

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное  
учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Инженерная школа ядерных технологий

Направление 01.04.02 «Прикладная математика и информатика»

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**  
**«Знакомство с Android Studio и Git»**

по дисциплине:  
**Программирование мобильных приложений**

**Исполнитель:**  
студент группы 0BM92

Е. В. Петрович \_\_\_\_\_  
Дата сдачи: \_\_\_\_\_

**Руководитель:**  
доцент,  
кандидат технических наук

А. В. Погребной \_\_\_\_\_  
Дата проверки: \_\_\_\_\_

Томск - 2019

|  |    |
|--|----|
| Оглавление   |    |
| <b>Цель работы</b> .....   | 2  |
| <b>Задание</b> .....   | 2  |
| <b>Основная часть</b> .....                                      | 2  |
| Создание нового проекта .....                                    | 2  |
| Создание репозитория «Git» .....                                 | 5  |
| Добавление в проект модулей для лабораторных работ .....         | 6  |
| Настройка зависимостей модулей .....                             | 9  |
| Создание кнопок для переходов к модулям лабораторных работ ..... | 10 |
| Запуск и тестирование приложения .....                           | 11 |
| <b>Заключение</b> .....  | 12 |
| <b>Приложение</b> .....  | 12 |
| Ссылка на проект .....   | 12 |

## Цель работы

Знакомство с интегрированной средой разработки мобильных приложений «Android Studio» и системой управления версиями «Git».

## Задание

Продублировать структуру проекта аналогично проекту из репозитория <https://github.com/ekzee/tpu-android-courses>. Проект залить на Github. В проекте должен быть модуль app и модули lab1, lab2,..., lab6. В модуле app находится MainActivity с переходами в activity из модулей lab1, lab2... выполненные по аналогии с Lab1Activity (с заголовком, содержащим номер модуля). Ознакомиться с документацией в модулях app, lab1 и \*.gradle файлах.

## Основная часть

### *Создание нового проекта*

1. После запуска «Android Studio» в окне приветствия выбираем пункт меню «Start a new Android Studio project» (Рис.1).

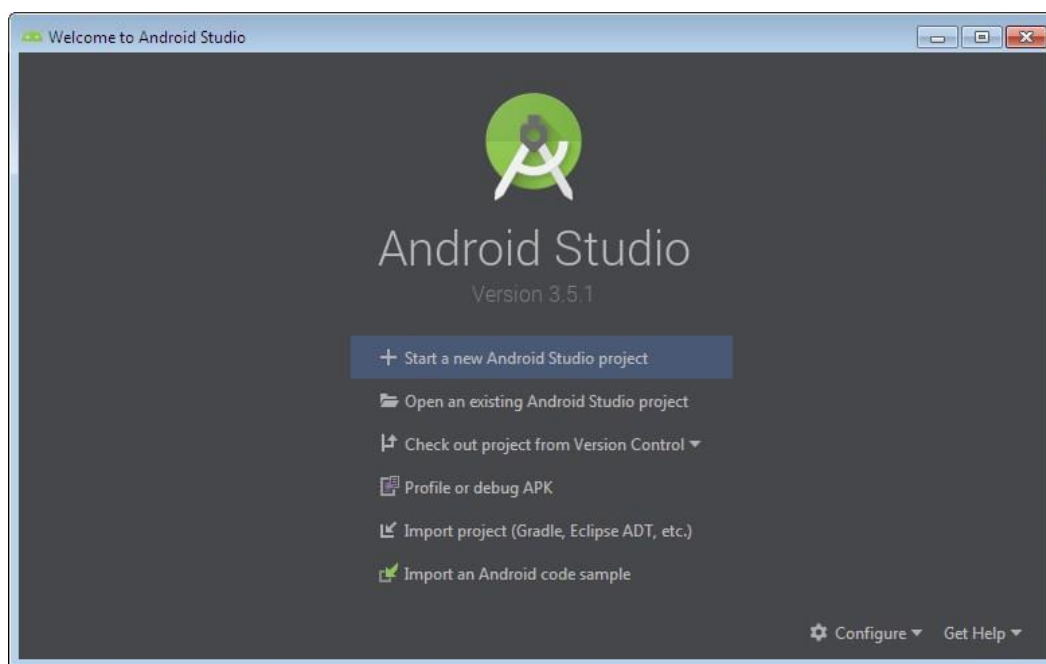


Рис. 1

Проект – это совокупность связанных файлов, содержащих код, графику, специальные файлы настройки и конфигурации, при помощи которых будет собрано приложение.

2. В окне «Choose your project» из предложенного списка выбираем шаблон, по которому будет строиться новый проект. В нашем случае выбираем «Empty Activity» для устройств «Phone and Tablet» (Телефон и Планшет) (Рис. 2).

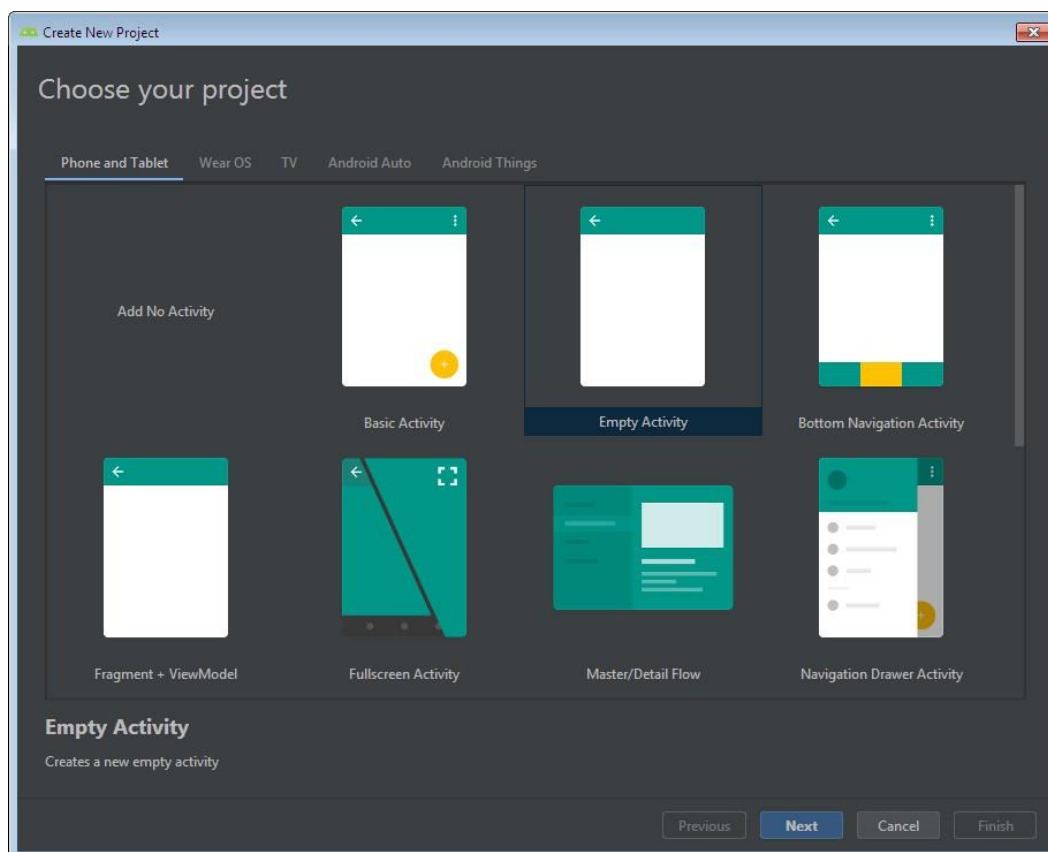


Рис. 2

3. В окне «Configure your project» вводим имя приложения - Name, имя пакета приложения – Package name и имя папки проекта – Save location. Все классы в Java проектах организуются в структуру пакетов. Пакет – логическая сущность, представляющая собой пространство имен. Существует общепринятая схема, где первая часть имени пакета должна состоять из перевёрнутого доменного имени разработчика класса. Так как доменные имена в интернете уникальны, соблюдение этого правила обеспечивает уникальность имён пакетов и предотвращает конфликты имен классов. Для того, чтобы повторить заданную структуру проекта, выбираем имя пакета «ru.tpu.courses.android» (Рис. 3).

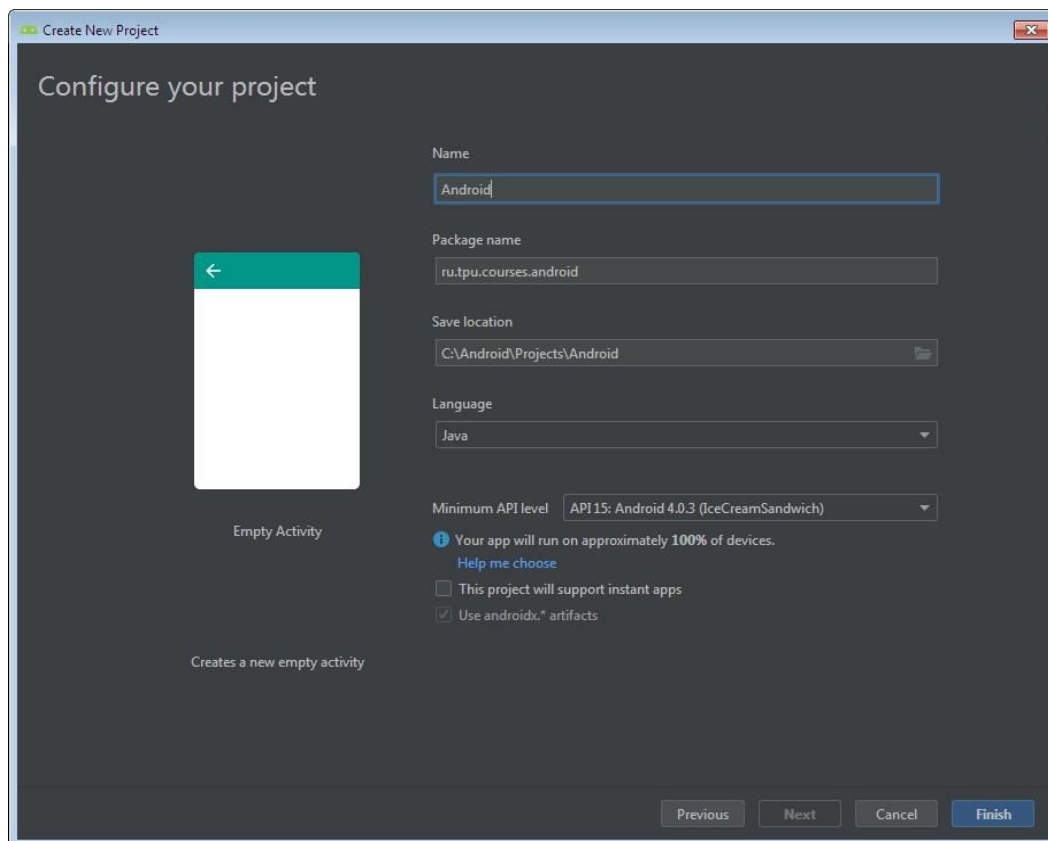


Рис. 3

После нажатия кнопки «Finish» «Android Studio» создает все необходимые для проекта файлы и открывает окно разработки (Рис. 4).

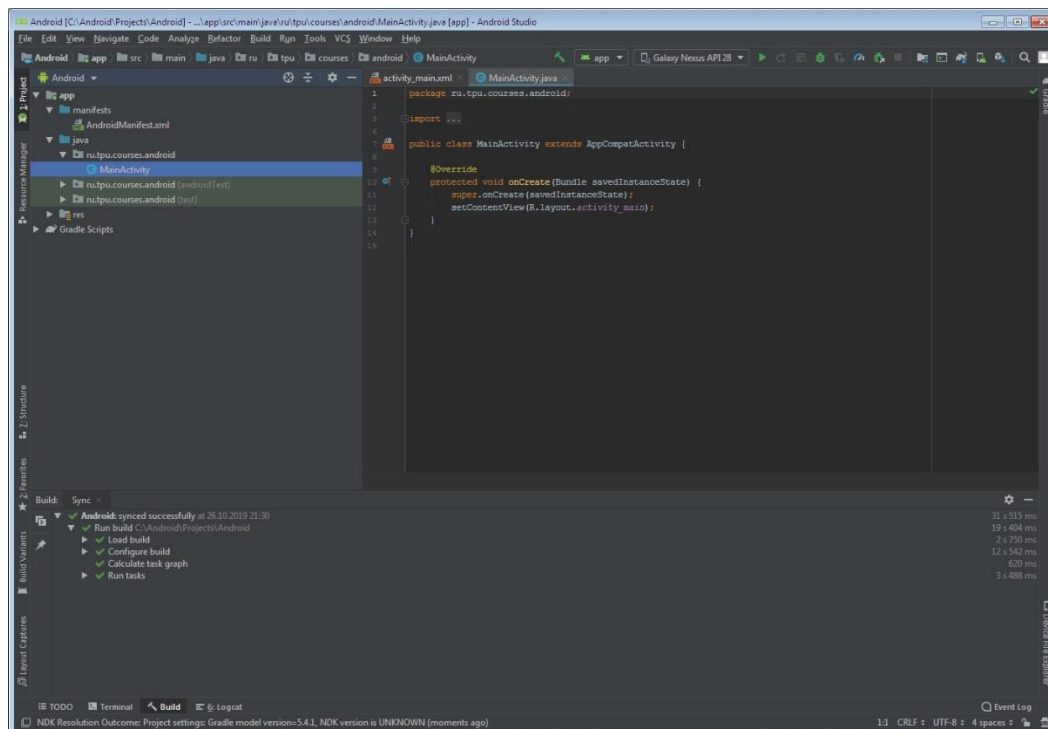


Рис. 4

### Создание репозитория «Git»

Для того, чтобы поставить проект под управление системы контроля версий «Git» создаем локальный репозиторий и репозиторий на сайте «Github». Для удобной работы с «Git» на компьютере установлен пакет «Git flow», который представляет собой набор скриптов, позволяющий автоматизировать работу по [модели ветвления Винсента Дриссена](#).

1. Для создания локального репозитория в командной строке вводим «git flow init» (Рис. 5).

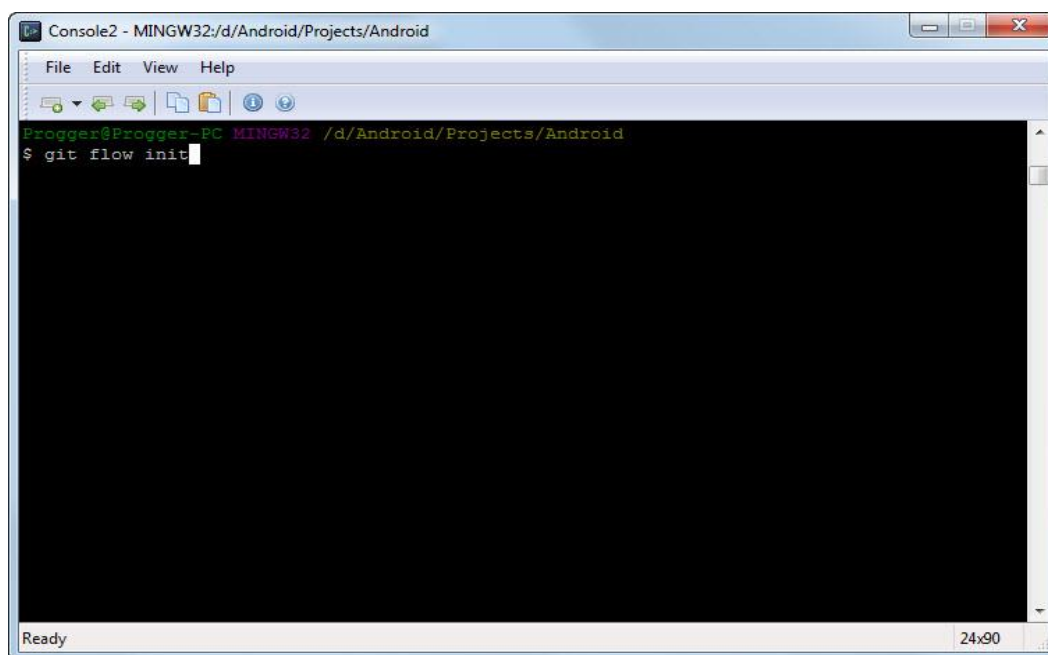


Рис. 5

«Git» создает локальный репозиторий и в нем две ветки «develop» и «master» (Рис. 6).

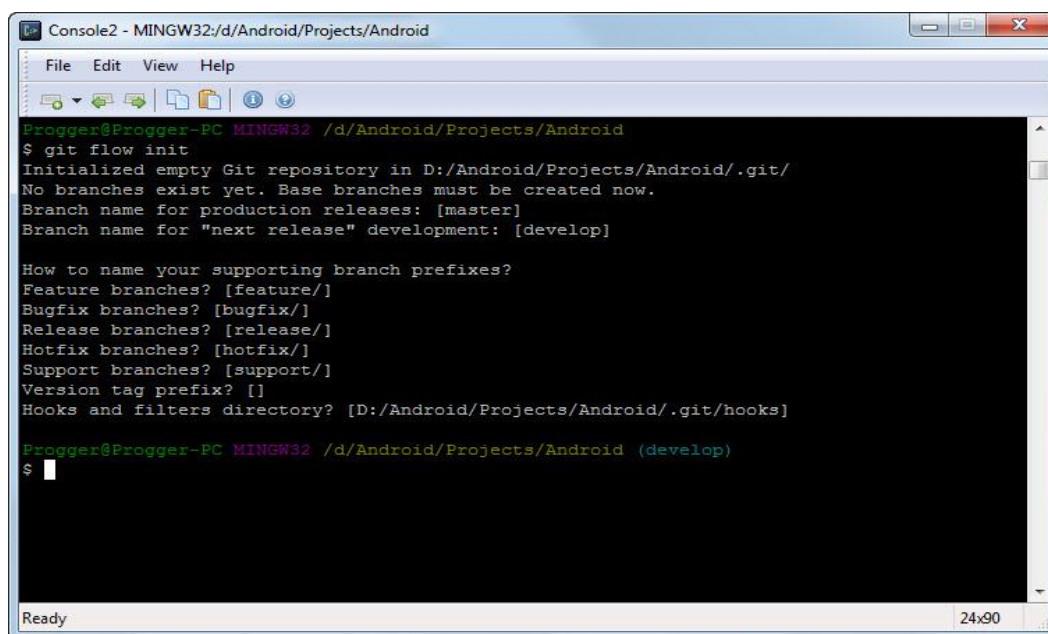
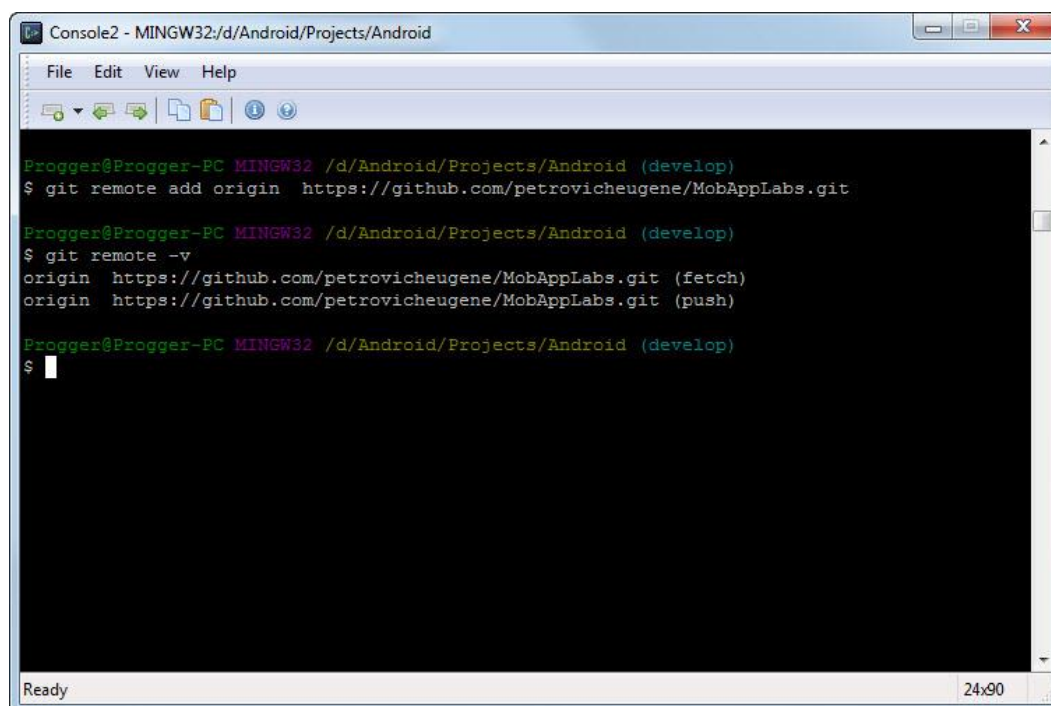


Рис. 6

2. Создаём репозиторий для лабораторных работ на сайте «Github» и привязываем к нему наш локальный репозиторий командой «git remote add origin <repository url>». Проверяем результат командой «git remote -v». (Рис. 7).



```
Progger@Progger-PC MINGW32 /d/Android/Projects/Android (develop)
$ git remote add origin https://github.com/etrovicheugene/MobAppLabs.git

Progger@Progger-PC MINGW32 /d/Android/Projects/Android (develop)
$ git remote -v
origin https://github.com/etrovicheugene/MobAppLabs.git (fetch)
origin https://github.com/etrovicheugene/MobAppLabs.git (push)

Progger@Progger-PC MINGW32 /d/Android/Projects/Android (develop)
$
```

Рис. 7

Далее все изменения в рабочем проекте можно сохранять в локальном репозитории командами «git add» и «git commit» и в удаленном репозитории на сайте «Github» командой «git push origin». Также есть возможность восстановить рабочий проект из репозитория до любого сохраненного состояния.

### ***Добавление в проект модулей для лабораторных работ***

Создадим в проекте для каждой лабораторной работы отдельный модуль.

1. Выбираем пункт меню File->New->New Module. В открывшемся окне «New Module» выбираем шаблон «Android Library» (Рис. 8).

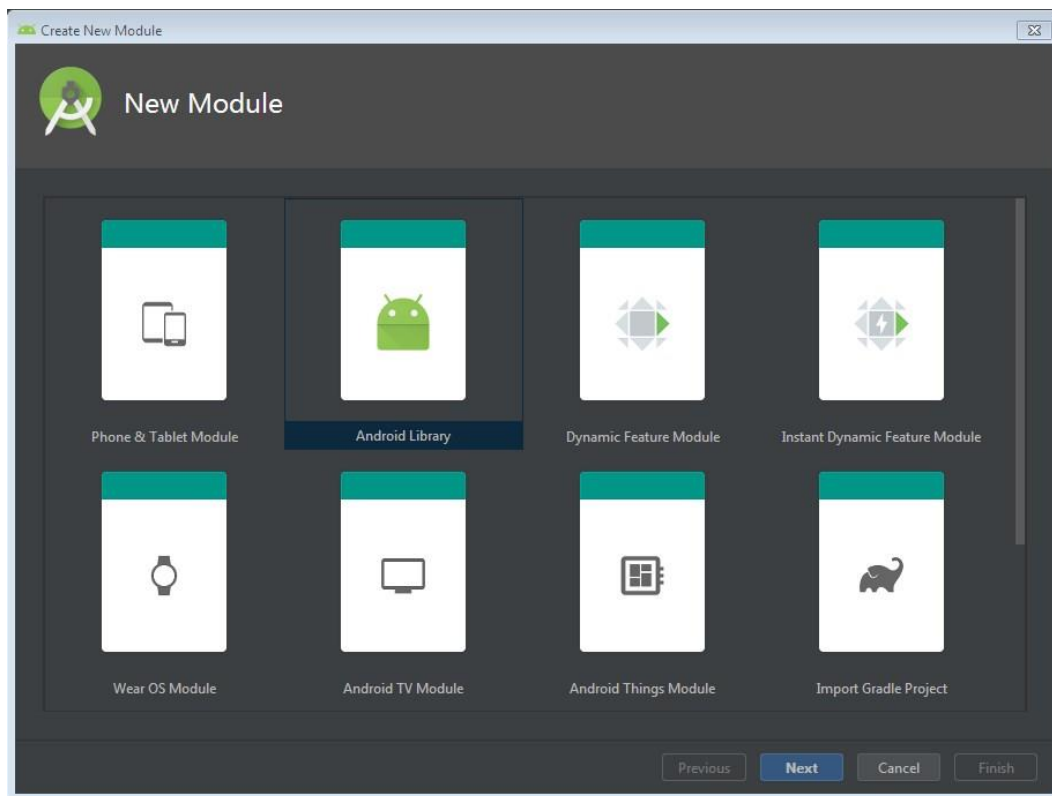


Рис. 8

2. В окне «Android Library» настраиваем заголовок, имя и имя пакета каждого модуля (Рис. 9).
3. Далее в каждый модуль добавим свою Activity. Для этого на каждом появившемся в проекте модуле открываем контекстное меню и выбираем пункты New->Activity->Empty Activity. В окне «Configure Activity» меняем имя Activity на соответствующее, например, для модуля «lab1» - «Lab1Activity» (Рис. 10).
4. В Activity каждой лабораторной работы добавляем статический метод Intent, который создает и возвращает объект класса Activity:

```
public static Intent newIntent(@NonNull Context context) {
    return new Intent(context, Lab1Activity.class);
}
```

В результате получаем заданную структуру модулей проекта (Рис. 11).



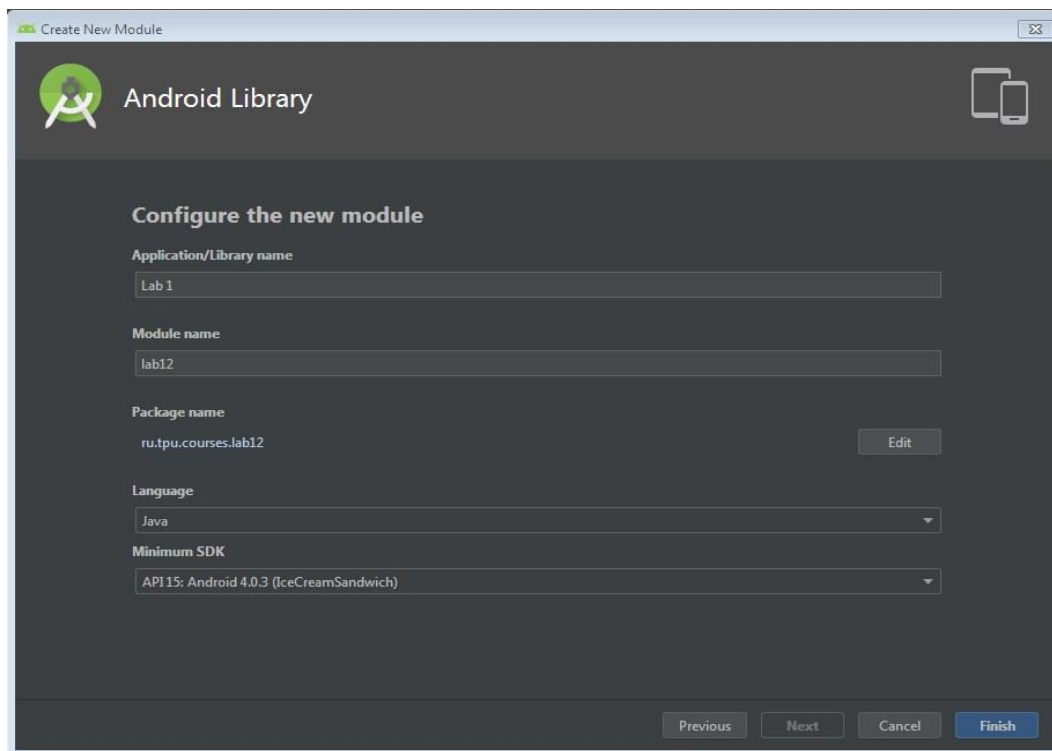


Рис. 9

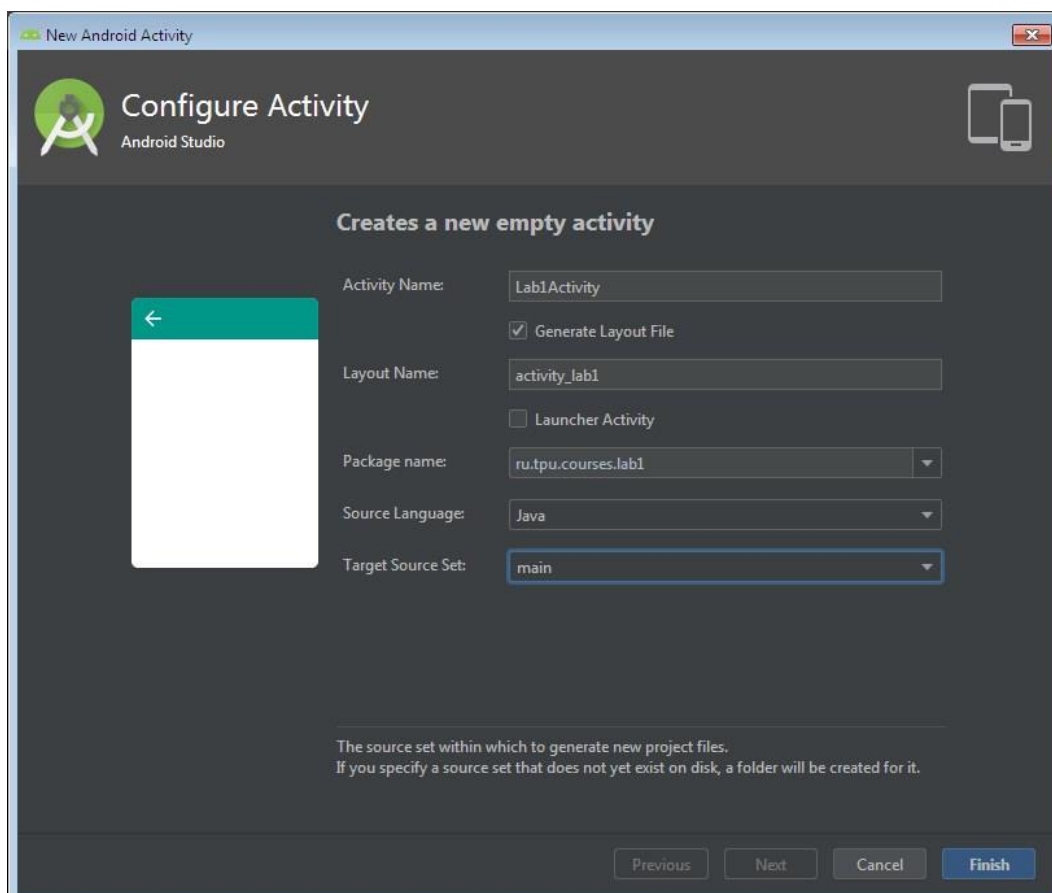


Рис. 10

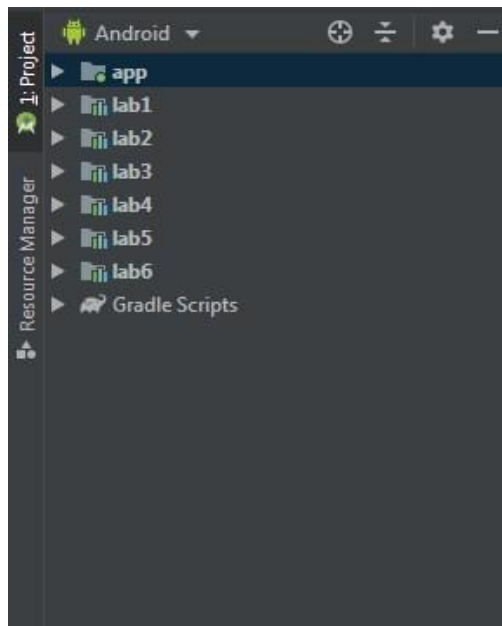


Рис. 11

### **Настройка зависимостей модулей**

Для того, чтобы функциональность одного модуля была доступна для кода другого модуля, необходимо в использующий модуль внедрить зависимость от используемого модуля. В «Android Studio» это удобно сделать при помощи редактора структуры проекта.

1. Для этого выбираем пункт меню File->Project Structure. В открывшемся окне на левой панели выбираем пункт «Dependencies», на панели «Modules» - модуль «app» и на панели «Declared Dependencies» нажимаем +.
2. В открывшемся окне отмечаем все модули «lab» (Рис. 12). Нажимаем ОК. В файле сборки build.gradle модуля «app» появились инструкции сборки модулей «lab» (Рис. 13).

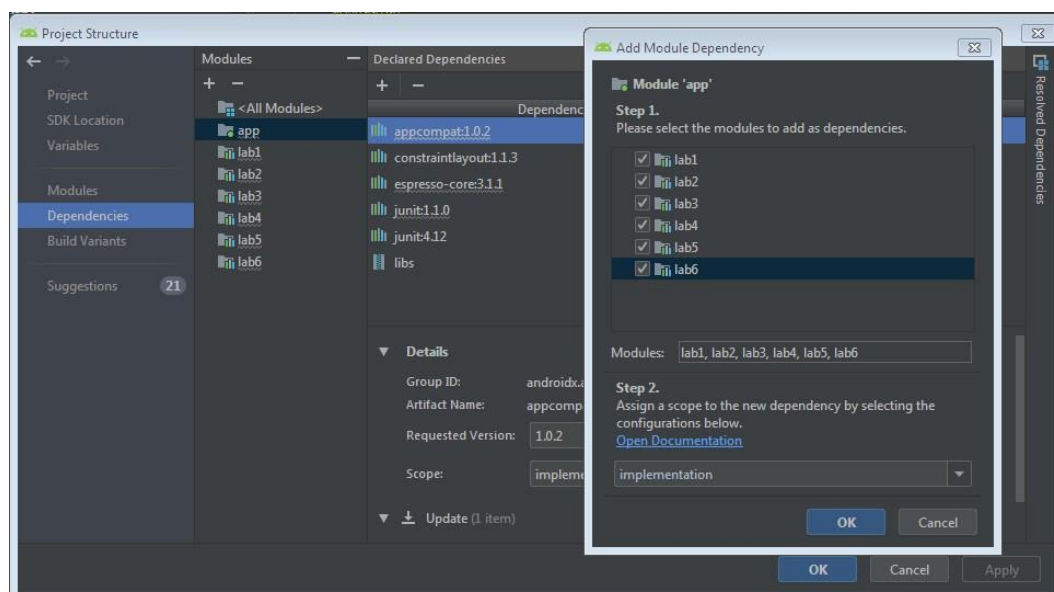


Рис. 12

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.0.2'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.0'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'
    implementation project(path: ':lab2')
    implementation project(path: ':lab4')
    implementation project(path: ':lab3')
    implementation project(path: ':lab5')
    implementation project(path: ':lab1')
    implementation project(path: ':lab6')
}
```

Рис. 13

### ***Создание кнопок для переходов к модулям лабораторных работ***

Создадим на макете «MainActivity» для каждой лабораторной работы кнопку при нажатии на которую будет выполнен переход на Activity соответствующей лабораторной работы. Для этого воспользуемся макетным редактором «Android Studio».

1. Заменим макет по умолчанию на макет вертикальный `LinearLayout` и перетаскиванием разместим на нем 6 кнопок «`Button`» (Рис. 14).
2. Для каждой кнопки на панели атрибутов назначим значения атрибутов высоты и ширины «`wrap_content`» и значения `id` для каждой кнопки повторяет имя соответствующего ей модуля лабораторной работы. Например, для лабораторной работы №1 это будет «`lab1`» и т.д.

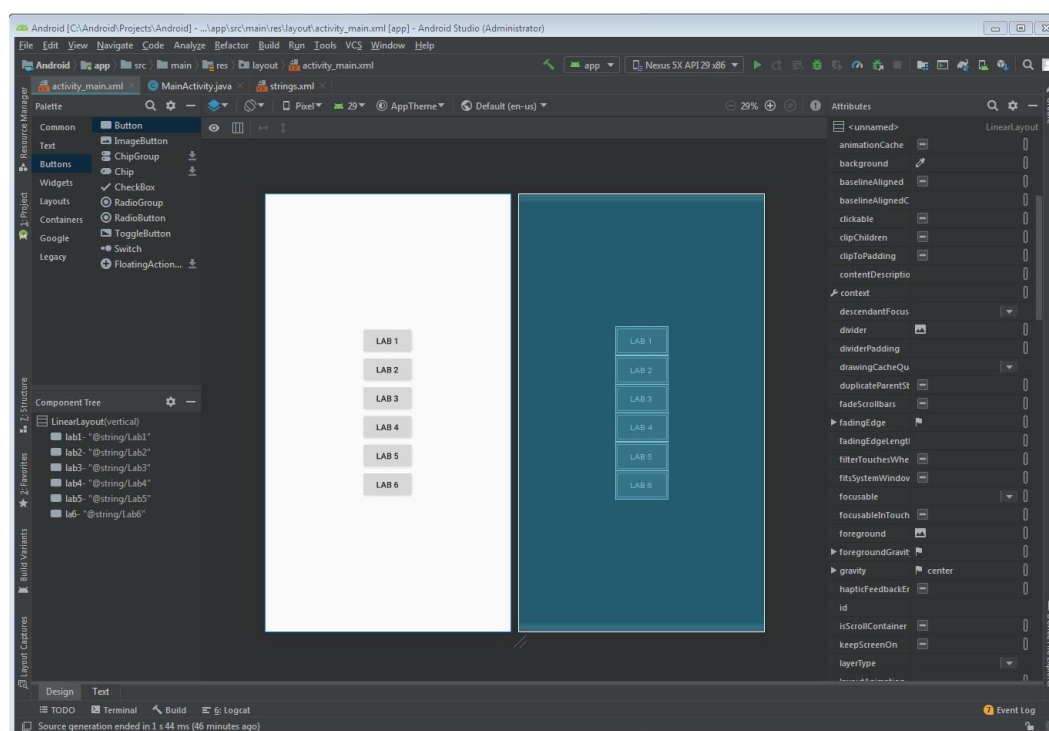


Рис. 14

3. В методе `onCreate()` класса `MainActivity`, который вызывается после создания класса, каждой кнопке назначаем «слушателя» - `Activity`, которая будет запущена при нажатии:

```
findViewById(R.id.lab2).setOnClickListener((v) ->  
startActivity(Lab2Activity.newIntent(this)));
```

### *Запуск и тестирование приложения*

Собранное приложение запускаем на реальном устройстве Samsung Galaxy Note 9 и проверяем переходы на `Activity` лабораторных работ (Рис. 15, Рис. 16, Рис. 17).

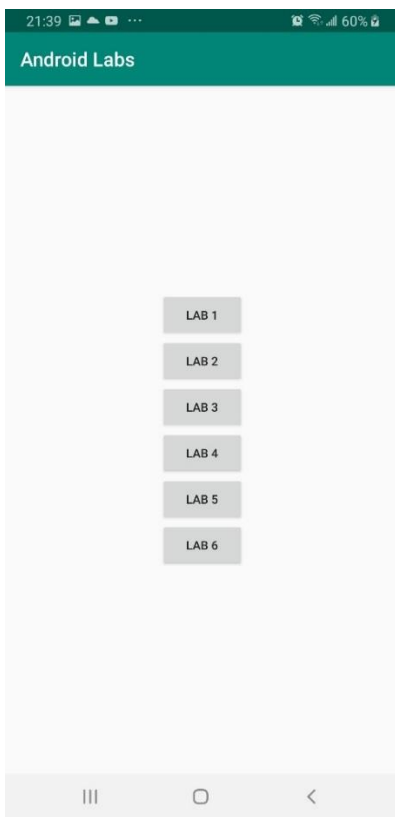


Рис. 15

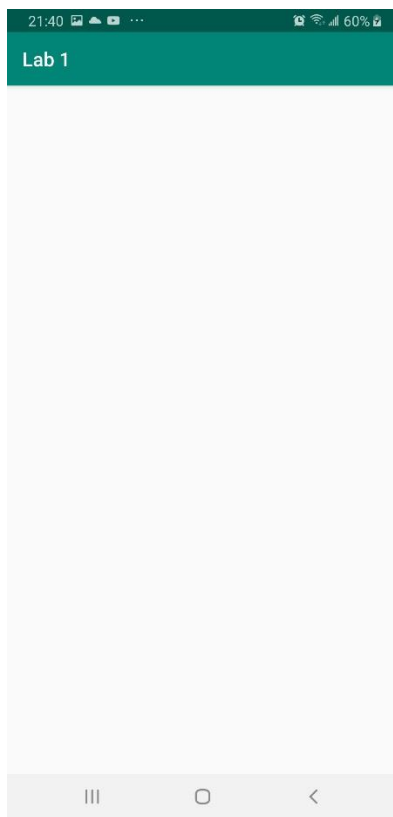


Рис. 16

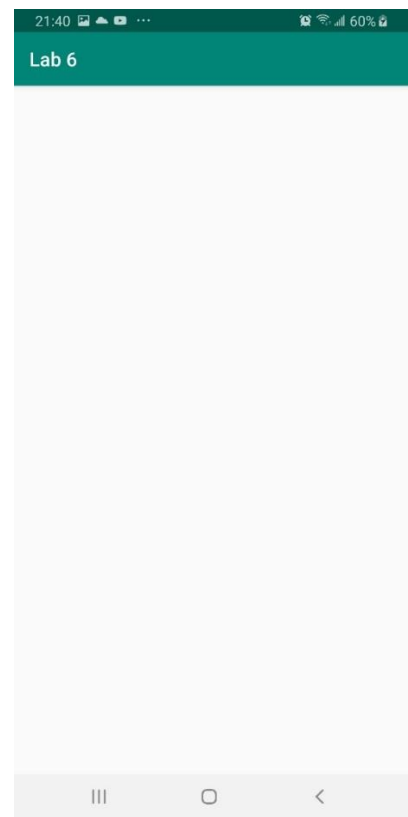


Рис. 17

## Заключение

В работе в интегрированной среде разработки «Android Studio» создан проект мобильного приложения с модульной структурой. Настроен макет основной Activity для выполнения переходов на макеты Activity других модулей. Проект протестирован на реальном устройстве и загружен на сайт «Github».

## Приложение

*Ссылка на проект*

<https://github.com/etrovicheugene/MobAppLabs.git>