Передача Bitmap между активностями

2-й курс/Закрытая зона

Существует несколько способов передачи изображения между активностями.

Если ваше изображение берётся из ресурсов или из файла, то проблем нет. Вам нужно указать при передаче во вторую активность ссылку на идентификатор ресурса или путь к файлу. Но представьте себе, что вы что-то нарисовали на изображении и получили новый **Bitmap**. Тут есть несколько вариантов. По мере накопления материала будем добавлять примеры.

Начнём с первого примера. Можно создать класс со статичным объектом, который будет содержать ссылку на **Bitmap**. Далее мы передаём в класс нужное изображение, и вторая активность получит возможность получить из класса сохранённую картинку.

package ru.alexanderklimov.testapplication;

import android.graphics.Bitmap;

public class BitmapHolder {

public static Bitmap sBitmap = null;

}

Создадим разметку для первой активности. Первая кнопка загрузит изображение из галереи, вторая - передаст изображение во вторую активность.

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context=".MainActivity">

<Button

android:id="@+id/buttonLoadImage"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Загрузить картинку" />

<TextView

android:id="@+id/textViewSourceUri"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

<Button

android:id="@+id/buttonStartActivity"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Передать" />

<ImageView

android:id="@+id/imageView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" />

</LinearLayout>

Код для первой активности.

package ru.alexanderklimov.testapplication;

import ...

public class MainActivity extends ActionBarActivity {

final int RQS\_LOADIMAGE = 1;

final int RQS\_ACTIVITY = 2;

private TextView mSourceTextView;

private ImageView mImageView;

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mSourceTextView = (TextView)findViewById(R.id.textViewSourceUri);

Button loadButton = (Button)findViewById(R.id.buttonLoadImage);

Button startActivityButton = (Button)findViewById(R.id.buttonStartActivity);

mImageView = (ImageView)findViewById(R.id.imageView);

loadButton.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View view) {

Intent intent = new Intent(Intent.ACTION\_PICK,

android.provider.MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI);

startActivityForResult(intent, RQS\_LOADIMAGE);

}});

startActivityButton.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View view) {

Intent intent = new Intent(MainActivity.this, SecondActivity.class);

startActivityForResult(intent, RQS\_ACTIVITY);

}});

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if(resultCode == RESULT\_OK){

switch (requestCode){

case RQS\_LOADIMAGE:

Uri uri = data.getData();

mSourceTextView.setText(uri.toString());

try {

// сохраняем в объекте

BitmapHolder.sBitmap = BitmapFactory.decodeStream(

getContentResolver().openInputStream(uri));

} catch (FileNotFoundException e) {

e.printStackTrace();

}

break;

case RQS\_ACTIVITY:

mImageView.setImageBitmap(BitmapHolder.sBitmap);

break;

}

}

}

}

Во второй активности разместите **ImageView** и код, который загрузит картинку.

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_second);

ImageView secondImageView = (ImageView)findViewById(R.id.imageView2);

secondImageView.setImageBitmap(BitmapHolder.sBitmap);

}

Пример является демонстрационным. В нём нет проверок, он не подходит для больших картинок.

Также есть вариант передачи через **Parcelable**, но он ничем не лучше этого примера.

Класс **Bitmap** поддерживаете интерфейс **Parcelable** и мы можем передать объект через намерение.

Intent intent = new Intent(this, NewActivity.class);

intent.putExtra("BitmapImage", bitmap);

Принимающая сторона получает объект.

Intent intent = getIntent();

Bitmap bitmap = (Bitmap) intent.getParcelableExtra("BitmapImage");

Существует лимит на передачу в 1 мб.

Также есть вариант передачи через массив. Изображение конвертируется в массив байтов, а затем передаётся:

ByteArrayOutputStream stream = new ByteArrayOutputStream();

bitmap.compress(Bitmap.CompressFormat.PNG, 100, stream);

byte[] byteArray = stream.toByteArray();

Intent intent = new Intent(this, SecondActivity.class);

intent.putExtra("image", byteArray);

startActivity(intent);

Вторая активность принимает.

Bitmap bitmap;

byte[] byteArray = getIntent().getByteArrayExtra("image");

bitmap = BitmapFactory.decodeByteArray(byteArray, 0, byteArray.length);

У **putExtra()** есть ограничение, о котом я говорил выше, и слишком большой объём он передать не может. Опять нужно следить за размером данных.

Самым оптимальным, вероятно, будет такой вариант, когда мы сохраняем картинку на диске в виде файла (внутренний или внешний накопитель) и передаём ссылку на него.

При небольших картинках можно использовать любой вариант.