|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе №1**

по дисциплине «Разработка мобильных приложений»

|  |  |
| --- | --- |
| **Выполнил:**  Студент группы ИКБО-26-21 | Бондарь А.Р. |
| **Проверил:** | доцент Синицын И.В. |

Москва 2022 г.

**Цель практической работы**

В этой практической работе рассмотрим создание первого приложения для Android. Научимся создавать проекты и запускать приложение в режиме отладки. Так же узнаем об основах разработки приложения для Android, включая создание простого пользовательского интерфейса и обработку ввода информации пользователем.

# Создание проекта в Android Studio

Чтобы создать новый проект в Android Studio, надо отрыть данное приложение. Щелкнуть на кнопку New Project на экране приветствия. Или если открыт другой проект, щелкнуть меню File и выберать New Project.

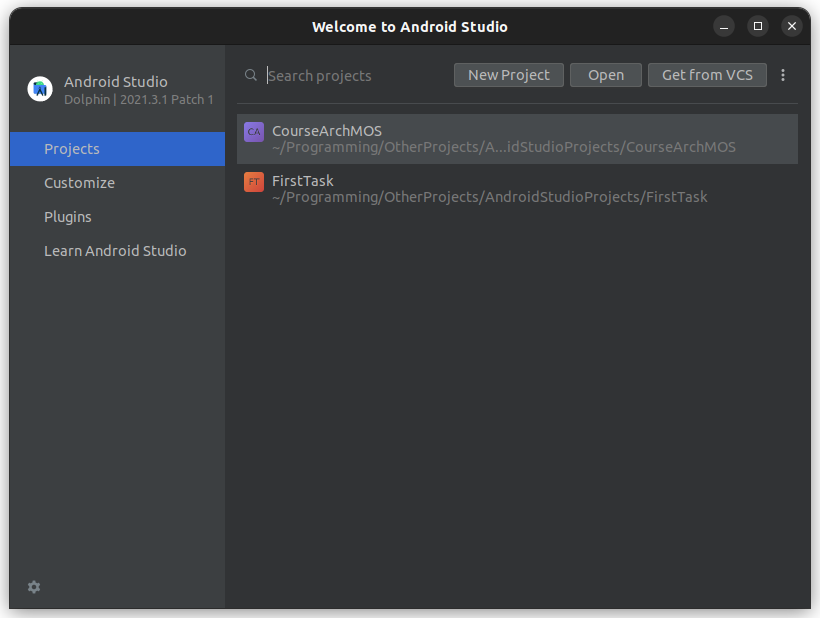


Рисунок 1.1 – Создание проекта в Android Studio

В передложенном списке стартовых activity выбираем наиболее подходящий и далее необходимо заполнить поля в окне Configure your new project.

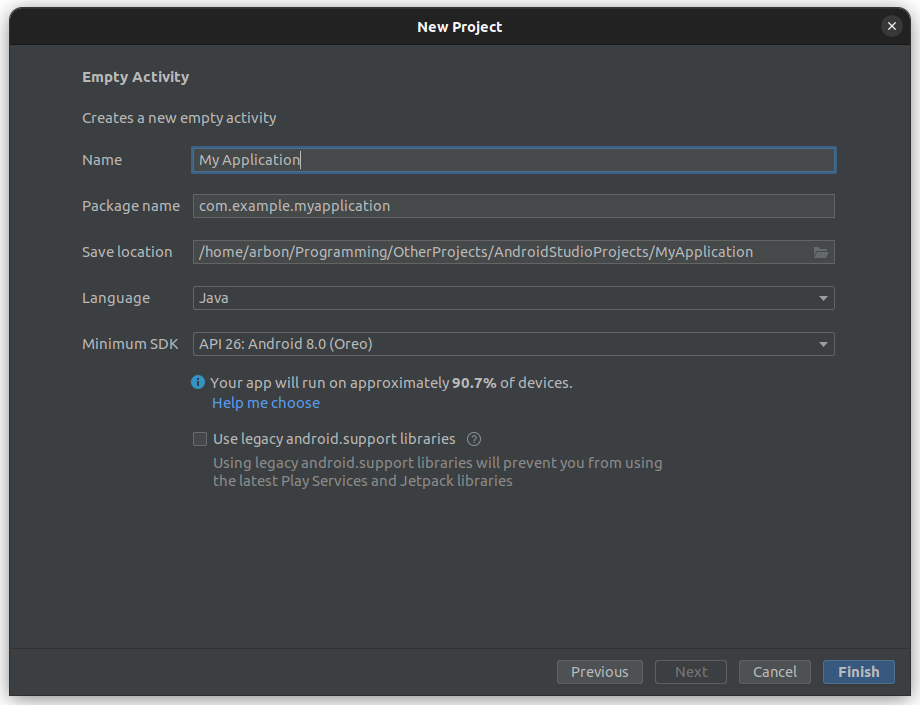


Рисунок 1.2 – Настройка проекта

# Создание разметки

Создание разметки в XML файлах предпочтительнее, чем в исходном коде по нескольким причинам, но главным образом из-за необходимости создания различных файлов разметки для устройств с различными размерами экрана.

## Создание линейной разметки (Linear Layout)

1. В Android Studio открыли файл res/layout/activity\_my.xml

2. В окне предпросмотра щелкнули по иконке Hide , чтобы скрыть окно.

3. Удалили элемент <TextView>.

4. Изменили элемент <RelativeLayout> на <LinearLayout>.

5. Добавили атрибут android:orientation и установите для него значение "horizontal".

6. Удалите атрибуты android:padding и tools:context.

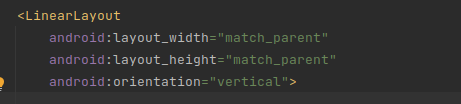


Рисунок 2.1.1 – Настройка разметри

## **Добавление текстового поля**

Как и для каждого объекта типа View, необходимо указать некоторые XML атрибуты, характерные для элемента EditText.

1. В файле activity\_my.xml, создали внутри <LinearLayout> элемент <EditText> и указали для него атрибут android:id со значением @+id/textView.

2. Создали атрибуты layout\_width и layout\_height со значением «match\_parent» и «wrap\_content», соответственно.

3. Создали атрибут hint и указали в качестве значения строковый объект с названием input\_suggestion.

В результате элемент <EditText> должен выглядеть следующим образом:

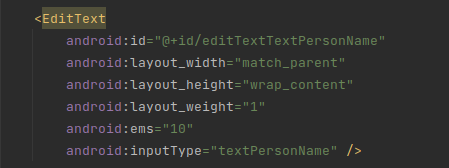


Рисунок 2.2.1 – Настройка текстового поля

## **Объекты ресурсов**

Объект ресурса имеет уникальное целочисленное значение, связанное с ресурсами приложения, такими как растровые изображения, файлы разметки или строки.

Каждый ресурс соответствует объекту ресурса, объявленному в файле gen/R.java.

По умолчанию строковые ресурсы хранятся в файле res/values/strings.xml. Добавили новый ресурс "input\_suggestion" и указали для него значение “Введите свой текст”.

1. В Android Studio открыли файл res/values/strings.xml

2. Добавили строку с названием "input\_suggestion" и значением “Введите свой текс”.

3. Добавили строку с названием "buttonText\_view" и значением “Отправить”.

4. Удалили строку, содержащую надпись "hello world".

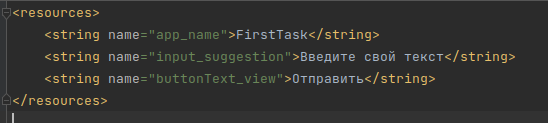


Рисунок 2.3.1 – Создание ресурсов

## **Добавление кнопки**

1. В Android Studio открыли файл res/layout/activity\_my.xml.
2. Создали внутри <LinearLayout> элемент <Button> сразу после элемента <EditText>.
3. Задали ширину как "wrap\_content", чтобы она зависела от надписи внутри кнопки.
4. Добавили атрибут android:text и указали в качестве значения строковый ресурс "buttonText\_view".

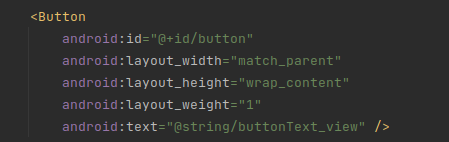


Рисунок 2.4.1 – Создание кнопки

## **Растягиваем поле ввода**

Для того, чтобы растянуть элемент EditText на все свободное пространство в файле activity\_my.xml добавили элементу <EditText> атрибут layout\_weight со значением 1, а атрибуту layout\_width установим значение 0dp.

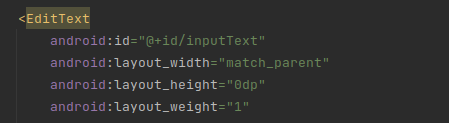


Рисунок 2.5.1 – Растянули поля ввода

## Запуск другого явление

Другое явление будет запускаться при нажатии кнопки в текущем.

Для этого в java-классе объявили объект кнопки и присвоили ему найденную по id кнопку расположенную в разметке. Далее установили обработчик событий на эту кнопу. Так что при ее нажатии, будет создаваться намеренье, которому передается текущий контекст и класс явление, которое будет запускаться.

Также в намеренье, через метод putExtra, сохраниться текст введенный пользователем в поля ввода.

И в конце вызывается метод startActivity с новосозданным намереньем.

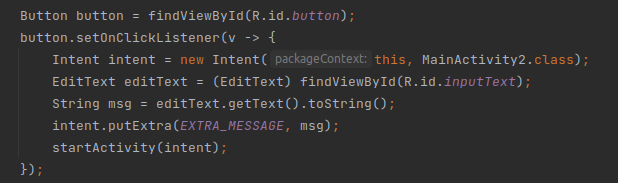


Рисунок 2.6.1 – Запуск нового явления

## **Создание нового явления в Android Studio**

1. В Android Studio щелкнули правой кнопкой мыши по пакету java/com.mycompany.myfirstapp и выберали New > Activity > Blank Activity.
2. В окне Choose options настроили новое явление и нажали Finish.
3. Открыли появившийся файл MainActivity2.java и удалили все лишние методы, оставив только onCreate.

## Отображение **сообщения в новом явлении**

Чтобы отобразить сообщение, нужно:

1. В разметку добавить виджет <TextView>.
2. В классе создать соответствующий объект текстовой надписи.
3. Получить объект намеренья.
4. Установить текст виджету, получив текст из намеренья с помощью метода getStringExtra.

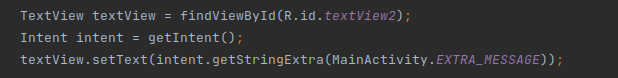


Рисунок 2.8.1 – Код java для отображение сообщения

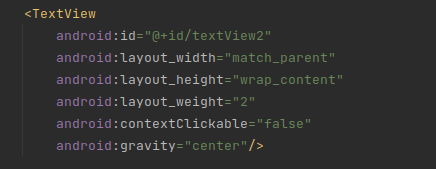


Рисунок 2.8.2 – Разметка для отображение сообщения

# Вывод

В ходе практической работы было создано первое приложение для Android. Научились создавать проекты и запускать приложение в режиме отладки. Так же узнали об основах разработки приложения для Android, такие как создание разметки с текстовыми полями, полями ввода текста и кнопки, и обработку ввода информации пользователем.