH: *C:\Program Files\Java\jdk1.8.1_xx\bin*

Значение в JAVA_HOME: *C:\Program Files\Java\jdk1.8.1_xx*

3) После проверьте установку:



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

C:\>java -version
java version "1.8.0_31"
Java(TM) SE Runtime Environment (build 1.8.0_31-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)

C:\>
```

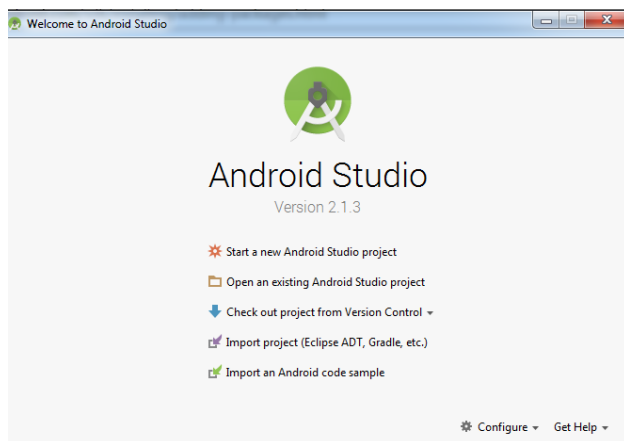
Изучите опции и параметры команд java, javac.

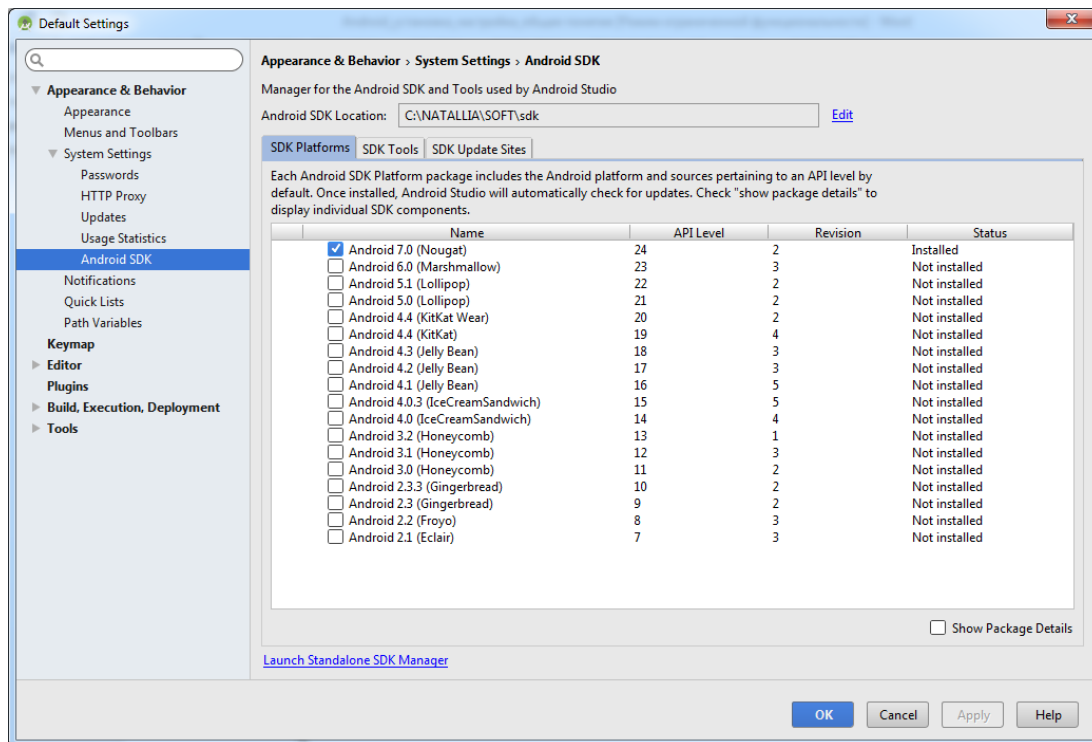
4) Установите Android Studio

<https://developer.android.com/studio/index.html>

5) Установите дополнительные компоненты.

Компоненты любой платформы можно получить при помощи Android SDK Manager. В Android Studio выполните команду Tools\Android\SDK Manager. (Меню Tools отображается только при наличии открытого проекта. Если вы еще не создали проект, SDK Manager также можно вызвать с экрана Welcome; выберите вариант Configure\SDK Manager.



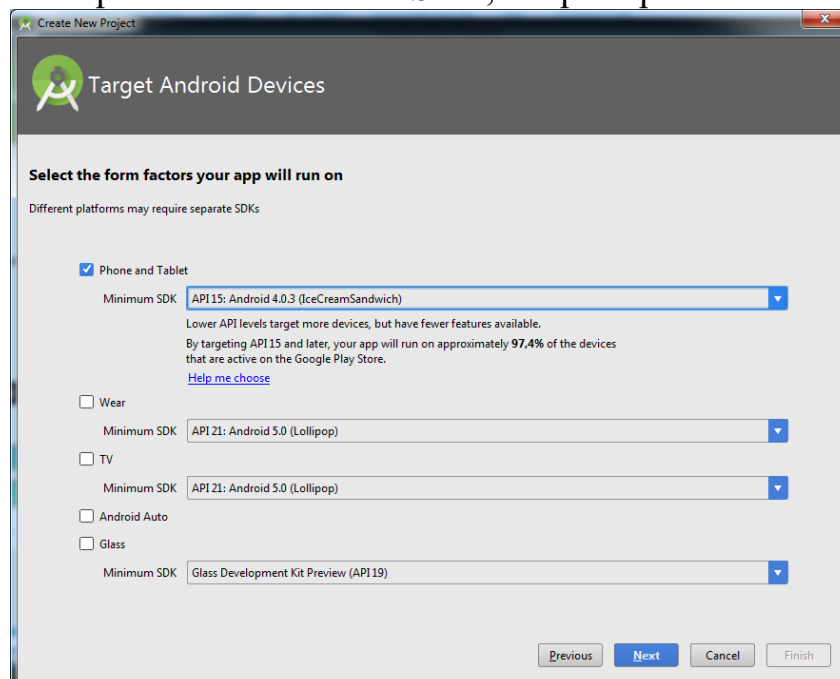


Выберите из SDK Platforms версии Android, которые вам понадобятся. Выберите из SDK Tools дополнительные инструменты в зависимости от устройств, которые вы будете использовать для тестирования ваших приложений (например Google....).

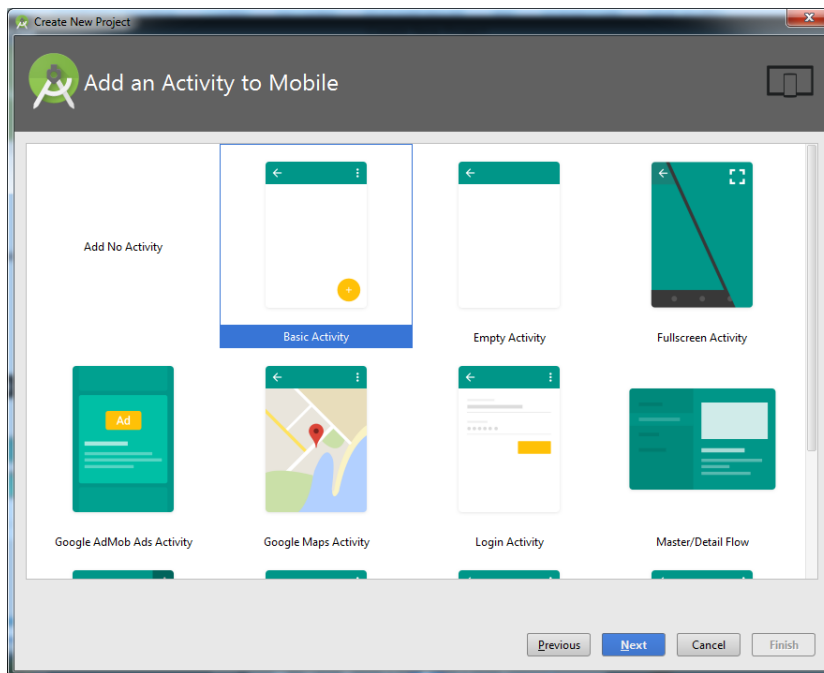
6) Пробное приложение.

Создайте новый проект. При задании Company domain используйте: фео.fit.bstu.by

Выберите минимальный SDK, например можно оставить по умолчанию



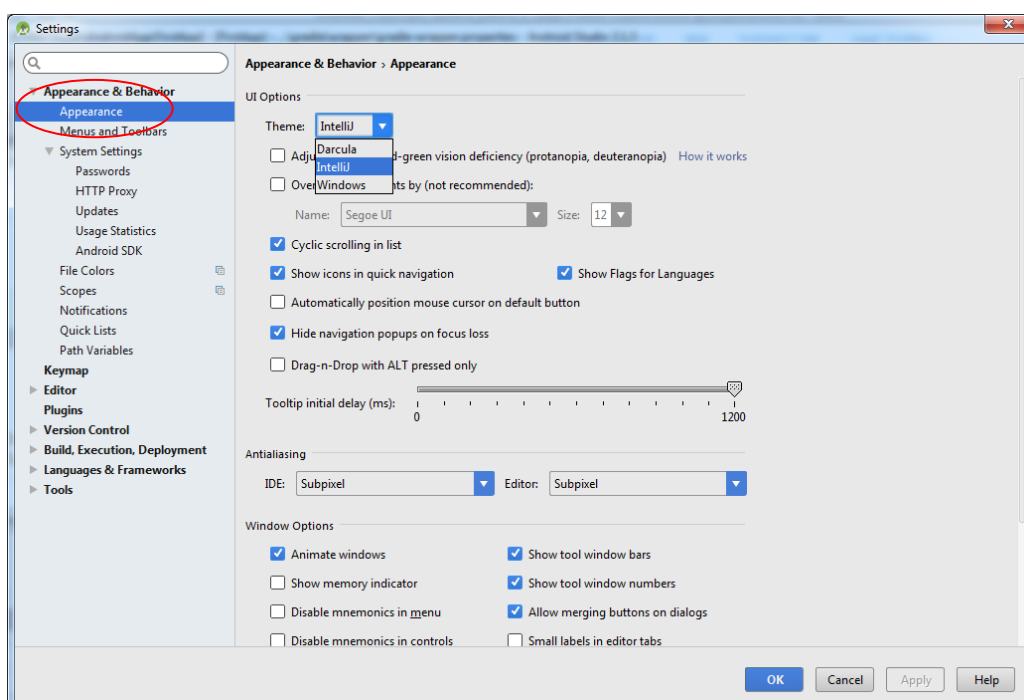
Выберите шаблон. Например Basic Activity



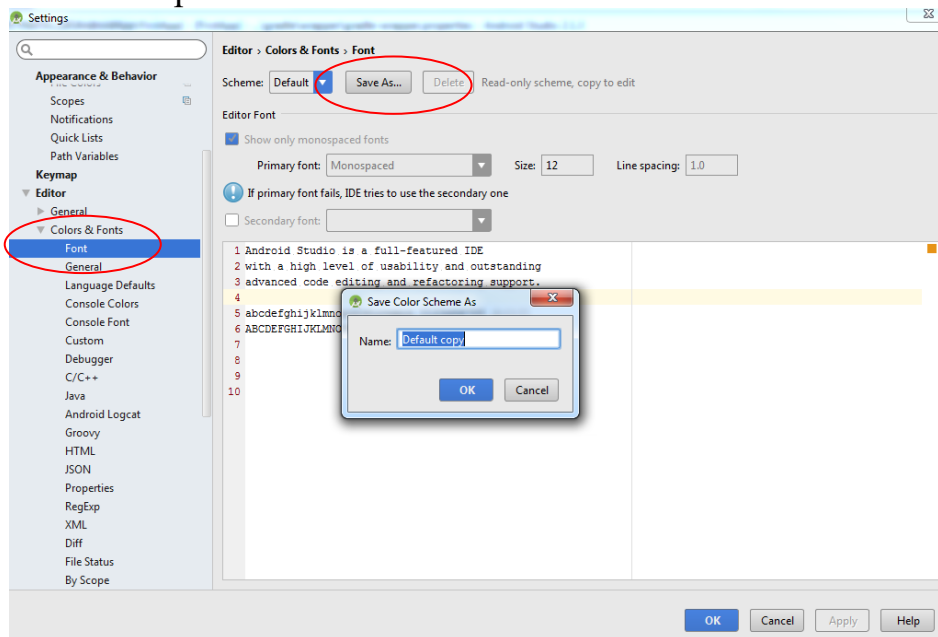
Имена для Activity можно оставить заданные по умолчанию.
Вспомните состав панетли Project.
Найдите класс активности и запустите проект на эмуляторе.

2 часть

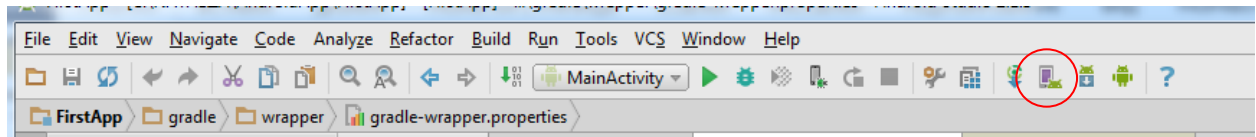
- 1) На основе созданного пустого проекта (работ 1) разберитесь с назначением папок libs, src, build, gradle.
- 2) **Настройте пользовательский интерфейс**
Перейдите File->Settings (ctrl-alt-s) затем Appearance



Выберите тему, можете подобрать себе цветовую схему, шрифт и размер. Editor -> Font. Предварительно нужно сохранить файл с настройками.



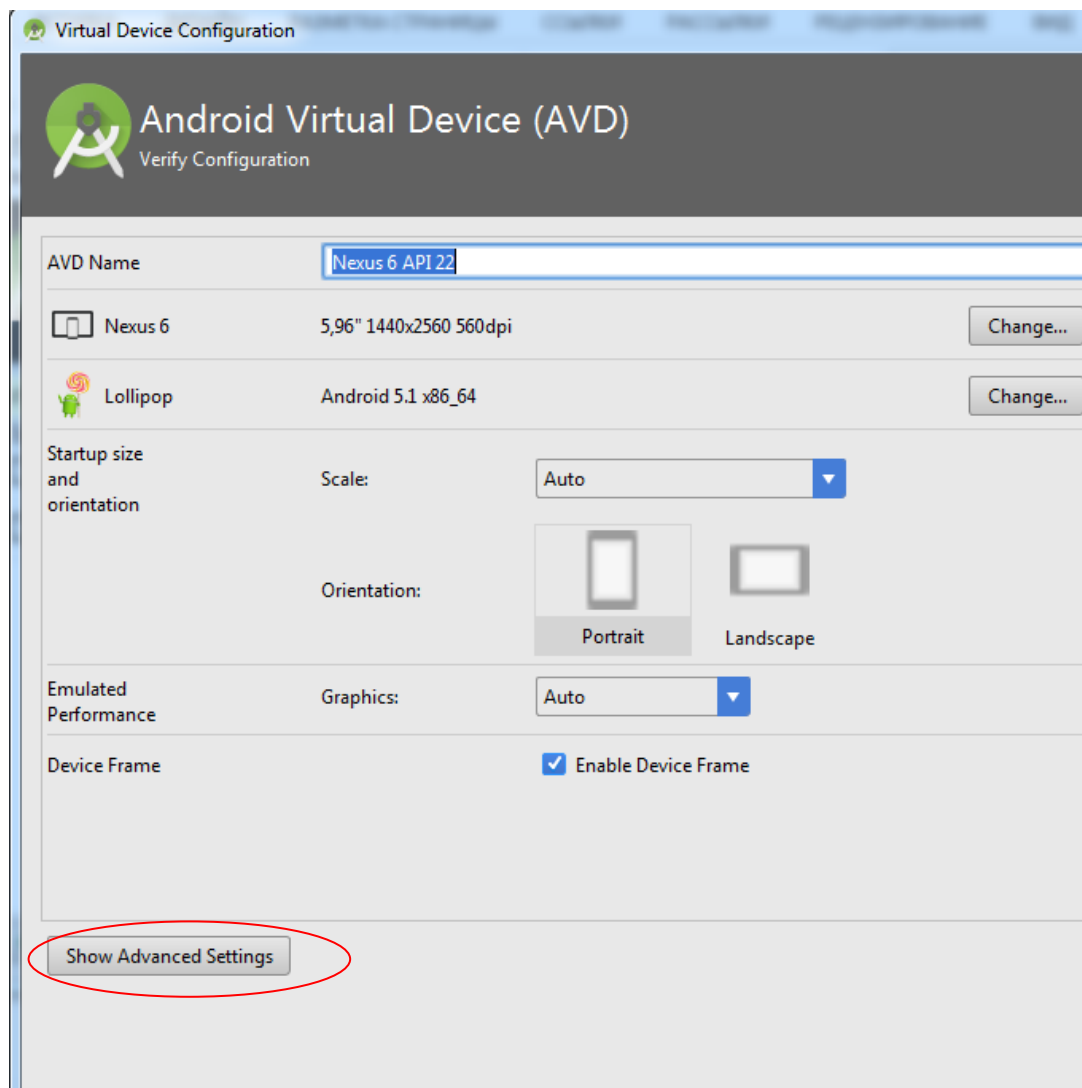
3) **Работа с эмуляторами и устройствами.** Перейдите в AVD Manager. Через Tools->Android-> AVD Manager или через пиктограмму.



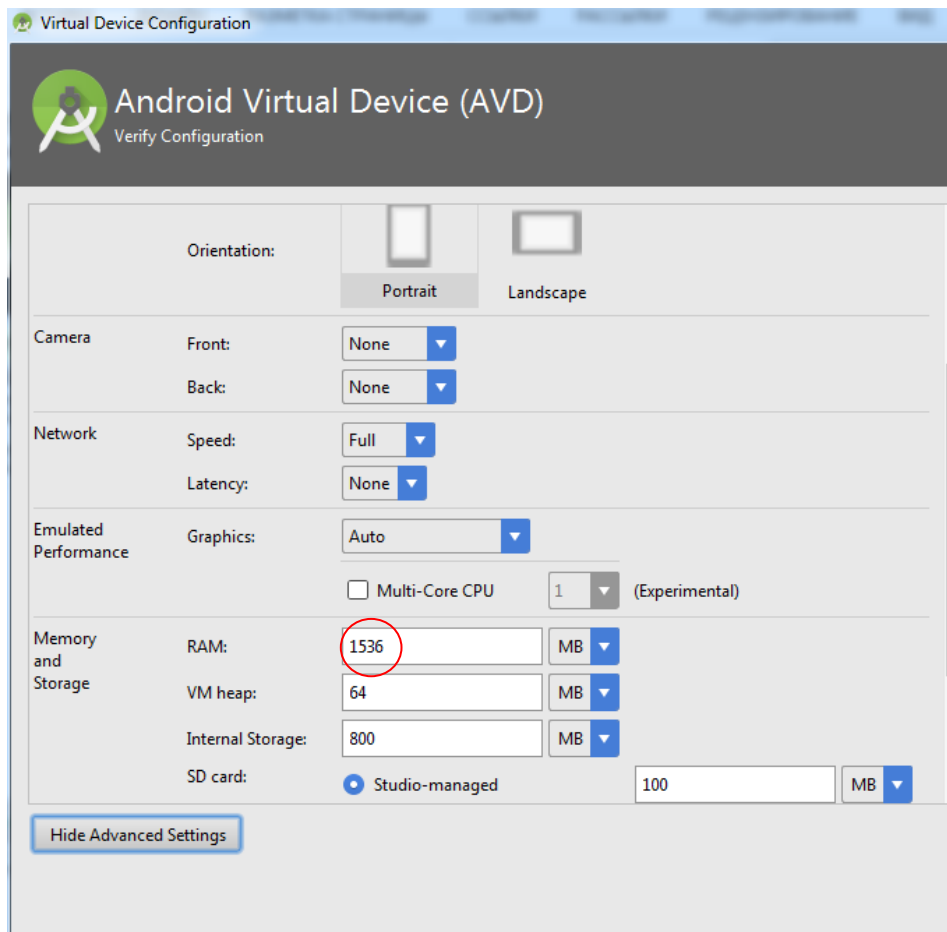
Изучите возможности управления виртуальными устройствами (хотя они медленные): добавления новых, удаления, дублирования, изменения параметров.

Type	Name	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	AVD GalaxyNexus ToolsForApa...	720 × 1280: xhdpi	19	Android 4.4	arm	200 MB	
	Nexus 6 API 22	1440 × 2560: 560dpi	22	Android 5.1 (Google APIs)	x86_64	650 MB	

Посмотрите дополнительные настройки

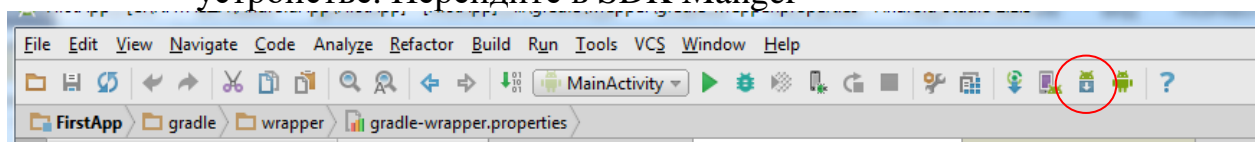


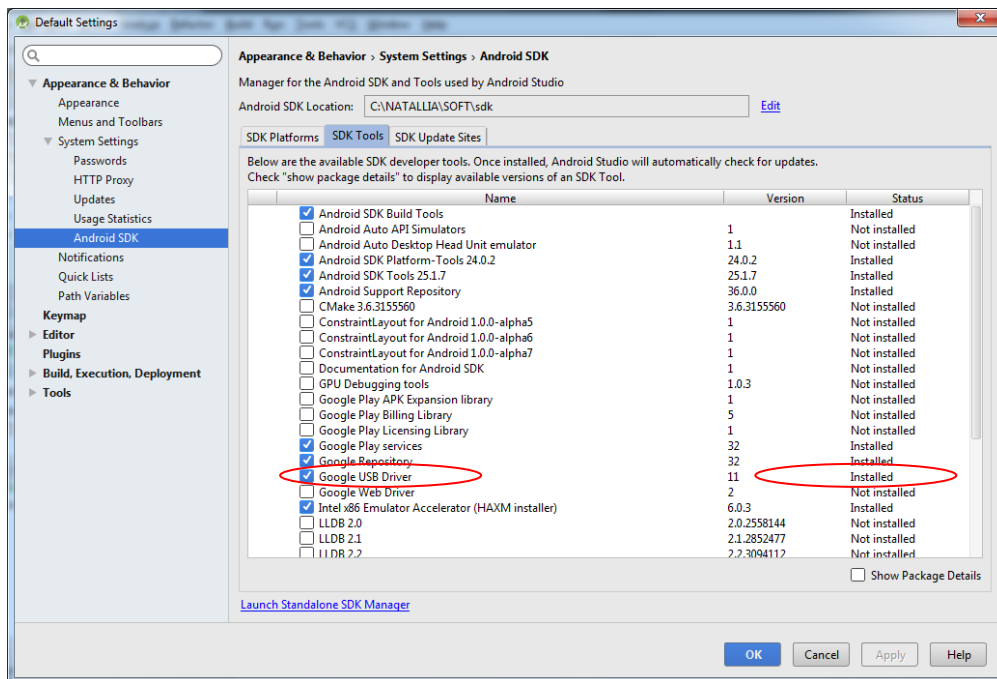
Можете изменить значения заданные для RAM и т.п.



Количество установленных виртуальных устройств зависит от количества свободной памяти в вашей системе.

Убедитесь что у вас установлен Google USB Driver и отладка по USB на устройстве. Перейдите в SDK Manager





Запустите приложение на устройстве.

В окне монитора вы увидите что приложение загружено и установлено на вашем устройстве.



При возникновении проблем обратитесь

<https://developer.android.com/studio/run/device.html>

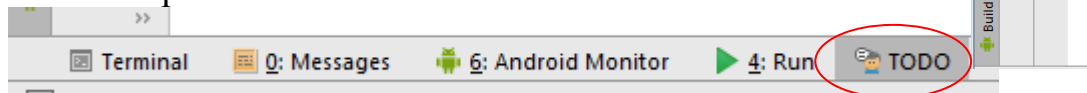
4) Изучите меню.

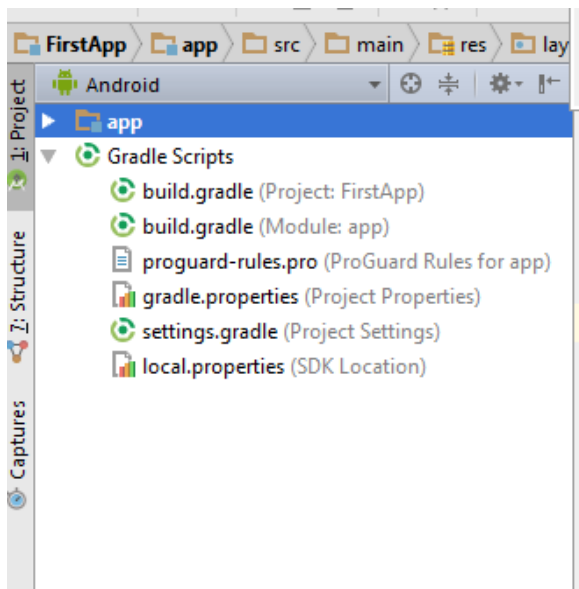
Для ускорения работы используйте

https://resources.jetbrains.com/assets/products/intellij-idea/IntelliJIDEA_ReferenceCard.pdf

Разберитесь с левой боковой панелью. Поясните назначение Structure, Captures, Favourites, Build Variants.

Добавьте в класс комментарий, который начинается с TODO. Найдите на нижней панели TODO. Перейдите по комментарию.





Перейдите в `gradle.properties` - он содержит свойства процесса построения.

Перейдите в `proguard-rules.pro` -там содержатся правила обфускации и оптимизации проекта. Обычно они закомментированы.

И оставшиеся два файла `build.gradle`. В нем содержится информация о версии SDK, используемой для компиляции, минимальной версии, версии приложения и тп.

```

buildToolsVersion "24.0.2"

defaultConfig {
    applicationId "by.bstu.pnv.firstapp"
    minSdkVersion 15
    targetSdkVersion 24
    versionCode 1
    versionName "1.0"
}

```

Еще один раздел – зависимости

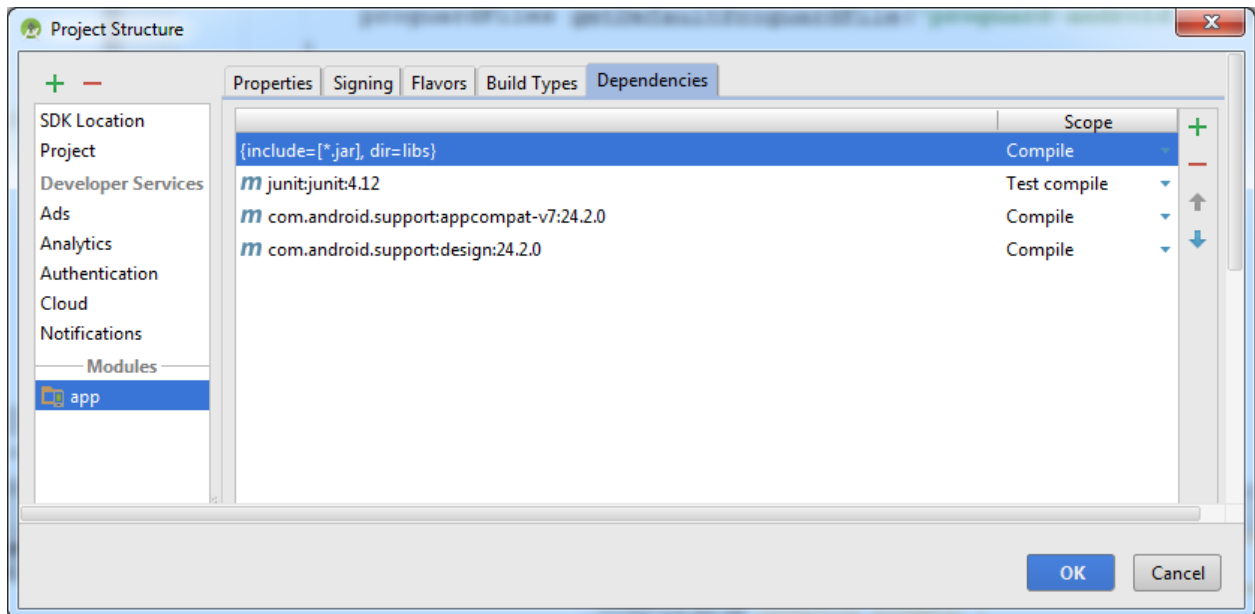
В нем содержится информация о включенных в `libs` jar файлов. При включении новой зависимости он а будет автоматически здесь прописана.

```

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:24.2.0'
    compile 'com.android.support:design:24.2.0'
}

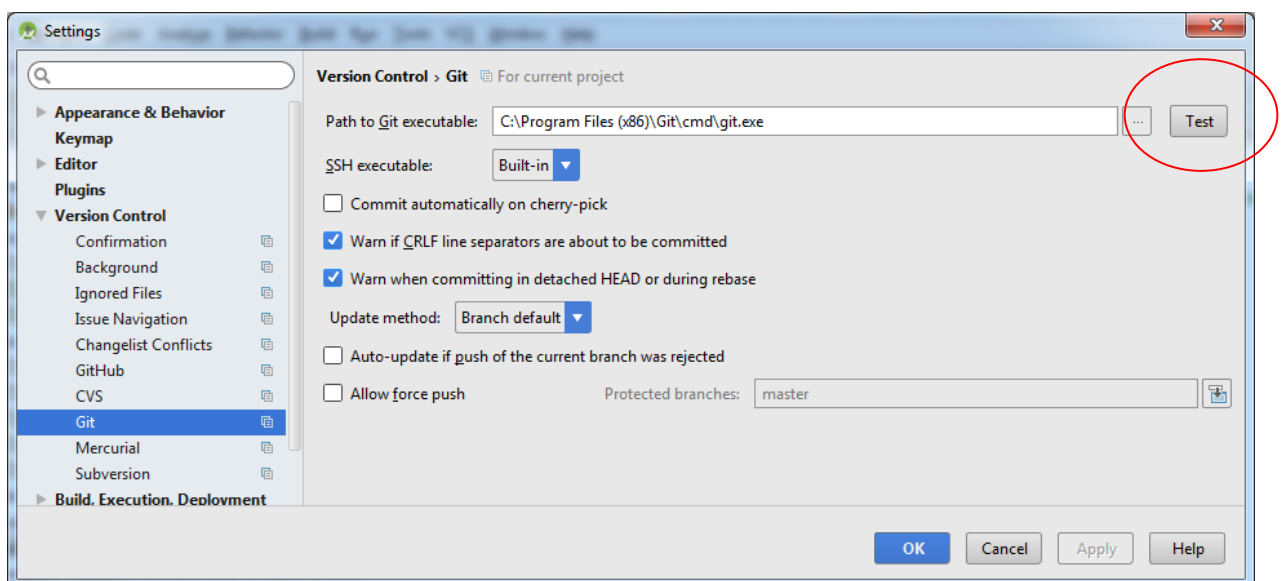
```

Для подключения новых зависимостей можно использовать также File->Project Structure. И знаками + - добавлять или удалять их.

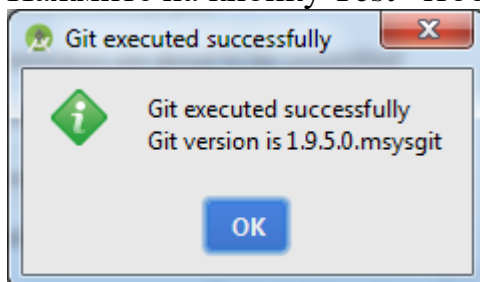


6) Интеграция с Git

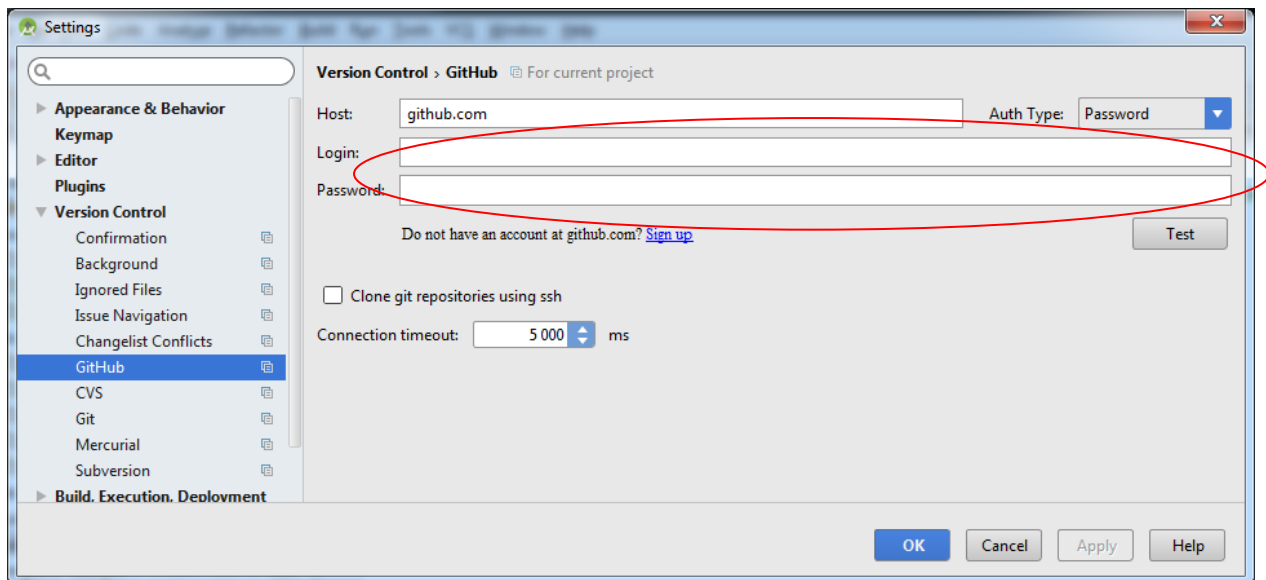
Перейдите в Settings -> Version Control ->Git



Нажмите на кнопку Test чтобы проверить что он установлен

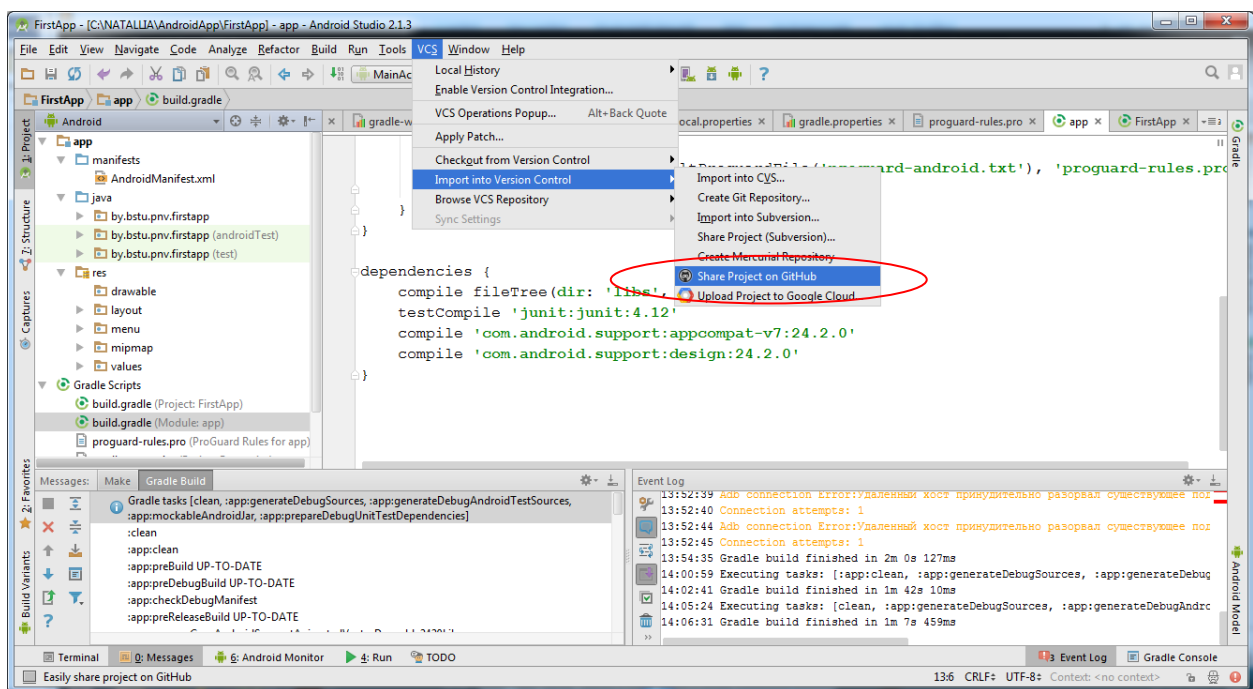


Перейдите на GitHub



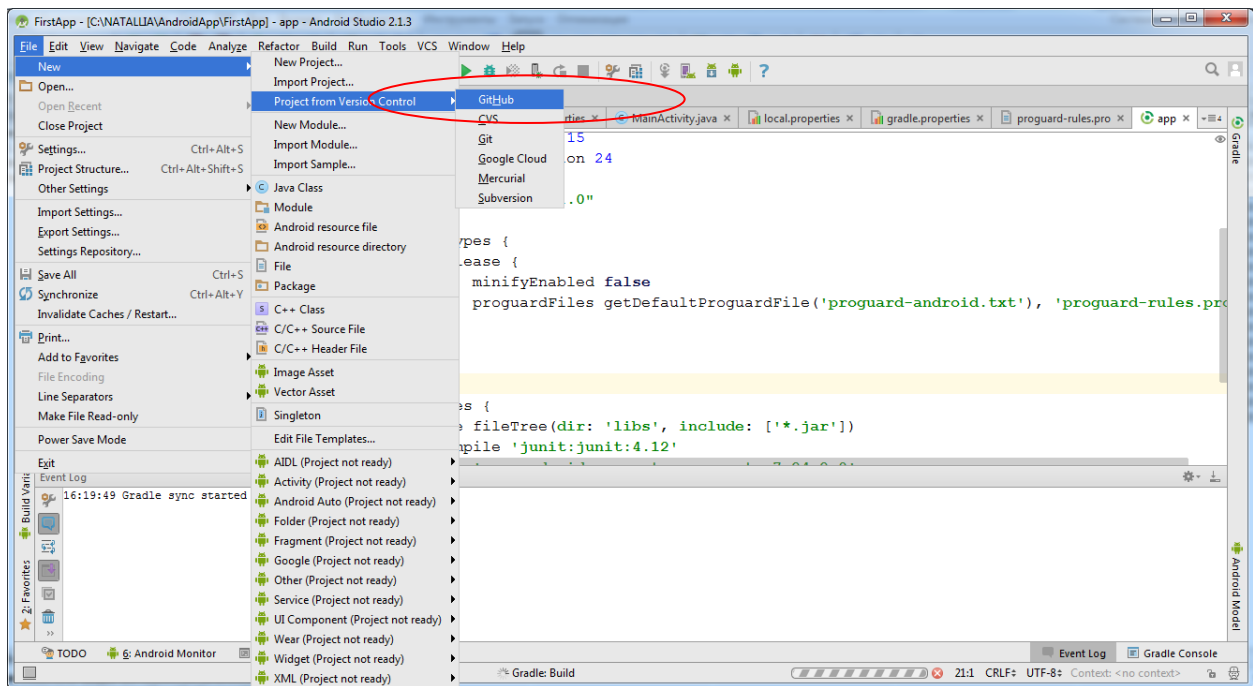
Ведите логин и пароль и проверьте подключение (аккаунт вы заводили)

Выполните публикацию проекта в удаленном репозитории
VCS -> Import into Version Control -> Share Project on GitHub



Задайте имя проекту и выполните Share.

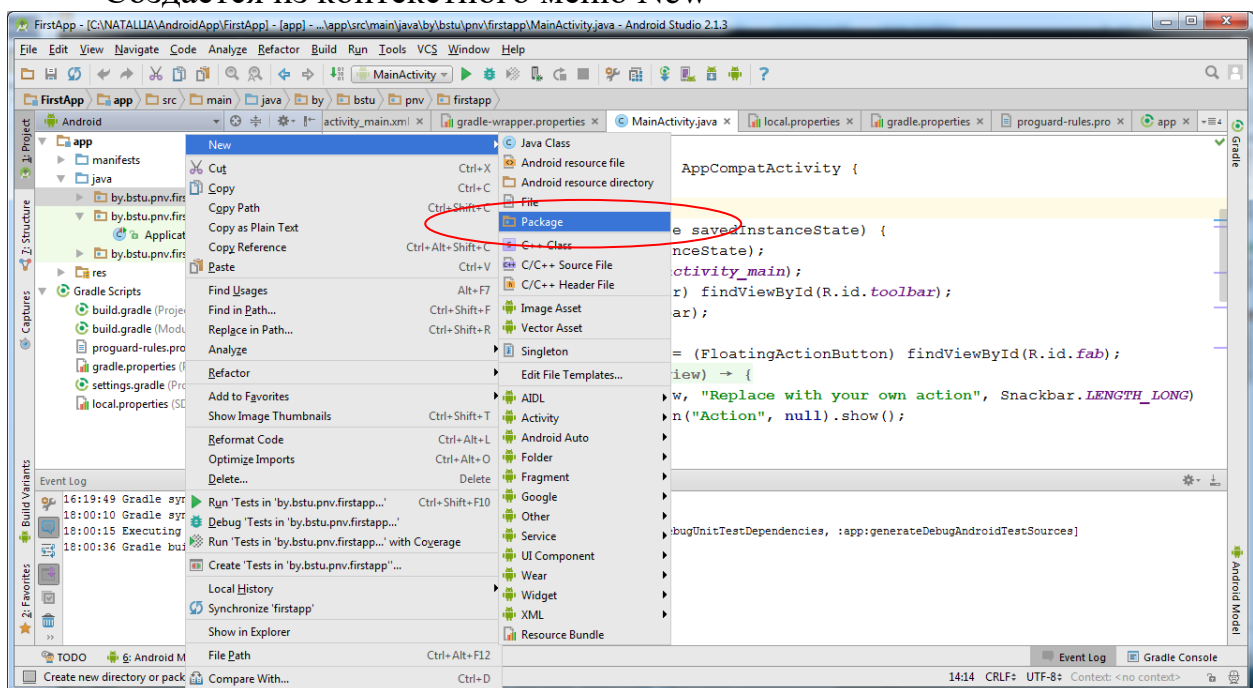
Удалите текущий проект с диска. Запустите Studio и выберите Project from VS -> Github



Клонируйте репозиторий и откройте его.

7) Создание пакетов и классов

Создайте новый пакет и назовите его в соответствии с вашей фамилией
Создается из контекстного меню New



Затем сойздайте новый класс (аналогично) со следующим методом.

```
public class TextFunction {

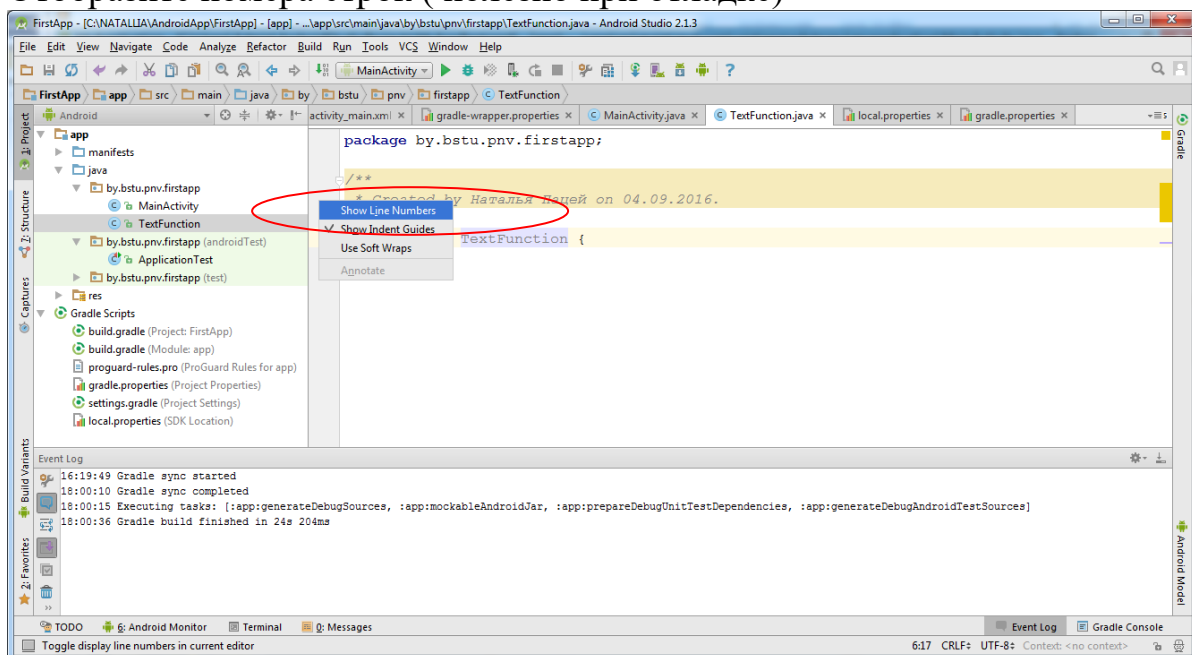
    public String getValue()
    {return "Hello from First project";}

}
```

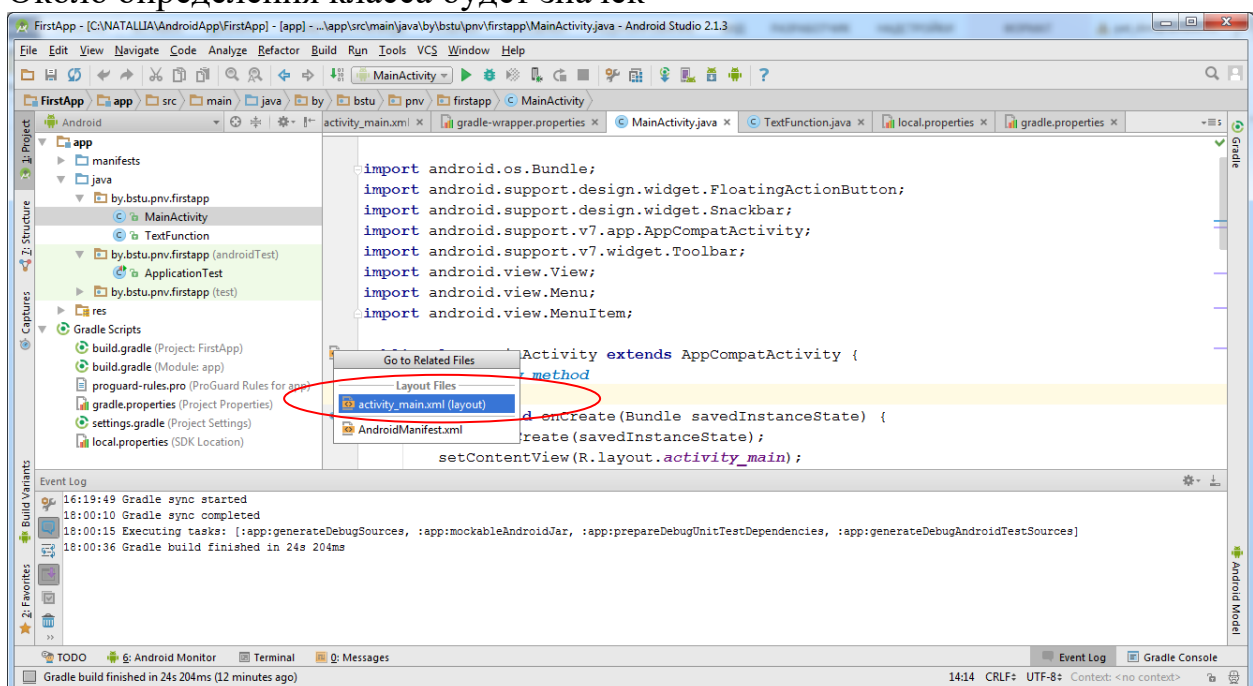
Переименуйте его – Refactor – Rename (Shift-F6).

Переместите созданный класс в пакет. Обратите внимание как поменялось имя package

Отобразите номера строк (полезно при отладке)



Около определения класса будет значек



Кликнув по нему можно быстро перейти на связанный xml файл – выполните и отредактируйте layout файл. Добавьте к нему id

```
<TextView
    android:id="@+id/Newtext"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!" />
```

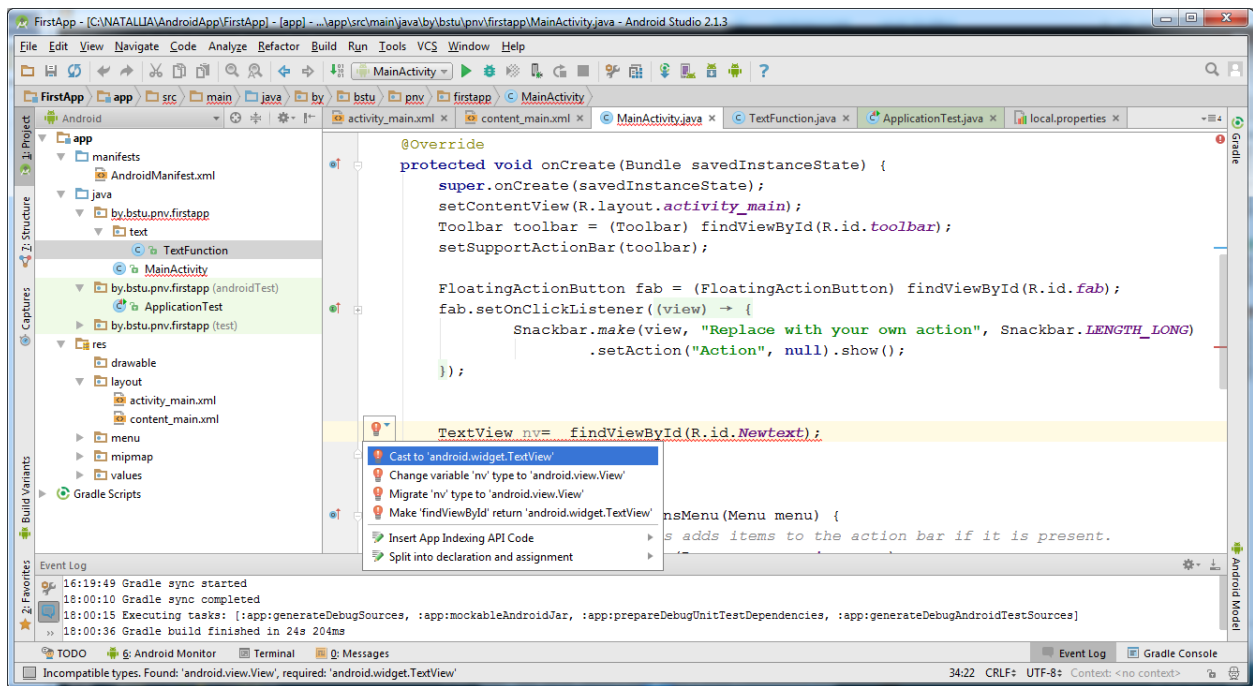
Теперь добавьте в класс MainActivity в метод onCreate след код.

```
public class MainActivity extends AppCompatActivity {
    //TODO Add new method
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

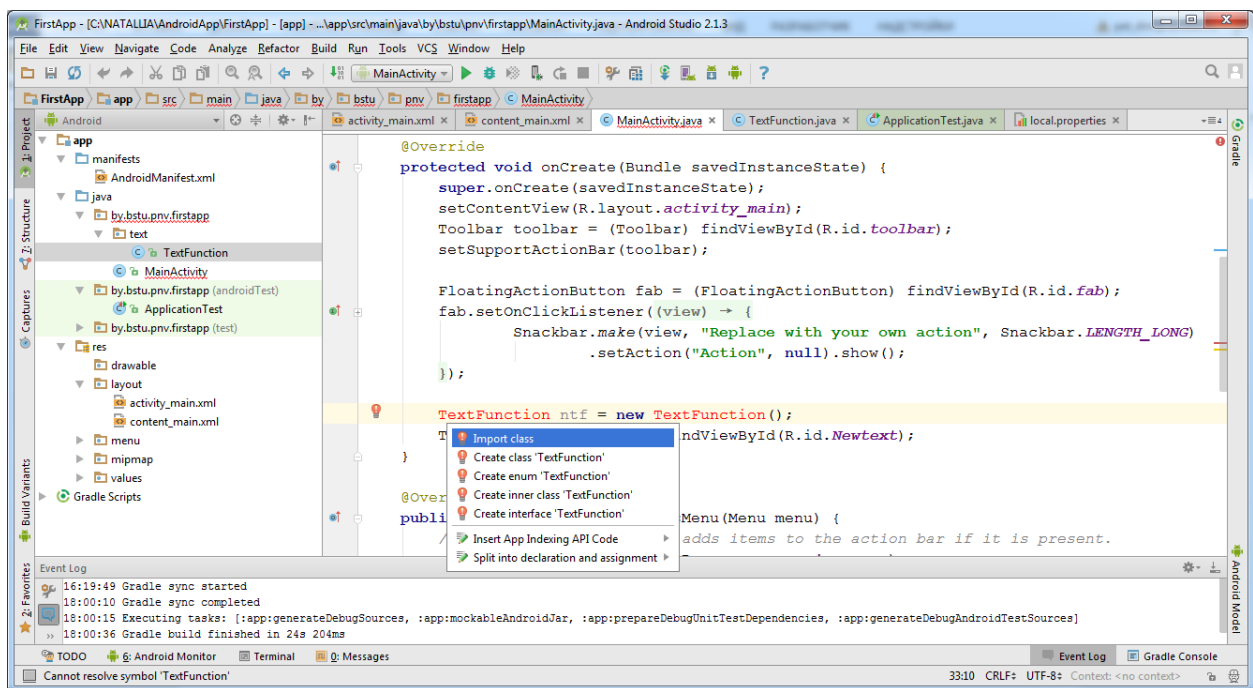
        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
                    Snackbar.LENGTH_LONG, R.id.fab).show();
            }
        });

        TextView nv= findViewById(R.id.Newtext);
    }
}
```

Нажмите на красную лампочку и устраните ошибки выбрав нужный вариант устранения



Теперь создайте объект созданного класса и выполните import class для импорта класса



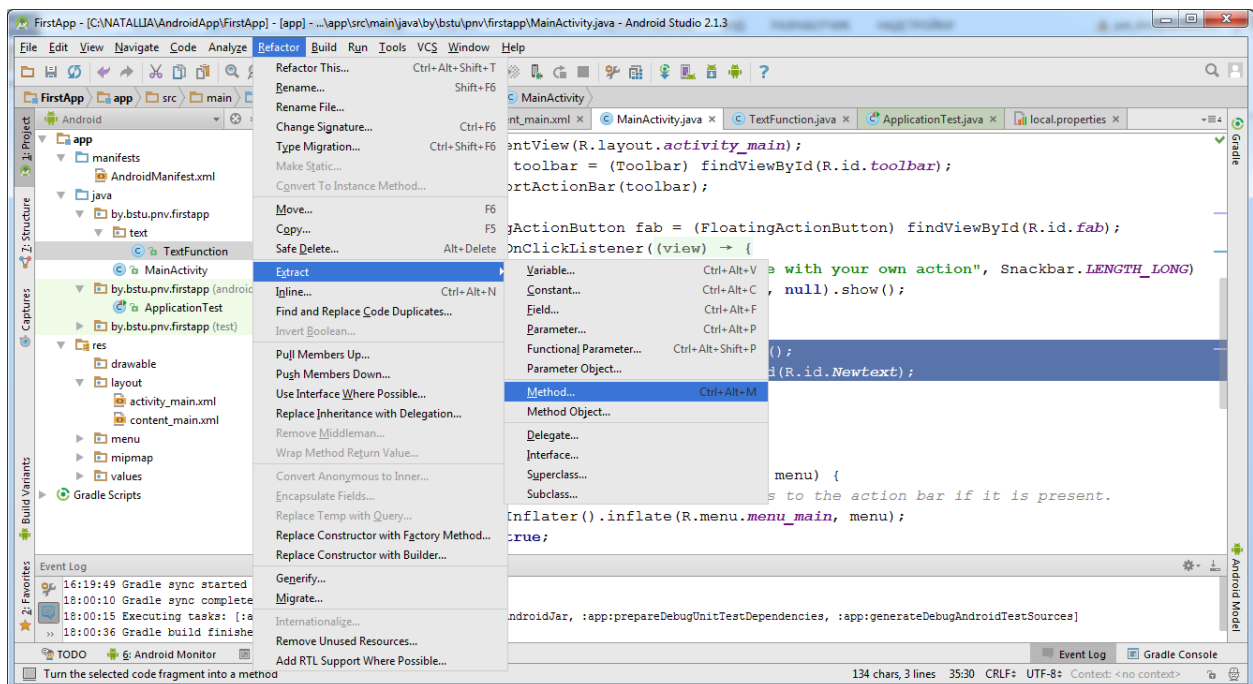
Допишите следующий текст

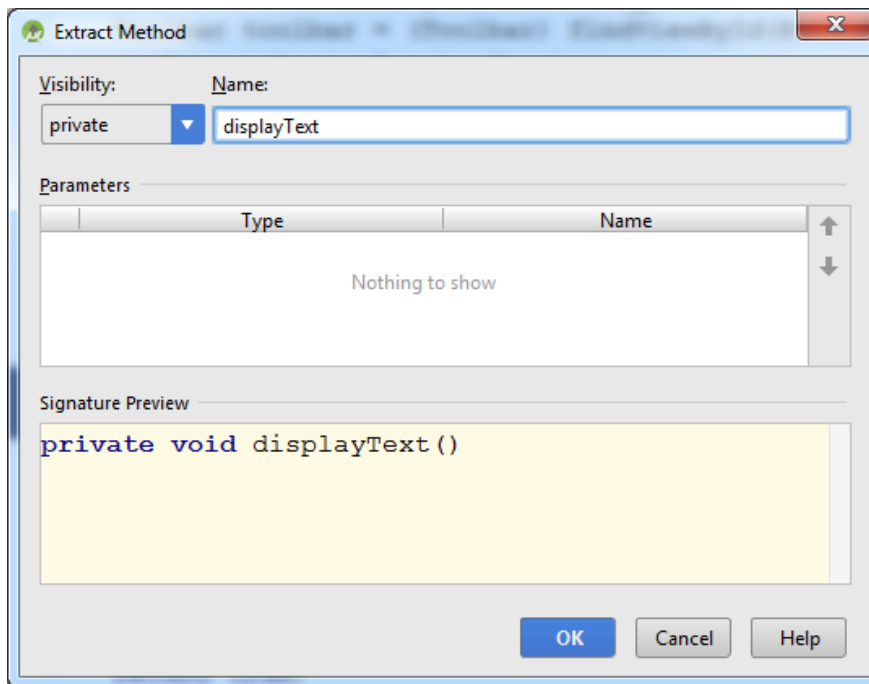

```
fab.setOnClickListener((view) → {
    Snackbar.make(view, "Replace with your own action",
        .setAction("Action", null).show();
});
```

```
TextFunction ntf = new TextFunction();
TextView nv= (TextView) findViewById(R.id.Newtext);
nv.setText(ntf.getValue());
}
```

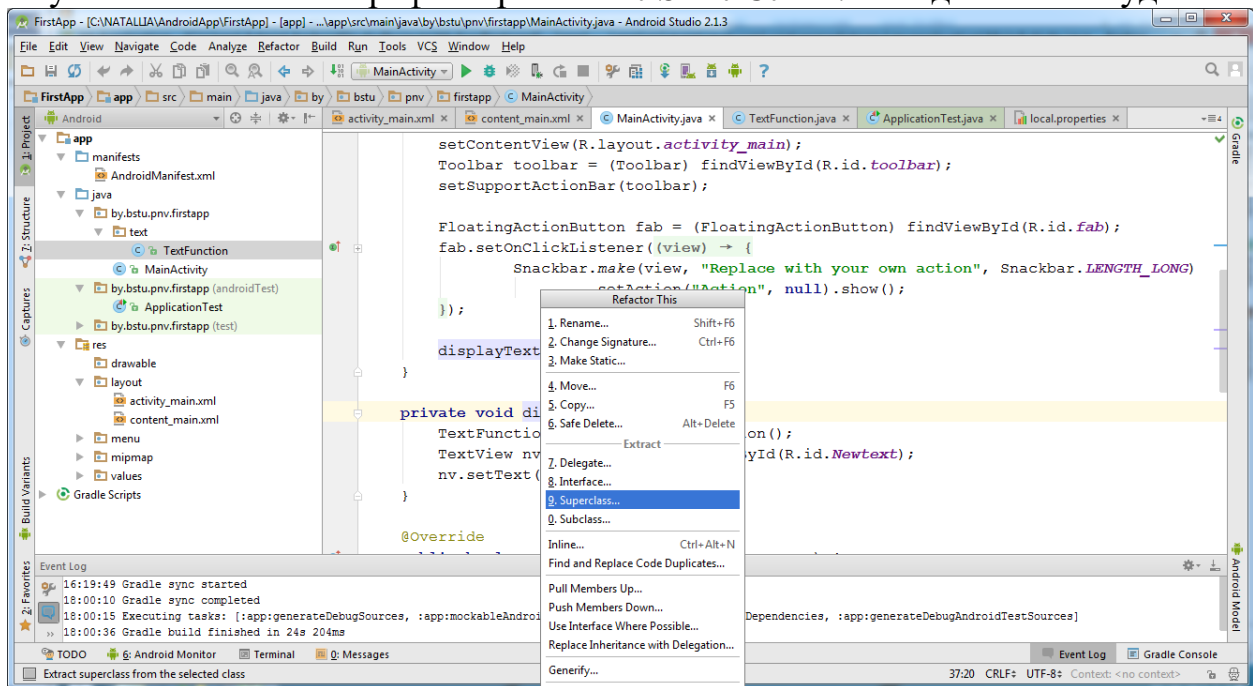
```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
```

Сделайте рефакторинг: выделите написанный фрагмент в метод.
Для этого выделите код. Refacto -> Extract->Method. Задайте имя методу.



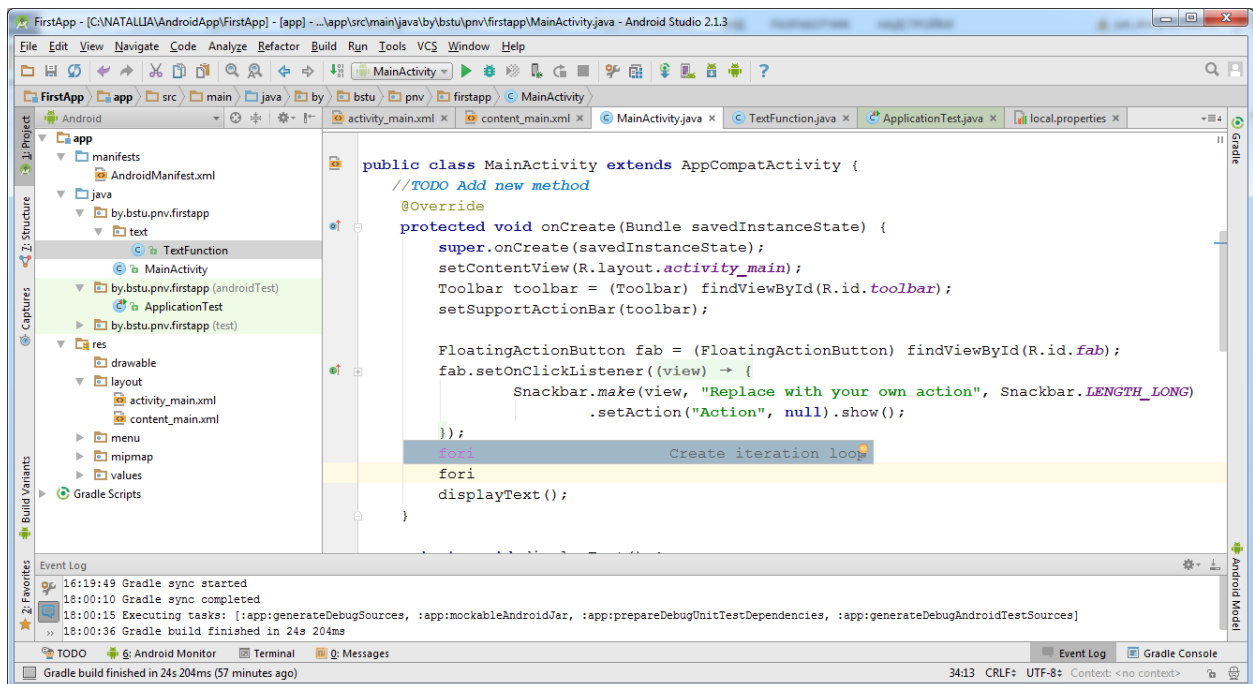


Изучите возможности рефакторинга Alt-Shift-Ctrl-T. Они достаточно удобные.



8) Использование средств отладки

Создайте цикл. Наберите в методе onCreate
for



Поменяйте имя переменной в выделенной области

```

    });

    for (int i = 0; i < ; i++) {

    }

```

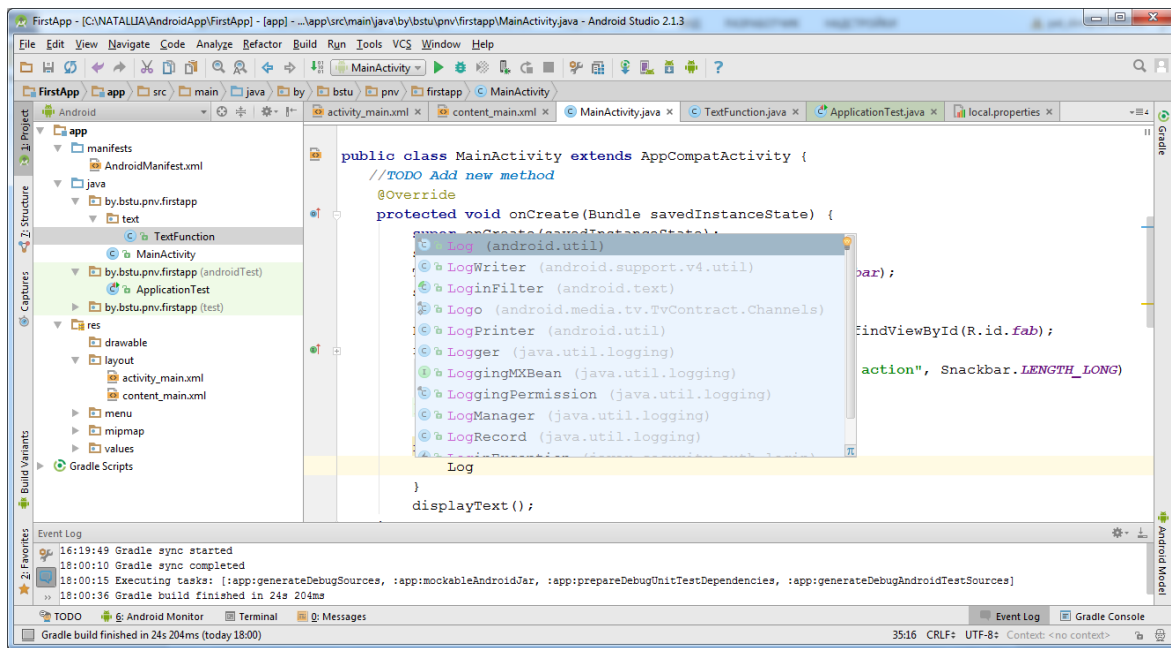
Нажмите enter и вставьте диапазон

```

    for (int count = 0; count < ; count++) {

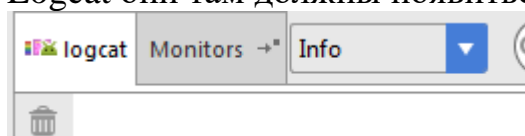
```

Добавим вывод сообщения в Log. Определите класс для авто импорта.



```
for (int count = 0; count < 10; count++) {
    Log.d("MainActivity", "Counter"+count);
}
```

Запустите в эмуляторе и посмотрите сообщения в android monitor
Logcat они там должны появиться



Поменяйте уровень log на debug.

Установите точку останова и запустите приложение в отладчике Run- -Debug (Shift-f9).

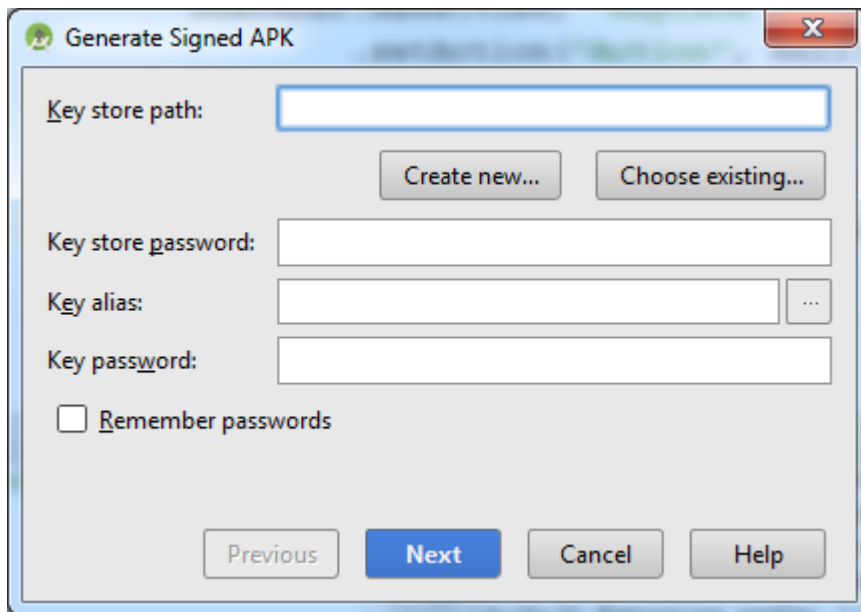
В окне Debugger посмотрите как изменяются переменная. Зайдите в Меню Run посмотрите доступные команды. Выполните Resume Programm.

Добавьте count в окно Watches.

Во время отладки щелкните по точке основа и установите условие установки допустим count==8. Выполните Resume Programm.

9) Публикация приложения.

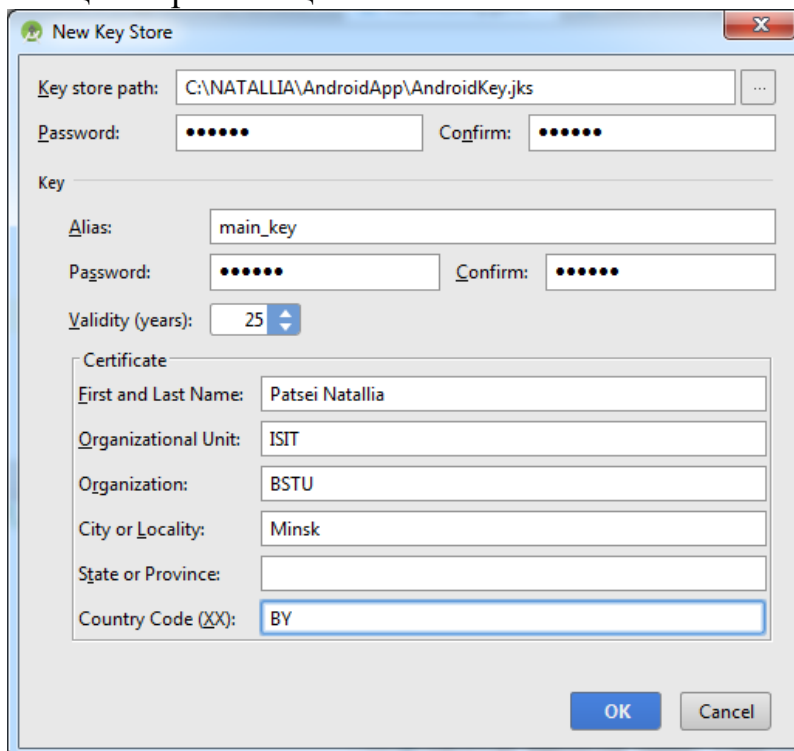
Из меню Build –Generate Signed APK



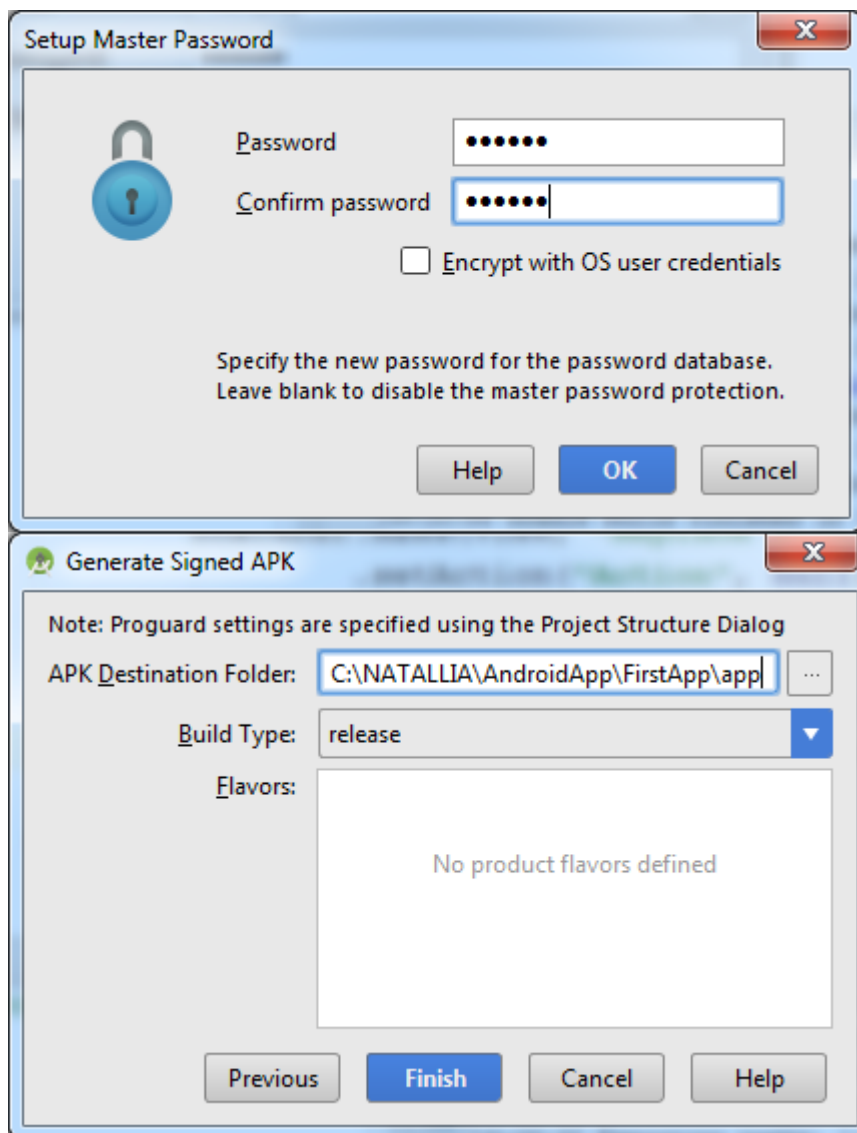
Можно выбрать или сгенерировать.

При генерации заполните все поля

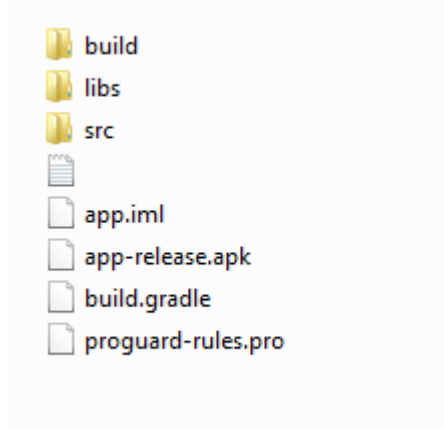
Обратите внимание на путь хранения, его можно скопировать во внешнее хранилище.



Создайте мастер пароль если понадобится



После Finish у вас будет создан .apk файл.



А ключ который вы сгенерировали будет использоан для подписания в макрете вашего приложения.

- 10) Для эффективной работы с Android Studio Изучите ключи

Description	Windows/Linux	Mac
General		
Save all	Control + S	Command + S
Synchronize	Control + Alt + Y	Command + Option + Y
Maximize/minimize editor	Control + Shift + F12	Control + Command + F12
Add to favorites	Alt + Shift + F	Option + Shift + F
Inspect current file with current profile	Alt + Shift + I	Option + Shift + I
Quick switch scheme	Control + ` (backquote)	Control + ` (backquote)
Open settings dialogue	Control + Alt + S	Command + , (comma)
Open project structure dialog	Control + Alt + Shift + S	Command + ; (semicolon)
Switch between tabs and tool window	Control + Tab	Control + Tab
Navigating and Searching Within Studio		
Search everything (including code and menus)	Press Shift twice	Press Shift twice
Find	Control + F	Command + F
Find next	F3	Command + G
Find previous	Shift + F3	Command + Shift + G
Replace	Control + R	Command + R

Find action	Control + Shift + A	Command + Shift + A
Search by symbol name	Control + Alt + Shift + N	Command + Option + O
Find class	Control + N	Command + O
Find file (instead of class)	Control + Shift + N	Command + Shift + O
Find in path	Control + Shift + F	Command + Shift + F
Open file structure pop-up	Control + F12	Command + F12
Navigate between open editor tabs	Alt + Right/Left Arrow	Control + Right/Left Arrow
Jump to source	F4 / Control + Enter	F4 / Command + Down Arrow
Open current editor tab in new window	Shift + F4	Shift + F4
Recently opened files pop-up	Control + E	Command + E
Recently edited files pop-up	Control + Shift + E	Command + Shift + E
Go to last edit location	Control + Shift + Backspace	Command + Shift + Backspace
Close active editor tab	Control + F4	Command + W
Return to editor window from a tool window	Esc	Esc
Hide active or last active tool window	Shift + Esc	Shift + Esc
Go to line	Control + G	Command + L
Open type hierarchy	Control + H	Control + H

Open method hierarchy	Control + Shift + H	Command + Shift + H
Open call hierarchy	Control + Alt + H	Control + Option + H
Writing Code		
Generate code (getters, setters, constructors, hashCode/equals, toString, new file, new class)	Alt + Insert	Command + N
Override methods	Control + O	Control + O
Implement methods	Control + I	Control + I
Surround with (if...else / try...catch / etc.)	Control + Alt + T	Command + Option + T
Delete line at caret	Control + Y	Command + Backspace
Collapse/expand current code block	Control + minus/plus	Command + minus/plus
Collapse/expand all code blocks	Control + Shift + minus/plus	Command + Shift + minus/plus
Duplicate current line or selection	Control + D	Command + D
Basic code completion	Control + Space	Control + Space
Smart code completion (filters the list of methods and variables by expected type)	Control + Shift + Space	Control + Shift + Space
Complete statement	Control + Shift + Enter	Command + Shift + Enter
Quick documentation lookup	Control + Q	Control + J
Show parameters for selected method	Control + P	Command + P

Go to declaration (directly)	Control + B or Control + Click	Command + B or Command + Click
Go to implementations	Control + Alt + B	Command + Alt + B
Go to super-method/super-class	Control + U	Command + U
Open quick definition lookup	Control + Shift + I	Command + Y
Toggle project tool window visibility	Alt + 1	Command + 1
Toggle bookmark	F11	F3
Toggle bookmark with mnemonic	Control + F11	Option + F3
Comment/uncomment with line comment	Control + /	Command + /
Comment/uncomment with block comment	Control + Shift + /	Command + Shift + /
Select successively increasing code blocks	Control + W	Option + Up
Decrease current selection to previous state	Control + Shift + W	Option + Down
Move to code block start	Control + [Option + Command + [
Move to code block end	Control +]	Option + Command +]
Select to the code block start	Control + Shift + [Option + Command + Shift + [
Select to the code block end	Control + Shift +]	Option + Command + Shift +]
Delete to end of word	Control + Delete	Option + Delete

Delete to start of word	Control + Backspace	Option + Backspace
Optimize imports	Control + Alt + O	Control + Option + O
Project quick fix (show intention actions and quick fixes)	Alt + Enter	Option + Enter
Reformat code	Control + Alt + L	Command + Option + L
Auto-indent lines	Control + Alt + I	Control + Option + I
Indent/unindent lines	Tab/Shift + Tab	Tab/Shift + Tab
Smart line join	Control + Shift + J	Control + Shift + J
Smart line split	Control + Enter	Command + Enter
Start new line	Shift + Enter	Shift + Enter
Next/previous highlighted error	F2 / Shift + F2	F2 / Shift + F2
Build and Run		
Build	Control + F9	Command + F9
Build and run	Shift + F10	Control + R
Apply changes (with Instant Run)	Control + F10	Control + Command + R
Debugging		
Debug	Shift + F9	Control + D
Step over	F8	F8
Step into	F7	F7
Smart step into	Shift + F7	Shift + F7
Step out	Shift + F8	Shift + F8
Run to cursor	Alt + F9	Option + F9
Evaluate expression	Alt + F8	Option + F8

Resume program	F9	Command + Option + R
Toggle breakpoint	Control + F8	Command + F8
View breakpoints	Control + Shift + F8	Command + Shift + F8
Refactoring		
Copy	F5	F5
Move	F6	F6
Safe delete	Alt + Delete	Command + Delete
Rename	Shift + F6	Shift + F6
Change signature	Control + F6	Command + F6
Inline	Control + Alt + N	Command + Option + N
Extract method	Control + Alt + M	Command + Option + M
Extract variable	Control + Alt + V	Command + Option + V
Extract field	Control + Alt + F	Command + Option + F
Extract constant	Control + Alt + C	Command + Option + C
Extract parameter	Control + Alt + P	Command + Option + P
Version Control / Local History		
Commit project to VCS	Control + K	Command + K
Update project from VCS	Control + T	Command + T
View recent changes	Alt + Shift + C	Option + Shift + C
Open VCS popup	Alt + ` (backquote)	Control + V