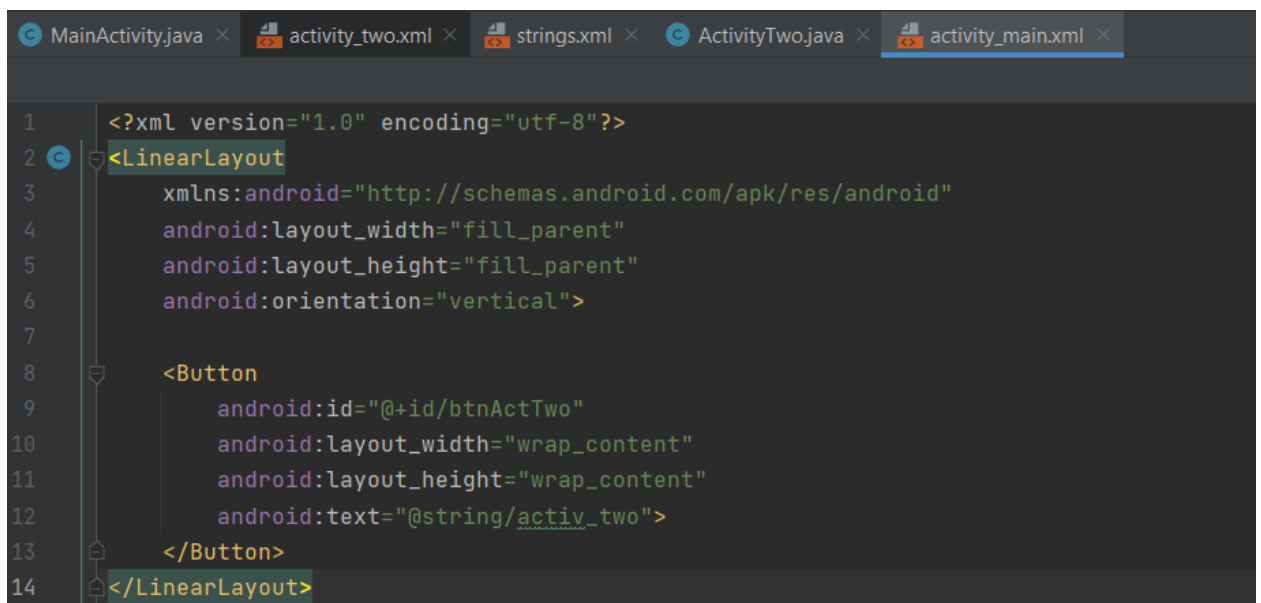


Отчёт по проделанной работе на семинаре по мобильной разработке (22.10.2021).

(выполнил Валяев Георгий Анатольевич)

Не менее важным в процессе разработки мобильного приложения является создание и подключение новых активностей к главному окну приложения. В данном отчёте нам предстоит ознакомиться с этим процессом и научиться связывать вызовы активностей друг с другом при помощи кнопок с соответствующими подписями.

Создадим пустой проект и в activity_main.xml зададим линейное расположение (LinearLayout) простой кнопки (название занесём в переменную ресурсного файла strings.xml), при помощи которой мы будем переходить на 2 окно активности.

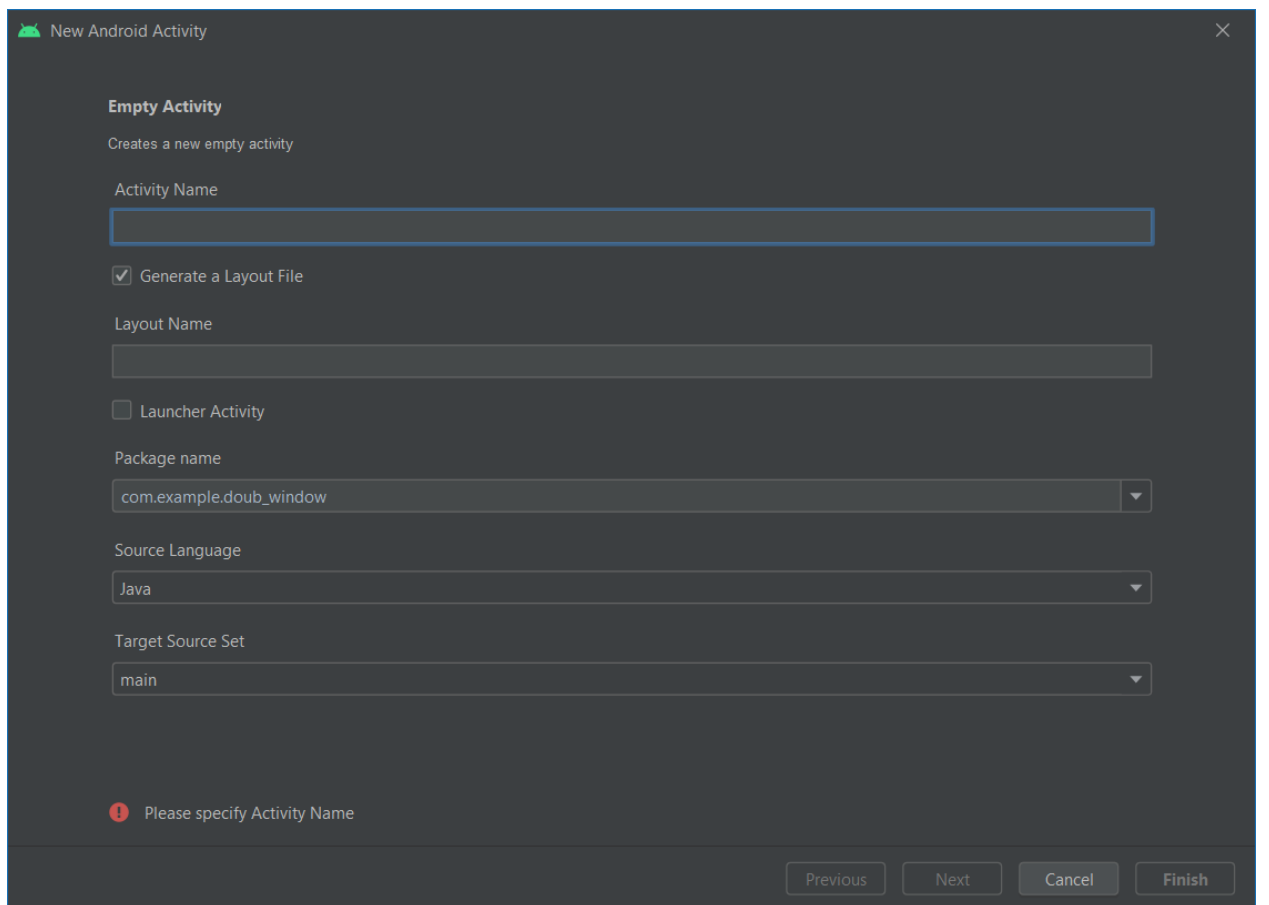
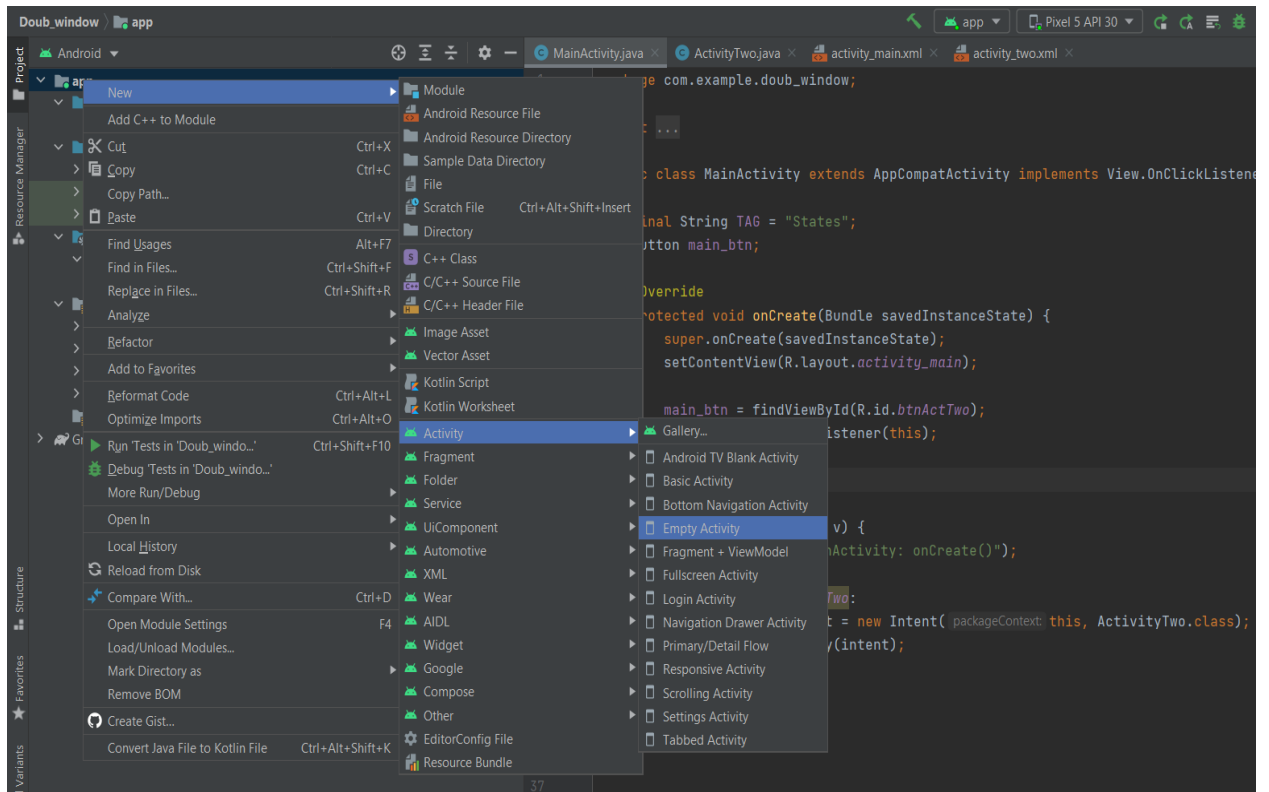


```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:layout_width="fill_parent"
5      android:layout_height="fill_parent"
6      android:orientation="vertical">
7
8      <Button
9          android:id="@+id/btnActTwo"
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="@string/activ_two">
13     </Button>
14 </LinearLayout>
```

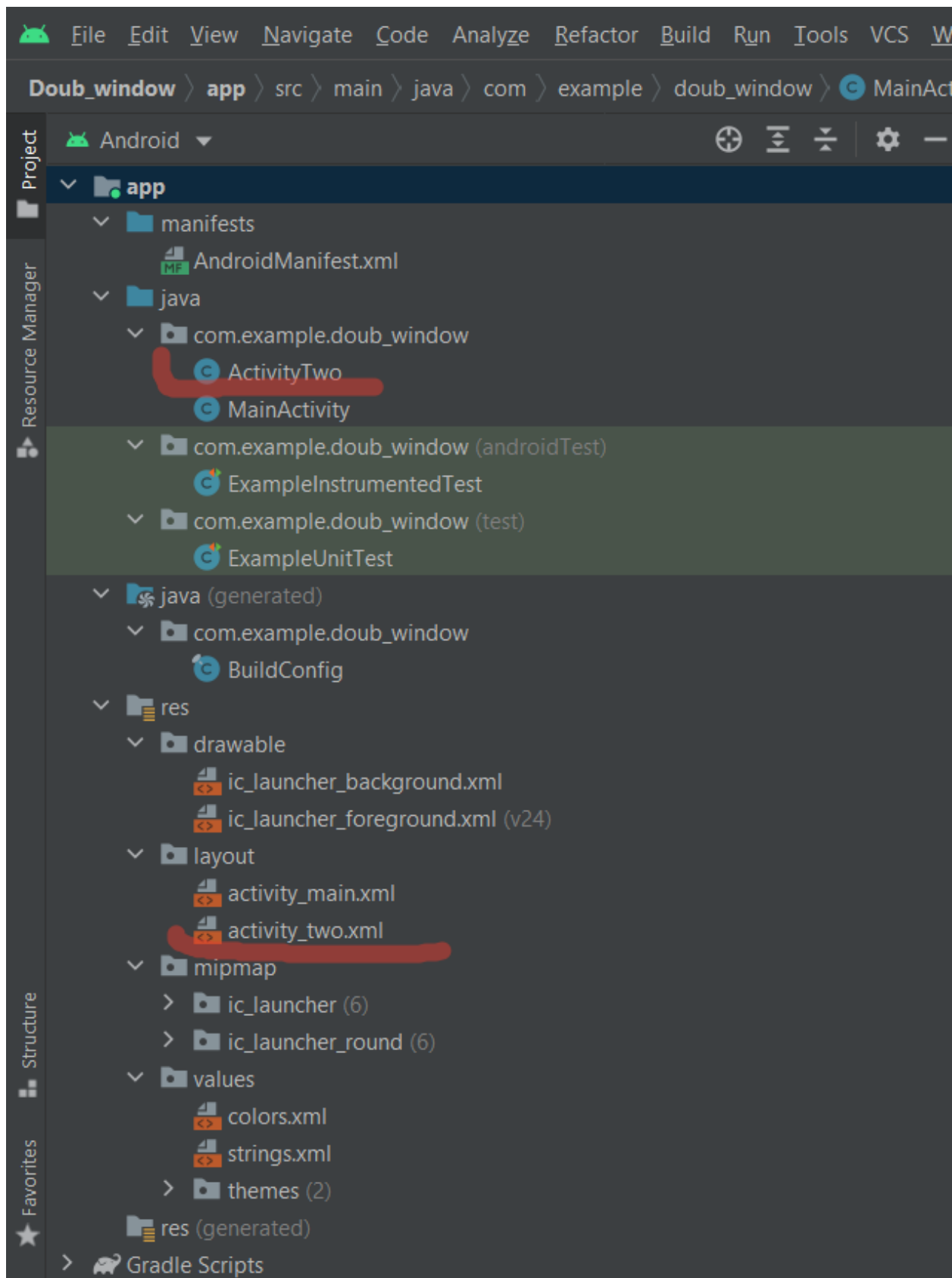
Далее в MainActivity.java создаём переменную кнопки, соответствующий обработчик нажатия кнопки через метод и присваивание данных о нажатии переменной типа Button, при помощи которой можно будет переключаться между активностями:

```
MainActivity.java x ActivityTwo.java x activity_main.xml x activity_two.xml x
1 package com.example.doub_window;
2
3 import ...
10
11 public class MainActivity extends AppCompatActivity implements View.OnClickListener{
12
13     final String TAG = "States";
14     Button main_btn;
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_main);
20
21         main_btn = findViewById(R.id.btnActTwo);
22         main_btn.setOnClickListener(this);
23     }
24
25     @Override
26     public void onClick(View v) {
27         Log.d(TAG, msg: "MainActivity: onCreate()");
28         switch (v.getId()) {
29             case R.id.btnActTwo:
30                 Intent intent = new Intent( packageContext: this, ActivityTwo.class);
31                 startActivity(intent);
32                 break;
33             default:
34                 break;
35         }
36     }
```

Пришло время создать 2 окно активности. Переходим в описание главной папки app, жмём правой кнопкой мыши на неё и выбираем New Activity -> Empty Activity и задаём название активности – «ActivityTwo».



В результате чего мы теперь имеем 2 новых файла проекта – ActivityTwo и activity_two.xml, а также новые строки активности в файле манифеста приложения:



```
MainActivity.java x AndroidManifest.xml x strings.xml x activity_main.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.doub_window">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="Doub_window"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportsRtl="true"
11        android:theme="@style/Theme.Doub_window">
12        <activity
13            android:name=".ActivityTwo"
14            android:exported="true" />
15        <activity
16            android:name=".MainActivity"
17            android:exported="true">
18            <intent-filter>
19                <action android:name="android.intent.action.MAIN" />
20
21                <category android:name="android.intent.category.LAUNCHER" />
22            </intent-filter>
23        </activity>
24    </application>
25
26 </manifest>
```

Далее перейдём в activity_two.xml и зададим обычное текстовое поле с информацией перехода пользователем на 2 окно активности приложения:

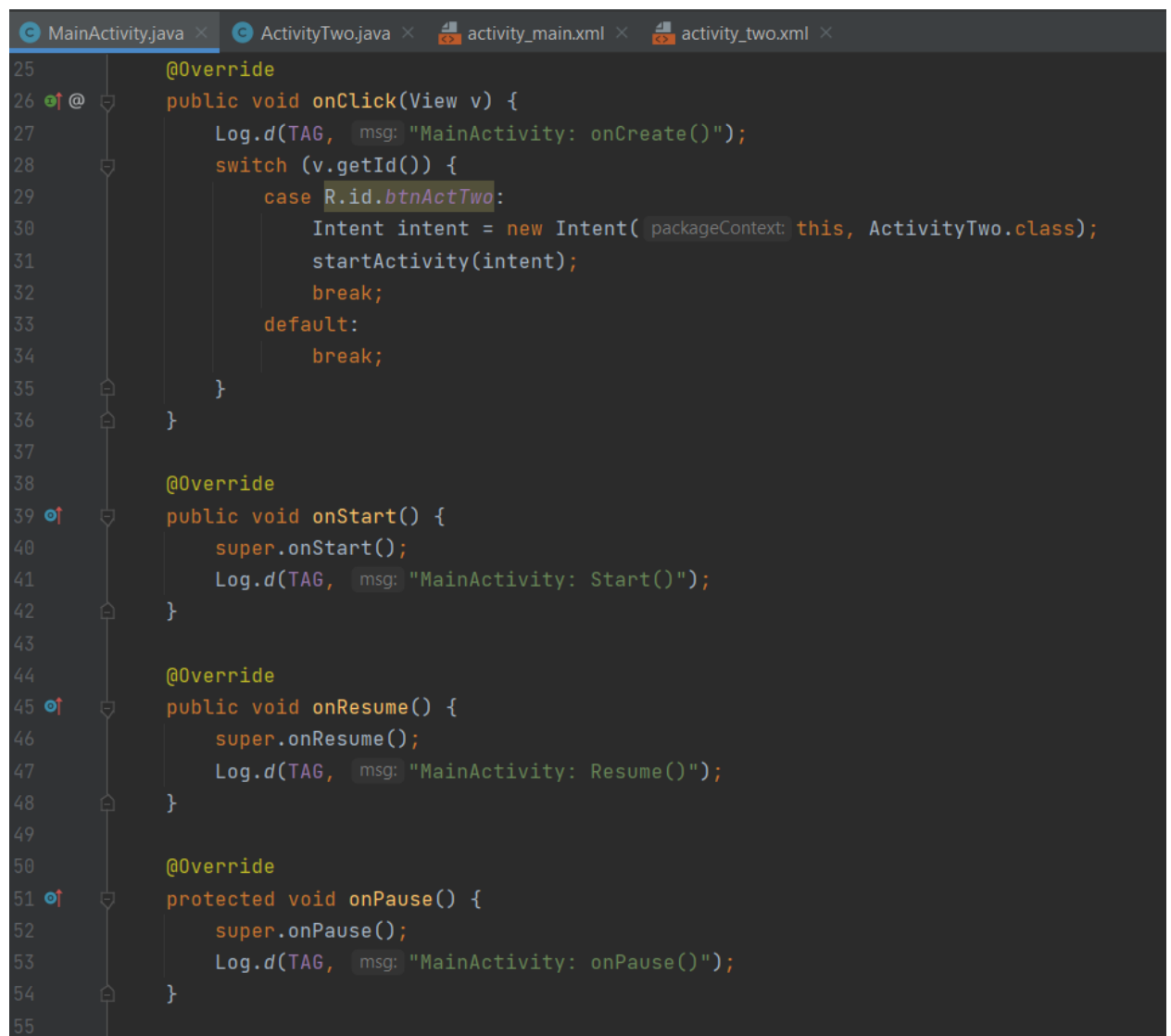
```
MainActivity.java x ActivityTwo.java x activity_main.xml x activity_two.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical">
7
8     <TextView
9         android:id="@+id/textView1"
10        android:layout_width="wrap_content"
11        android:layout_height="wrap_content"
12        android:text="This is Activity Two">
13    </TextView>
14
15 </LinearLayout>
```

Для более полного понимания жизненного цикла активности нужно прописать в java-коде главной активности системные сообщения о стадии жизненного цикла на протяжении пользования приложения пользователем.

Сначала для этого пропишем в начале объявление глобальной строковой переменной:

```
13      final String TAG = "States";
```

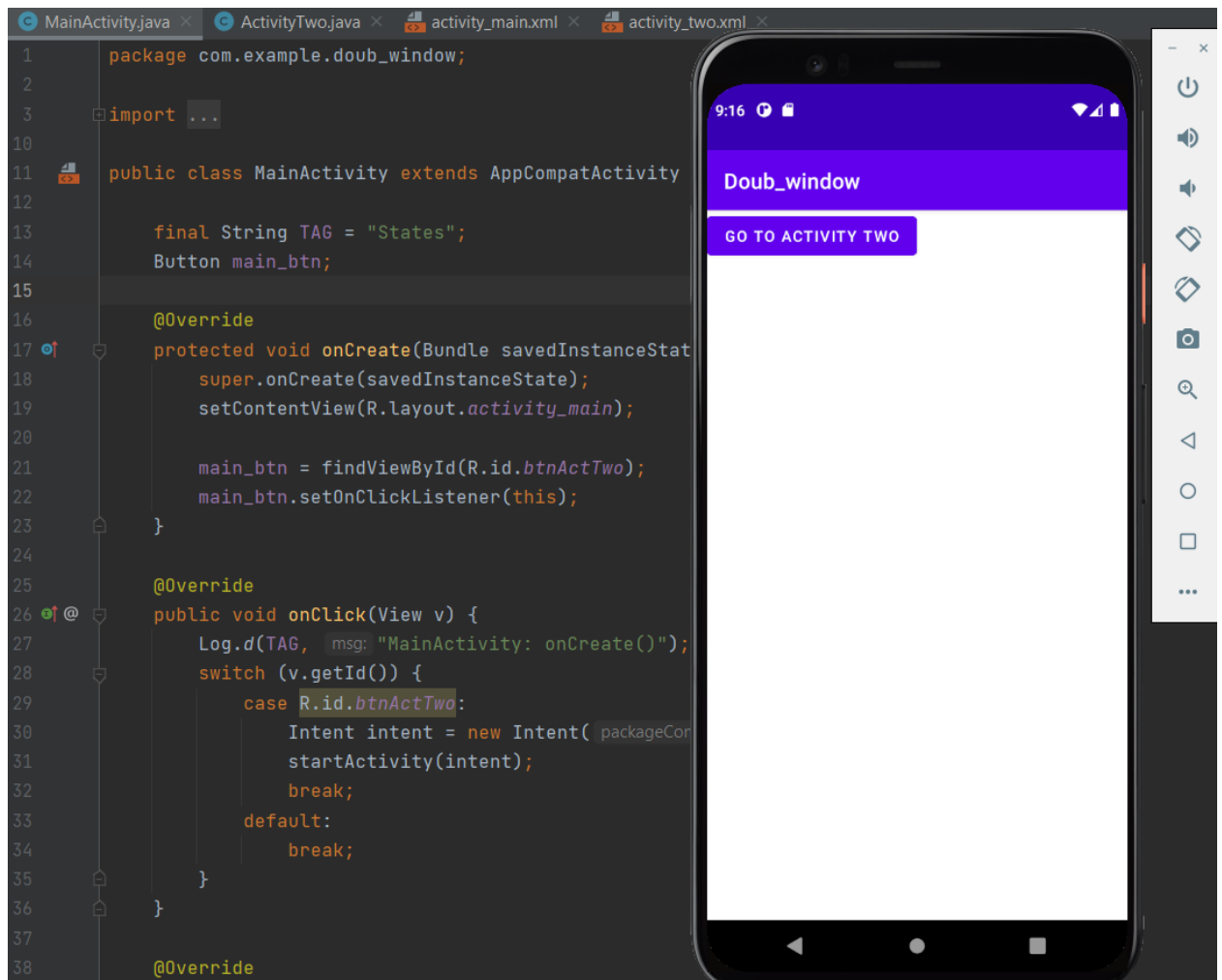
Затем опишем основные сообщения в консоль о стадиях жизненного цикла активности:

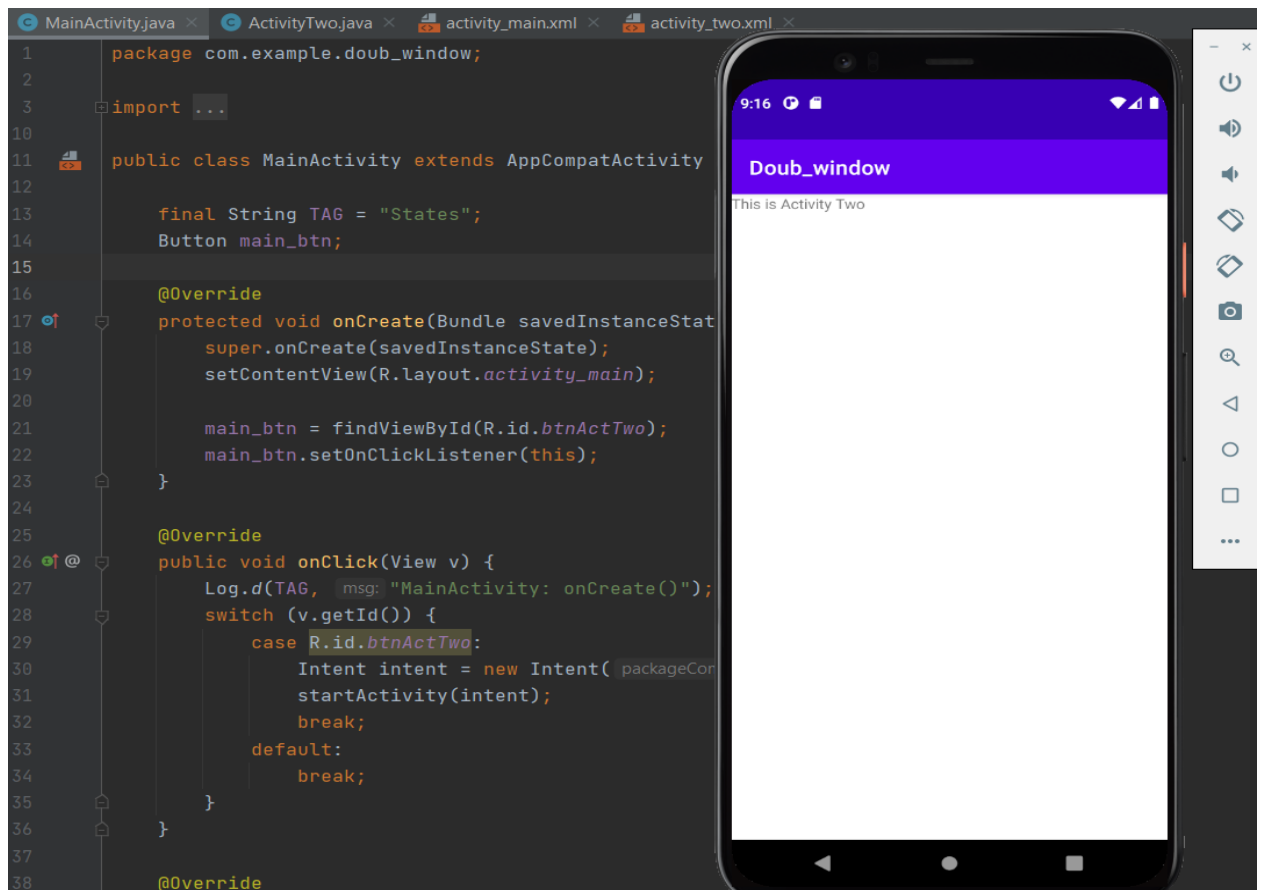


```
MainActivity.java x ActivityTwo.java x activity_main.xml x activity_two.xml x
25      @Override
26      public void onClick(View v) {
27          Log.d(TAG, msg: "MainActivity: onCreate()");
28          switch (v.getId()) {
29              case R.id.btnActTwo:
30                  Intent intent = new Intent( packageContext: this, ActivityTwo.class);
31                  startActivity(intent);
32                  break;
33              default:
34                  break;
35          }
36      }
37
38      @Override
39      public void onStart() {
40          super.onStart();
41          Log.d(TAG, msg: "MainActivity: Start()");
42      }
43
44      @Override
45      public void onResume() {
46          super.onResume();
47          Log.d(TAG, msg: "MainActivity: Resume()");
48      }
49
50      @Override
51      protected void onPause() {
52          super.onPause();
53          Log.d(TAG, msg: "MainActivity: onPause()");
54      }
55
```

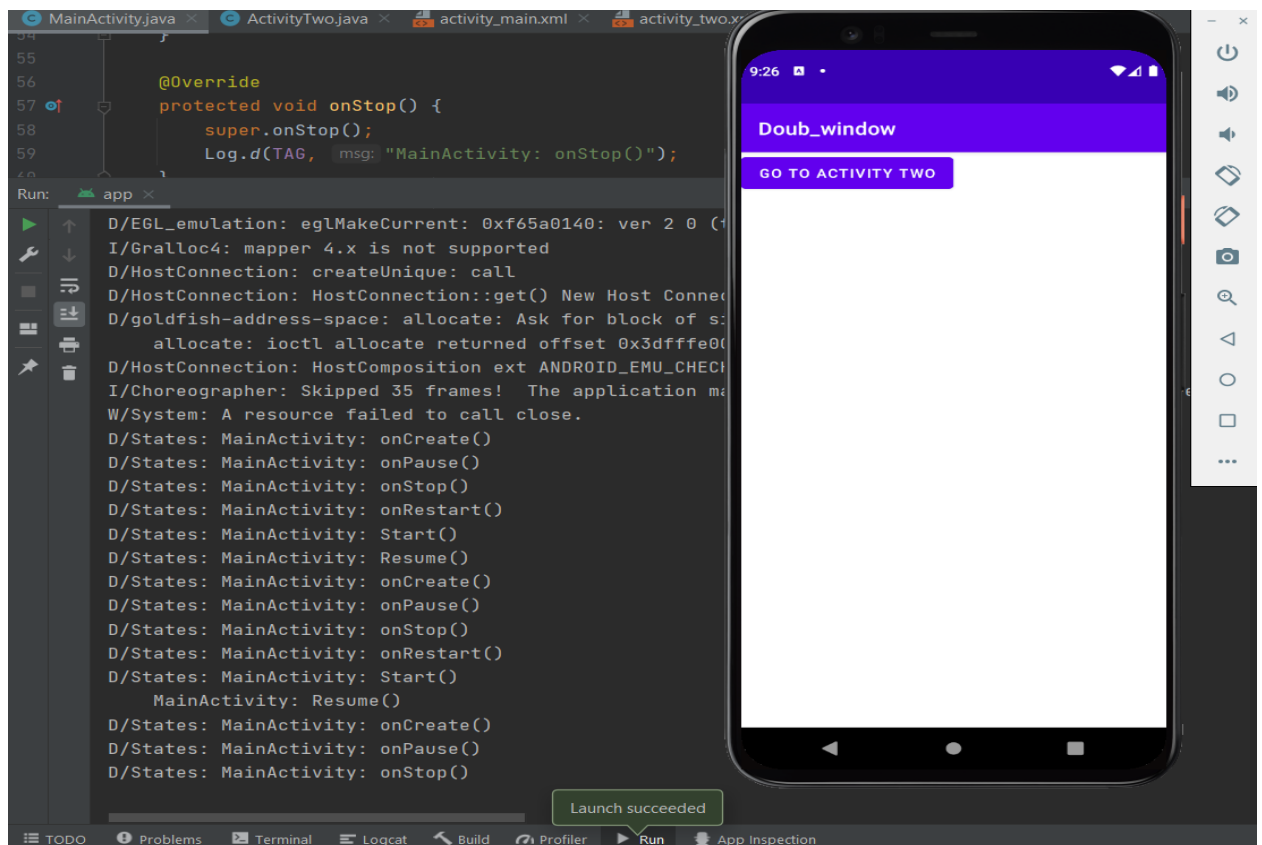
```
MainActivity.java x ActivityTwo.java x activity_main.xml x activity_two.xml x
46         super.onResume();
47         Log.d(TAG, msg: "MainActivity: Resume()");
48     }
49
50     @Override
51     protected void onPause() {
52         super.onPause();
53         Log.d(TAG, msg: "MainActivity: onPause()");
54     }
55
56     @Override
57     protected void onStop() {
58         super.onStop();
59         Log.d(TAG, msg: "MainActivity: onStop()");
60     }
61
62     @Override
63     protected void onDestroy() {
64         super.onDestroy();
65         Log.d(TAG, msg: "MainActivity: onDestroy()");
66     }
67
68     @Override
69     protected void onRestart() {
70         super.onRestart();
71         Log.d(TAG, msg: "MainActivity: onRestart()");
72     }
73
74 }
```

Таким образом, мы можем получить следующий результат:





Далее посмотрим в консоли все сообщения о стадиях жизненного цикла активности и убедимся в их правильности:



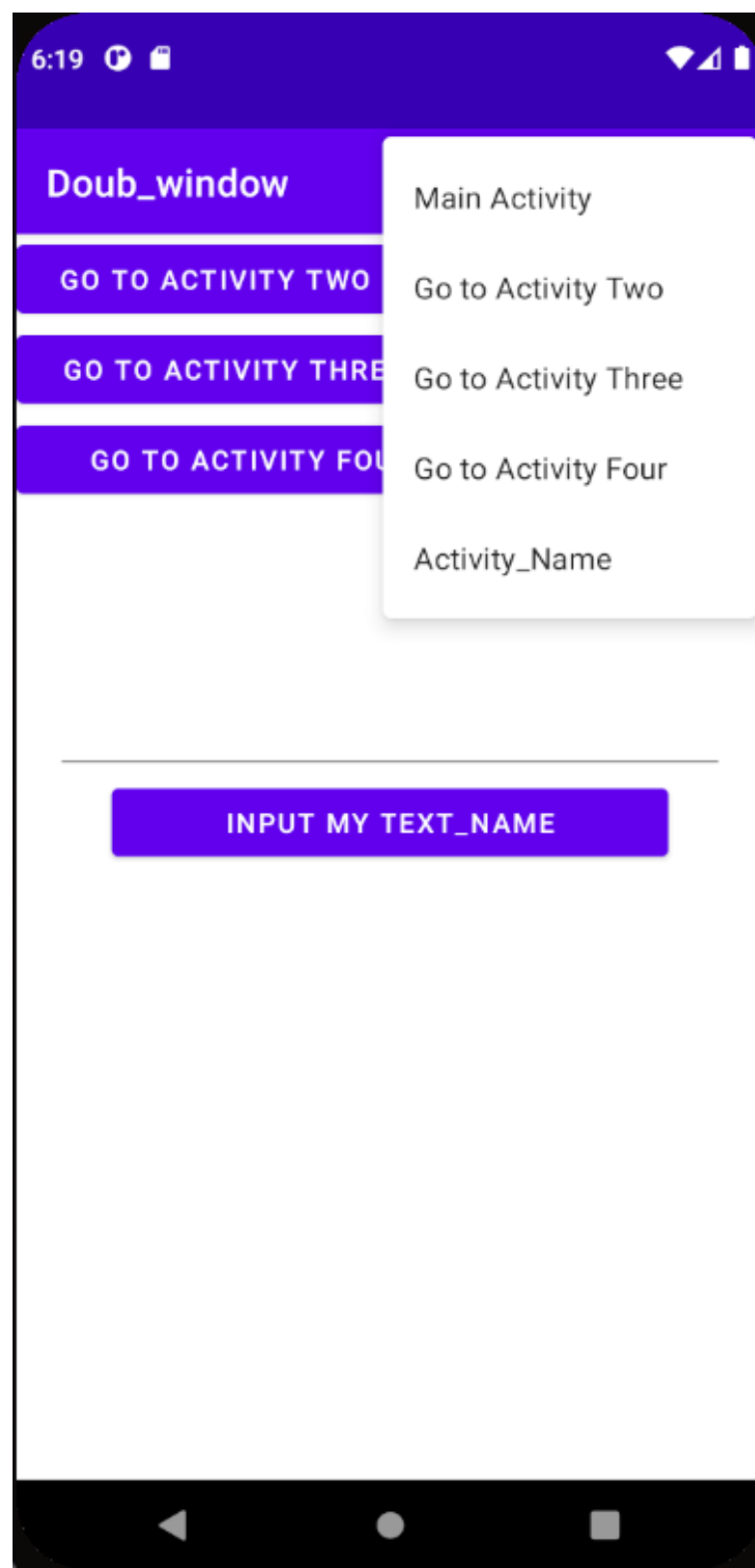
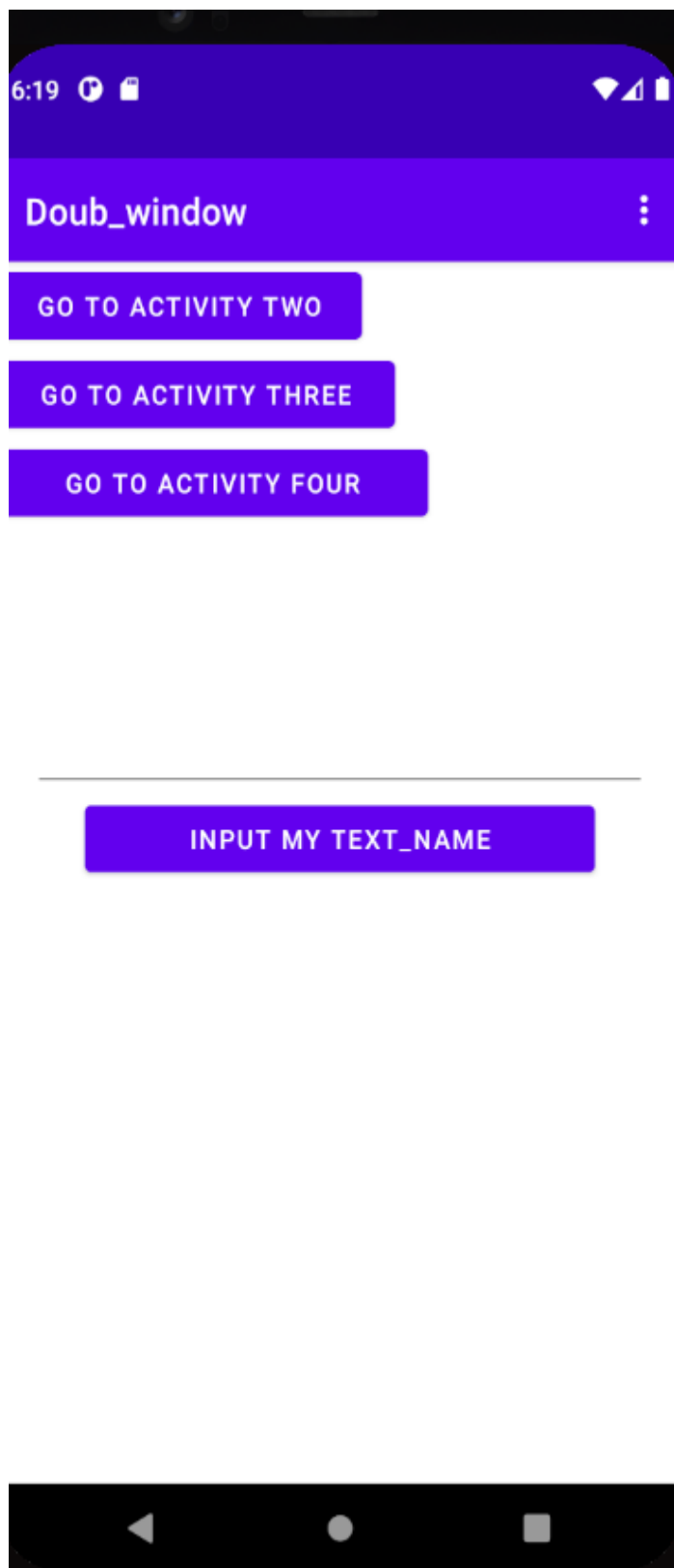
Перейдём к выполнению дополнительных заданий:

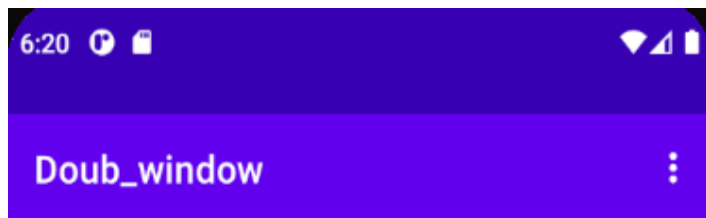
Дополнительные задания

1. Создайте приложение, состоящее из четырех активностей и реализуйте переходы между ними.
2. Реализуйте переходы между активностями используя меню приложения. Меню должно быть описано в XML файле и быть общим для всех четырех активностей.
3. (*) Реализуйте передачу данных между активностями. В первом окне создайте текстовое поле для ввода имени и кнопку. При нажатии на кнопку должно открываться второе окно, в котором отображается имя, введенное пользователем.

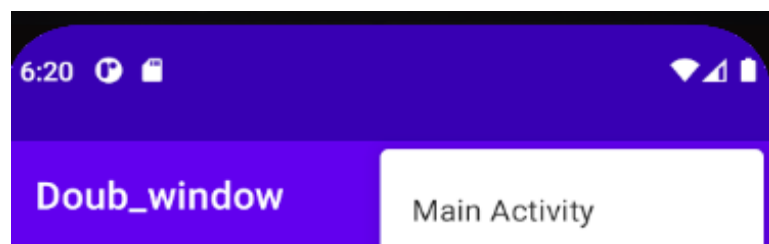
Результат работы приложения:

Все возможности, которые были реализованы в этом приложении – это переключение на 3 другие активности с главной при помощи кнопок и при помощи пунктов меню главной активности + это также возможность ввести информацию в текстовое поле и нажать на кнопку, в результате чего пользователь перекинется на 5 активность с уже введенными пользователем данными – это может быть и имя, и фамилия, и отчество, и также многое другое.



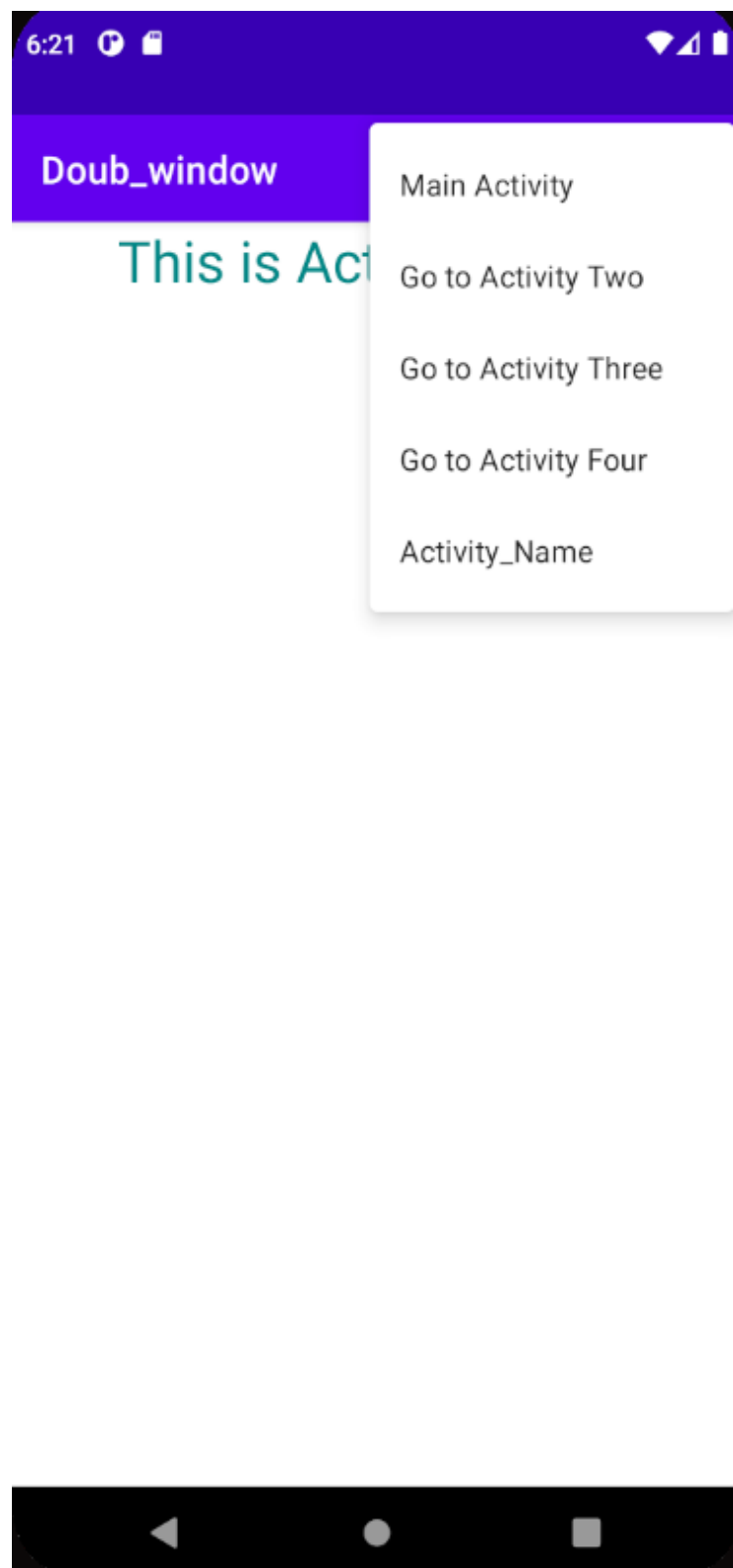


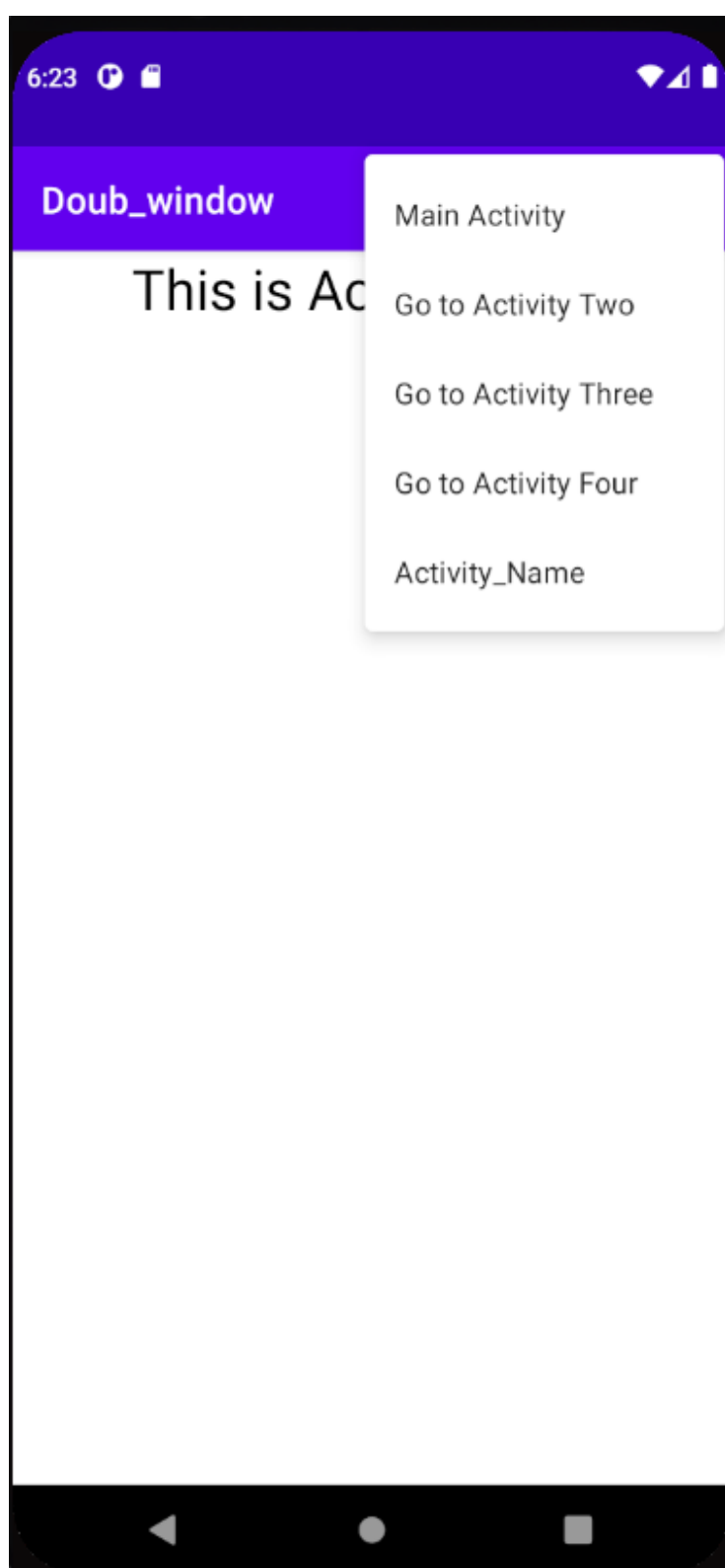
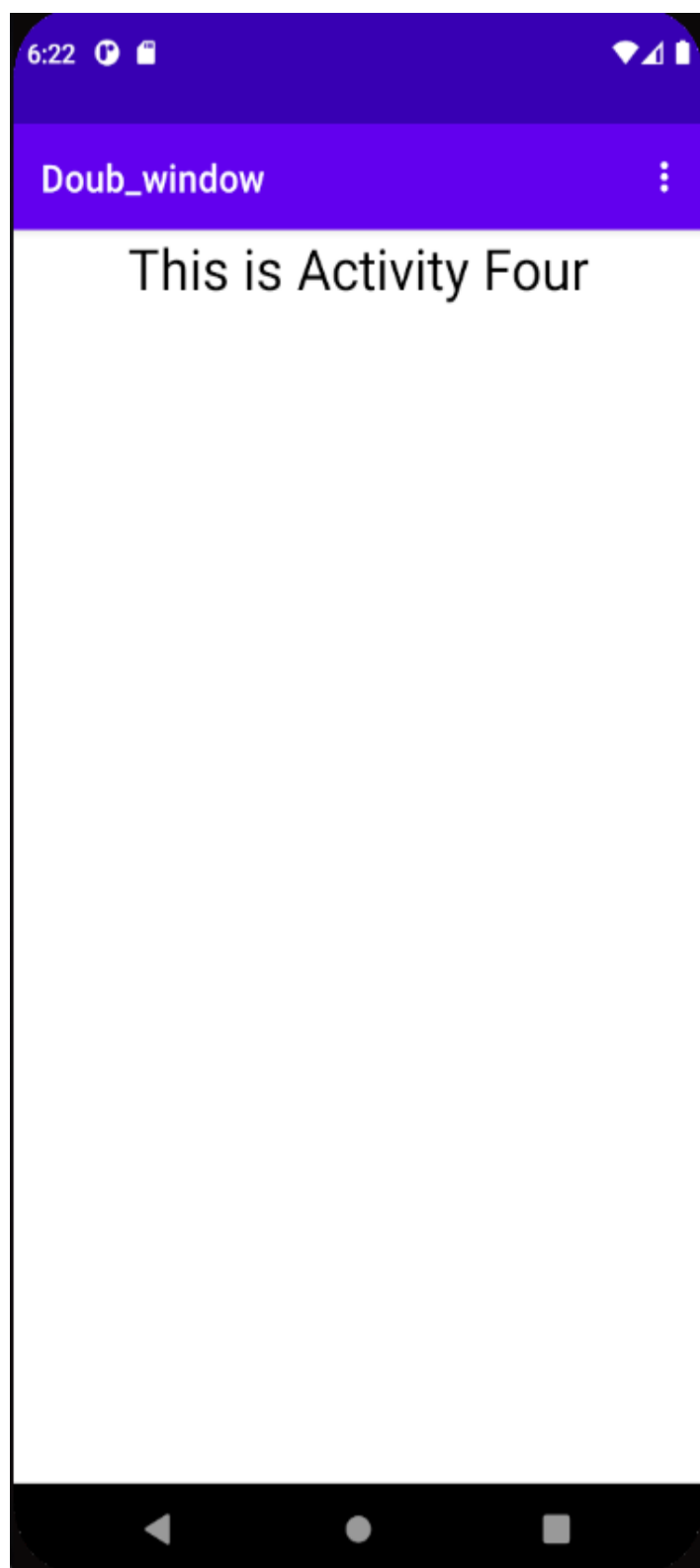
This is Activity Two

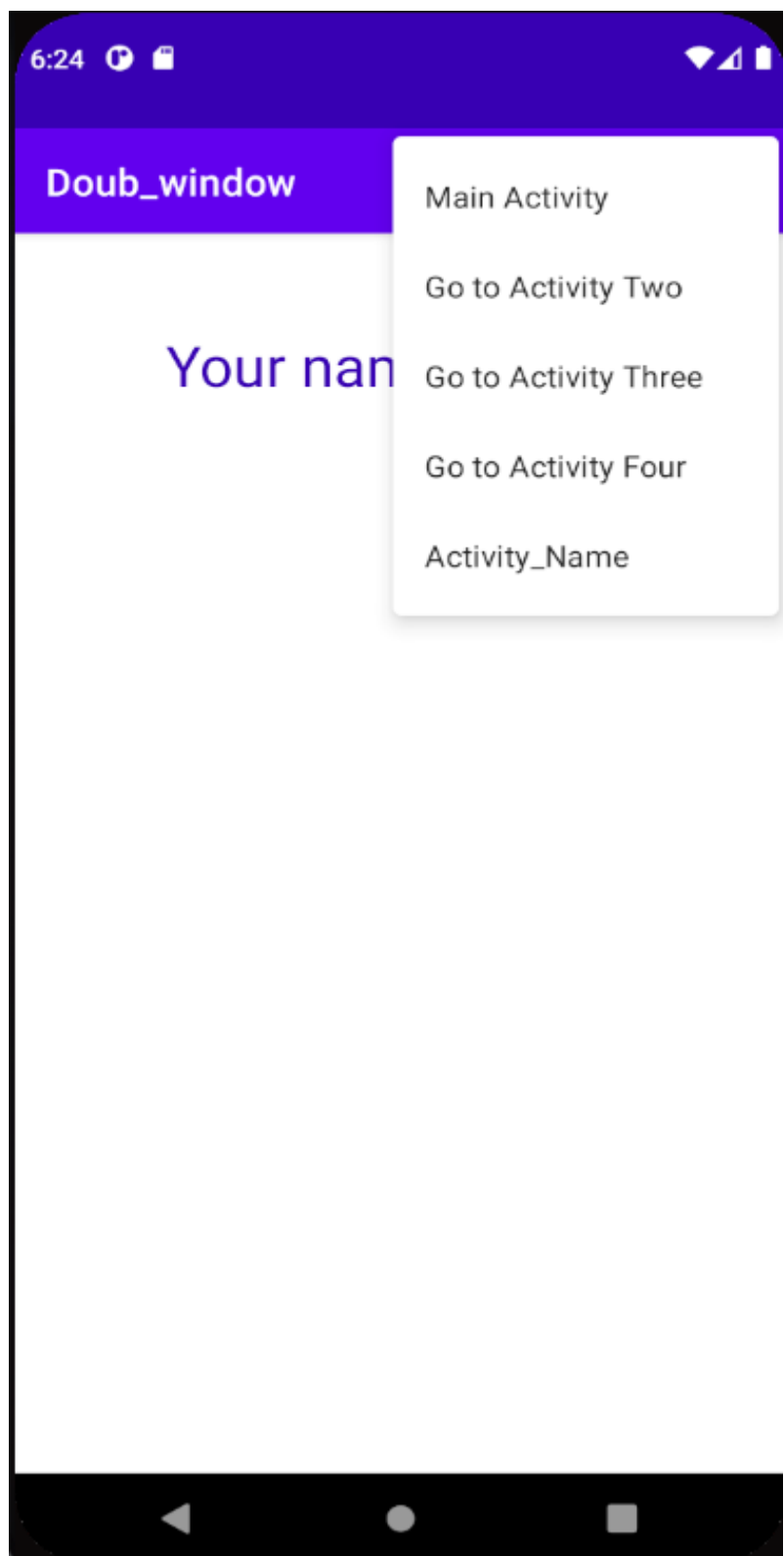
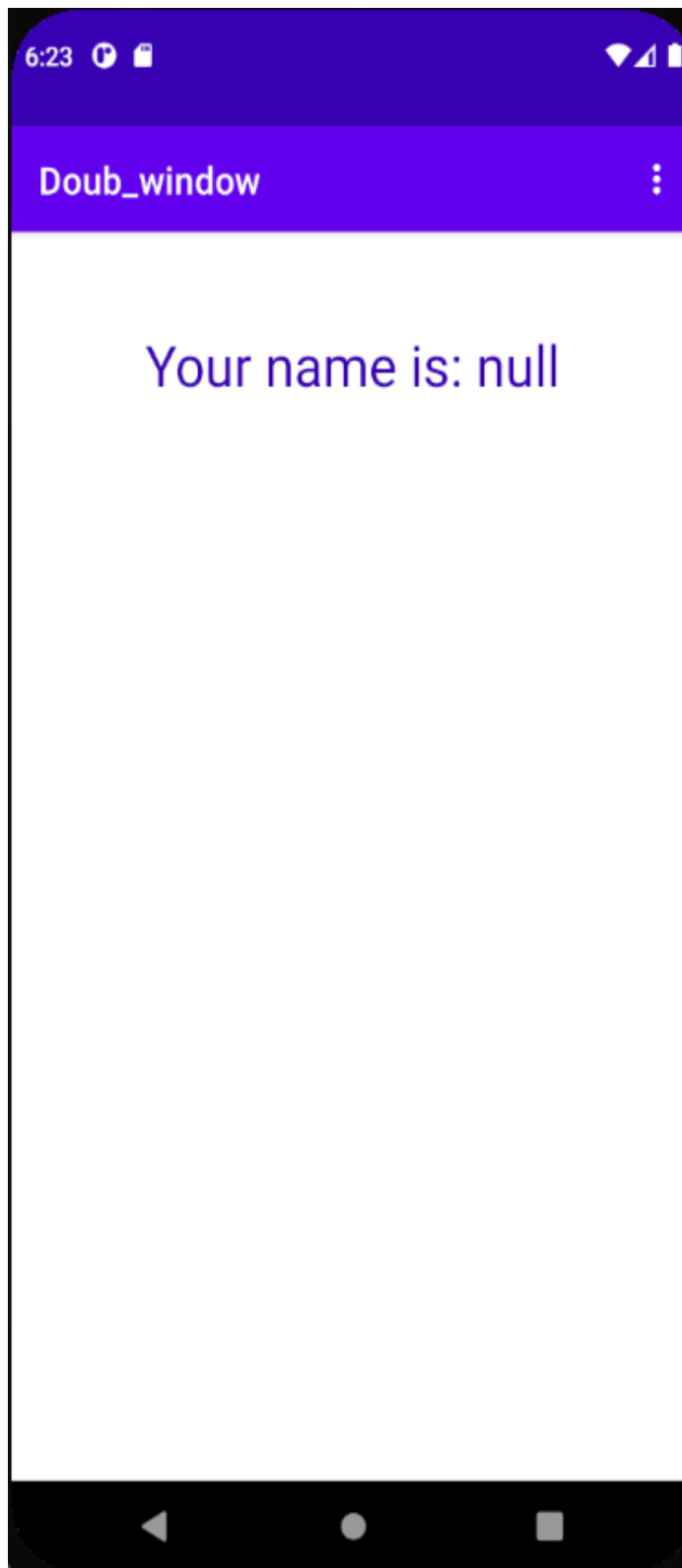


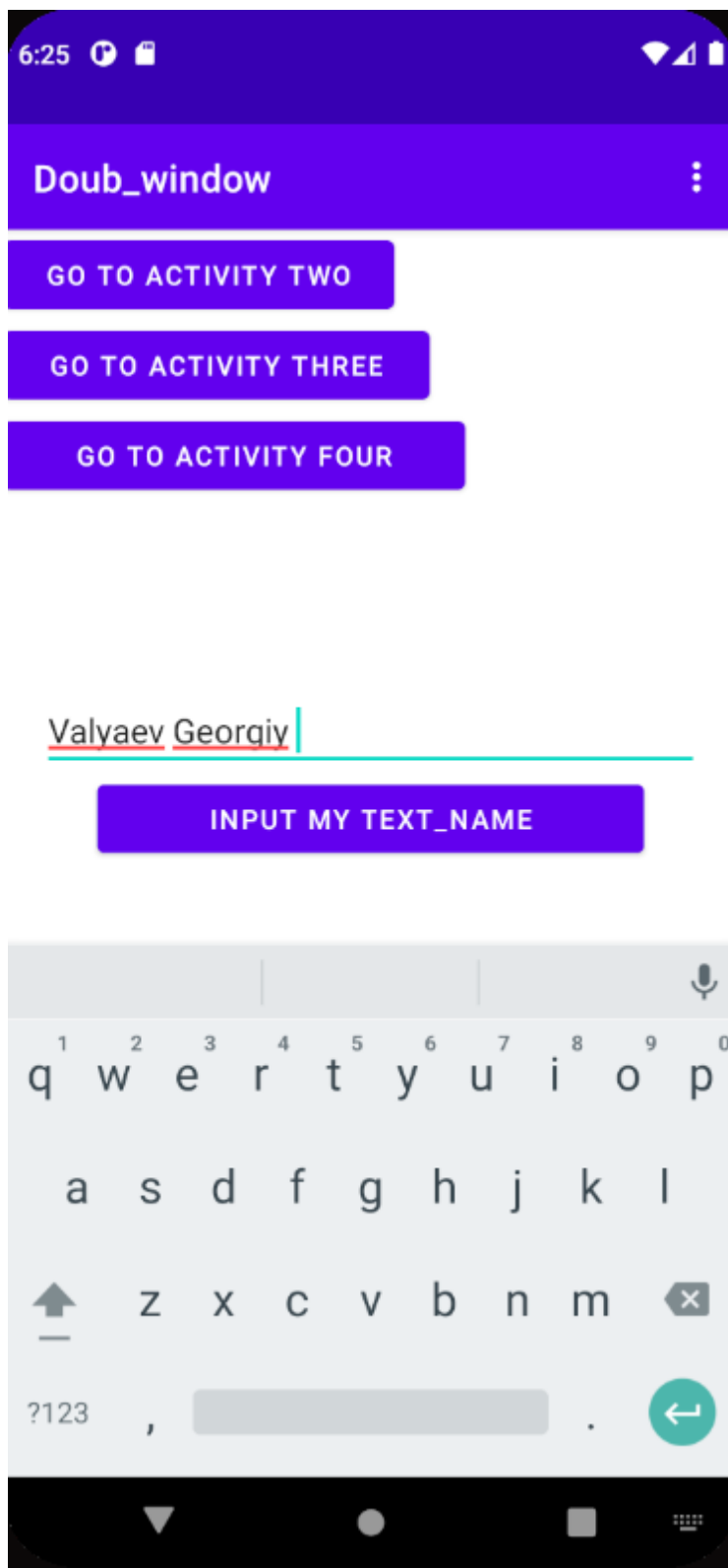
This is Activity Two

- Main Activity
- Go to Activity Two
- Go to Activity Three
- Go to Activity Four
- Activity_Name









Вследствие обилия активностей в этой работе (около 5), которые немного, но похожи друг на друга, предлагаю полностью ознакомиться со всем моим кодом на аккаунте GitHub. Прилагаю все необходимые вам ссылки:

Все файлы активности с Java-кодом:
https://github.com/Valyaevgeorgiy/Android_projects/tree/main/Doub_window/app/src/main/java/com/example/doub_window

Все файлы активности с XML-кодом:
https://github.com/Valyaevgeorgiy/Android_projects/tree/main/Doub_window/app/src/main/res/layout

Папка с XML-файлом меню приложения и его пунктами:
https://github.com/Valyaevgeorgiy/Android_projects/blob/main/Doub_window/app/src/main/res/menu/activity_menu_main.xml

Файл манифеста приложения (AndroidManifest.xml):
https://github.com/Valyaevgeorgiy/Android_projects/blob/main/Doub_window/app/src/main/AndroidManifest.xml