

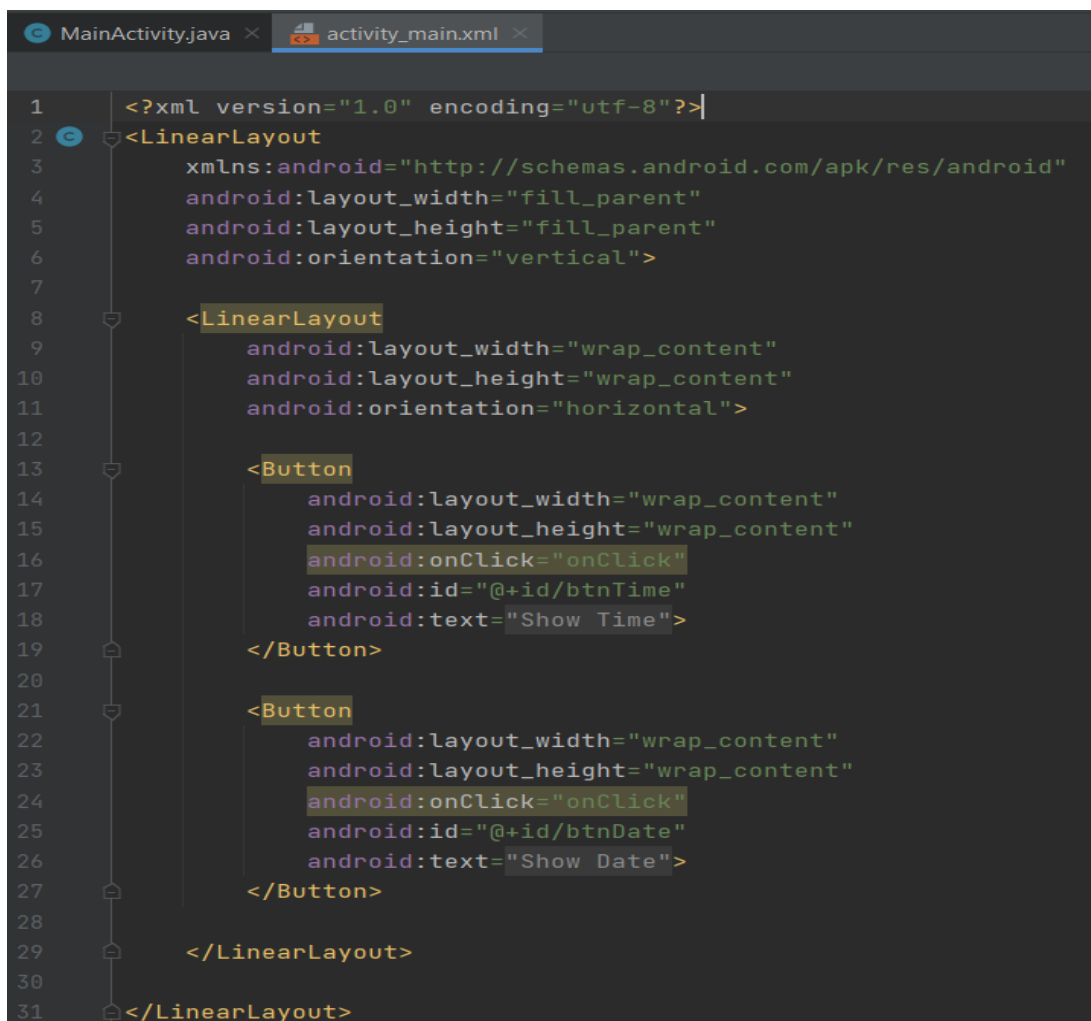
Отчёт по проделанной работе на семинаре по мобильной разработке (11.11.2021).

(выполнил Валяев Георгий Анатольевич)

В этой работе можно освоить важный навык в разработке мобильного приложения под Android – умение неявного вызова активности и создания своих собственных фильтров намерений, собственно, для запуска других, немало важных компонентов приложения.

Во-первых, попытаемся освоить этот навык на базовом небольшом проекте – мобильном приложении, которое при нажатии кнопок будет показывать неявно активности с отображением времени и даты.

Для этого первым шагом опишем кнопки и обработчик тапа в файлах первой активности (MainActivity.java и activity_main.xml):



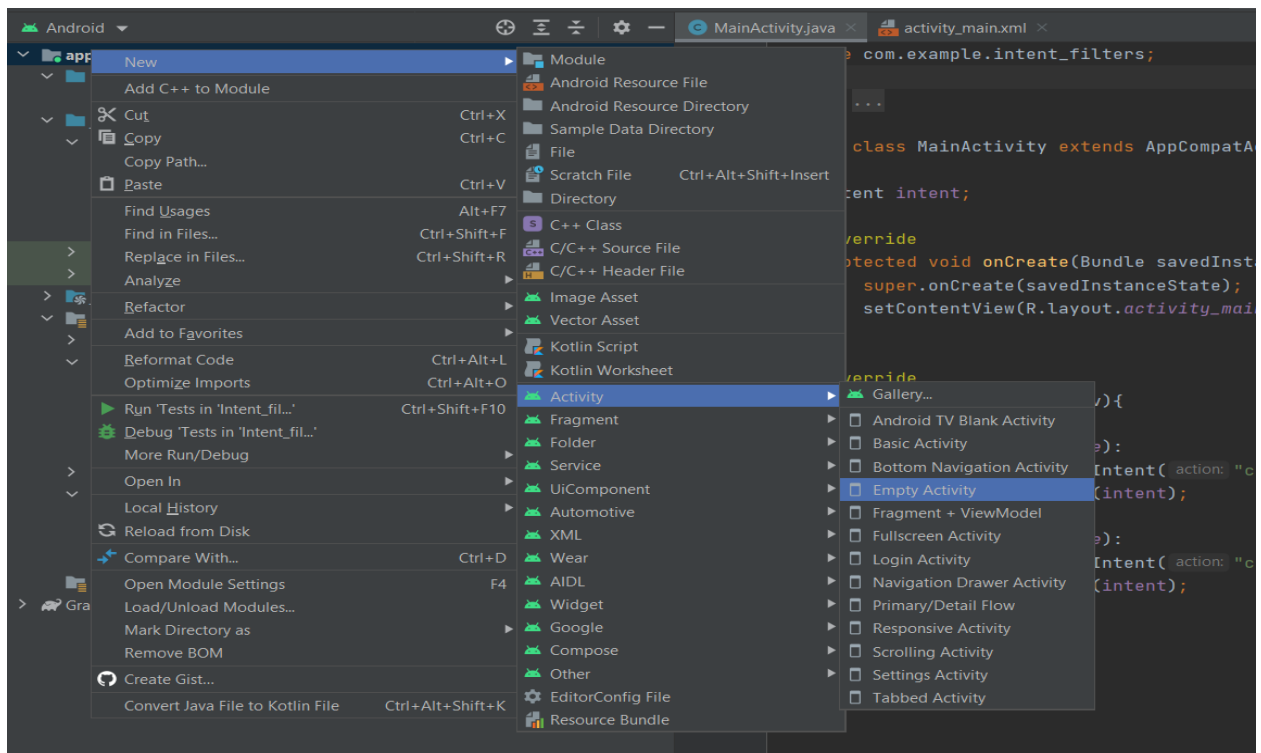
```
1  <?xml version="1.0" encoding="utf-8"?>|
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:layout_width="fill_parent"
5      android:layout_height="fill_parent"
6      android:orientation="vertical">
7
8      <LinearLayout
9          android:layout_width="wrap_content"
10         android:layout_height="wrap_content"
11         android:orientation="horizontal">
12
13         <Button
14             android:layout_width="wrap_content"
15             android:layout_height="wrap_content"
16             android:onClick="onClick"
17             android:id="@+id/btnTime"
18             android:text="Show Time">
19         </Button>
20
21         <Button
22             android:layout_width="wrap_content"
23             android:layout_height="wrap_content"
24             android:onClick="onClick"
25             android:id="@+id/btnDate"
26             android:text="Show Date">
27         </Button>
28
29     </LinearLayout>
30
31 </LinearLayout>
```

```
MainActivity.java × activity_main.xml ×
1 package com.example.intent_filters;
2
3 import ...
9
10 public class MainActivity extends AppCompatActivity implements View.OnClickListener
11
12     Intent intent;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18     }
19
20     @Override
21     public void onClick(View v){
22         switch (v.getId()){
23             case (R.id.btnTime):
24                 intent = new Intent( action: "com.example.intent.action.showtime");
25                 startActivity(intent);
26                 break;
27             case (R.id.btnDate):
28                 intent = new Intent( action: "com.example.intent.action.showdate");
29                 startActivity(intent);
30                 break;
31         }
32     }
33 }
```

В XML-файле описано вертикальное линейное расположение из горизонтального линейного расположения 2 кнопок, которые будут неявно вызывать активности приложения с показом даты и времени по UTC.

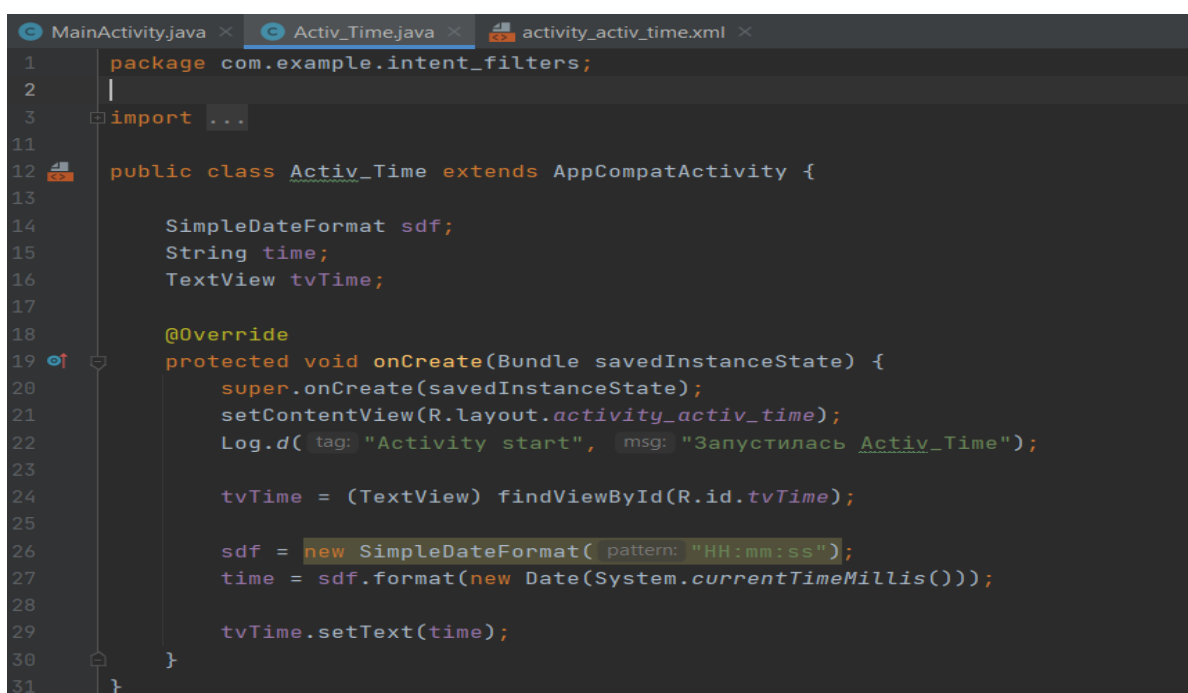
Перейдём к Java-файлу. Здесь инициализируется переменная типа Intent, после чего объявляется главный метод создания главного окна приложения + обработчик тапа с анализом нажатых кнопок выводением уже других активностей через startActivity(intent);.

Затем при помощи таких манипуляций...



...мы создаём необходимые нам активности: `Activ_Time.java`, `Activ_Date.java` и `Info.java` (она будет нам необходима для вызова активности в зависимости от нажатой кнопки на главной активности).

Рассмотрим подробно каждую созданную активность и прописанный код по их работе с приёмом вызова из главной активности и начнём с `Activ_Time.java`, `activity_activ_time.xml` и `Activ_Date.java`, `activity_activ_date.xml`, которые очень похожи друг с другом по структуре файла:



```
MainActivity.java x Activ_Time.java x activity_activ_time.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:layout_width="match_parent"
10             android:layout_height="wrap_content"
11             android:textSize="30sp"
12             android:textAlignment="center"
13             android:id="@+id/tvTime"
14             android:text="This is second Activity - Time Activity">
15
16     </TextView>
17 </LinearLayout>
```

```
MainActivity.java x Activ_Time.java x activity_activ_time.xml x Activ_Date.java x activity_activ_date.xml x
1      package com.example.intent_filters;
2
3      import ...
4
11
12      public class Activ_Date extends AppCompatActivity {
13
14          SimpleDateFormat sdf;
15          String date;
16          TextView tvDate;
17
18          @Override
19          protected void onCreate(Bundle savedInstanceState) {
20              super.onCreate(savedInstanceState);
21              setContentView(R.layout.activity_activ_date);
22              Log.d( tag: "Activity start", msg: "Запустилась Activ_Date");
23
24              tvDate = (TextView) findViewById(R.id.tvDate);
25
26              sdf = new SimpleDateFormat( pattern: "dd.MM.yyyy");
27              date = sdf.format(new Date(System.currentTimeMillis()));
28
29              tvDate.setText(date);
30          }
31      }
```

```
MainActivity.java x Activ_Time.java x activity_activ_time.xml x Activ_Date.java x activity_activ_date.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:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:id="@+id/tvDate"
12        android:textAlignment="center"
13        android:textSize="30sp"
14        android:text="This is third Activity - Date Activity">
15    </TextView>
16
17 </LinearLayout>
```

Итак, в XML-коде (даты и времени) обыкновенным способом описано линейное расположение текстового окна с выводимой информацией по центру с 30 размером шрифта и полным заполнением экрана по ширине.

А вот в Java-коде (даты и времени) сначала мы инициализируем 3 переменные типа TextView, String и SimpleDateFormat, которые уже в главном методе создания активности мы будем использовать напрямую для присваивания переменной id TextView в XML-файле, расчёта и форматирования времени (даты) для уже последующей отправки в переменную TextView и, наконец, отображения при нажатии кнопки в главной активности приложения.

Теперь рассмотрим 4 активность, которая будет реагировать на нажатие 1 из 2 кнопок на главной активности (Info):

```
MainActivity.java x Info.java x activity_info.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:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:textAlignment="center"
12        android:textSize="30sp"
13        android:id="@+id/tvInfo"
14        android:text="Show Info">
15    </TextView>
16
17 </LinearLayout>
```

```

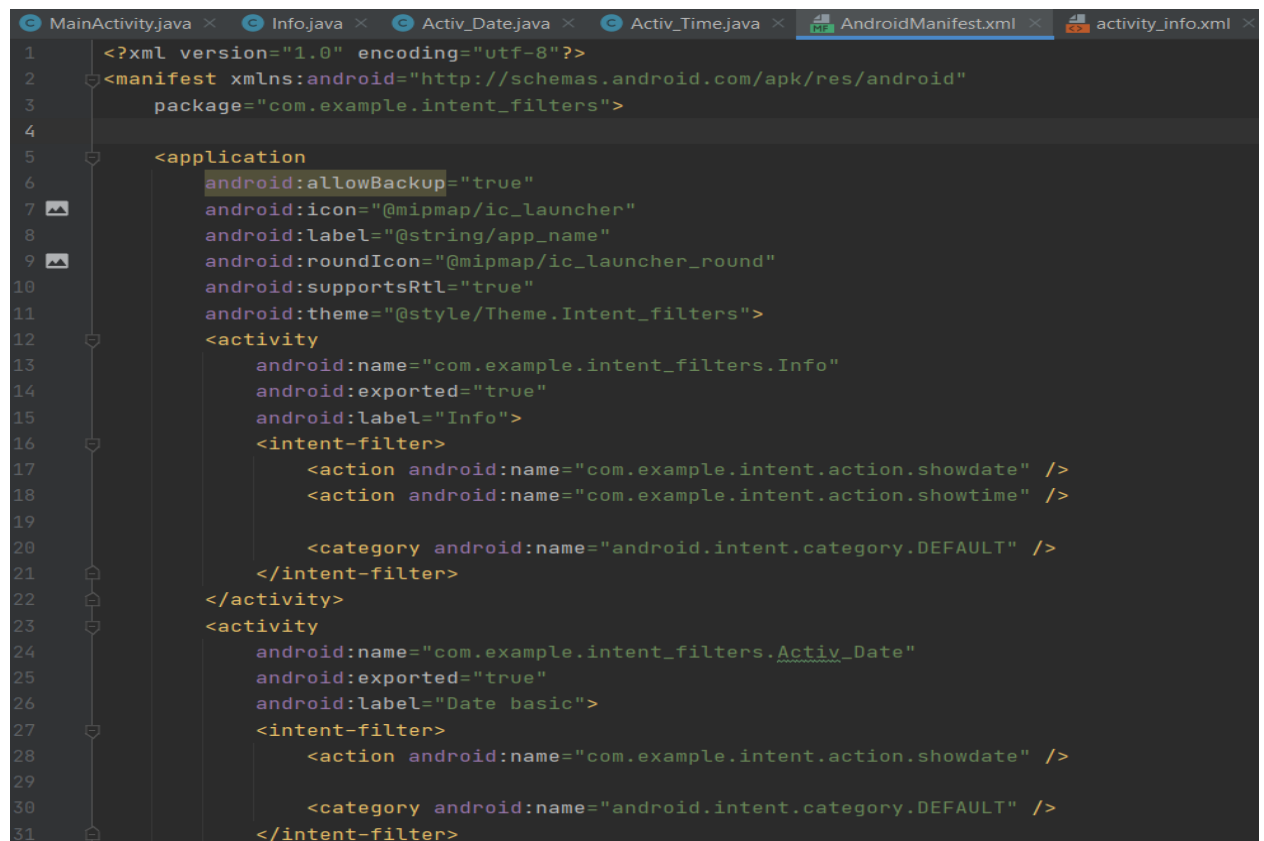
C MainActivity.java x Info.java x activity_info.xml x
1 package com.example.intent_filters;
2 |
3 import ...
11
12 public class Info extends AppCompatActivity {
13
14     SimpleDateFormat sdf;
15     TextView tvView, tvDate;
16
17     String format = "", textInfo = "", datetime = "", lname = "";
18
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_info);
23
24         //tvView = findViewById(R.id.tvView);
25
26         // получаем Intent, который вызывал это Activity
27         Intent intent = getIntent();
28
29         // читаем из него action
30         String action = intent.getAction();
31
32         // в зависимости от action заполняем переменные
33         if (action.equals("com.example.intent.action.showtime")) {
34             format = "HH:mm:ss";
35             textInfo = "Time: ";
36         }
37         else if (action.equals("com.example.intent.action.showdate")) {
38             format = "dd.MM.yyyy";
39             textInfo = "Date: ";
40         }
41
42         // в зависимости от содержимого переменной format
43         // получаем дату или время в переменную datetime
44         sdf = new SimpleDateFormat(format);
45         datetime = sdf.format(new Date(System.currentTimeMillis()));
46
47         tvDate = (TextView) findViewById(R.id.tvInfo);
48         tvDate.setText(textInfo + datetime);
49
50         //lname = intent.getStringExtra("lname");
51         //tvView.setText("Your name is: " + lname);
52     }
53 }

```

XML-файл весьма прост – линейное вертикальное расположение одного текстового окна с выводимой информацией о нажатой кнопке и функционале.

Java-код уже смотрится объёмнее (серым выделен тот участок кода, который сейчас нам не нужен, но точно понадобится в дальнейших примерах): инициализируем переменные текстового окна, SimpleDateFormat, и некоторое количество строковых переменных. Затем в главном методе создания получаем интент активности, которую вызвали ранее, и читаем из него action, после чего определяем, что нужно выводить и (в зависимости от action) заполняем созданные строковые переменные. В конце рассчитываем и форматируем выводимые на активность данные и передаём их уже переменной TextView в конце метода.

Перед тем, как получить результат, разумеется добавим в общий файл манифеста приложения (AndroidManifest.xml) к тегам всех активностей теги интент-фильтров уже с тегами action и category для появления всплывающих окон приложения с подсказками пользователю об уже выбранных активностях и новых для выбора:



```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3    package="com.example.intent_filters">
4
5    <application
6        android:allowBackup="true"
7        android:icon="@mipmap/ic_launcher"
8        android:label="@string/app_name"
9        android:roundIcon="@mipmap/ic_launcher_round"
10       android:supportsRtl="true"
11       android:theme="@style/Theme.Intent_filters">
12
13       <activity
14           android:name="com.example.intent_filters.Info"
15           android:exported="true"
16           android:label="Info">
17           <intent-filter>
18               <action android:name="com.example.intent.action.showdate" />
19               <action android:name="com.example.intent.action.showtime" />
20
21               <category android:name="android.intent.category.DEFAULT" />
22           </intent-filter>
23       </activity>
24       <activity
25           android:name="com.example.intent_filters.Activ_Date"
26           android:exported="true"
27           android:label="Date basic">
28           <intent-filter>
29               <action android:name="com.example.intent.action.showdate" />
30
31               <category android:name="android.intent.category.DEFAULT" />
32           </intent-filter>
33       </activity>
34   </application>
35 </manifest>
```

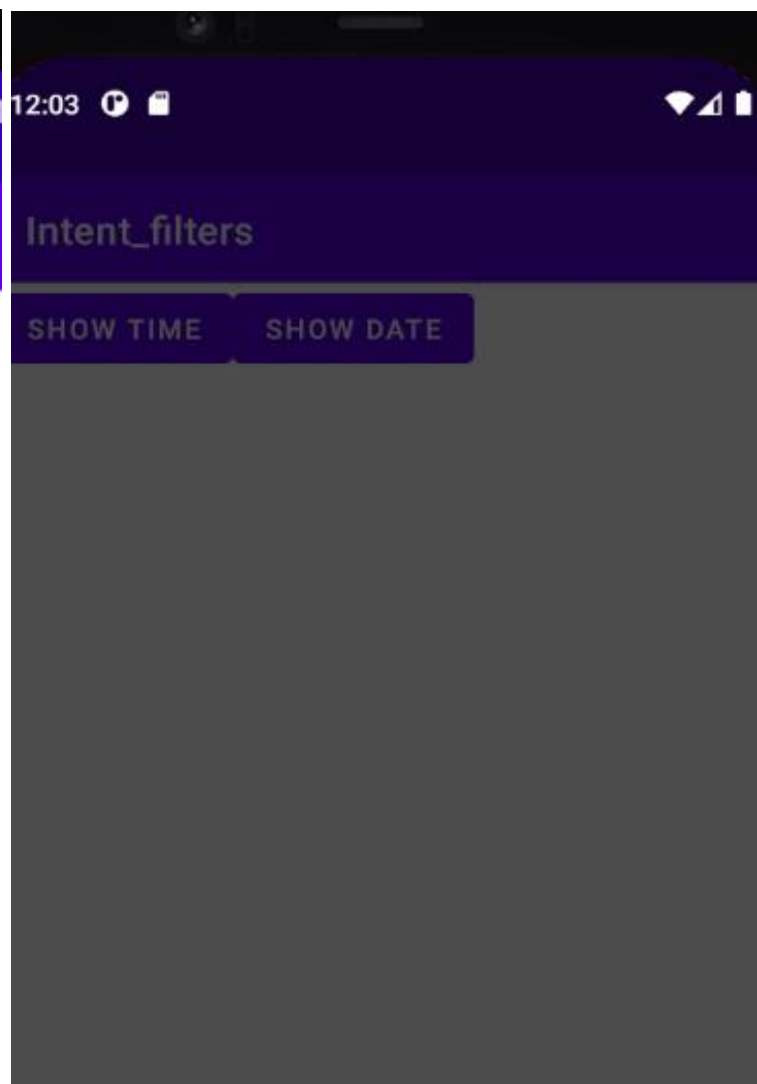
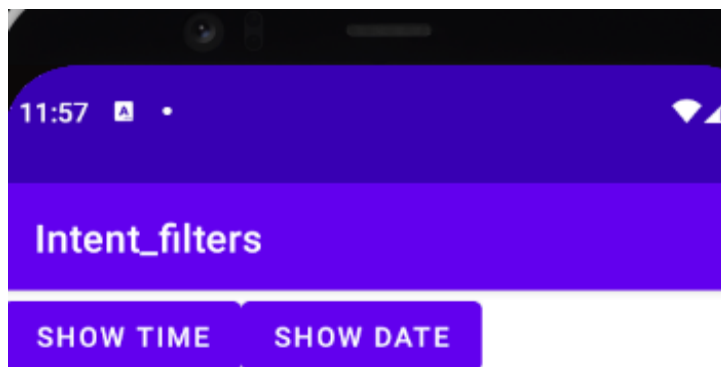
```
<intent-filter>
    <action android:name="com.example.intent.action.showdate" />

    <category android:name="android.intent.category.DEFAULT" />
</intent-filter>
</activity>
<activity
    android:name="com.example.intent_filters.Activ_Time"
    android:exported="true"
    android:label="Time basic">
    <intent-filter>
        <action android:name="com.example.intent.action.showtime" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity
    android:name="com.example.intent_filters.MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>

</manifest>
```

Наблюдаем за полученным результатом (на след. странице):



Complete action using



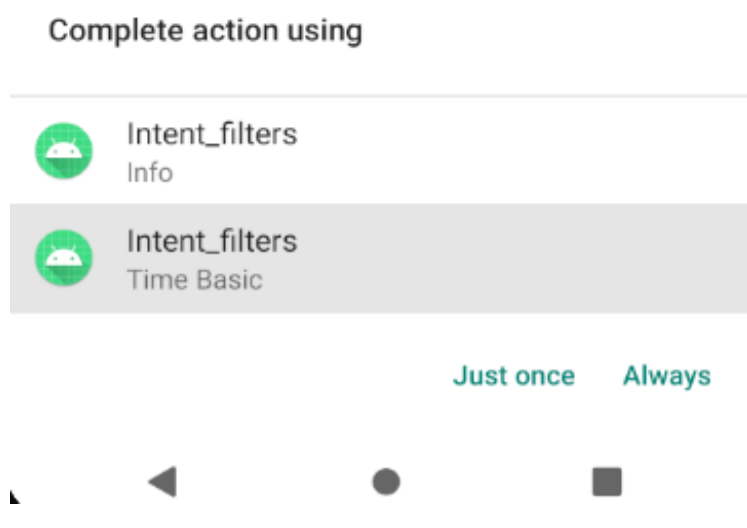
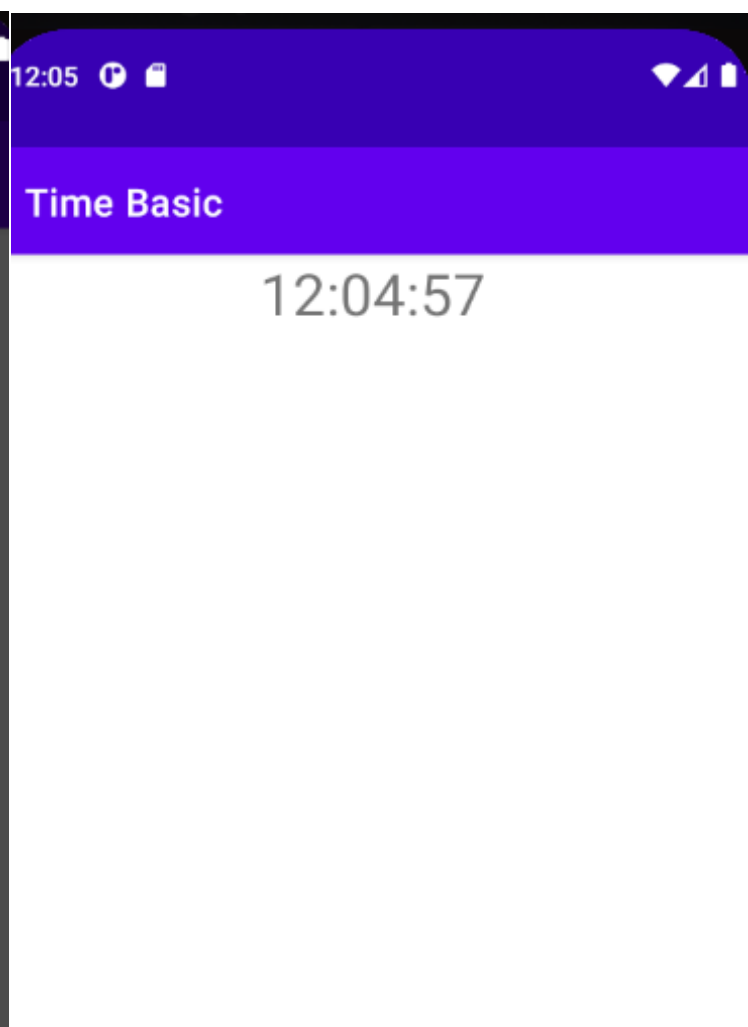
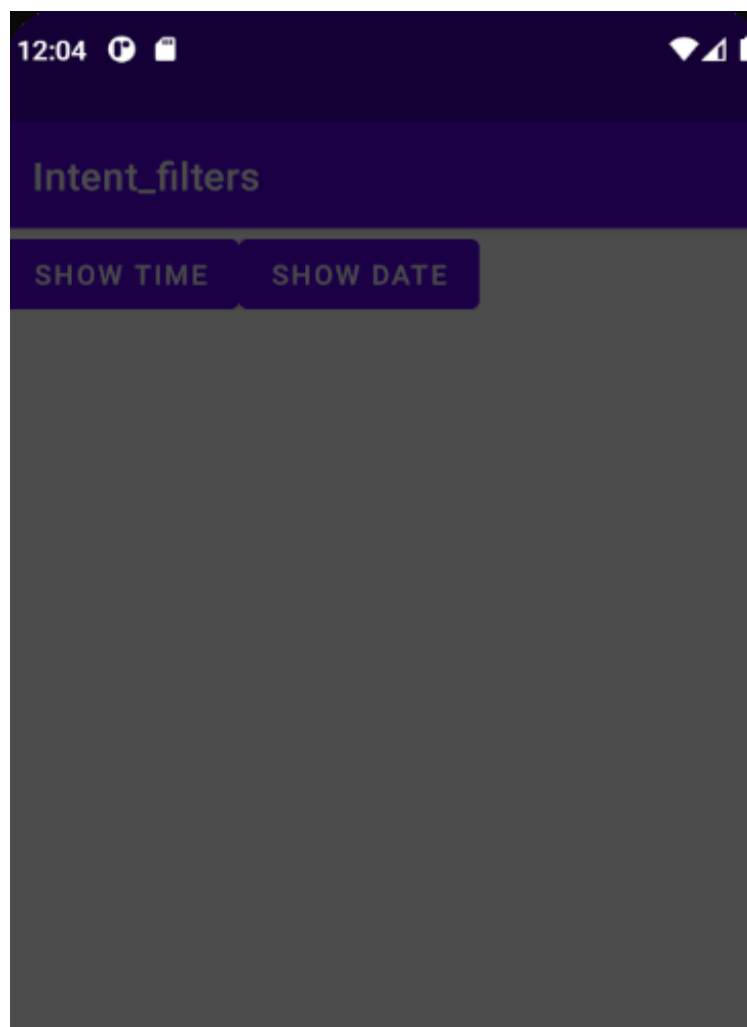
Intent_filters
Info



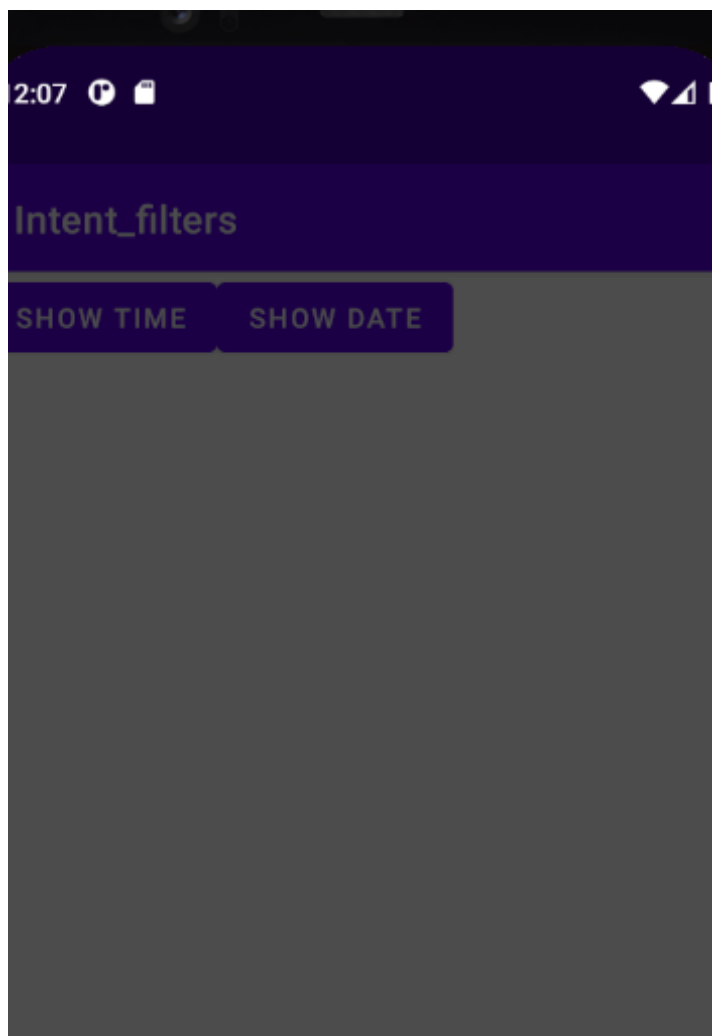
Intent_filters
Time Basic

Just once Always





Здесь уже при повторном нажатии кнопок будет сама система запомнит ваш выбор и предложит использовать ещё раз ранее использованную активность или уже на постоянной основе:



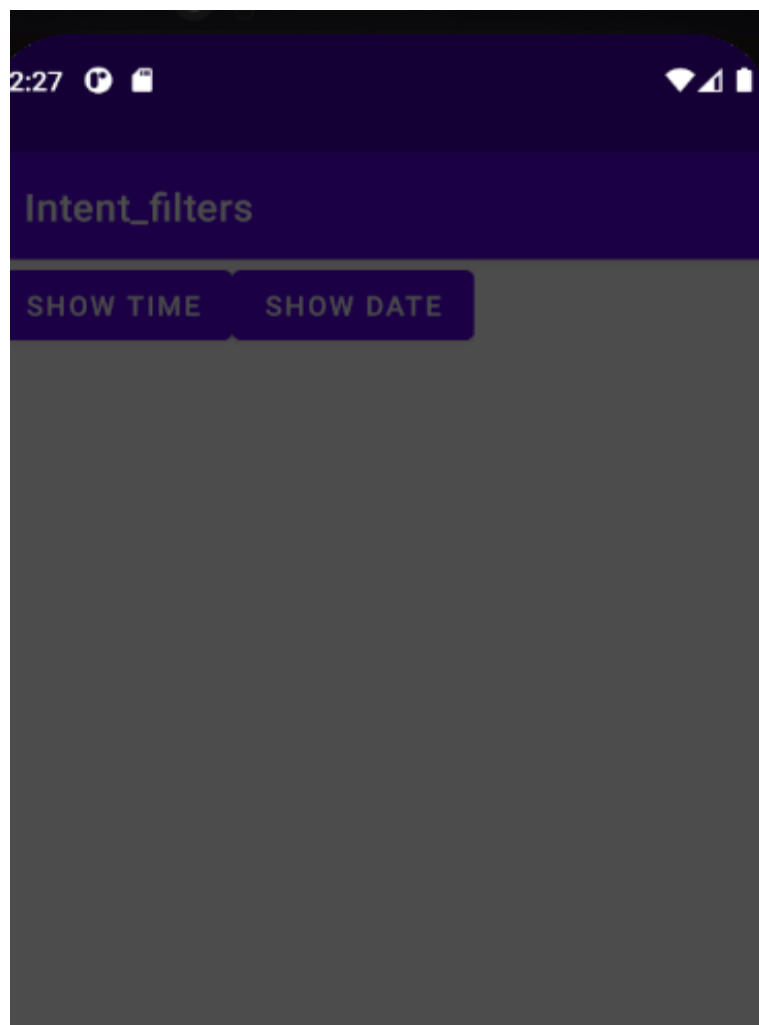
Complete action using Intent_filters

Just once Always

Use a different app



Intent_filters
Time Basic



Info

Date: 17.11.2021



Complete action using Intent_filters

Just once Always

Use a different app



Intent_filters
Date basic





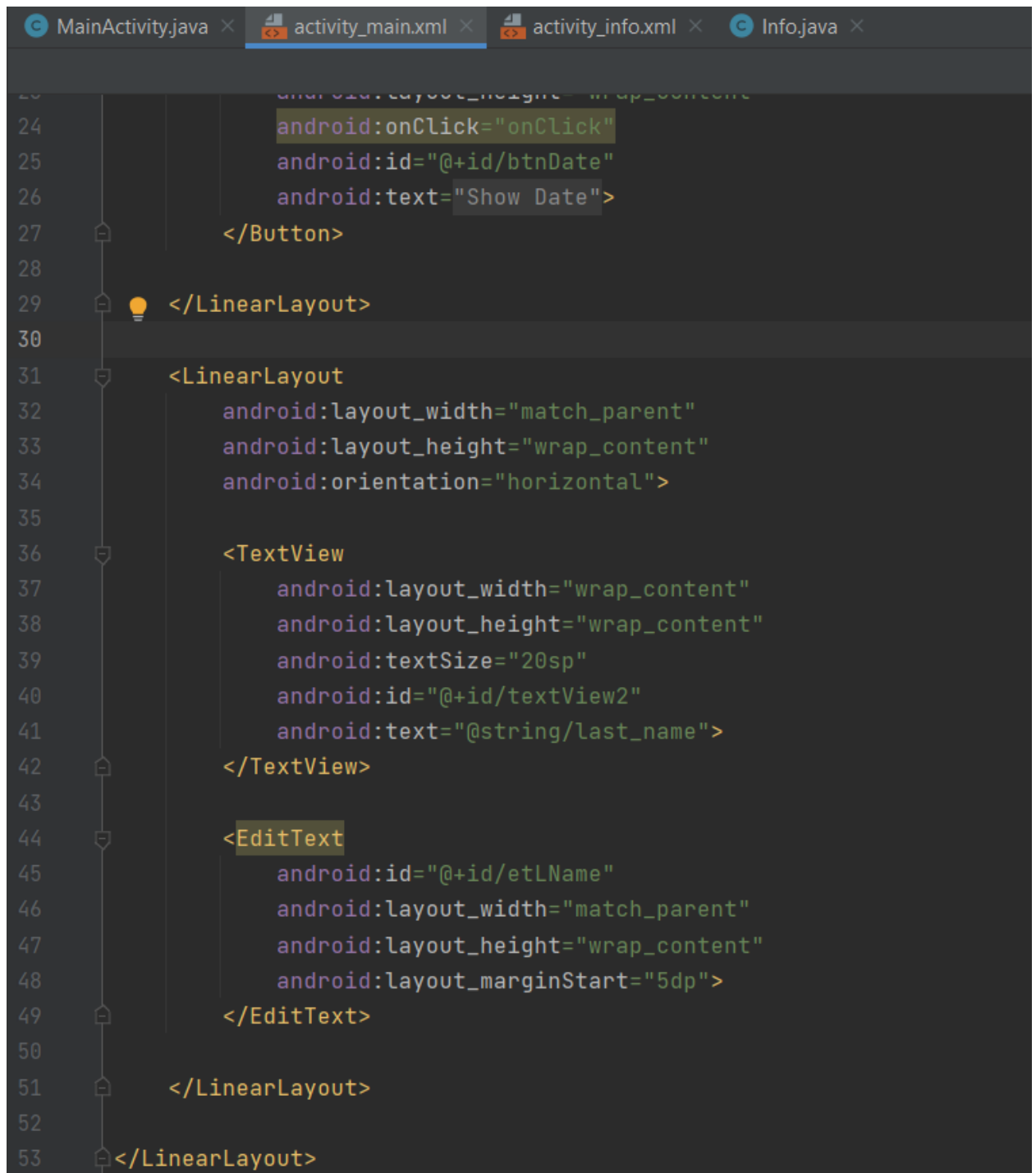
17.11.2021



В целом приложение работает, всё отлично!

Далее перейдём к возможности передавать данные в запущенный по интенту компонент. Для этого создадим поле для текстового ввода в схеме главного

окна (при этом добавив после прошлого расположения с кнопками новое горизонтальное линейное расположение поле с текстовым пояснением, что нужно ввести, и само поле для ввода):



```
23         android:layout_height="wrap_content"
24         android:onClick="onClick"
25         android:id="@+id/btnDate"
26         android:text="Show Date">
27     </Button>
28
29 </LinearLayout>
30
31 <LinearLayout
32     android:layout_width="match_parent"
33     android:layout_height="wrap_content"
34     android:orientation="horizontal">
35
36     <TextView
37         android:layout_width="wrap_content"
38         android:layout_height="wrap_content"
39         android:textSize="20sp"
40         android:id="@+id/textView2"
41         android:text="@string/last_name">
42     </TextView>
43
44     <EditText
45         android:id="@+id/etLName"
46         android:layout_width="match_parent"
47         android:layout_height="wrap_content"
48         android:layout_marginStart="5dp">
49     </EditText>
50
51 </LinearLayout>
52
53 </LinearLayout>
```

Далее в MainActivity.java инициализируем переменную, которая будет принимать и считывать данные, введённые пользователем в поле для ввода, и при помощи «intent.putExtra("lname", etLName.getText().toString());» в течение

перехода на другую активность будем переносить также и введенные данные с поля ввода:

```
MainActivity.java × activity_main.xml × activity_info.xml × Info.java ×
1      package com.example.intent_filters;
2
3      import ...
11
12     public class MainActivity extends AppCompatActivity implements View.OnClickListener{
13
14         Intent intent;
15         EditText etLName;
16
17         @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21
22         etLName = findViewById(R.id.etLName);
23     }
24
25     @Override
26     public void onClick(View v){
27         switch (v.getId()){
28             case (R.id.btnTime):
29                 intent = new Intent( action: "com.example.intent.action.showtime");
30                 intent.putExtra( name: "lname", etLName.getText().toString());
31                 startActivity(intent);
32                 break;
33             case (R.id.btnDate):
34                 intent = new Intent( action: "com.example.intent.action.showdate");
35                 intent.putExtra( name: "lname", etLName.getText().toString());
36                 startActivity(intent);
37                 break;
38         }
39     }
40 }
```

Далее отредактируем файлы активности, которая вызывается при нажатии любой из 2 кнопок на главной активности (Info.java и activity_info.xml):

```
MainActivity.java × activity_main.xml × Info.java × activity_info.xml ×
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:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:textAlignment="center"
12        android:textSize="30sp"
13        android:id="@+id/tvInfo"
14        android:text="Show Info">
15    </TextView>
16
17    <TextView
18        android:id="@+id/tvView"
19        android:layout_width="wrap_content"
20        android:layout_height="wrap_content"
21        android:layout_gravity="center_horizontal"
22        android:layout_marginTop="20dp"
23        android:text="@string/tv"
24        android:textSize="20sp">
25    </TextView>
26
27 </LinearLayout>
```

Здесь мы просто добавляем текстовое окно с информацией, которую получаем при переключении от главной активности приложения.


```

MainActivity.java × activity_main.xml × Info.java × activity_info.xml ×
1  package com.example.intent_filters;
2
3  import ...
12
13  public class Info extends AppCompatActivity {
14
15      SimpleDateFormat sdf;
16      TextView tvView, tvDate;
17
18      String format = "", textInfo = "", datetime = "", lname = "";
19
20      @Override
21  protected void onCreate(Bundle savedInstanceState) {
22      super.onCreate(savedInstanceState);
23      setContentView(R.layout.activity_info);
24
25      tvView = findViewById(R.id.tvView);
26
27      // получаем Intent, который вызывал это Activity
28      Intent intent = getIntent();
29
30      // читаем из него action
31      String action = intent.getAction();
32
33      Log.d( tag: "Action", msg: "Action is: " + action);
34
35      // в зависимости от action заполняем переменные
36
37      if (action.equals("com.example.intent.action.showtime")) {
38          format = "HH:mm:ss";
39          textInfo = "Time: ";
40      }

```

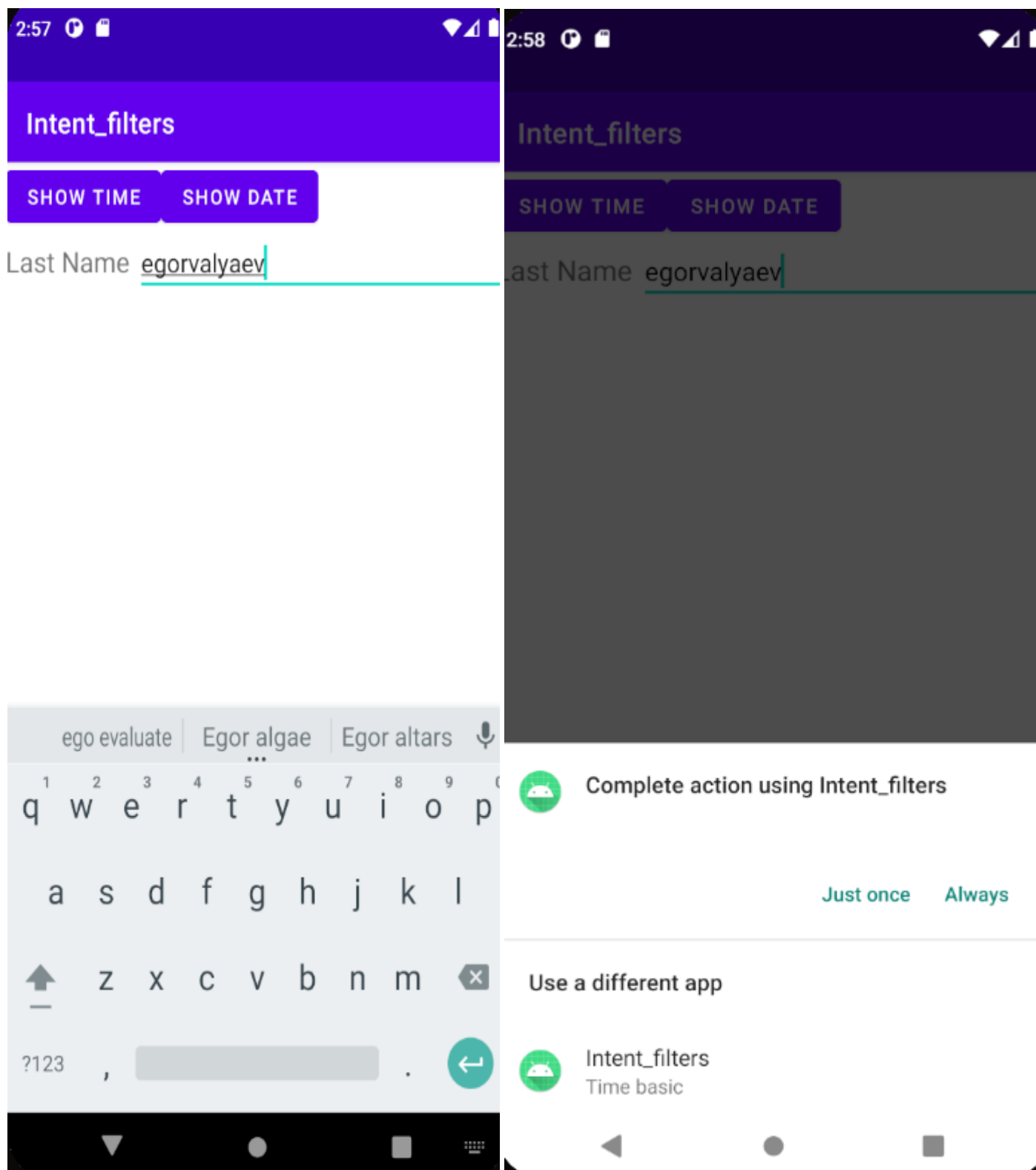
```

40      }
41      else if (action.equals("com.example.intent.action.showdate")) {
42          format = "dd.MM.yyyy";
43          textInfo = "Date: ";
44      }
45
46      // в зависимости от содержимого переменной format
47      // получаем дату или время в переменную datetime
48      sdf = new SimpleDateFormat(format);
49      datetime = sdf.format(new Date(System.currentTimeMillis()));
50
51      tvDate = (TextView) findViewById(R.id.tvInfo);
52      tvDate.setText(textInfo + datetime);
53
54      lname = intent.getStringExtra( name: "lname");
55      tvView.setText("Your name is: " + lname);
56  }
57  }

```

В этом файле мы добавляем переменную lname, которой уже передали текстовую информацию, введённую пользователем на главной активности, и в конце кода просто получаем эти данные и отправляем их уже переменной, которая отвечает за текстовое окно активности Info.

Получаем следующий результат:





Time: 14:58:14

Your name is: egorvalyaev

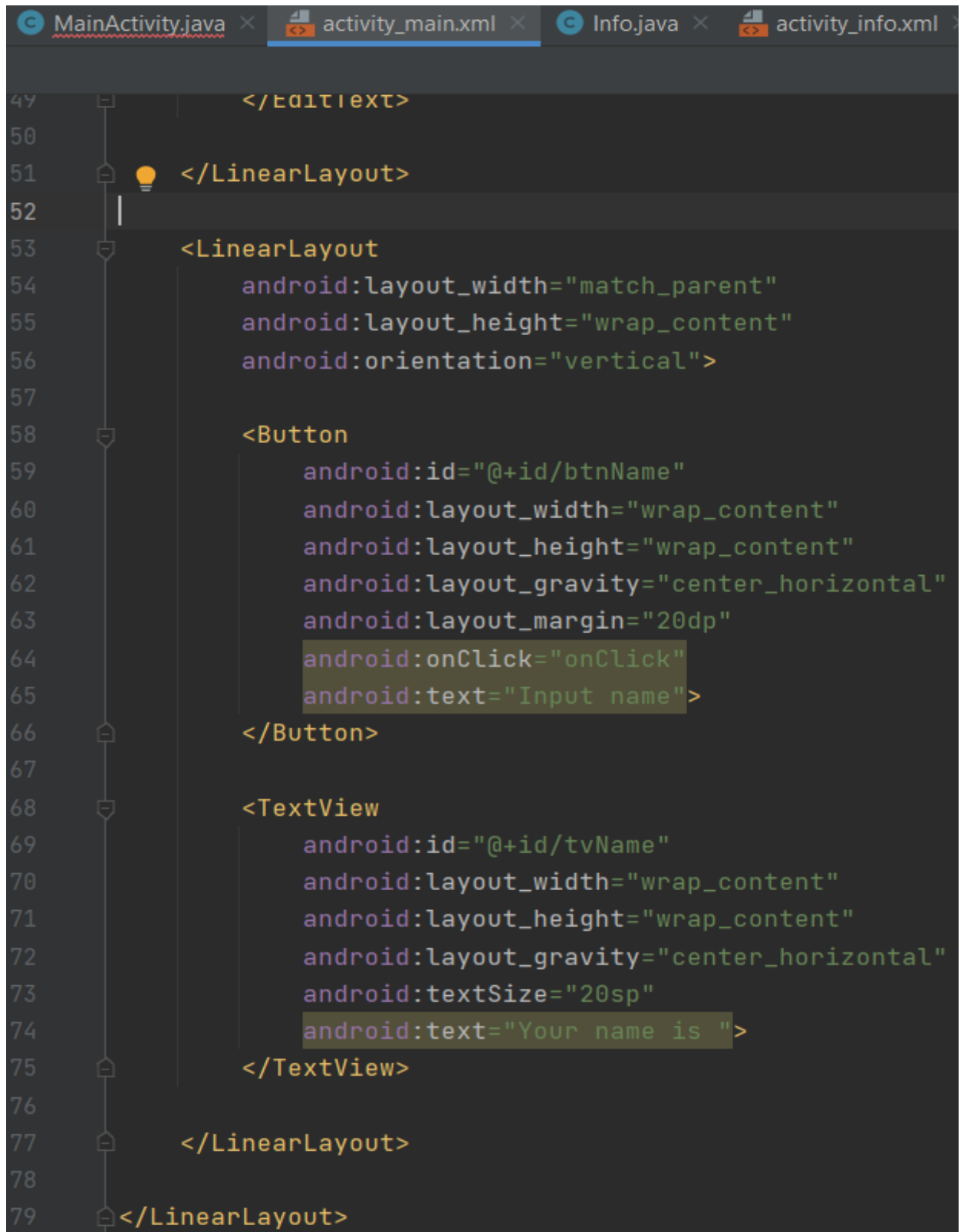


Date: 17.11.2021

Your name is: egorvalyaev

Отлично, результат именно тот, который мы и ожидали увидеть!

Часто требуется запустить активность для того, чтобы получить из неё какой-либо результат. Для этого изучим механизм возврата значения из активности. Добавим в главное окно надпись и кнопку, по нажатию на которую мы будем переходить в другую активность и вводить текст, а при возврате в главную активность этот текст будет отображаться:



```
49      </EditText>
50
51  </LinearLayout>
52
53  <LinearLayout
54      android:layout_width="match_parent"
55      android:layout_height="wrap_content"
56      android:orientation="vertical">
57
58      <Button
59          android:id="@+id/btnName"
60          android:layout_width="wrap_content"
61          android:layout_height="wrap_content"
62          android:layout_gravity="center_horizontal"
63          android:layout_margin="20dp"
64          android:onClick="onClick"
65          android:text="Input name">
66      </Button>
67
68      <TextView
69          android:id="@+id/tvName"
70          android:layout_width="wrap_content"
71          android:layout_height="wrap_content"
72          android:layout_gravity="center_horizontal"
73          android:textSize="20sp"
74          android:text="Your name is ">
75      </TextView>
76
77  </LinearLayout>
78
79 </LinearLayout>
```

Теперь в java-файле инициализируем соответствующую переменную TextView, в главном методе найдём по id объект приложения TextView и затем (в обработчике тапа) добавим обработку нового нажатия на кнопку с id btnName:

```
MainActivity.java x activity_main.xml x Info.java x activity_info.xml x
1 package com.example.intent_filters;
2
3 import ...
11
12 public class MainActivity extends AppCompatActivity implements View.OnClickListener{
13
14     Intent intent;
15     EditText etLName;
16     TextView tvName;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22
23         etLName = findViewById(R.id.etLName);
24         tvName = findViewById(R.id.tvName);
25     }
26
27     @Override
28     public void onClick(View v){
29         switch (v.getId()){
30             case (R.id.btnTime):
31                 intent = new Intent( action: "com.example.intent.action.showtime");
32                 intent.putExtra( name: "lname", etLName.getText().toString());
33                 startActivity(intent);
34                 break;
35             case (R.id.btnDate):
36                 intent = new Intent( action: "com.example.intent.action.showdate");
37                 intent.putExtra( name: "lname", etLName.getText().toString());
38                 startActivity(intent);
39                 break;
40             case (R.id.btnName):
41                 intent = new Intent( packageContext: this, NameActivity.class);
42                 startActivityForResult(intent, requestCode: 1);
43                 break;
44         }
45     }
```

Из-за возникающей ошибки об отсутствии соответствующей активности создадим её и расположим необходимые элементы для ввода текста:

```
MainActivity.java × activity_main.xml × NameActivity.java × activity_name.xml ×
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      <LinearLayout
9          android:id="@+id/linearLayout1"
10         android:layout_width="match_parent"
11         android:layout_height="wrap_content"
12         android:layout_margin="10dp">
13
14         <TextView
15             android:id="@+id/textView1"
16             android:layout_width="wrap_content"
17             android:layout_height="wrap_content"
18             android:text="@string/name">
19         </TextView>
20
21         <EditText
22             android:id="@+id/etName"
23             android:layout_width="wrap_content"
24             android:layout_height="wrap_content"
25             android:layout_marginLeft="10dp"
26             android:layout_weight="1">
27             <requestFocus>
28             </requestFocus>
29         </EditText>
30
31     </LinearLayout>
```

Тут мы создаём линейное расположение текста с именем и полем для ввода пользователем имени, а также кнопку для отправки введенных данных.

```

20
21     <EditText
22         android:id="@+id/etName"
23         android:layout_width="wrap_content"
24         android:layout_height="wrap_content"
25         android:layout_marginLeft="10dp"
26         android:layout_weight="1">
27         <requestFocus>
28         </requestFocus>
29     </EditText>
30
31 </LinearLayout>
32 <Button
33     android:id="@+id/btnOK"
34     android:layout_width="wrap_content"
35     android:layout_height="wrap_content"
36     android:layout_gravity="center_horizontal"
37     android:onClick="onClick"
38     android:text="OK">
39 </Button>
40 </LinearLayout>
41

```

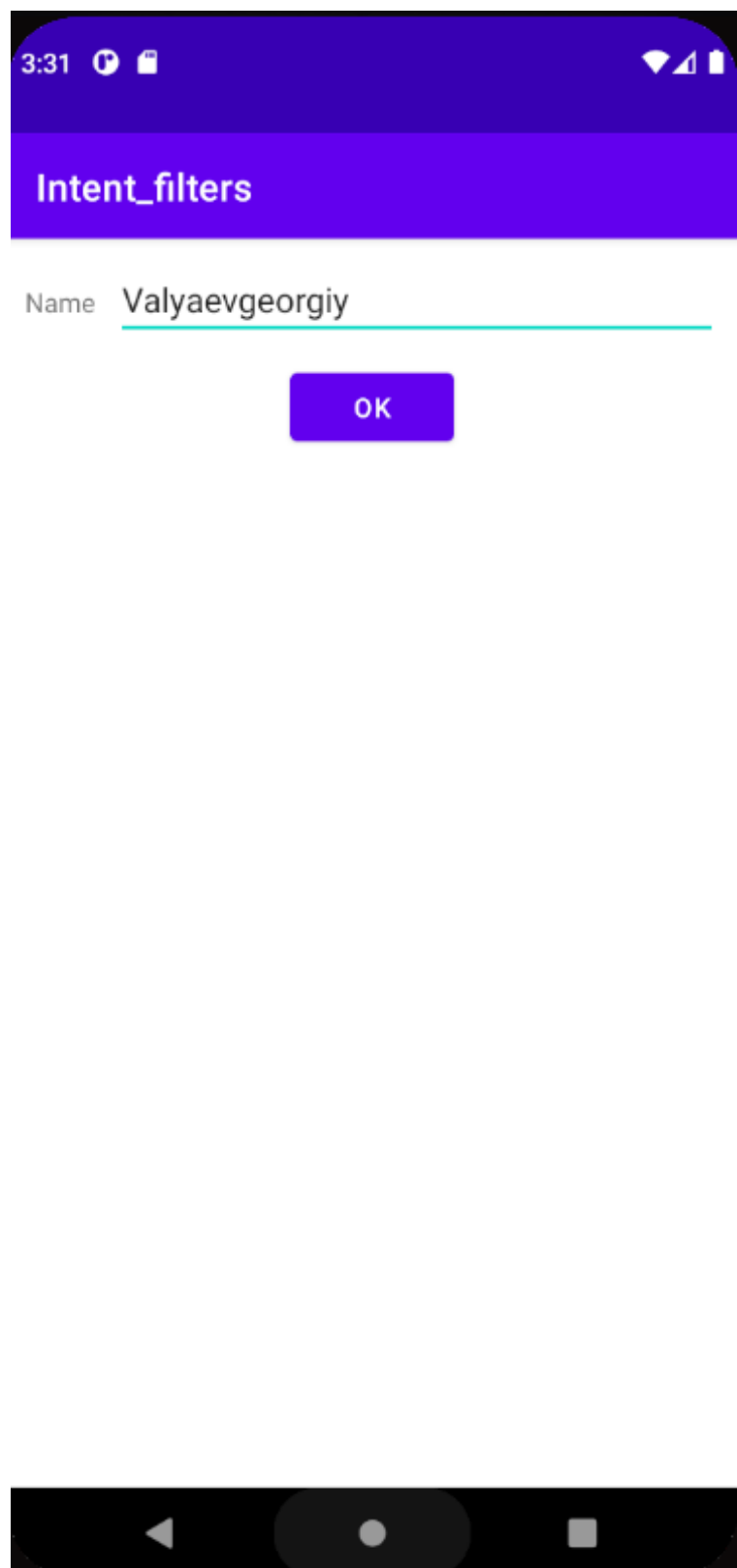
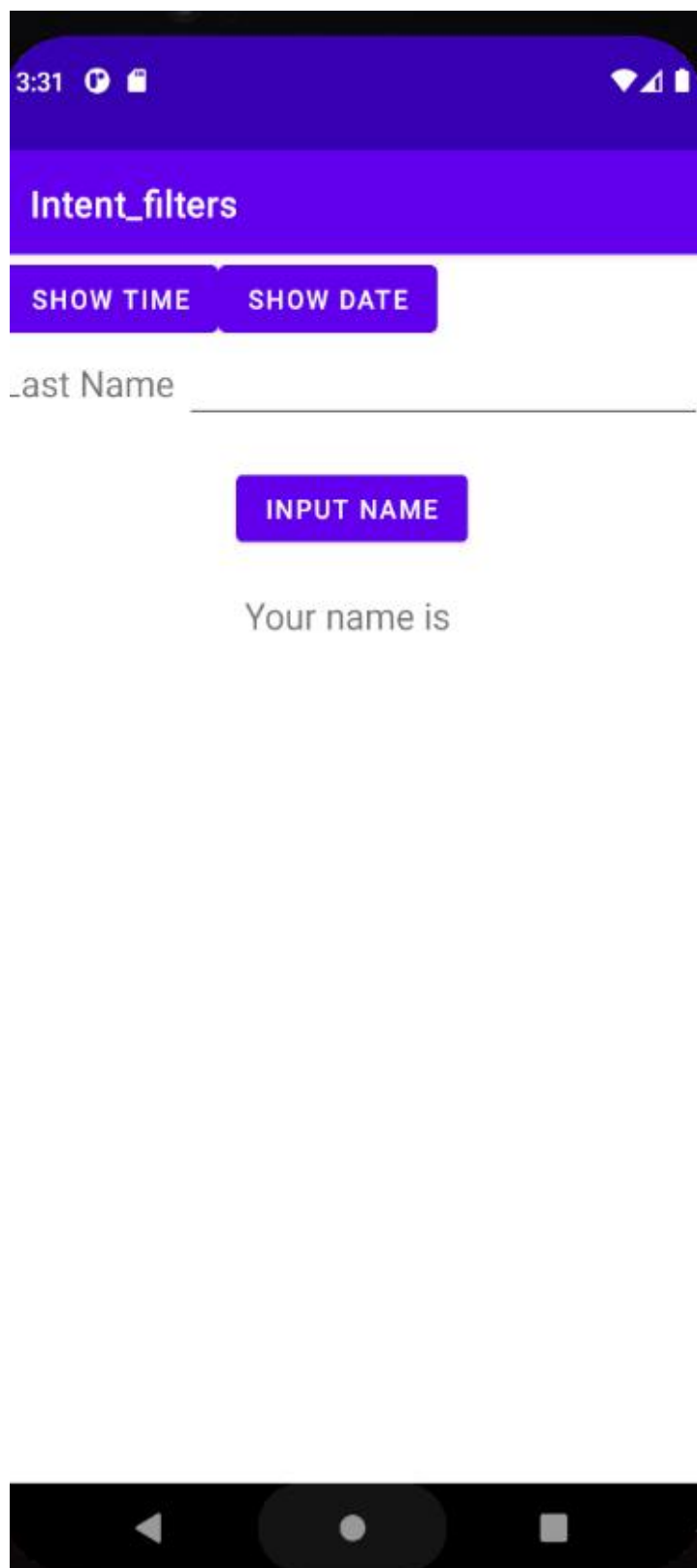
Если говорить про файл Name Activity.java, то тут всё просто: инициализируем переменную с текстового ввода информации, в главном методе её распознаём по id объекта приложения, создаём обработчик тапа, в котором создаём интент, в который поместим возвращаемую информацию:

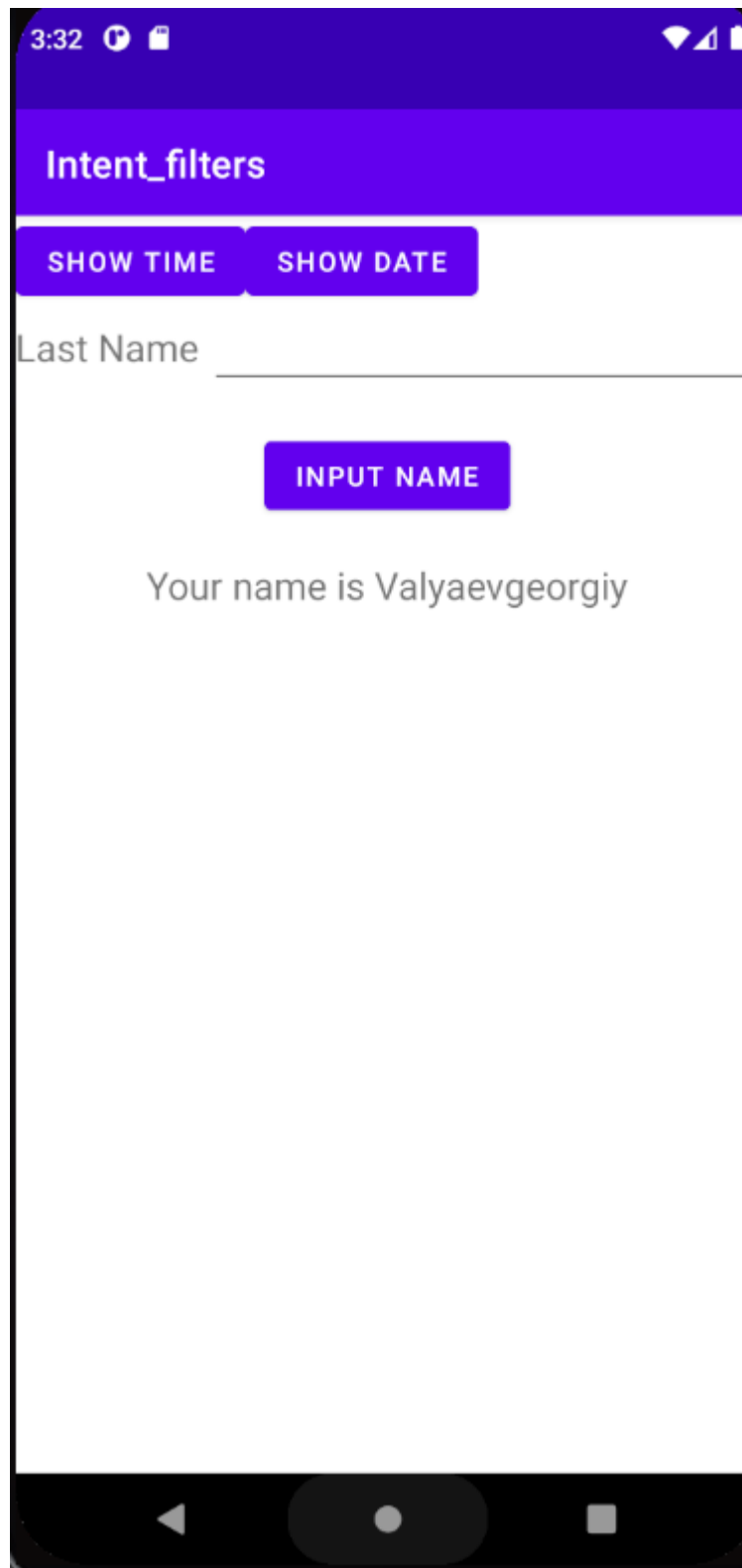
```
MainActivity.java × activity_main.xml × NameActivity.java × activity_name.xml ×
1 package com.example.intent_filters;
2
3 import ...
9
10 public class NameActivity extends AppCompatActivity {
11
12     EditText etName;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_name);
18
19         etName = findViewById(R.id.etName);
20     }
21
22     public void onClick(View v){
23         Intent intent = new Intent();
24         intent.putExtra("name", etName.getText().toString());
25         setResult(RESULT_OK, intent);
26         finish();
27     }
28 }
```

Теперь вернёмся к главной активности в java-код и добавляем метод, который будет обрабатывать обратный интент:


```
MainActivity.java x activity_main.xml x NameActivity.java x activity_name.xml x
28 public void onClick(View v){
29     switch (v.getId()){
30         case (R.id.btnTime):
31             intent = new Intent( action: "com.example.intent.action.showtime");
32             intent.putExtra( name: "lname", etLName.getText().toString());
33             startActivity(intent);
34             break;
35         case (R.id.btnDate):
36             intent = new Intent( action: "com.example.intent.action.showdate");
37             intent.putExtra( name: "lname", etLName.getText().toString());
38             startActivity(intent);
39             break;
40         case (R.id.btnName):
41             intent = new Intent( packageContext: this, NameActivity.class);
42             startActivityForResult(intent, requestCode: 1);
43             break;
44     }
45 }
46
47 @Override
48 public void onActivityResult(int requestCode, int resultCode, Intent data){
49     super.onActivityResult(requestCode, resultCode, data);
50     if (data == null){
51         return;
52     }
53     String name = data.getStringExtra( name: "name");
54     tvName.setText("Your name is " + name);
55 }
56 }
```

Осталось только проверить работу нашего итогового приложения:





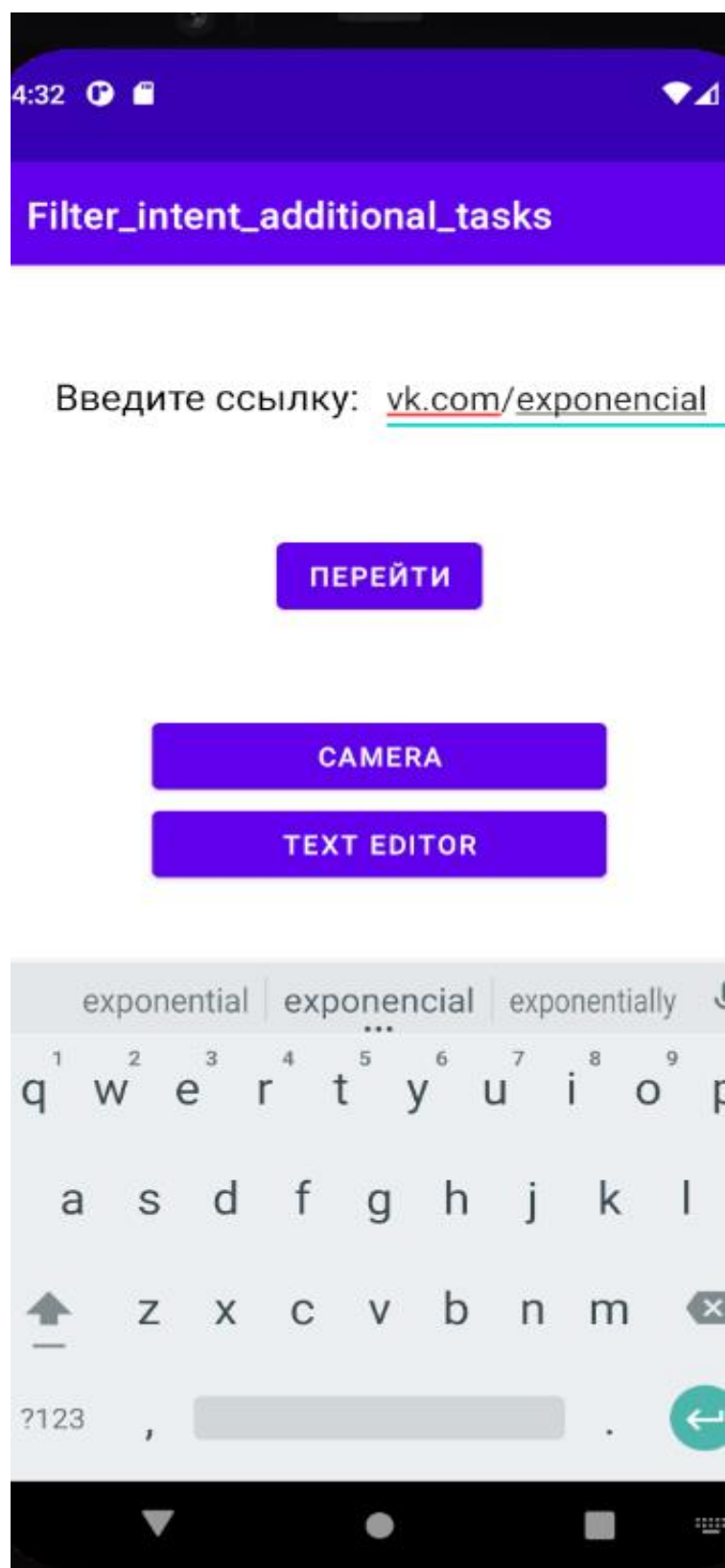
Перейдём теперь к дополнительным заданиям:

Дополнительные задания

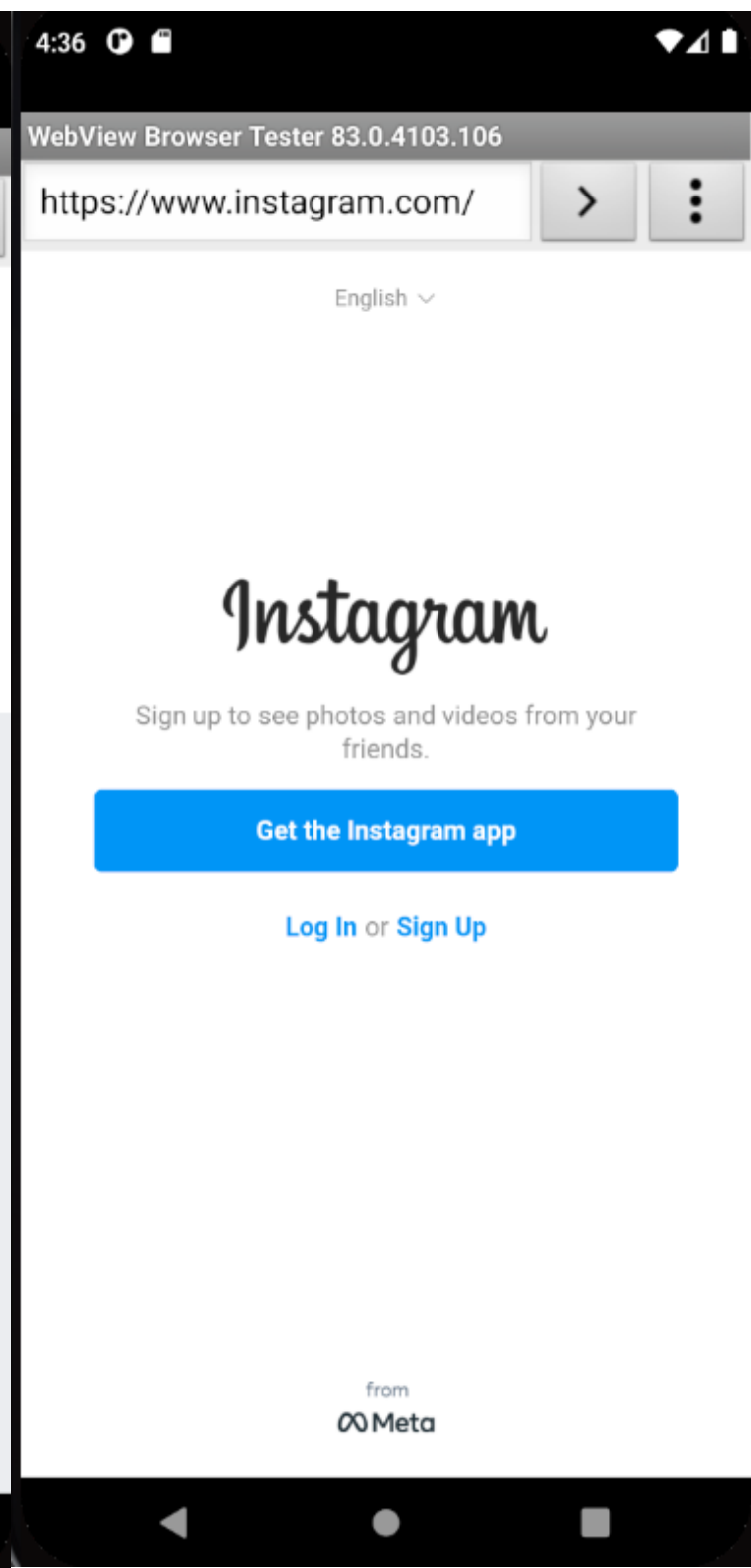
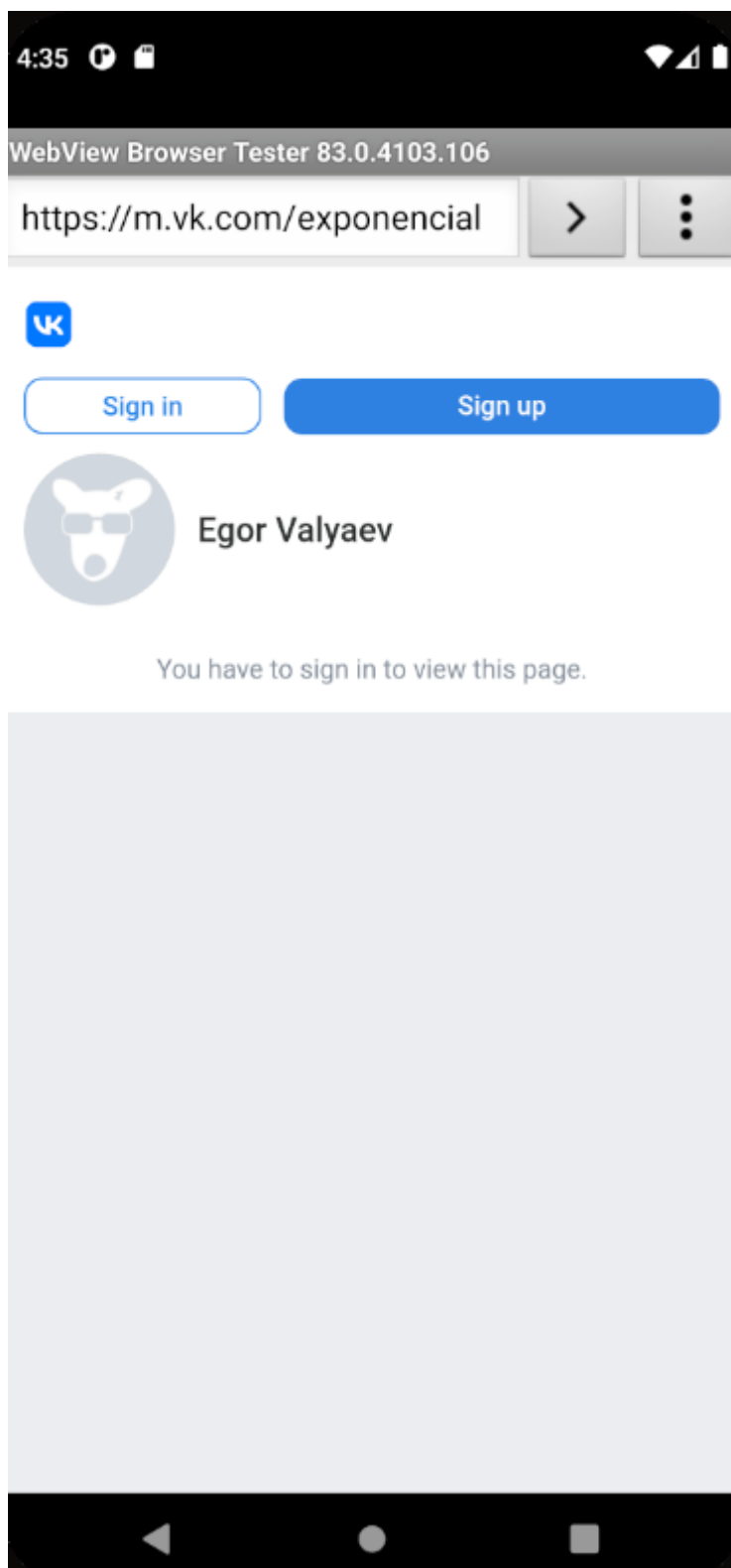
1. Создайте приложение, на главном окне которого будет расположено поле ввода текста и при нажатии на кнопку “перейти” будет запускаться браузер по введенному пользователем адресу.
2. Создайте приложение, отвечающее на какое-либо стандартное системное действие. Проверьте его работоспособность.
3. Создайте приложение, которое выводит текстовую надпись и предлагает выбрать цвет и выравнивание надписи. Выбор должен производиться в двух разных активностях. При возврате в основную активность форматирование надписи должно меняться.
4. (*) Создайте приложение, запускающее приложение камеры. Когда пользователь делает снимок, он должен вернуться в наше приложение, и оно должно отобразить его в виде миниатюры на экране.

Проверим работу приложения:

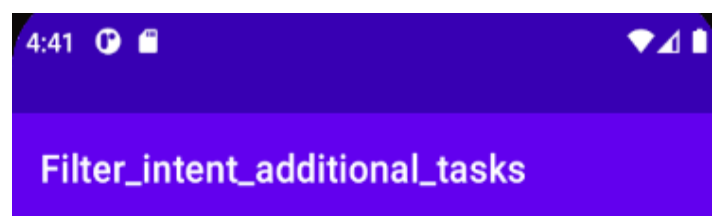
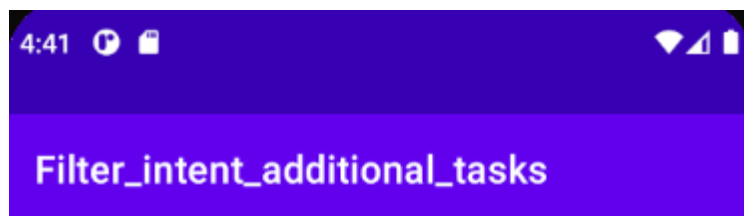
Сперва можно ввести ссылку и попасть в браузер по этой ссылке туда, куда хотелось попасть (потому что добавление `https://` производится автоматически), что просится в 1 дополнительном задании в качестве выполнения.



Вот примерные результаты работы (загрузившиеся веб-страницы, не выходя при этом из приложения):



Далее перейдём к следующей активности (), которая открывает нам сразу приложение «Камера» на смартфоне, а также в приложение занести получившуюся фотографию:



ЗАПУСК КАМЕРЫ

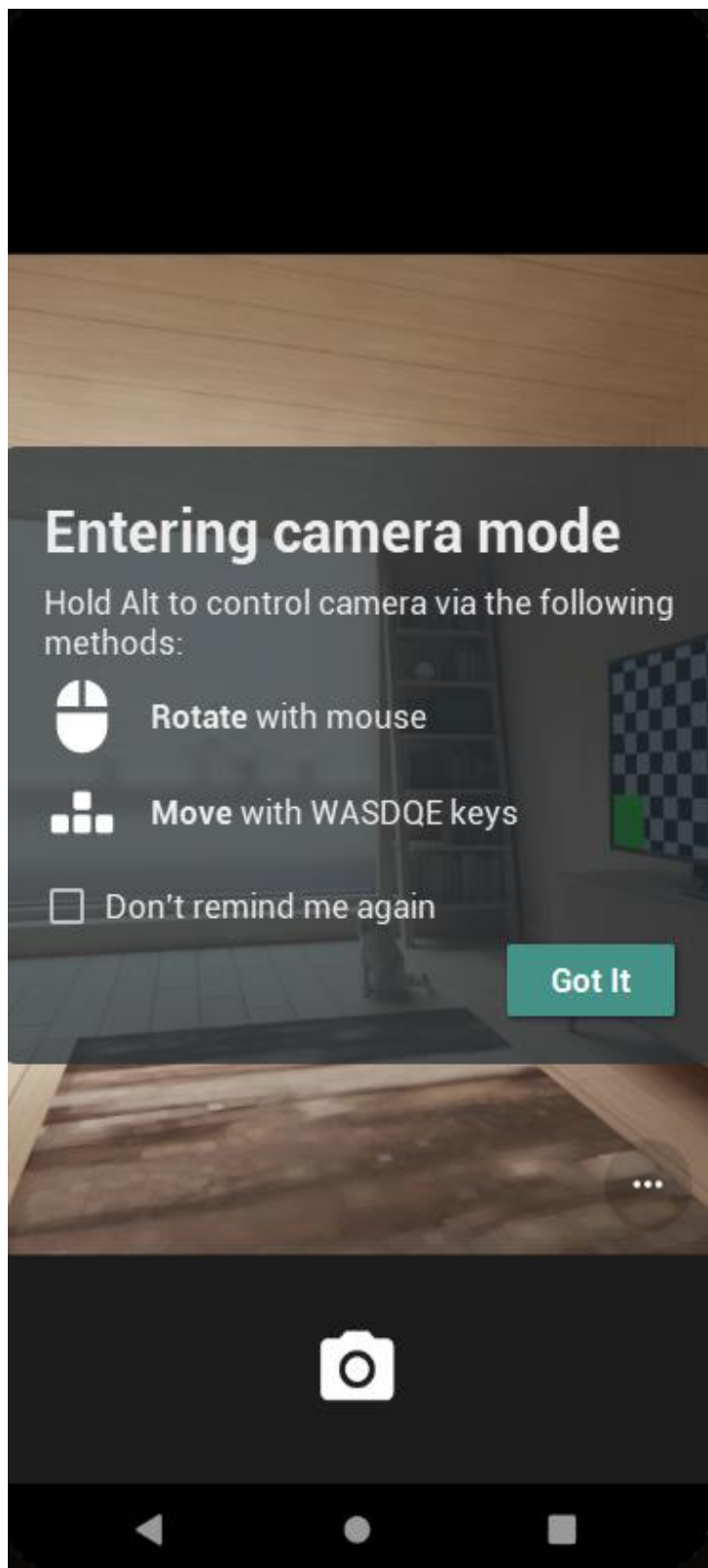
Введите ссылку: instagram.com/

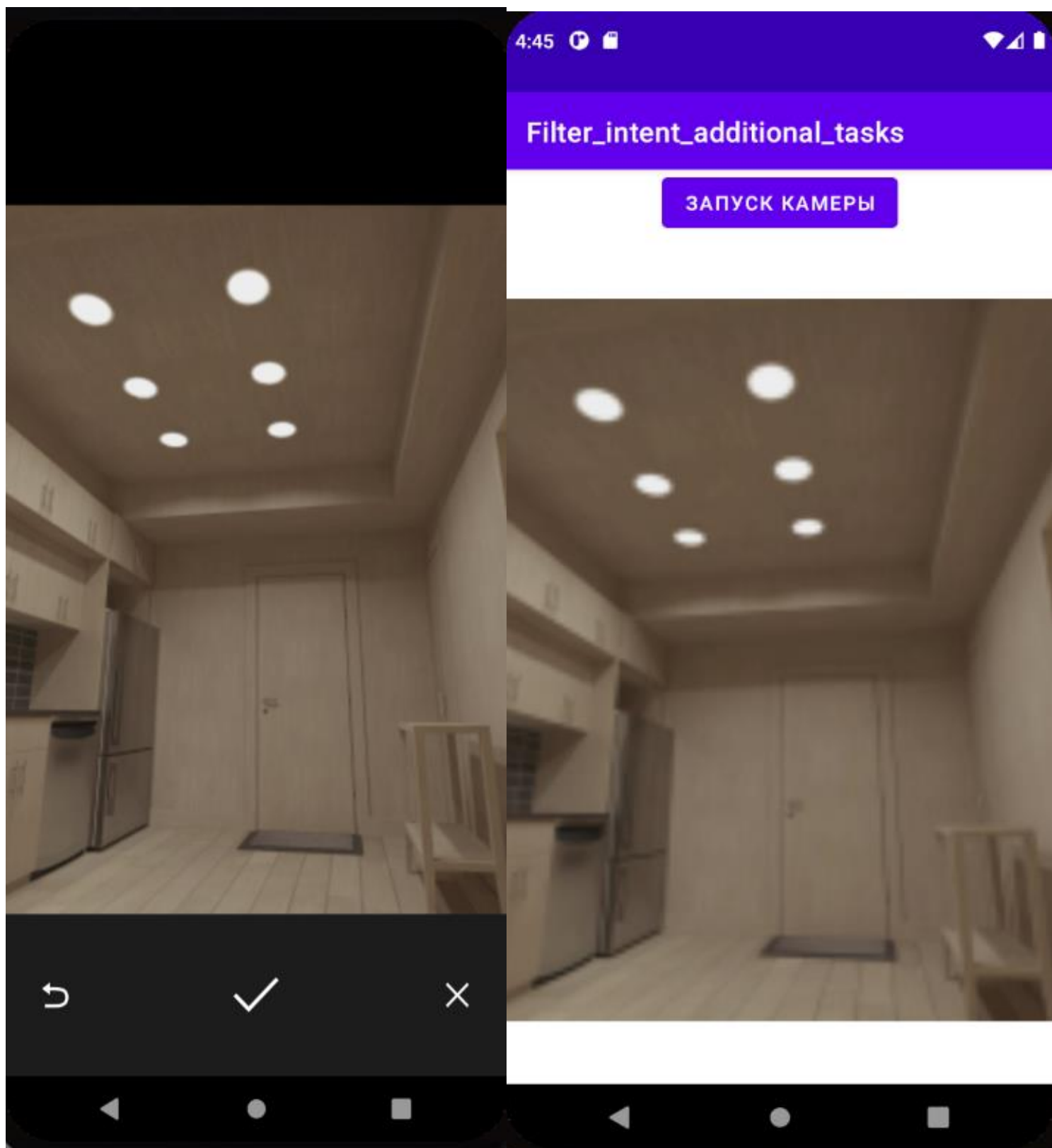
ПЕРЕЙТИ

CAMERA

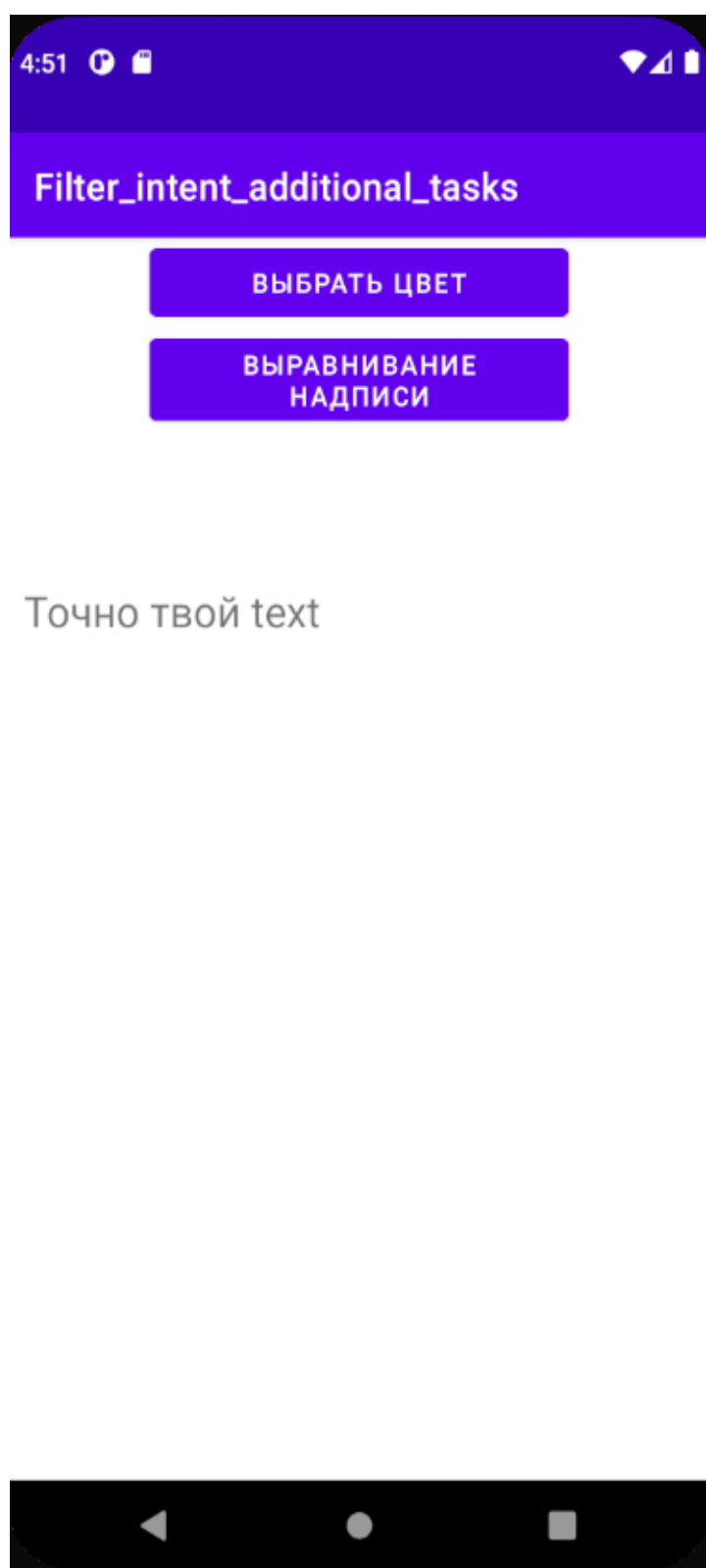
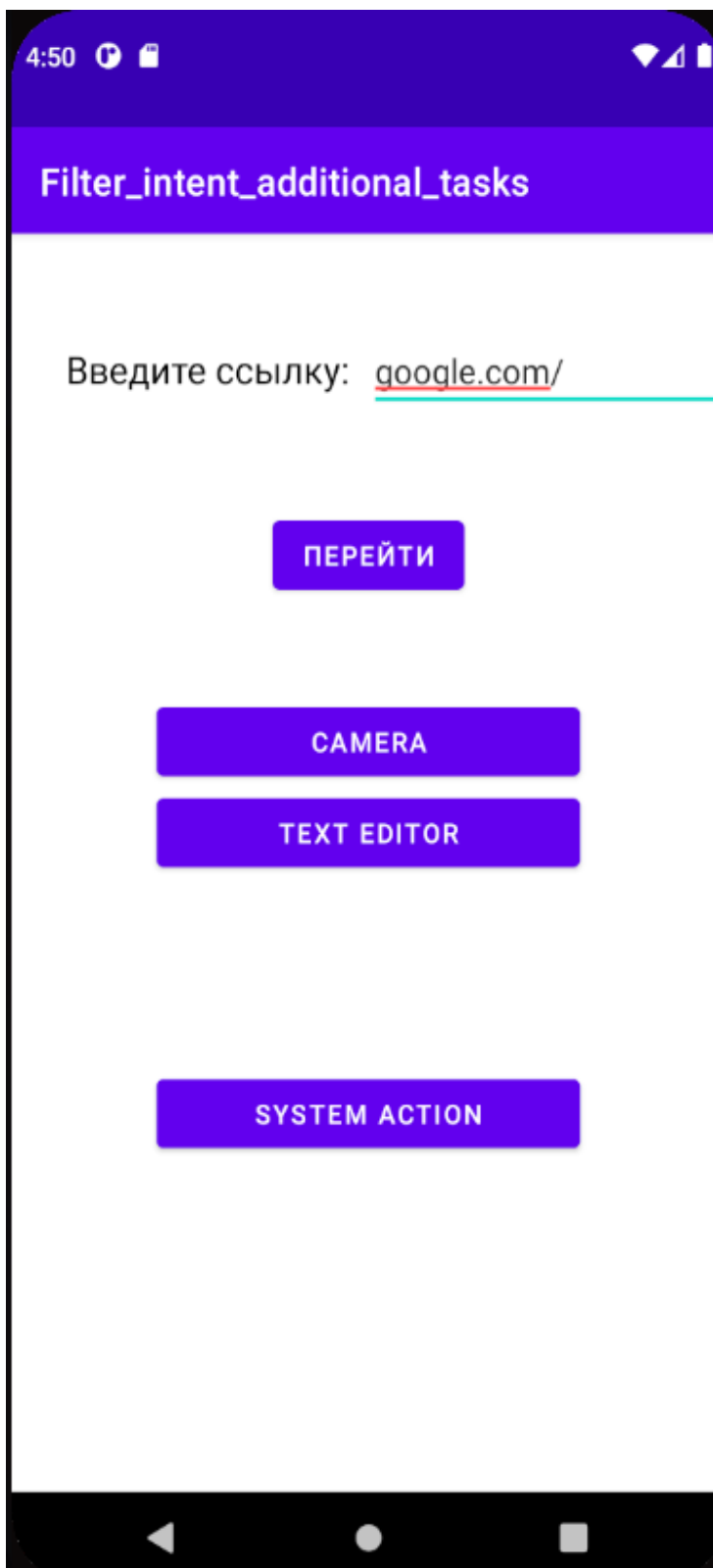
TEXT EDITOR

SYSTEM ACTION





Теперь рассмотрим следующую интересную возможность приложения — это простой редактор текста (по цвету и выравниванию):



4:52



4:53



Filter_intent_additional_tasks

Filter_intent_additional_tasks

Выберите цвет надписи:

Красный

Синий

Жёлтый

Зелёный

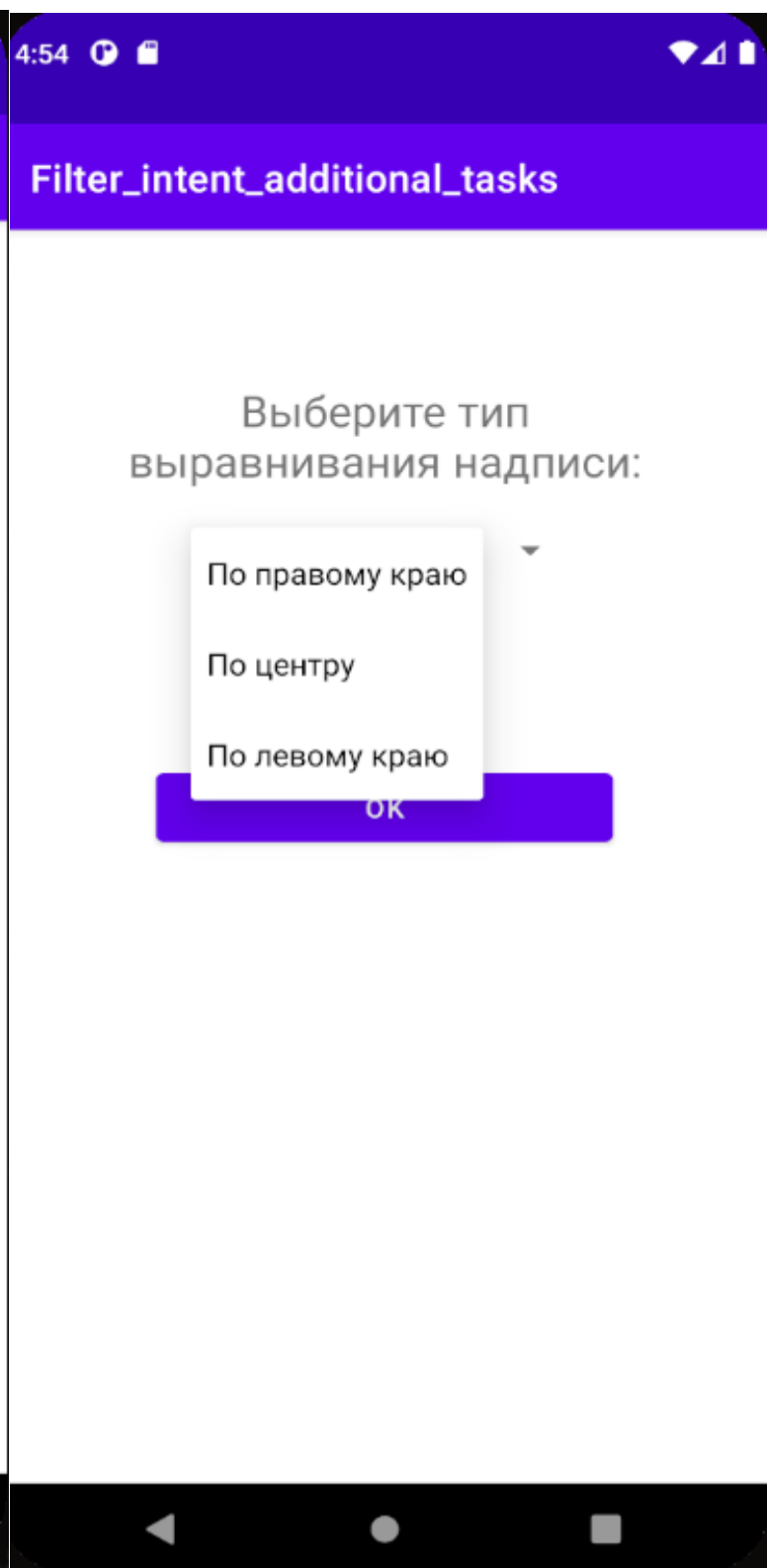
Чёрный

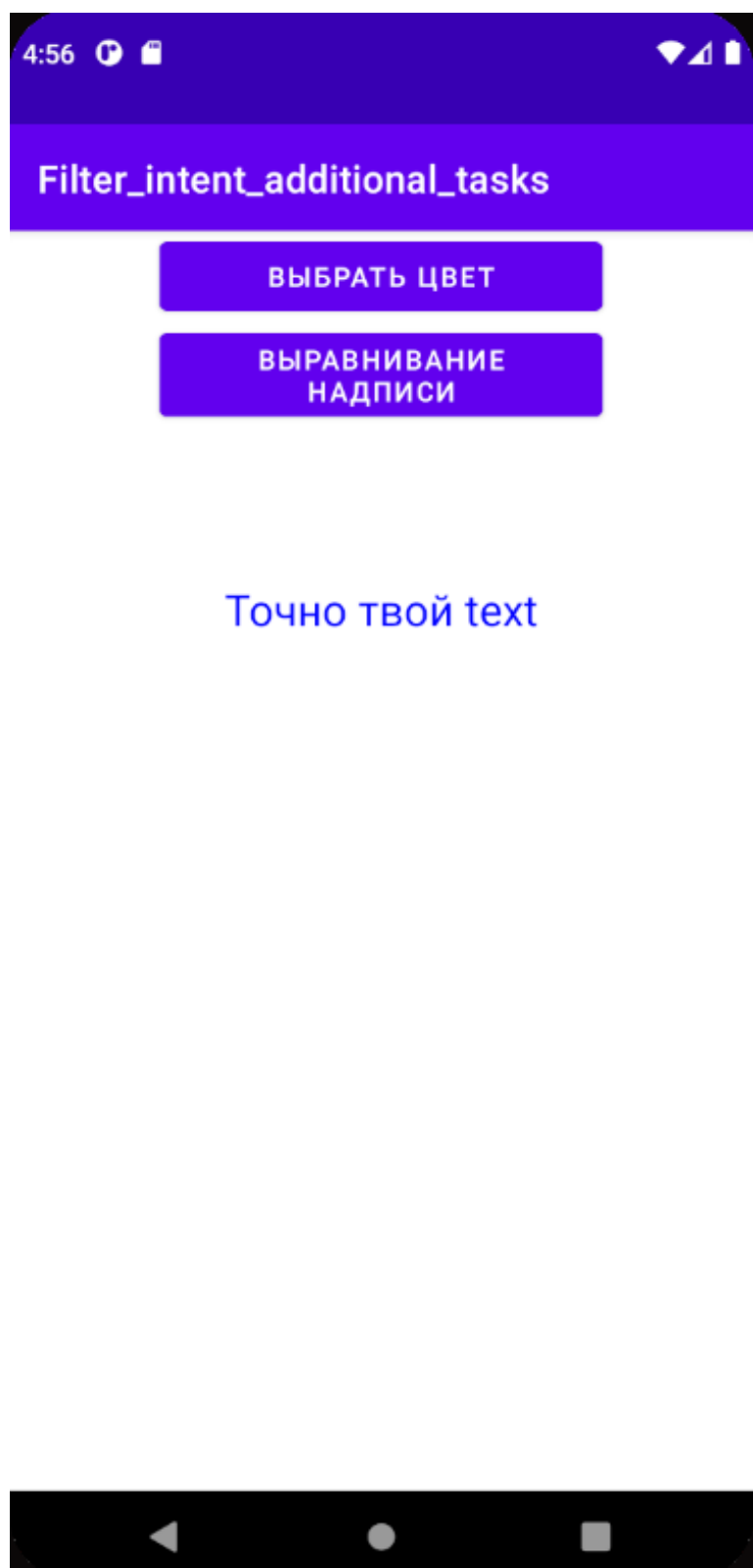
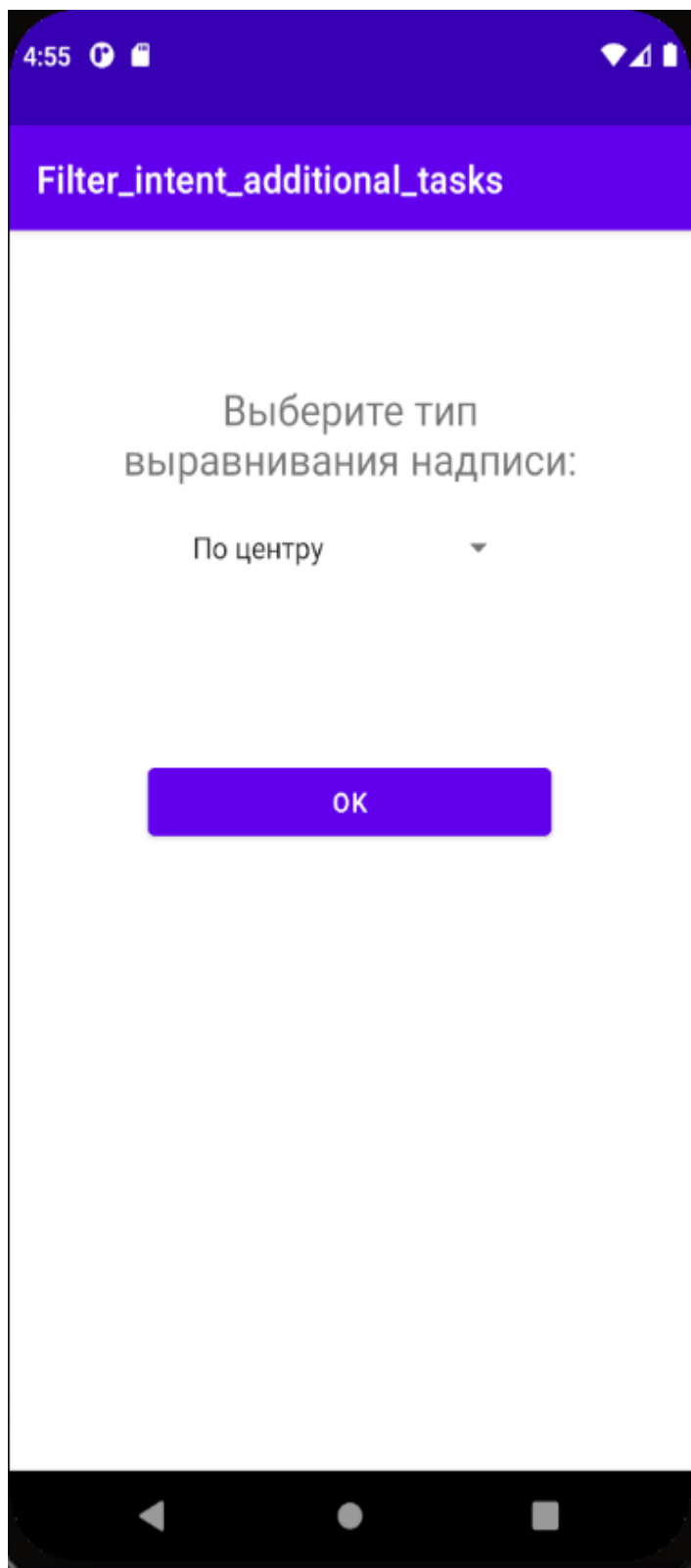
Розовый

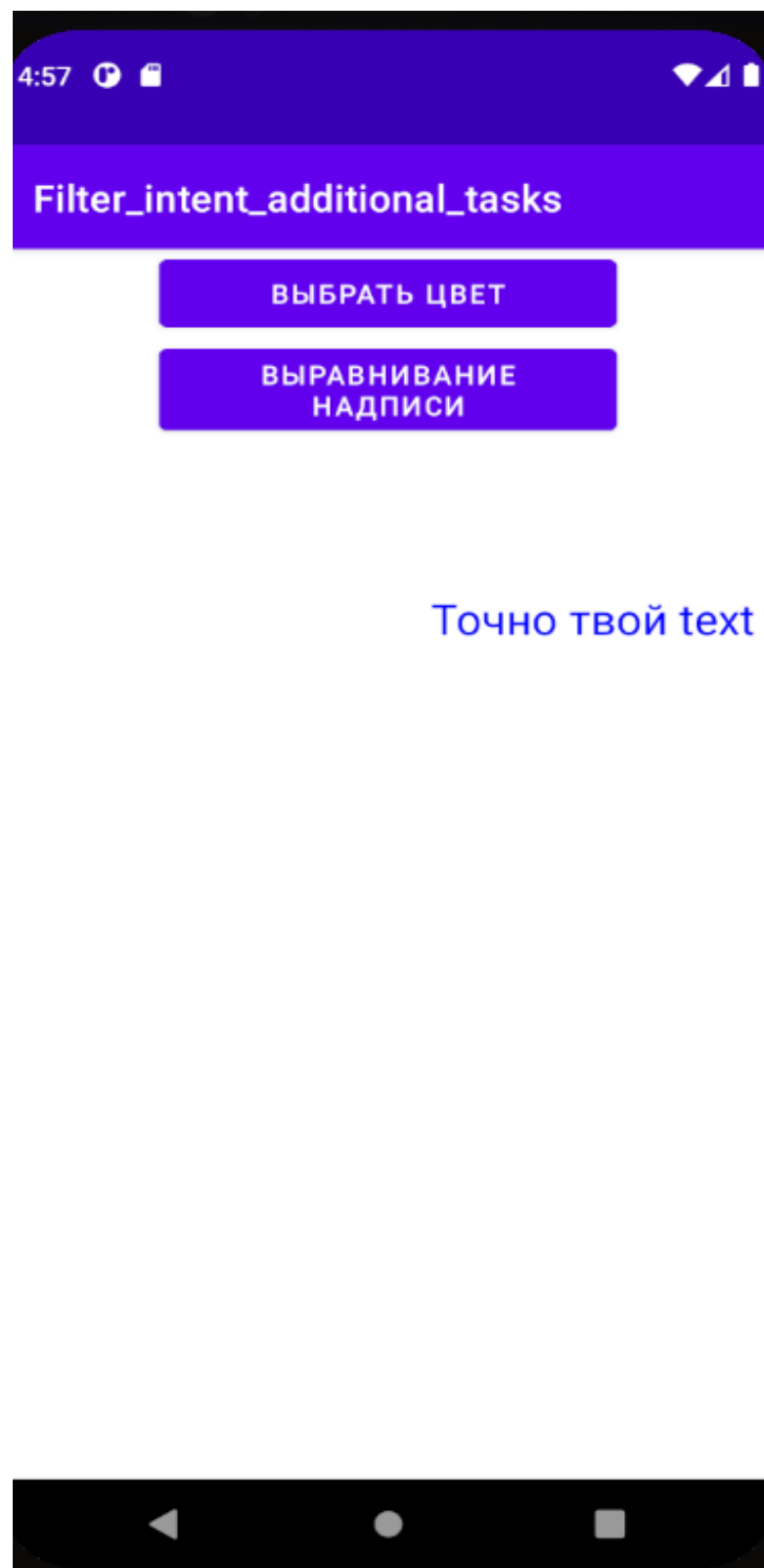
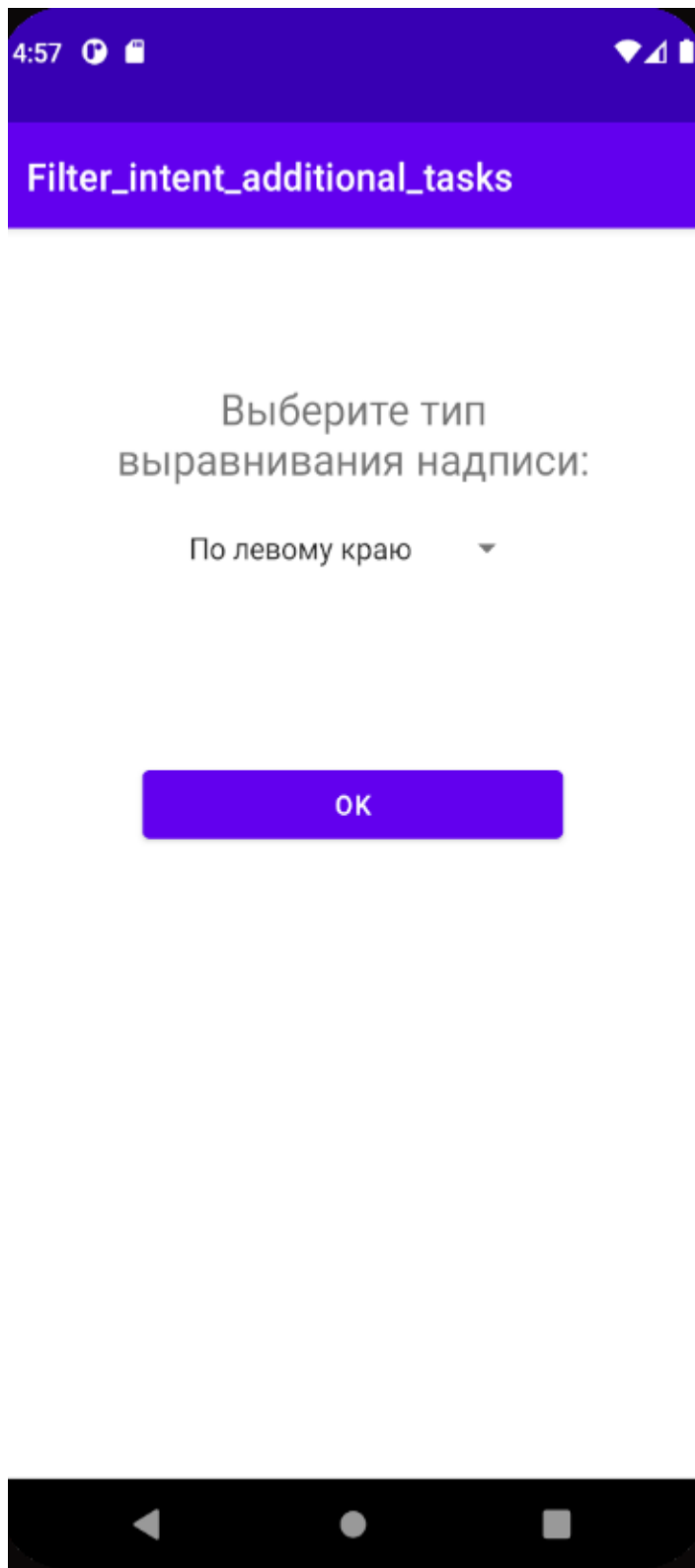
Выберите цвет надписи:

Синий

OK

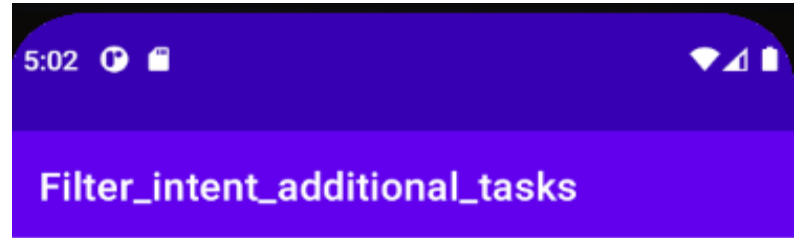
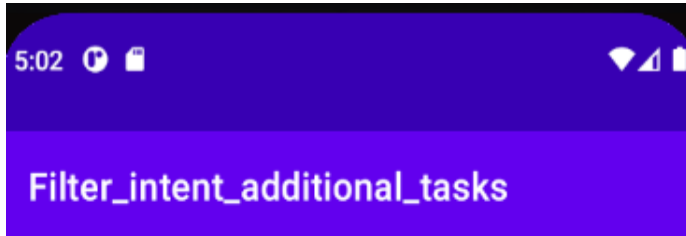


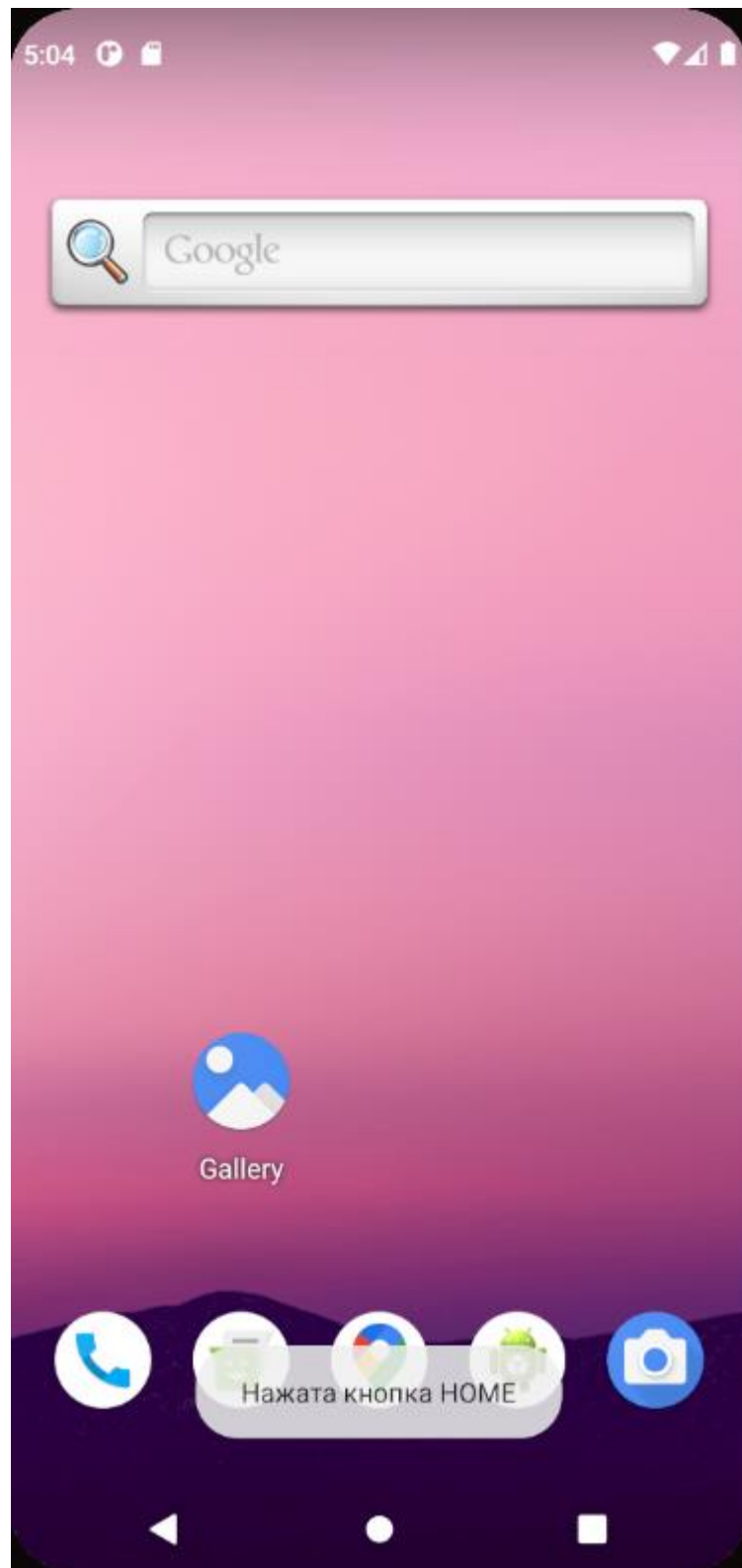




И в качестве последнего элемента функционала приложения стоит указать активность `SystemAction`, которая при нажатии центральной кнопки HOME

пользователем приложение сможет обработать это системное действие, и пользователь получит уведомление как раз-таки от Toast, что он совершил переход домой и вышел в целом из приложения.





С целью экономии места файла и удобства пользования и просмотра уже моего кода всего приложения (оно насчитывает порядка 6 активностей, почти связанных друг с другом), прикладываю ссылку на папки своего репозитория на GitHub со своими пояснениями, что и где находится:

Основная часть заданий

Все файлы с java-кодом:

https://github.com/Valyaevgeorgiy/Android_projects/tree/main/Intent_filt/Intent_filters/app/src/main/java/com/example/intent_filters

Все файлы с XML-кодом:

https://github.com/Valyaevgeorgiy/Android_projects/tree/main/Intent_filt/Intent_filters/app/src/main/res/layout

Файл со строковым ресурсом:

https://github.com/Valyaevgeorgiy/Android_projects/blob/main/Intent_filt/Intent_filters/app/src/main/res/values/strings.xml

Файл манифеста приложения (AndroidManifest.xml):

https://github.com/Valyaevgeorgiy/Android_projects/blob/main/Intent_filt/Intent_filters/app/src/main/AndroidManifest.xml

Дополнительные задания

Все файлы с java-кодом:

https://github.com/Valyaevgeorgiy/Android_projects/tree/main/Intent_filt/Filter_intent_additional_tasks/app/src/main/java/com/example/filter_intent_additional_tasks

Все файлы с XML-кодом:

https://github.com/Valyaevgeorgiy/Android_projects/tree/main/Intent_filt/Filter_intent_additional_tasks/app/src/main/res/layout

Файл со строковым ресурсом:

https://github.com/Valyaevgeorgiy/Android_projects/blob/main/Intent_filt/Filter_intent_additional_tasks/app/src/main/res/values/strings.xml

Файл манифеста приложения (AndroidManifest.xml):

https://github.com/Valyaevgeorgiy/Android_projects/blob/main/Intent_filt/Filter_intent_additional_tasks/app/src/main/AndroidManifest.xml