

Федеральное агентство образования Российской Федерации
Государственное образовательное учреждение
высшего профессионального образования
«Владимирский государственный университет»
Кафедра ИСПИ

Лабораторная работа №3
по дисциплине «»
«Переход между активностями. Работа с Intent»

Разработал:

Владимир, 2016

Цель работы

Изучить функционал, позволяющий осуществлять переход между активностями приложения. Научиться работать с Intent.

Теоретические сведения

К примеру, мы имеем две Activity. Их названия ActivityOne и ActivityTwo. Для перехода на вторую активность с первой используется следующий код.

```
Intent intent = new Intent(this, ActivityTwo.class);
startActivity(intent);
```

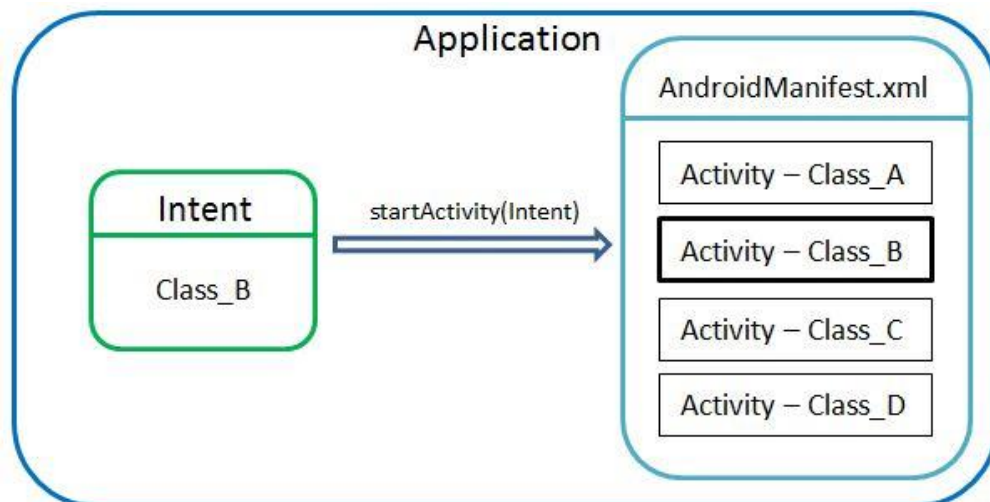
Что такое Intent

В нашем случае Intent – это объект, в котором мы прописываем, какое Activity нам необходимо вызвать. После чего мы передаем этот Intent-объект метод startActivity, который находит соответствующее Activity и показывает его. При создании Intent мы использовали конструктор Intent (Context packageContext, Class cls) с двумя параметрами.

Первый параметр – это Context. Activity является подклассом Context, поэтому мы можем использовать ее – this. Вкратце, Context – это объект, который предоставляет доступ к базовым функциям приложения таким как: доступ к ресурсам, к файловой системе, вызов Activity и т.д

Второй параметр – имя класса. При создании записи Activity в манифест-файле мы указываем имя класса. И теперь если мы укажем тот же класс в Intent – то система, просмотрев манифест-файл обнаружит соответствие и покажет соответствующий Activity.

Вызов Activity с помощью такого Intent – это **явный** вызов. Т.е. с помощью класса мы **явно** указываем какое Activity хотели бы увидеть. Это обычно используется **внутри одного приложения**. Схематично это можно изобразить так:



Здесь мы создаем **Intent**, в качестве параметра передаем ему **класс Class_B**. Далее вызываем метод **startActivity** с созданным **Intent** в качестве параметра. Метод проверяет **AndroidManifest** на наличие **Activity** связанной с классом **Class_B** и если находит, то отображает. Все это в пределах одного приложения.

Существует также **неявный** вызов Activity. Он отличается тем, что при создании **Intent** мы используем **не класс**,

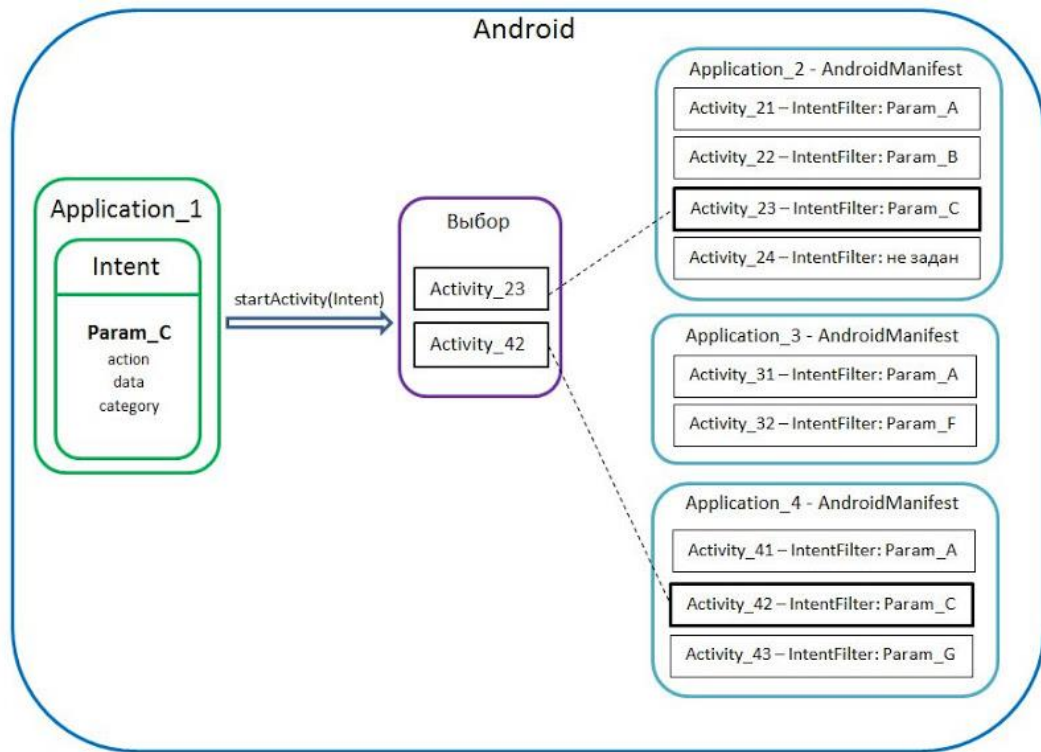
а **заполняем** параметры **action, data, category** определенными значениями.

Комбинация этих значений определяют цель, которую мы хотим достичь.

Например: отправка письма, открытие гиперссылки, редактирование текста, просмотр картинки, звонок по определенному номеру и т.д. В свою очередь для **Activity** мы прописываем **Intent Filter** - это набор тех же параметров: **action, data, category** (но значения уже свои - зависят от того, что умеет делать Activity). И если параметры нашего

Intent **совпадают** с условиями этого фильтра, то Activity вызывается. Но при этом поиск уже идет **по всем Activity всех приложений** в системе. Если находится **несколько**, то система предоставляет вам **выбор**, какой именно программой вы хотите воспользоваться. Схематично это можно изобразить так:

Intent **совпадают** с условиями этого фильтра, то Activity вызывается. Но при этом поиск уже идет **по всем Activity всех приложений** в системе. Если находится **несколько**, то система предоставляет вам **выбор**, какой именно программой вы хотите воспользоваться. Схематично это можно изобразить так:



В Application_1 создается **Intent**, заполняются параметры **action**, **data**, **category**. Для удобства, получившийся набор параметров назовем **Param_C**. С помощью **startActivity** этот **Intent** отправляется на поиски **подходящей** Activity, которая сможет выполнить то, что нам нужно (т.е. то, что определено с помощью Param_C). В системе есть разные приложения, и в каждом из них несколько Activity. Для некоторых Activity определен Intent Filter (наборы Param_A, Param_B и т.д.), для некоторых нет. Метод **startActivity** **сверяет** набор параметров **Intent** и наборы параметров **Intent Filter** для каждой **Activity**. Если наборы **совпадают** (Param_C для обоих), то Activity считается **подходящей**. Если в итоге нашлась только одна Activity – она и отображается. Если же нашлось **несколько** подходящих Activity, то пользователю выводится **список**, где он может сам выбрать какое приложение ему использовать.

Например, если в системе установлено несколько музыкальных плееров, и вы запускаете mp3, то система выведет вам список Activity, которые умеют играть музыку и попросит выбрать, какое из них использовать. А те Activity,

которые умеют редактировать текст, показывать картинки, звонить и т.п. будут проигнорированы.

Порядок выполнения работы

1. Создать новый Android Application Project;
2. Добавить активности;
3. Написать логику работы приложения
4. Выполнить индивидуальные задания
5. Оформить отчет

Пример

Создадим проект:

Project name: P0241_TwoActivityState

Build Target: Android 2.3.3

Application name: TwoActivityState

Package name: ru.startandroid.develop.p0241twoactivitystate

Create Activity: MainActivity

В main.xml пишем следующее:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello">
    </TextView>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Go to Activity Two"
        android:id="@+id/btnActTwo">
    </Button>
</LinearLayout>
```

Кнопка “Go to Activity Two” будет вызывать второе Activity.

Откроем MainActivity.java и пишем туда все **методы**, на этот раз, включая **onRestart**, и в методах прописываем запись в **логи**. Также описываем и находим **кнопку**, присваиваем ей **обработчик**. В методе **onClick** пока ничего не пишем.

```
package ru.startandroid.develop.p0241twoactivitystate;
```

```

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class MainActivity extends Activity implements OnClickListener {

    final String TAG = "States";

    Button btnActTwo;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        btnActTwo = (Button) findViewById(R.id.btnActTwo);
        btnActTwo.setOnClickListener(this);

        Log.d(TAG, "MainActivity: onCreate()");
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d(TAG, "MainActivity: onRestart()");
    }

    @Override
    protected void onStart() {
        super.onStart();
        Log.d(TAG, "MainActivity: onStart()");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.d(TAG, "MainActivity: onResume()");
    }

    @Override
    protected void onPause() {
        super.onPause();
        Log.d(TAG, "MainActivity: onPause()");
    }

    @Override
    protected void onStop() {
        super.onStop();
        Log.d(TAG, "MainActivity: onStop()");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d(TAG, "MainActivity: onDestroy()");
    }

    @Override
    public void onClick(View v) {
    }
}

```

Создаем второе Activity. Назовем ее **ActivityTwo**. Надо **создать класс** с таким именем и с **суперклассом** `android.app.Activity`, и прописать новое Activity в **манифест-файле**. Также надо создать layout-файл, назовем его `two.xml` и заполним этим кодом:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is Activity Two">
    </TextView>
</LinearLayout>
```

Просто `TextView` с текстом, чтобы было понятно, что это `ActivityTwo`.

Создаем класс. Код `ActivityTwo.java`:

```
package ru.startandroid.develop.p0241twoactivitystate;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;

public class ActivityTwo extends Activity {

    final String TAG = "States";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.two);
        Log.d(TAG, "ActivityTwo: onCreate()");
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d(TAG, "ActivityTwo: onRestart()");
    }

    @Override
    protected void onStart() {
        super.onStart();
        Log.d(TAG, "ActivityTwo: onStart()");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.d(TAG, "ActivityTwo: onResume()");
    }
}
```

```

@Override
protected void onPause() {
    super.onPause();
    Log.d(TAG, "ActivityTwo: onPause()");
}

@Override
protected void onStop() {
    super.onStop();
    Log.d(TAG, "ActivityTwo: onStop()");
}

@Override
protected void onDestroy() {
    super.onDestroy();
    Log.d(TAG, "ActivityTwo: onDestroy()");
}
}

```

Не забудьте добавить запись об **ActivityTwo** в манифест. И теперь мы можем дописать код метода **onClick** в **MainActivity.java**, прописав там вызов **ActivityTwo**

```

@Override
public void onClick(View v) {
    Intent intent = new Intent(this, ActivityTwo.class);
    startActivity(intent);
}

```

(добавляете только подчеркнутый код).

Вот что должно получиться:





Задания

1. Выполнить пример из лабораторной работы.
2. Выполнить одно из следующих заданий:
 - 2.1. Добавить переход на третью активность. Добавить переход с третьей активности на первую.
 - 2.2. Реализовать передачу данных при переходе между активностями.

Требования к оформлению отчета

Отчет должен содержать:

1. титульный лист;
2. цель работы;
3. задание;
4. описание выполнения примера
5. описание выполнения индивидуального задания со скриншотами.
6. выводы.

Контрольные вопросы

1. Каково одно из главных преимуществ платформы Android?
2. Какие инструменты входят в состав Android SDK?
3. Что такое эмулятор Android?

Дополнительная литература

1. <http://www.startandroid.ru/ru/uroki/vse-uroki-spiskom/>
2. <http://developer.alexanderklimov.ru/android/>
- 3.