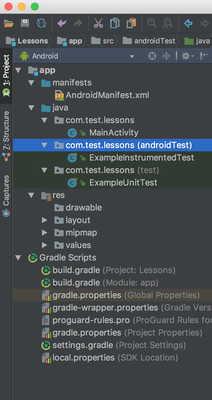
Давайте разберемся, из чего же состоит наш проект.

Откройте вкладку **Project** и разверните дерево проекта:

[](https://skillberg.com/media/uploads/2017/11/17/53cbe6d732694047b4b79e6ac4abac67.png)Структура проекта Android

В первую очередь, проект состоит из модулей. По-умолчанию создается один модуль — app, но вы можете добавить дополнительные модули.

Первый файл, который мы видим — AndroidManifest.xml. В этом файле указываются компоненты и некоторые параметры приложения, и мы будем активно использовать его в следующих уроках.

Далее идет "ветка" java. В ней вы видите несколько пакетов. Первый содержит исходный код нашего приложения. Пока что в нем только один файл — MainActivity.java. Это главный экран нашего приложения.

Остальные два пакета содержат автоматически сгенерированные примеры тестов для кода. Тесты нам пока не нужны, но в одном из уроков мы основательно ими займемся.

Далее вы видите директорию res. В ней содержатся ресурсы приложения — строки, иконки, изображения и т.д. Мы обязательно подробно рассмотрим тему ресурсов, но позже.

Далее вы видите Gradle Scripts. Что такое **Gradle**?

Gradle

**Gradle** — это система сборки, используемая в том числе в Android. Он позволяет гибко конфигурировать процесс сборки и удобно управлять зависимостями проекта.

В виртуальной "директории" Gradle Scripts содержатся, как ни странно, скрипты Gradle — файлы, в которых задаются правила сборки проекта.

В первую очередь, откройте settings.gradle. В этом файле вы увидите всего одну строчку:

include ':app'

В нем перечислены модули, которые нужно включить в сборку.

Далее — build.gradle (Тот, у которого написано **Project: Lessons**). Это скрипт с правилами сборки проекта.

buildscript {

repositories { // Maven-репозитории для системы сборки

jcenter()

}

dependencies { // Зависимости системы сборки

classpath 'com.android.tools.build:gradle:3.0.0-beta3'

}

}

allprojects {

repositories { // Maven-репозитории, общие для всех модулей

jcenter()

google()

}

}

task clean(type: Delete) { // Что делать при "очистке" проекта

delete rootProject.buildDir

}

В этом файле описываются правила, общие для всего проекта.

Далее — build.gradle модуля app. В нем описаны правила сборки конкретного модуля.

// Это означает, что мы собираем приложение

// В ином случае мы можем собирать библиотеку

apply plugin: 'com.android.application'

// Правила сборки Android-приложения

android {

// Версия SDK, с которой будет собрано приложение

compileSdkVersion 26

// Версия инструментов сборки

buildToolsVersion "26.0.1"

// Стандартная конфигурация

defaultConfig {

applicationId "com.test.lessons"

minSdkVersion 17

targetSdkVersion 26

versionCode 1

versionName "1.0"

testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"

}

// Типы сборки

buildTypes {

// Правила для релизной сборки

release {

// Это нам пока не нужно

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

}

}

}

// Зависимости модуля

dependencies {

// Собираем все jar-файлы из директории libs

// В более ранних версиях вместо implementation использовалась директива compile

implementation fileTree(dir: 'libs', include: ['\*.jar'])

// Зависимости для тестов

androidTestImplementation('com.android.support.test.espresso:espresso-core:2.2.2', {

exclude group: 'com.android.support', module: 'support-annotations'

})

// Библиотеки Android Support

implementation 'com.android.support:appcompat-v7:26.0.2'

implementation 'com.android.support.constraint:constraint-layout:1.0.2'

// Зависимости для тестов

testImplementation 'junit:junit:4.12'

}

Рассмотрим поподробнее секцию **defaultConfig**.

* applicationId — имя пакета приложения, его идентификатор. У каждого приложения должно быть уникальное имя пакета.
* minSdkVersion — минимальная версия Android, на которой будет работать приложение. В Android постоянно добавляются новые функции (и удаляются некоторые старые), и поддерживать все версии не имеет смысла. Поэтому стоит всегда ограничивать минимальную версию. Как понять, какую версию поддерживать, а какую нет? Смотрите [тут](https://developer.android.com/about/dashboards/index.html) распространённость версий, и старайтесь охватить ~95% устройств. На момент написания этого урока не имеет смысла поддерживать устройства с версией меньше 17, т.к. их всего около 4%.
* targetSdkVersion — максимальная версия SDK, на которой приложение гарантированно будет работать. Старайтесь всегда поддерживать самые новые версии (то есть значение должно быть максимальным из доступных).
* versionCode — версия приложения. При выпуске новой версии всегда увеличивайте версию, иначе Google Play просто не даст выгрузить её.
* versionName — текстовый вариант версии приложения. Тут, в принципе, можно писать что угодно.

В секции dependencies указываются зависимости модуля. Например, мы хотим подключить библиотеку — тогда нам нужно прописать туда зависимость, Gradle автоматически её скачает и соберет. Мы обязательно попробуем сделать это в одном из следующих уроков.

Вот, в принципе, и все. В следующем уроке я покажу вам, как изменять контент, отображаемый на экране. До встречи!