

Отчёт по проделанной работе на семинаре по мобильной разработке (10.03.2022).

(выполнил Валяев Георгий Анатольевич)

В наше время в работе большинства многофункциональных мобильных приложений не менее важную и большую (местами рутинную) роль занимают сервисы и приёмники широковещательных сообщений (первое – это небольшая программа как компонент приложения, которая выполняет конкретный пул долгих задач и операций в фоновом режиме без отображения на интерфейсах пользователю; а уже второе – некая архитектура, позволяющая обмениваться сообщениями и в какой-то мере данными, так как она по большей части выполняет функционал клиент-серверной архитектуры большинства приложений в мире).

В этой методичке в рамках знакомства со вторым компонентом (Broadcast Receiver) мы создадим довольно простое приложение, которое позволит нам отправлять данные и тут же их принимать и выводить их на экран через всплывающие уведомления Toast.

Приступим и начнём с activity_main.xml:

```
MainActivity.java x MyReceiver.java x activity_main.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="fill_parent"
5     android:layout_height="fill_parent"
6     android:orientation="vertical">
7
8     <TextView
9         android:id="@+id/tvInfo"
10        android:layout_width="wrap_content"
11        android:layout_height="wrap_content"
12        android:layout_marginTop="100sp"
13        android:text="@string/broadcast_app"
14        android:layout_gravity="center"
15        android:textColor="@color/purple_700"
16        android:textSize="25sp">
17    </TextView>
18
19    <Button
20        android:id="@+id/btnSendMessage"
21        android:layout_width="wrap_content"
22        android:layout_height="wrap_content"
23        android:layout_gravity="center"
24        android:layout_marginTop="150sp"
25        android:onClick="SendMessage"
26        android:text="@string/send_msg">
27    </Button>
28
29 </LinearLayout>
```

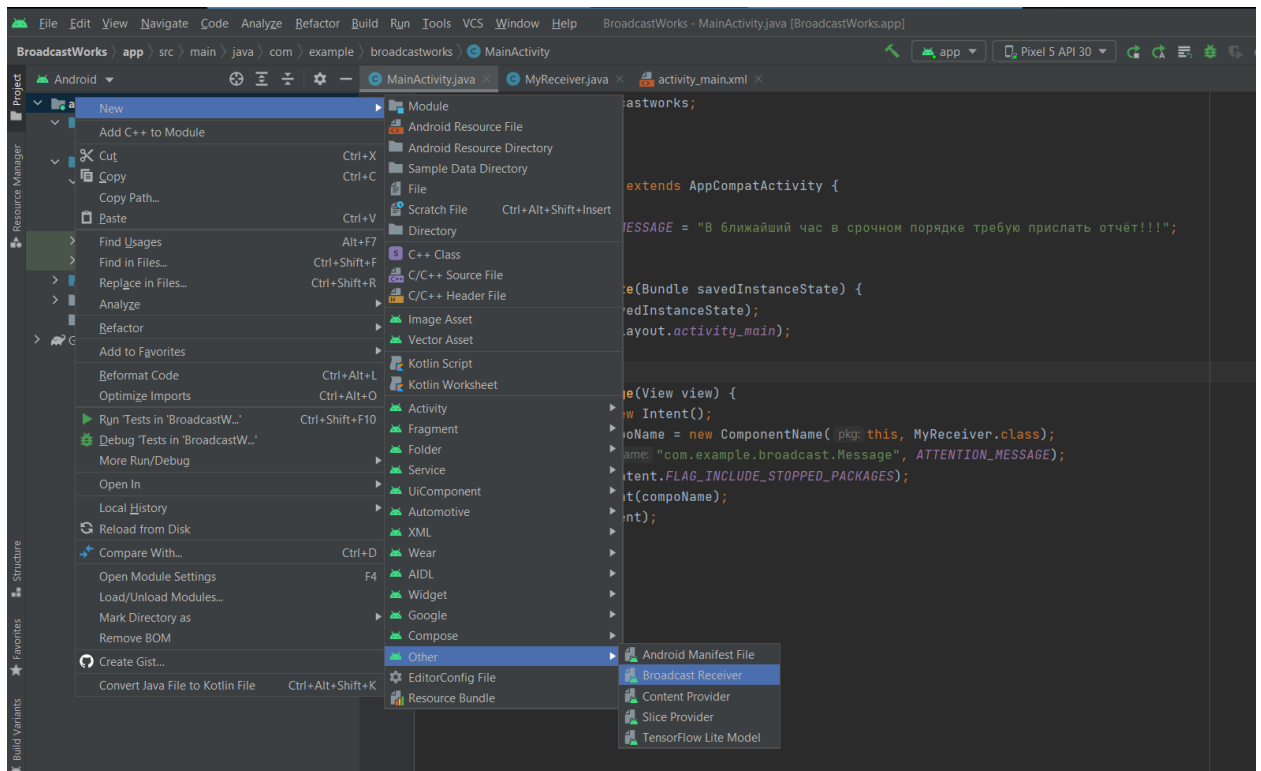
Довольно примитивный вид главной активности: текстовое окно с информацией о приложении с кнопкой, выводящей информацию, которую как раз-таки мы и будем отправлять через приёмник.

Перейдём в MainActivity.java:

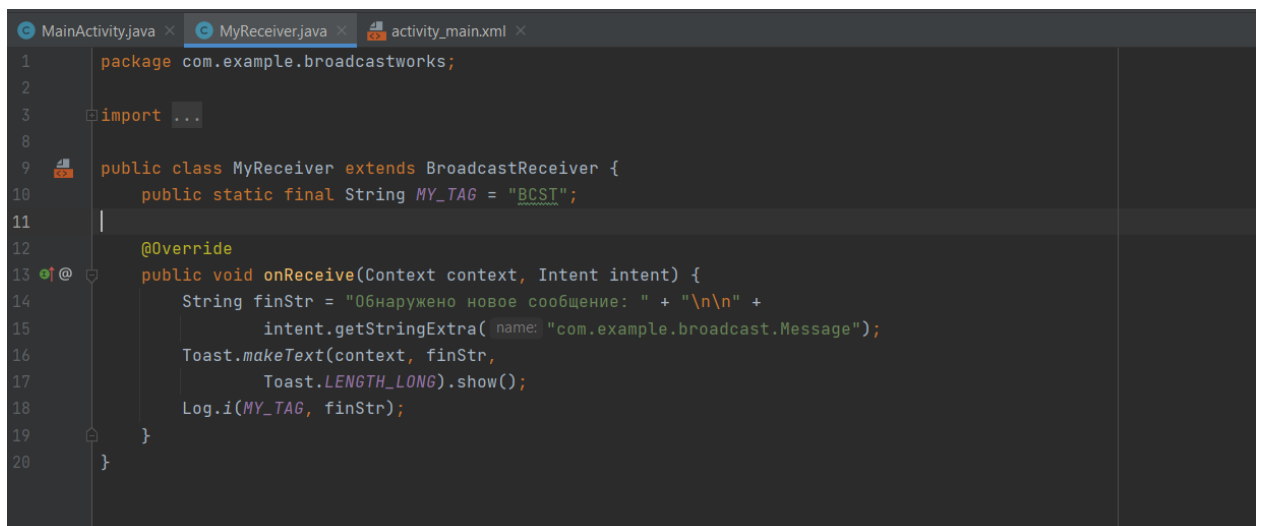
```
MainActivity.java x MyReceiver.java x activity_main.xml x
1 package com.example.broadcastworks;
2
3 import ...
4
5
6
7
8
9
10 public class MainActivity extends AppCompatActivity {
11     public static final
12         String ATTENTION_MESSAGE = "В ближайший час в срочном порядке требую прислать отчет!!!";
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18     }
19
20     public void SendMessage(View view) {
21         Intent intent = new Intent();
22         ComponentName compoName = new ComponentName( pkg: this, MyReceiver.class);
23         intent.putExtra( name: "com.example.broadcast.Message", ATTENTION_MESSAGE);
24         intent.addFlags(Intent.FLAG_INCLUDE_STOPPED_PACKAGES);
25         intent.setComponent(compoName);
26         sendBroadcast(intent);
27     }
28 }
```

Создаём постоянную неизменяемую строковую переменную, содержащую данные, которые мы будем передавать в дальнейшем. В методе `SendMessage` мы задаём простой интент, в который закладываем наши данные и присваиваем ему экземпляр объекта типа `ComponentName()`, связывающий нашу главную активность с приёмником. И в конце методом `sendBroadcast(intent)` отправляем приёмнику наш интент.

При создании приёмника можно брать готовый шаблон от Android Studio:

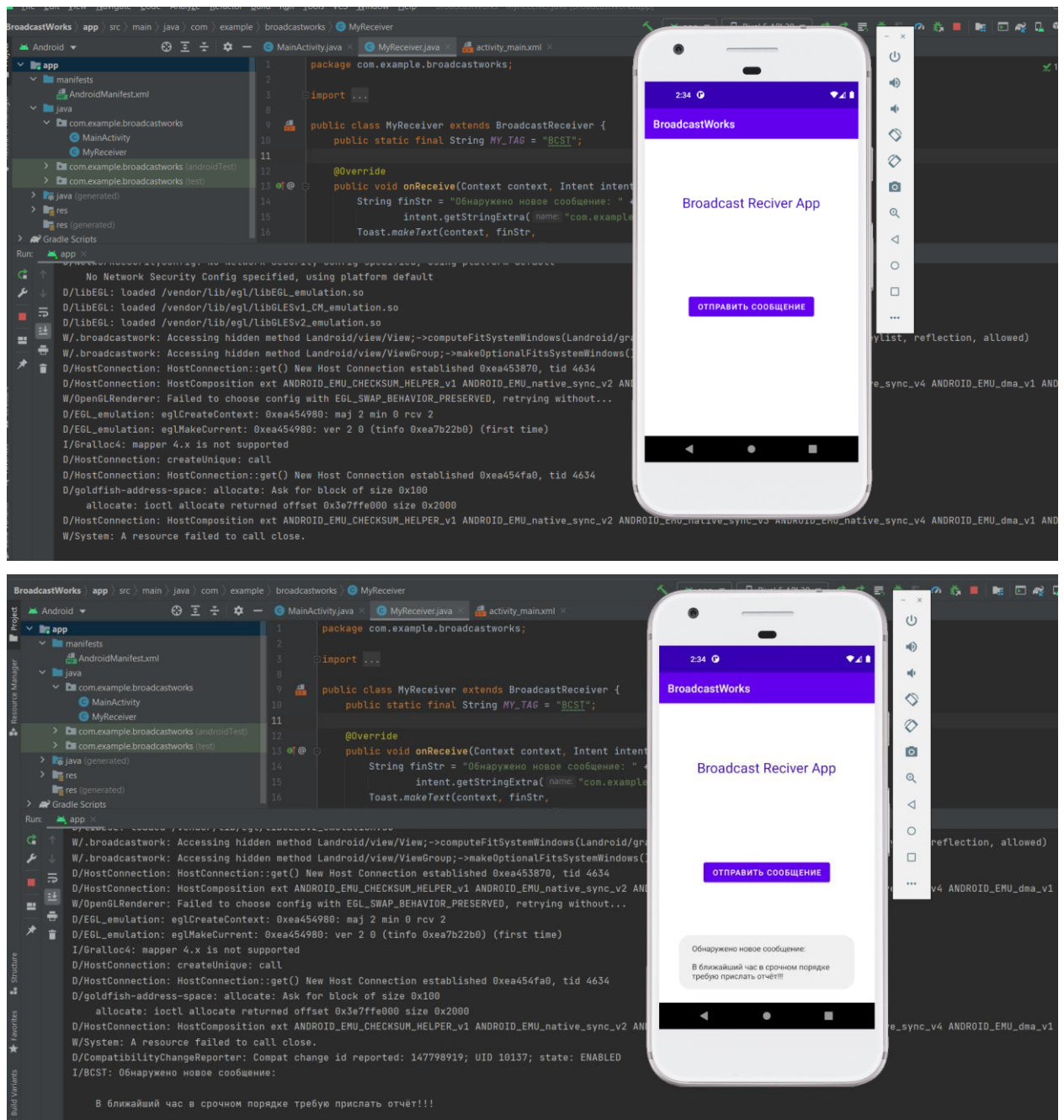


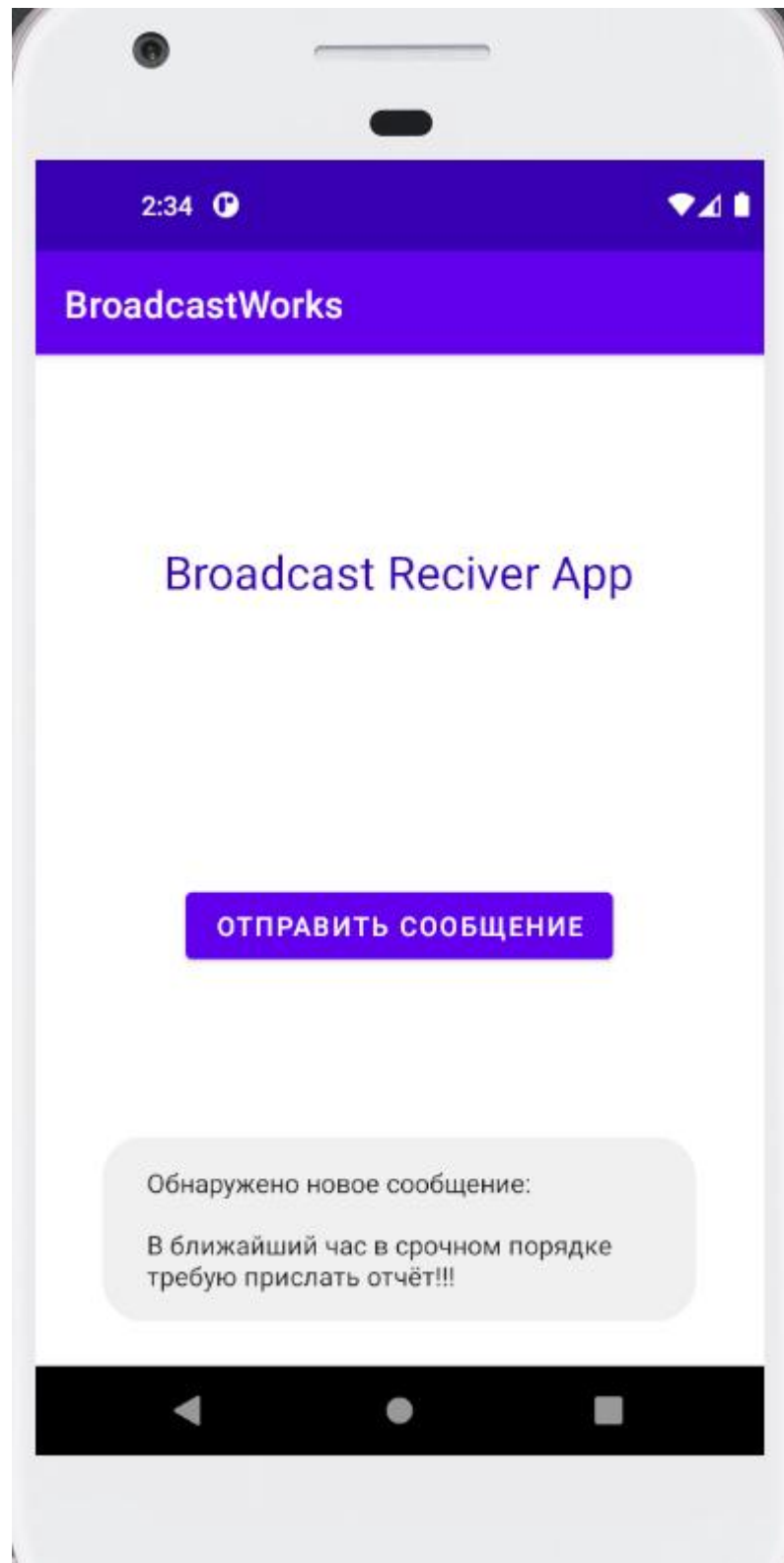
Рассмотрим последнюю активность `MyReceiver.java`:



Всё довольно просто: в главном методе `onReceive()` мы принимаем данные от интента из главной активности и формируем отправку сообщения в приложение через всплывающее уведомление (`Toast.makeText()`) и в консоль приложения (`Log.i()`).

Наблюдаем за результатами:





Работа успешно окончена!