[Вызываем Activity и получаем результат. Метод startActivityForResult](https://startandroid.ru/ru/uroki/vse-uroki-spiskom/68-urok-29-vyzyvaem-activity-i-poluchaem-rezultat-metod-startactivityforresult.html)

В этом уроке:

- вызываем Activity с возвратом результата

Бывает необходимость **вызвать Activity**, **выполнить**на нем какое-либо **действие**и **вернуться** с **результатом**. Например – при создании SMS. Вы жмете кнопку «добавить адресата», система показывает экран со списком из адресной книги, вы выбираете нужного вам абонента и возвращаетесь в экран создания SMS. Т.е. вы **вызвали экран выбора**абонента, а он **вернул**вашему экрану **результат**.

Об этом можно почитать [здесь](http://developer.android.com/reference/android/app/Activity.html#StartingActivities) и [здесь](http://developer.android.com/guide/topics/fundamentals/activities.html#StartingAnActivityForResult).

Давайте посмотрим на практике. Создадим приложение с двумя экранами. С первого экрана будем вызывать второй экран, там вводить данные, нажимать кнопку и возвращаться на первый экран с введенными данными. Например, будем таким образом запрашивать имя.

**Создадим** **проект:**

**Project name**: P0291\_SimpleActivityResult  
**Build Target**: Android 2.3.3  
**Application name**: SimpleActivityResult  
**Package name**: ru.startandroid.develop.p0291simpleactivityresult  
**Create Activity**: MainActivity

Открываем **main.xml** и нарисуем такой экран:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | <?xml version="1.0" encoding="utf-8"?>  <LinearLayout      xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="fill\_parent"      android:layout\_height="fill\_parent"      android:orientation="vertical">      <Button          android:id="@+id/btnName"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_gravity="center\_horizontal"          android:layout\_margin="20dp"          android:text="Input name">      </Button>      <TextView          android:id="@+id/tvName"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_gravity="center\_horizontal"          android:text="Your name is ">      </TextView>  </LinearLayout> |

На экране **TextView**, который будет отображать имя, и **кнопка**, которая будет вызывать экран для ввода.

Кодим **MainActivity.java**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | package ru.startandroid.develop.p0291simpleactivityresult;    import android.app.Activity;  import android.content.Intent;  import android.os.Bundle;  import android.view.View;  import android.view.View.OnClickListener;  import android.widget.Button;  import android.widget.TextView;    public class MainActivity extends Activity implements OnClickListener {      TextView tvName;    Button btnName;        /\*\* Called when the activity is first created. \*/      @Override      public void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.main);            tvName = (TextView) findViewById(R.id.tvName);          btnName = (Button) findViewById(R.id.btnName);          btnName.setOnClickListener(this);        }      @Override    public void onClick(View v) {      Intent intent = new Intent(this, NameActivity.class);      startActivityForResult(intent, 1);    }        @Override    protected void onActivityResult(int requestCode, int resultCode, Intent data) {      if (data == null) {return;}      String name = data.getStringExtra("name");      tvName.setText("Your name is " + name);    }  } |

Определяем TextView и кнопку, настраиваем обработчик. В методе обработчика **onClick**создаем **Intent**, указываем **класс**второго **Acivity**(которое создадим чуть позже, на ошибку не обращайте внимания). Для отправки используем [startActivityForResult](http://developer.android.com/reference/android/app/Activity.html" \l "startActivityForResult(android.content.Intent,%20int)" \t "_blank). Отличие от обычного **startActivity**в том, что **MainActivity**становится «**родителем**» для **NameActivity**. И когда NameActivity закрывается, вызывается метод [onActivityResult](http://developer.android.com/reference/android/app/Activity.html" \l "onActivityResult(int,%20int,%20android.content.Intent)" \t "_blank) в MainActivity, тем самым давая нам знать, что закрылось Activity, которое мы вызывали методом **startActivityForResult**.

В **startActivityForResult**в качестве параметров мы передаем **Intent**и **requestCode**. **requestCode**– необходим для идентификации. В этом уроке мы его укажем, но не будем использовать по назначению. В следующем же уроке разберемся подробнее, зачем он нужен.

В **onActivityResult**мы видим следующие параметры:   
**requestCode** – тот же идентификатор, что и в startActivityForResult. По нему определяем, с какого Activity пришел результат.   
**resultCode**– код возврата. Определяет успешно прошел вызов или нет.  
**data**– Intent, в котором возвращаются данные

**requestCode**и **resultCode**мы пока использовать не будем, подробнее рассмотрим их на следующем уроке. А из **data**мы будем получать объект по имени *name*и выводить значение в **TextView**.

Если мы извлекаем из Intent объект с именем *name*, значит надо, чтобы кто-то его туда положил. Этим займется **NameActivity**.

Создадим экран **name.xml**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35 | <?xml version="1.0" encoding="utf-8"?>  <LinearLayout      xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:orientation="vertical">      <LinearLayout          android:id="@+id/linearLayout1"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:layout\_margin="10dp">          <TextView              android:id="@+id/textView1"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:text="Name">          </TextView>          <EditText              android:id="@+id/etName"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:layout\_marginLeft="10dp"              android:layout\_weight="1">              <requestFocus>              </requestFocus>          </EditText>      </LinearLayout>      <Button          android:id="@+id/btnOK"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_gravity="center\_horizontal"          android:text="OK">      </Button>  </LinearLayout> |

В поле ввода будем вводить имя и жать кнопку OK.

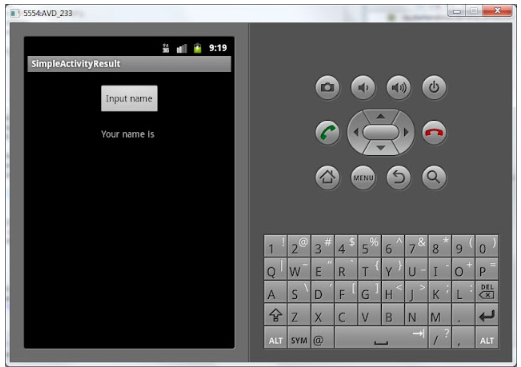
Создаем класс **NameActivity**и прописываем его в манифесте:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | package ru.startandroid.develop.p0291simpleactivityresult;    import android.app.Activity;  import android.content.Intent;  import android.os.Bundle;  import android.view.View;  import android.view.View.OnClickListener;  import android.widget.Button;  import android.widget.EditText;    public class NameActivity extends Activity implements OnClickListener {      EditText etName;    Button btnOK;      @Override    protected void onCreate(Bundle savedInstanceState) {      // TODO Auto-generated method stub      super.onCreate(savedInstanceState);      setContentView(R.layout.name);        etName = (EditText) findViewById(R.id.etName);      btnOK = (Button) findViewById(R.id.btnOK);      btnOK.setOnClickListener(this);    }      @Override    public void onClick(View v) {      Intent intent = new Intent();      intent.putExtra("name", etName.getText().toString());      setResult(RESULT\_OK, intent);      finish();    }  } |

Определяем поле ввода и кнопку, прописываем обработчик. В методе onClick мы **создаем Intent**и **помещаем**в него **данные**из поля ввода под именем name. Обратите внимание, мы никак **не адресуем** этот Intent. Т.е. ни **класс**, ни **action**мы **не указываем**. И получается, что непонятно куда пойдет этот Intent. Но метод [setResult](http://developer.android.com/reference/android/app/Activity.html" \l "setResult(int,%20android.content.Intent)" \t "_blank) знает, куда его адресовать - в «**родительское**» **Activity**, в котором был вызван метод **startActivityForResult**. Также в **setResult**мы передаем константу **RESULT\_OK**, означающую **успешное завершение**вызова. И именно она передастся в параметр **resultCode**метода **onActivityResult**в MainActivity.java. Это мы подробнее разберем на следующем уроке. Далее методом **finish** мы завершаем работу NameActivity, чтобы результат ушел в MainActivity.

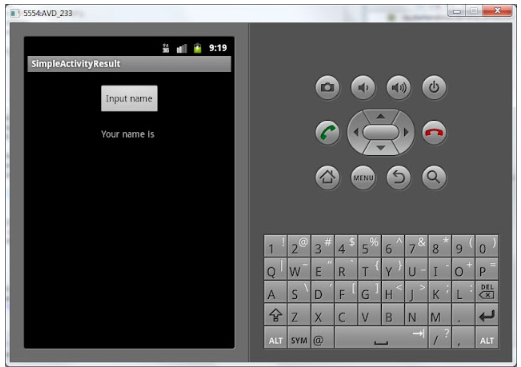
Все сохраним и запустим приложение.

Видим первый экран:

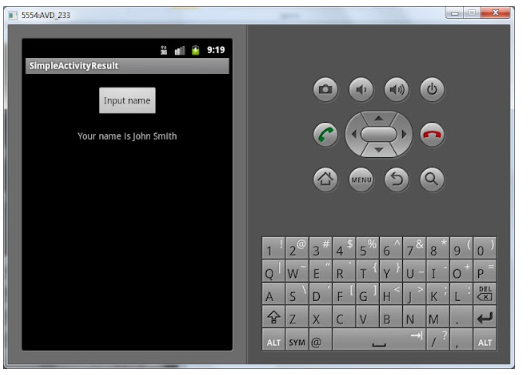


Жмем кнопку, чтобы попасть на экран ввода имени.

Вводим имя и жмем ОК



Снова первый экран, отобразивший полученные данные.



Попробуем подытожить. В **MainActivity**мы **создали Intent**с явным указанием на класс NameActivity. Запустили этот Intent с помощью метода **startActivityForResult**. NameActivity отобразилось, мы ввели имя и нажали кнопку. Создался **Intent**, в который **поместилось**введенное нами **имя**. Метод **setResult**знает, что Intent надо вернуть в **Activity**, которое выполнило вызов **startActivityForResult**, т.е. – MainActivity. В MainActivity за прием результатов с вызванных Activity отвечает метод **onActivityResult**. В нем мы распаковали Intent и отобразили полученные данные в TextView.

Пока необходимо просто понять схему вызова и возврата.

[Подробнее про onActivityResult. Зачем нужны requestCode и resultCode](https://startandroid.ru/ru/uroki/vse-uroki-spiskom/69-urok-30-podrobnee-pro-onactivityresult-zachem-nuzhny-requestcode-i-resultcode.html)

В этом уроке:

- разбираемся, зачем нужны requestCode и resultCode в onActivityResult

На прошлом уроке мы поверхностно рассмотрели, как вызвать **Activity**, и как сделать так, чтобы она **вернула результат**. Рассмотрим немного подробней этот механизм. Создадим приложение, которое будет вызывать два разных Activity и получать от них результат. Как мы помним, результат приходит в метод [onActivityResult](http://developer.android.com/reference/android/app/Activity.html" \l "onActivityResult(int,%20int,%20android.content.Intent)" \t "_blank). И **requestCode**используется, чтобы отличать друг от друга пришедшие результаты. А **resultCode**– позволяет определить успешно прошел вызов или нет.

Создадим проект:

**Project name**: P0301\_ActivityResult  
**Build Target**: Android 2.3.3  
**Application name**: ActivityResult  
**Package name**: ru.startandroid.develop.p0301activityresult  
**Create Activity**: MainActivity

Нарисуем экран в **main.xml**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | <?xml version="1.0" encoding="utf-8"?>  <LinearLayout      xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="fill\_parent"      android:layout\_height="fill\_parent"      android:orientation="vertical">      <TextView          android:id="@+id/tvText"          android:layout\_width="fill\_parent"          android:layout\_height="wrap\_content"          android:layout\_marginTop="20dp"          android:gravity="center\_horizontal"          android:text="Hello World"          android:textSize="20sp">      </TextView>      <LinearLayout          android:id="@+id/linearLayout1"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:layout\_margin="20dp"          android:orientation="horizontal">          <Button              android:id="@+id/btnColor"              android:layout\_width="match\_parent"              android:layout\_height="wrap\_content"              android:layout\_marginRight="5dp"              android:layout\_weight="1"              android:text="Color">          </Button>          <Button              android:id="@+id/btnAlign"              android:layout\_width="match\_parent"              android:layout\_height="wrap\_content"              android:layout\_marginLeft="5dp"              android:layout\_weight="1"              android:text="Alignment">          </Button>      </LinearLayout>  </LinearLayout> |

На экране **TextView** с текстом. И**две кнопки** для **выбора цвета**шрифта и **выравнивания**текста в TextView. Нажатие на кнопку будет **вызывать**Activity для выбора и получать обратно **результат**.

Давайте начнем кодить в **MainActivity.java**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | package ru.startandroid.develop.p0301activityresult;    import android.app.Activity;  import android.os.Bundle;  import android.view.View;  import android.view.View.OnClickListener;  import android.widget.Button;  import android.widget.TextView;    public class MainActivity extends Activity implements OnClickListener {      TextView tvText;    Button btnColor;    Button btnAlign;        /\*\* Called when the activity is first created. \*/      @Override      public void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.main);            tvText = (TextView) findViewById(R.id.tvText);            btnColor = (Button) findViewById(R.id.btnColor);          btnAlign = (Button) findViewById(R.id.btnAlign);            btnColor.setOnClickListener(this);          btnAlign.setOnClickListener(this);        }      @Override    public void onClick(View v) {      // TODO Auto-generated method stub      }  } |

Определили экранные элементы, прописали обработчик кнопкам и пока остановимся на этом.

Создадим два других Activity. Начнем с **Activity**для **выбора** **цвета**. Создадим layout-файл **color.xml**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | <?xml version="1.0" encoding="utf-8"?>  <LinearLayout      xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:orientation="horizontal">      <Button          android:id="@+id/btnRed"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:layout\_margin="5dp"          android:layout\_weight="1"          android:text="Red">      </Button>      <Button          android:id="@+id/btnGreen"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:layout\_margin="5dp"          android:layout\_weight="1"          android:text="Green">      </Button>      <Button          android:id="@+id/btnBlue"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:layout\_margin="5dp"          android:layout\_weight="1"          android:text="Blue">      </Button>  </LinearLayout> |

Создаем класс ColorActivity. **ColorActivity.java**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48 | package ru.startandroid.develop.p0301activityresult;    import android.app.Activity;  import android.content.Intent;  import android.graphics.Color;  import android.os.Bundle;  import android.view.View;  import android.view.View.OnClickListener;  import android.widget.Button;    public class ColorActivity extends Activity implements OnClickListener {      Button btnRed;    Button btnGreen;    Button btnBlue;      @Override    protected void onCreate(Bundle savedInstanceState) {      super.onCreate(savedInstanceState);      setContentView(R.layout.color);        btnRed = (Button) findViewById(R.id.btnRed);      btnGreen = (Button) findViewById(R.id.btnGreen);      btnBlue = (Button) findViewById(R.id.btnBlue);        btnRed.setOnClickListener(this);      btnGreen.setOnClickListener(this);      btnBlue.setOnClickListener(this);    }      @Override    public void onClick(View v) {      Intent intent = new Intent();      switch (v.getId()) {      case R.id.btnRed:        intent.putExtra("color", Color.RED);        break;      case R.id.btnGreen:        intent.putExtra("color", Color.GREEN);        break;      case R.id.btnBlue:        intent.putExtra("color", Color.BLUE);        break;      }      setResult(RESULT\_OK, intent);      finish();    }  } |

Как обычно определяем элементы, присваиваем обработчик кнопкам и реализуем onClick. В onClick мы создаем **Intent**, затем **определяем**, **кнопка**с каким цветом была нажата и помещаем в Intent объект с именем *color*и **значением**цвета. Ставим статус **RESULT\_OK**, указываем, что надо вернуть объект **intent** в качестве результата и **закрываем**Activity. Для значения цветов используем системные константы.

Аналогично создаем Activity для выбора выравнивания.

Layout-файл **align.xml**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | <?xml version="1.0" encoding="utf-8"?>  <LinearLayout      xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:orientation="horizontal">      <Button          android:id="@+id/btnLeft"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:layout\_margin="5dp"          android:layout\_weight="1"          android:text="Left">      </Button>      <Button          android:id="@+id/btnCenter"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:layout\_margin="5dp"          android:layout\_weight="1"          android:text="Center">      </Button>      <Button          android:id="@+id/btnRight"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:layout\_margin="5dp"          android:layout\_weight="1"          android:text="Right">      </Button>  </LinearLayout> |

**AlignActivity.java**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48 | package ru.startandroid.develop.p0301activityresult;    import android.app.Activity;  import android.content.Intent;  import android.os.Bundle;  import android.view.Gravity;  import android.view.View;  import android.view.View.OnClickListener;  import android.widget.Button;    public class AlignActivity extends Activity implements OnClickListener {      Button btnLeft;    Button btnCenter;    Button btnRight;      @Override    protected void onCreate(Bundle savedInstanceState) {      super.onCreate(savedInstanceState);      setContentView(R.layout.align);        btnLeft = (Button) findViewById(R.id.btnLeft);      btnCenter = (Button) findViewById(R.id.btnCenter);      btnRight = (Button) findViewById(R.id.btnRight);        btnLeft.setOnClickListener(this);      btnCenter.setOnClickListener(this);      btnRight.setOnClickListener(this);    }      @Override    public void onClick(View v) {      Intent intent = new Intent();      switch (v.getId()) {      case R.id.btnLeft:        intent.putExtra("alignment", Gravity.LEFT);        break;      case R.id.btnCenter:        intent.putExtra("alignment", Gravity.CENTER);        break;      case R.id.btnRight:        intent.putExtra("alignment", Gravity.RIGHT);        break;      }      setResult(RESULT\_OK, intent);      finish();    }  } |

Здесь все аналогично, как и в ColorActivity. Только работаем не с цветами, а с **выравниванием**. Не забудьте прописать оба Activity в манифесте.

Теперь можем завершить код в MainActivity.java. Добавим пару своих **констант**в класс для удобства:

|  |  |
| --- | --- |
| 1  2 | final int REQUEST\_CODE\_COLOR = 1;  final int REQUEST\_CODE\_ALIGN = 2; |

Эти константы далее будем использовать в качестве requestCode.

Допишем метод onClick:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | @Override  public void onClick(View v) {    Intent intent;    switch (v.getId()) {    case R.id.btnColor:      intent = new Intent(this, ColorActivity.class);      startActivityForResult(intent, REQUEST\_CODE\_COLOR);      break;    case R.id.btnAlign:      intent = new Intent(this, AlignActivity.class);      startActivityForResult(intent, REQUEST\_CODE\_ALIGN);      break;    }  } |

Мы определяем, какая **кнопка**была нажата и посылаем **Intent**с ожиданием возврата результата. Два вызова отличаются **классом**вызываемого Activity и параметром **requestCode**в методе **startActivityForResult**. При вызове **ColorActivity**используем константу **REQUEST\_CODE\_COLOR**, а при вызове **AlignActivity**- **REQUEST\_CODE\_ALIGN**. Эту константу мы обратно получим в методе обработки результата – **onActivityResult**, и по ней сможем определить из **какого**именно **Activity**пришел **результат**.

Давайте реализуем метод onActivityResult в MainActivity.java:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | @Override  protected void onActivityResult(int requestCode, int resultCode, Intent data) {    // запишем в лог значения requestCode и resultCode    Log.d("myLogs", "requestCode = " + requestCode + ", resultCode = " + resultCode);    // если пришло ОК    if (resultCode == RESULT\_OK) {      switch (requestCode) {      case REQUEST\_CODE\_COLOR:        int color = data.getIntExtra("color", Color.WHITE);        tvText.setTextColor(color);        break;      case REQUEST\_CODE\_ALIGN:        int align = data.getIntExtra("alignment", Gravity.LEFT);        tvText.setGravity(align);        break;      }    // если вернулось не ОК    } else {      Toast.makeText(this, "Wrong result", Toast.LENGTH\_SHORT).show();    }  } |

Для наглядности пишем в лог значения переменных.

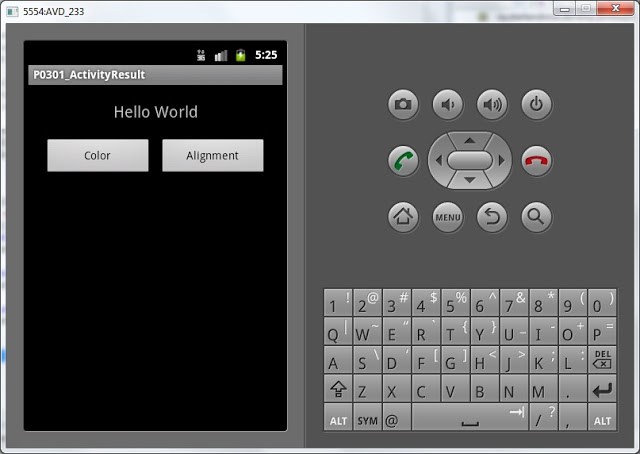
Вспоминаем, что в ColorActivity и AlignActivity в методе setResult мы ставили статус **RESULT\_OK**при отправке результата. Значит в **onActivityResult**нам надо ожидать этот статус, как обозначение успешного окончания вызова.

Если вызов прошел успешно (resultCode = RESULT\_OK), то мы смотрим значение **requestCode**. Если оно равно константе **REQUEST\_CODE\_COLOR**, то вспоминаем, что мы использовали эту константу в методе **startActivityForResult**, когда отправляли запрос на **выбор цвета**. Значит, нам пришел **результат**этого **выбора**. Мы берем **Intent**(data) и **извлекаем**из него значение объекта с именем *color*и присваиваем это значение **цвету**текста в TextView. Константа **Color.WHITE** в методе **getIntExtra**означает значение по умолчанию. Т.е. если в Intent не найдется объекта с именем *color*, то метод вернет белый (white) цвет.

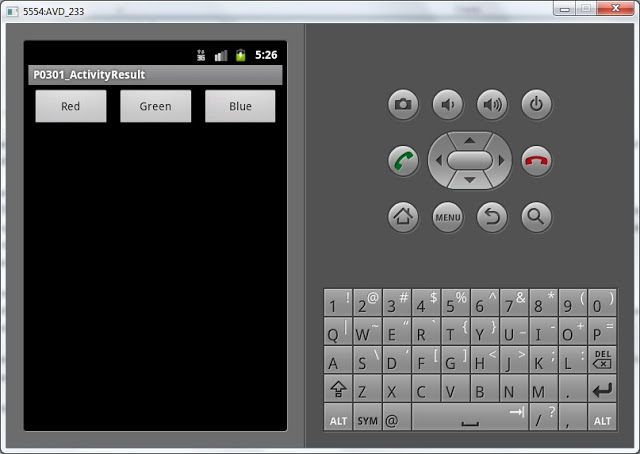
Аналогично для **REQUEST\_CODE\_ALIGN**. Эту константу мы использовали для запроса выбора **выравнивания**. И если в методе **onActivityResult**параметр **requestCode**= этой константе, значит пришел ответ на запрос выравнивания. И мы считываем это значение из Intent и присваиваем его атрибуту Gravity для TextView.

Если **resultCode**не равен **RESULT\_OK**, значит что-то пошло не так. Выводим на экран соответствующее сообщение. Этот случай может наступить, например, если на экране выбора не делать выбор, а нажать кнопку Назад.

Давайте все сохраним и запустим приложение.

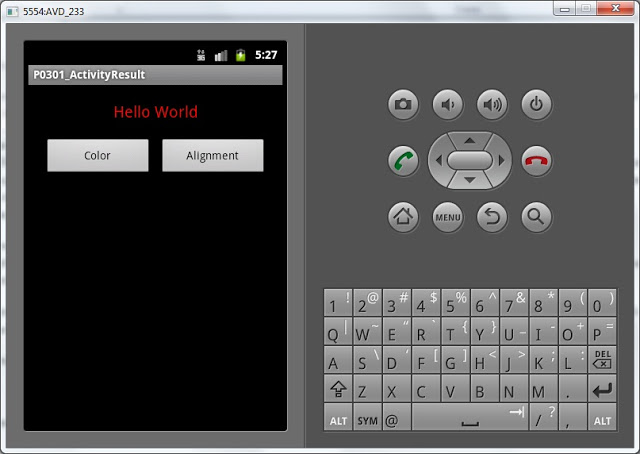


Нажмем Color



и выберем, например Red

Цвет изменился



смотрим лог:

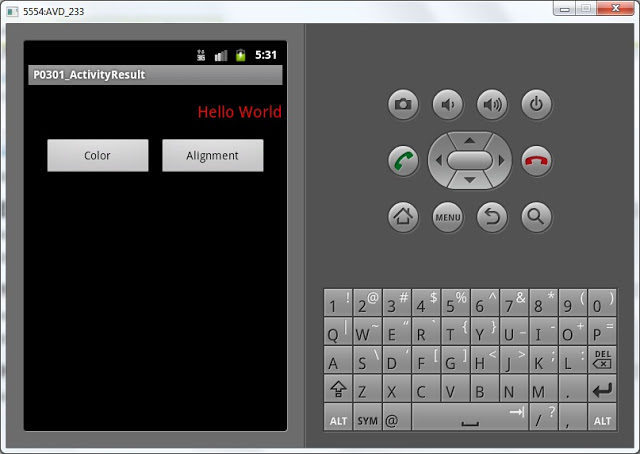
*requestCode = 1, resultCode = -1*

**requestCode**пришедший в метод **onActivityResult** равен 1. Все верно, это значение константы **REQUEST\_CODE\_COLOR**, которое мы использовали при вызове.

**resultCode**= -1 – это значение системной константы [RESULT\_OK](http://developer.android.com/reference/android/app/Activity.html#RESULT_OK)

Т.е. все верно, пришел ответ на запрос цвета, и его статус = **RESULT\_OK**.

Теперь жмем Alignment и выбираем Right, получаем выравнивание вправо:



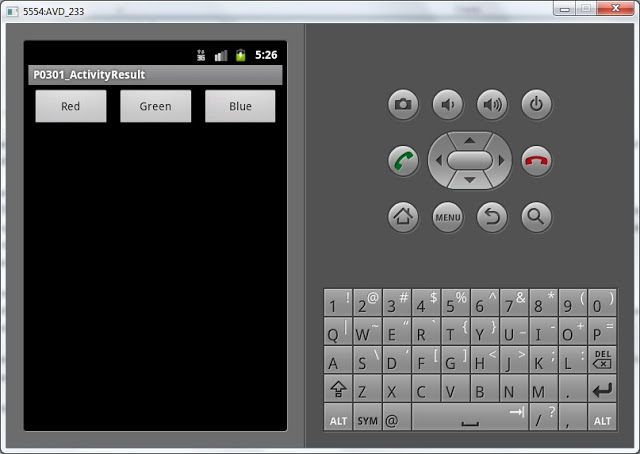
Смотрим лог:

*requestCode = 2, resultCode = -1*

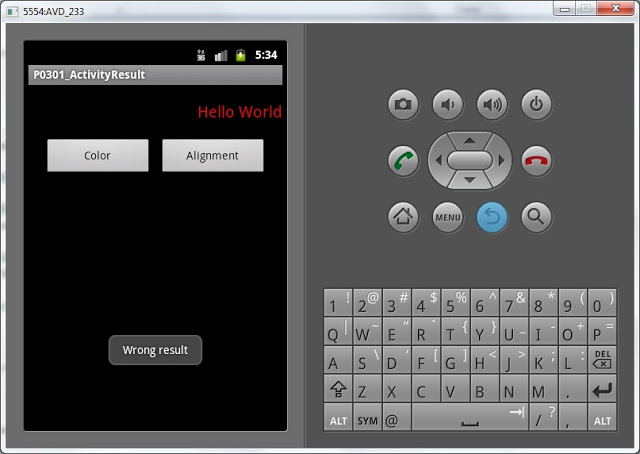
**requestCode**= 2, что равно константе **REQUEST\_CODE\_ALIGN**. Значит пришел ответ на запрос выравнивания.

**resultCode**= -1, т.е. RESULT\_OK.

Теперь снова жмем Color



но вместо того, чтобы выбрать цвет нажмем кнопку назад



Отобразилось наше сообщение об ошибке. Смотрим логи:

*requestCode = 1, resultCode = 0*

**requestCode**= 1 – все верно, мы запрашивали цвет (**REQUEST\_CODE\_COLOR**)

**resultCode**= 0, это значение константы [RESULT\_CANCELED](http://developer.android.com/reference/android/app/Activity.html#RESULT_CANCELED), значит вызов прошел неудачно

Ограничений на значение статуса в методе setResult нет. RESULT\_OK и RESULT\_CANCELED – системные общепринятые константы. Но вы можете свободно использовать свои значения, если в этом есть необходимость.

Итак, подведем итог.

**requestCode**– это в некотором роде ID запроса. Задается в методе **startActivityForResult**, и проверяется потом в **onActivityResult**, чтобы точно знать, на какой вызов пришел ответ.

**resultCode**– статус вызова. Задается в методе **setResult**, и проверяется в **onActivityResult**, чтобы понять насколько успешно прошел вызов. Если при вызове что-то пошло не так, то вернется системная константа **RESULT\_CANCELED**.

Полный код **MainActivity.java**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75 | package ru.startandroid.develop.p0301activityresult;    import android.app.Activity;  import android.content.Intent;  import android.graphics.Color;  import android.os.Bundle;  import android.util.Log;  import android.view.Gravity;  import android.view.View;  import android.view.View.OnClickListener;  import android.widget.Button;  import android.widget.TextView;  import android.widget.Toast;    public class MainActivity extends Activity implements OnClickListener {      final int REQUEST\_CODE\_COLOR = 1;    final int REQUEST\_CODE\_ALIGN = 2;      TextView tvText;    Button btnColor;    Button btnAlign;      /\*\* Called when the activity is first created. \*/    @Override    public void onCreate(Bundle savedInstanceState) {      super.onCreate(savedInstanceState);      setContentView(R.layout.main);        tvText = (TextView) findViewById(R.id.tvText);        btnColor = (Button) findViewById(R.id.btnColor);      btnAlign = (Button) findViewById(R.id.btnAlign);        btnColor.setOnClickListener(this);      btnAlign.setOnClickListener(this);    }      @Override    public void onClick(View v) {      Intent intent;      switch (v.getId()) {      case R.id.btnColor:        intent = new Intent(this, ColorActivity.class);        startActivityForResult(intent, REQUEST\_CODE\_COLOR);        break;      case R.id.btnAlign:        intent = new Intent(this, AlignActivity.class);        startActivityForResult(intent, REQUEST\_CODE\_ALIGN);        break;      }    }      @Override    protected void onActivityResult(int requestCode, int resultCode, Intent data) {      // запишем в лог значения requestCode и resultCode      Log.d("myLogs", "requestCode = " + requestCode + ", resultCode = " + resultCode);      // если пришло ОК      if (resultCode == RESULT\_OK) {        switch (requestCode) {        case REQUEST\_CODE\_COLOR:          int color = data.getIntExtra("color", Color.WHITE);          tvText.setTextColor(color);          break;        case REQUEST\_CODE\_ALIGN:          int align = data.getIntExtra("alignment", Gravity.LEFT);          tvText.setGravity(align);          break;        }        // если вернулось не ОК      } else {        Toast.makeText(this, "Wrong result", Toast.LENGTH\_SHORT).show();      }    }  } |