



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИИТ)

Кафедра математического обеспечения и стандартизации ИТ

ОТЧЁТ ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №2

по дисциплине

«Разработка мобильных приложений»

Отчет представлен к

рассмотрению:

Студенты группы ИНБО-04-20

«28» февраля 2022 г.

Ло В.Х.

(подпись)

Преподаватель

« »

2022 г.

Фандеев И.И.

(подпись)

Москва, 2022

СОДЕРЖАНИЕ

1.Создайте проект с Android Studio.....	3
2.Произвести запуск приложения.....	4
3. Создание простого пользовательского интерфейса.....	7
4.Создать второе явление.....	10

1. Создайте проект с Android Studio

- В Android Studio создать новый проект.
- В меню « File» выберите «New project» .
- В разделе « Set up a new project» заполните поля, как показано на рис 1, и нажмите « Finish » .

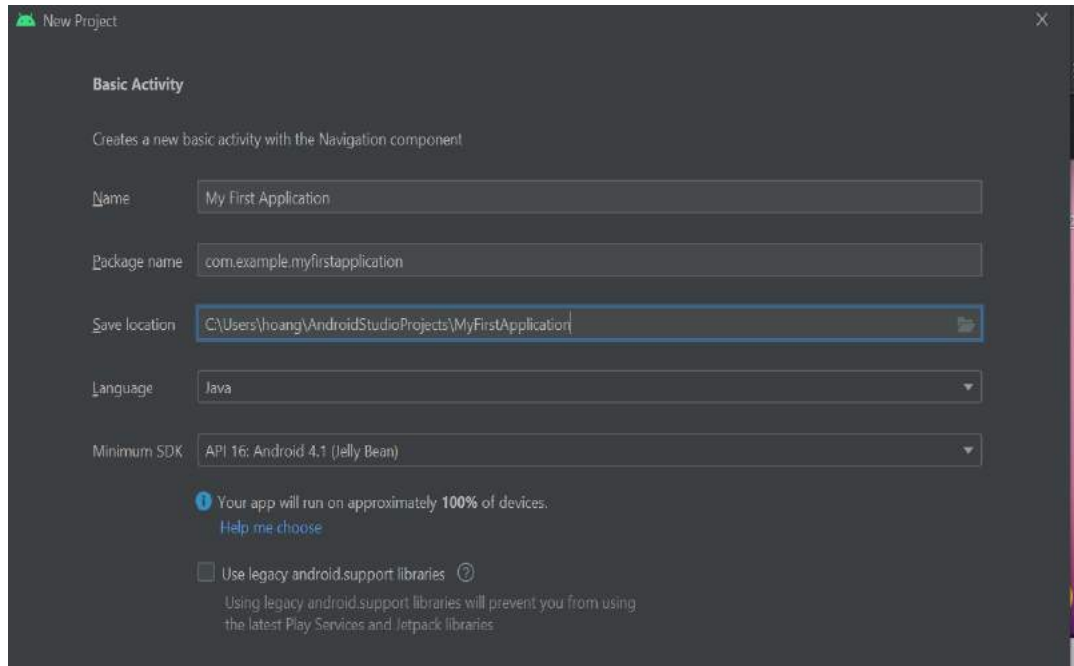


Рис 1. Настройка нового проекта в Android Studio.

Во-первых, мы должны решить, на каком устройстве Android мы хотим разработать наше приложение. Варианты: телефон и планшет, Wear OS (часы Android), телевизор и Android (рис 2).

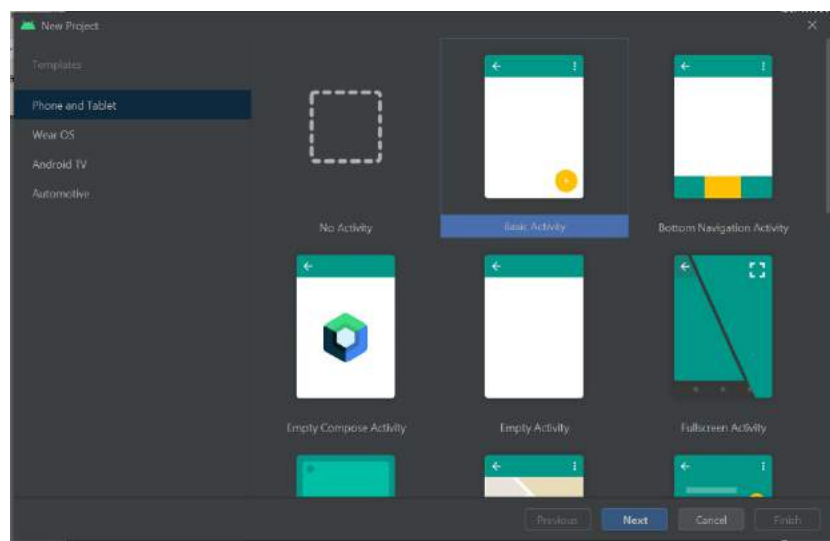


Рис 2. Blank activity

Взгляните на мой проект студии Android(рис 3).

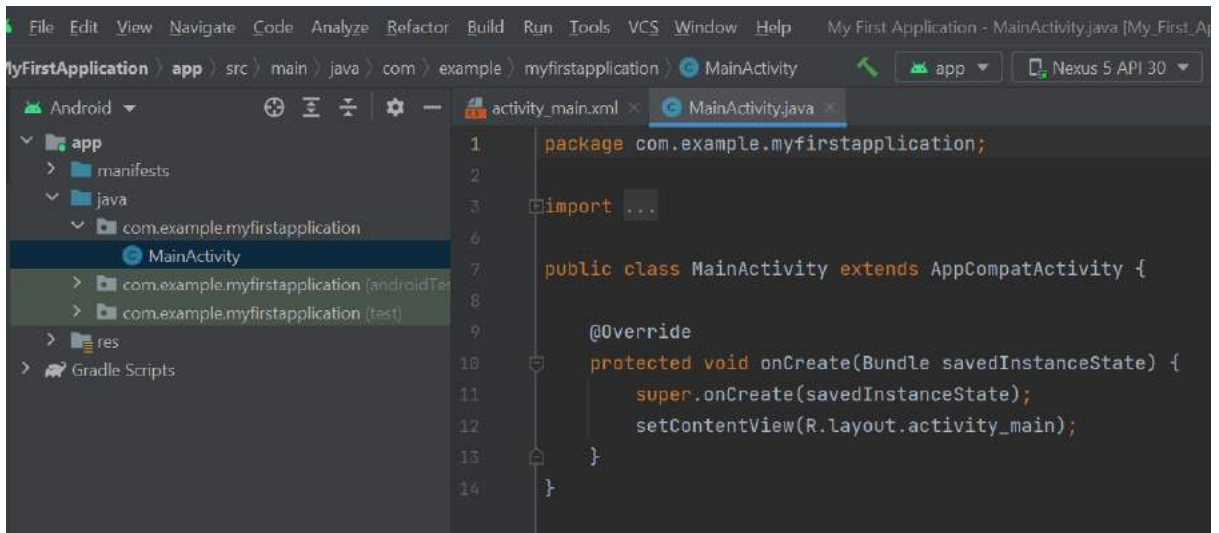


Рис 3. Мой проект студии Android

Создали новый проект студии Android. Android Studio создает для нас один XML-файл макета и один файл Java. Как видите, файл MainActivity.java уже открыт. Например, если бы вы выбрали Kotlin, то это был бы файл Kotlin.

2. Произвести запуск приложения

Запустить на эмуляторе

Независимо от того, используете ли вы Android Studio или командную строку, для запуска приложения в эмуляторе вам необходимо сначала создать виртуальное устройство Android (AVD). AVD — это конфигурация устройства для эмулятора Android, позволяющая моделировать конкретное устройство.

Создать AVD :

1. На главном Device Manager нажмите << Create a virtual device >>(рис 4).

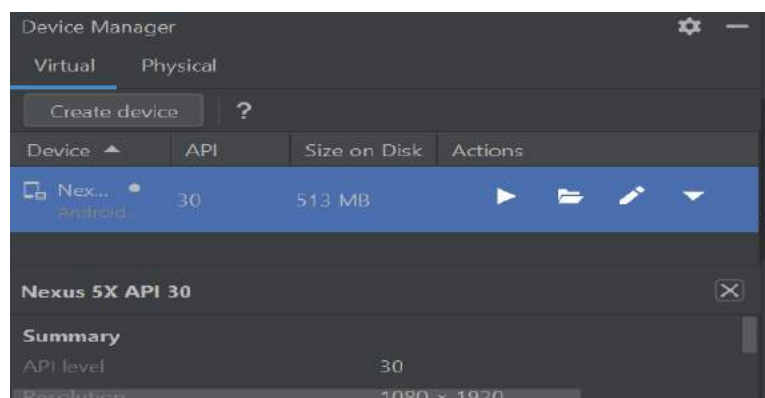


Рис 4. Device Manager

2. Выбрать тип устройства(рис 5) и приступайте к загрузке(рис 6).

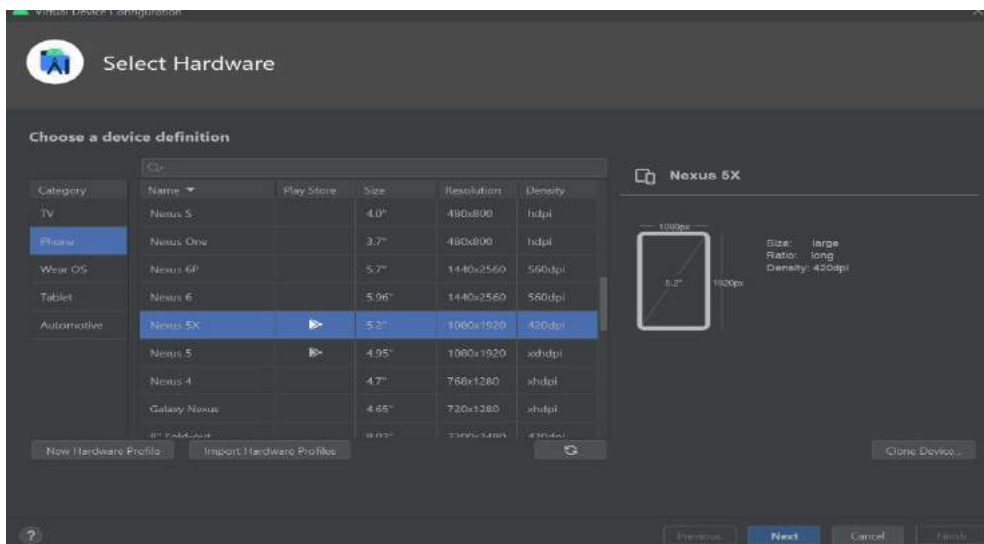


Рис 5. Тип устройства

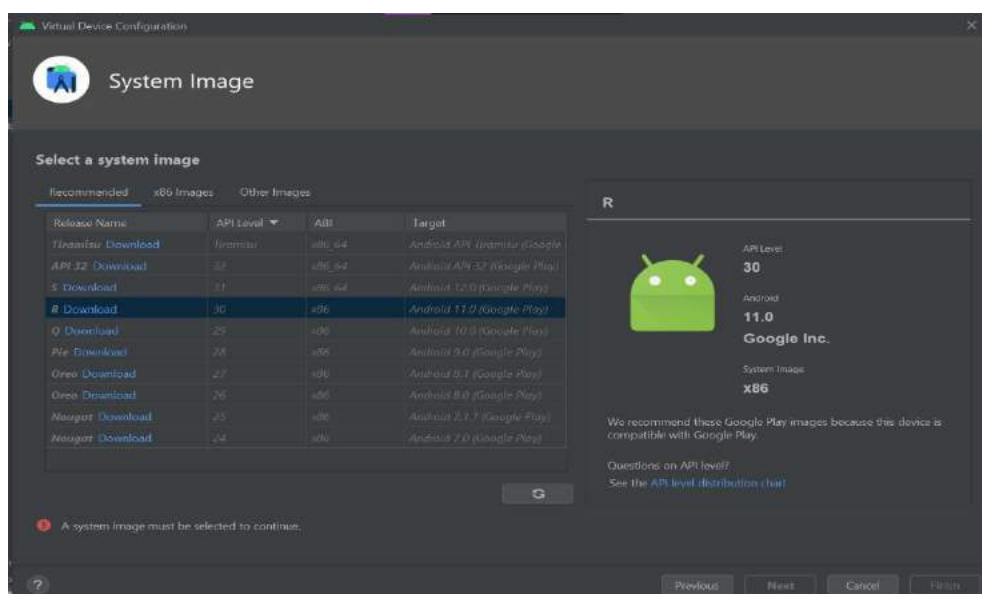



Рис 6. Загрузке

Запустите приложение из Android Studio

- В Android Studio выбрать свой проект и нажмите «Выполн»  на панели инструментов.
- В окне «Choose Device» нажмите кнопку-переключатель «Launch emulator».
- В раскрывающемся меню Android virtual device выберите созданный emulator и нажмите «ОК» (рис 7).

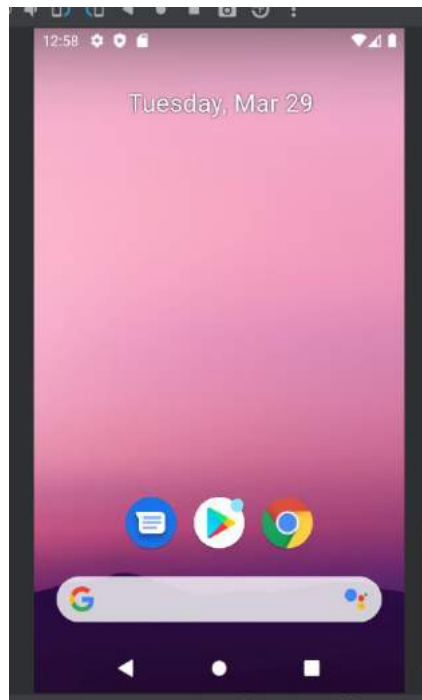


Рис 7. Android virtual device

3. Создание простого пользовательского интерфейса

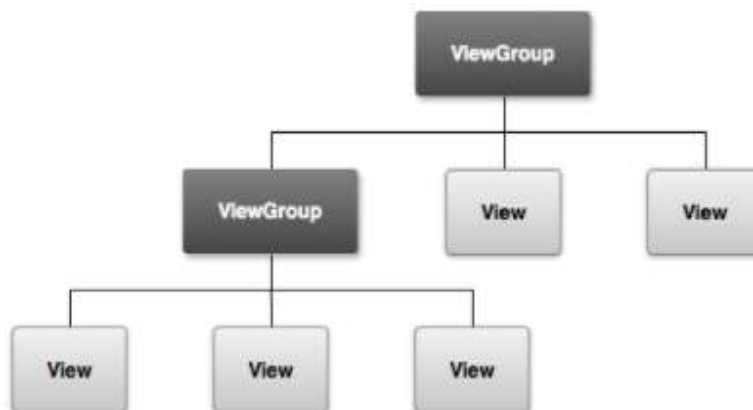


Рис 8. ViewGroup объекты формируют ответвления в макете и содержат другие Viewобъекты.

Создать линейную разметку (Linear Layout)

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    android:orientation="horizontal" >
</LinearLayout>

```

Рис 9. Linear Layout

Добавить текстовое поле



Рис 10. текстовое поле

Добавить строковые ресурсы

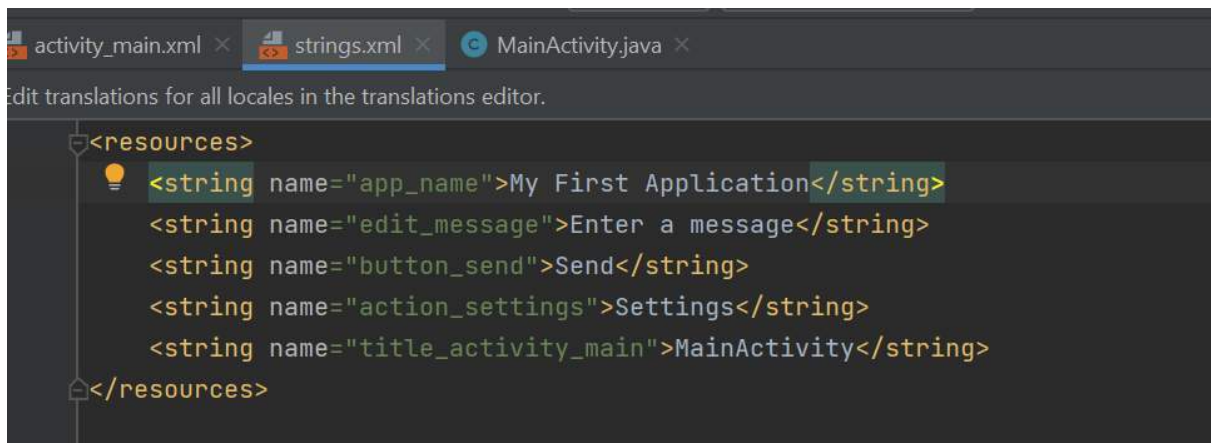


Рис 11. Добавить строковые ресурсы

Добавить кнопку

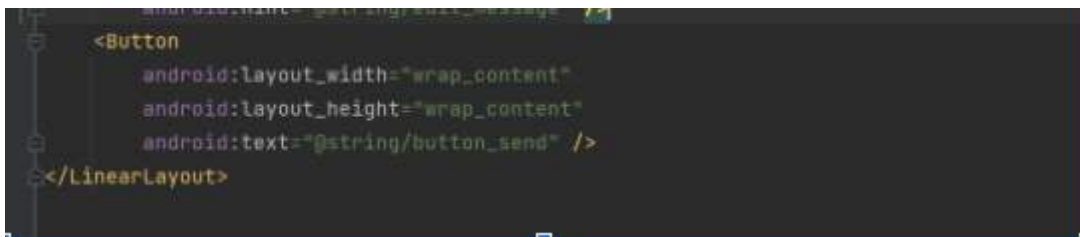


Рис 12. Кнопка

Сделайте так, чтобы поле ввода заполнило ширину экрана

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="horizontal" >
    <EditText
        android:id="@+id/edit_message"
        android:layout_width="8dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:hint="@string/edit_message" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_send"
    />
</LinearLayout>

```

Рис 13. file activity_main.xml

Запустить приложение

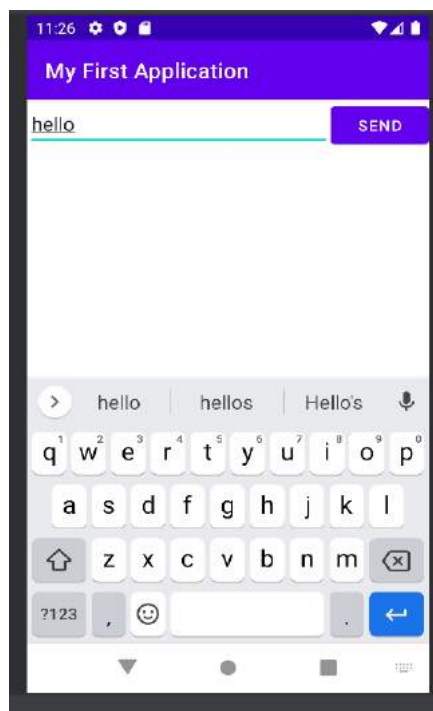


Рис 14.Результат

Запустить другое явление

Реализовать обработку нажатия на кнопку:

К <Button> элементу добавить android:onClick атрибут.
res/layout/activity_main.xml


```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_send"
    android:onClick="sendMessage" />
```

Рис 15. Button

Создать намерение

```
public class MainActivity extends AppCompatActivity {
    public final static String EXTRA_MESSAGE =
        "com.mycompany.myfirstapplication.MESSAGE";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void sendMessage(View view) {
        Intent intent = new Intent(this, DisplayMessageActivity.class);
        EditText editText = (EditText) findViewById(R.id.edit_message);
        String message = editText.getText().toString();
        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }
}
```

Рис 16. File MainActivity

4. Создать второе явление

Создать новое явление в Android Studio

В Android Studio в java каталоге выберите пакет com.mycompany.

myfirstapplication , щелкните правой кнопкой мыши и выберите New > Activity > Blank Activity ..

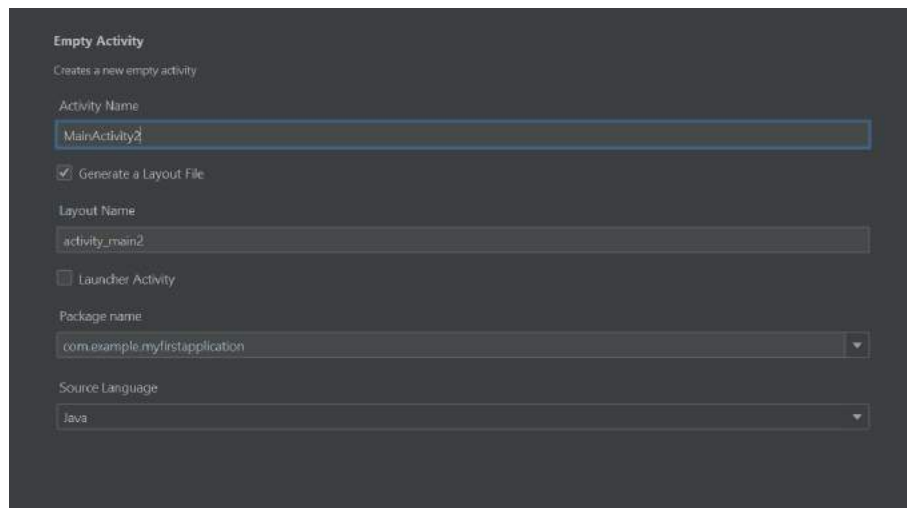


Рис 17. Button

Создать явления без Android Studio

Создать новый файл с именем DisplayMessageActivity.java в src/ каталоге проекта рядом с исходным MainActivity.java файлом.

Добавим в файл следующий код:

```
public class DisplayMessageActivity extends AppCompatActivity {

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    /**
     * A placeholder fragment containing a simple view.
     */
    public static class PlaceholderFragment extends Fragment {

        public PlaceholderFragment() { }

        @Override
        public View onCreateView(LayoutInflater inflater, ViewGroup container,
                                Bundle savedInstanceState) {
            View rootView = inflater.inflate(R.layout.fragment_display_message,
                container, false);
            return rootView;
        }
    }
}
```

Рис 18. File DisplayMessageActivity

Обеспечить получение намерения.

В свой strings.xml файл добавим заголовок нового действия следующим образом:

```
<resources>
    <string name="app_name">My First Application</string>
    <string name="edit_message">Enter a message</string>
    <string name="button_send">Send</string>
    <string name="action_settings">Settings</string>
    <string name="title_activity_main">MainActivity</string>
    <string name="sendMessage"></string>
    <string name="title_activity_display_message">My Message</string>
</resources>
```

Рис 19. File strings.xml

В файле манифеста AndroidManifest.xml внутри Application элемента добавим <activity>элемент для своего DisplayMessageActivity класса следующим образом

```
<activity
    android:name="com.example.myfirstapplication.DisplayMessageActivity"
    android:label="@string/title_activity_display_message"
    android:parentActivityName="com.example.myfirstapplication.MainActivity" />
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="com.example.myfirstapplication.MainActivity" />
</activity>
```

Рис 20. File AndroidManifest.xml

Обеспечить отображение сообщения.

Полный onCreate() метод на DisplayMessageActivity данный момент выглядит так:

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Get the message from the intent
    Intent intent=getIntent();
    String message=intent.getStringExtra(MainActivity.EXTRA_MESSAGE);

    // Create the text view
    TextView textView=new TextView(this);
    textView.setTextSize(40);
    textView.setText(message);

    // Set the text view as the activity layout
    setContentView(textView);
}

```

Рис 21. onCreate()метод

Запустить приложение

Когда откроется, введем сообщение в текстовое поле, нажмем «Send», и сообщение появится во втором действии.

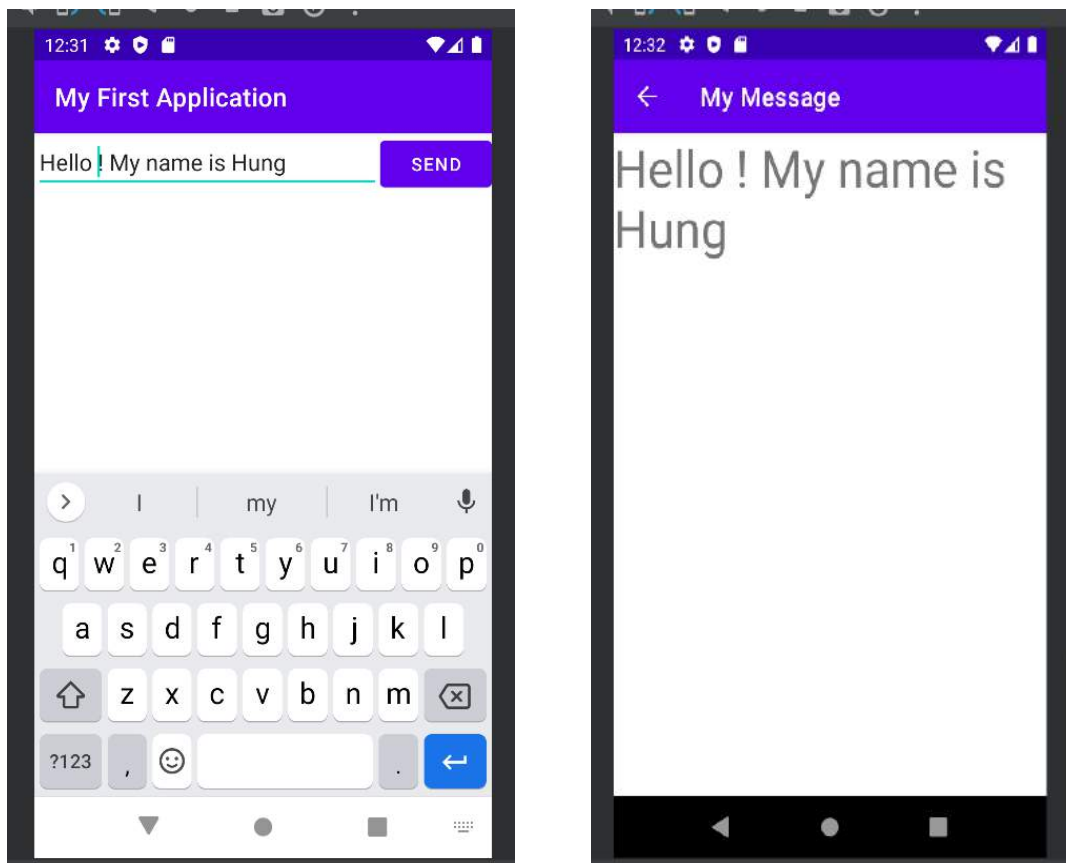


Рис 21. Оба действия в финальном приложении, работающем на Android

Вот и все, Я создал свое первое приложение для Android!