## Фильтры намерений

### Цель работы

Научиться использовать неявный вызов активности и создавать свои фильтры намерений (intent filters) для запуска независимых компонентов приложения.

### Контрольные вопросы

1. Какие существуют соглашения в порядке наименования действий?

Некоторые стандартные действия определены в классе Intent как константы ACTION\_string, где string — название действия. Указывать такие действие нужно так: android.Intent.ACTION\_string. Например, sndroid.Intent.ACTION\_VIEW. Для действий, которые определяет разработчик, лучше всего использовать имя пакета приложения в качестве префикса для обеспечения уникальности. Например, com.example.project.CUSTOM\_ACTION, где CUSTOM\_ACTION — имя действия.

1. Как передать информацию в активность используя неявный вызов?

Предать информацию в активность используя неявный вызов можно, указав ее вторым аргументом. Например, чтобы создать неявный интент с действием открытия ссылки в браузере, нужно указать адрес (объект Uri) вторым аргументом:

Intent openLinkIntent = new Intent(Intent.ACTION\_VIEW, address);

1. Какие еще параметры можно задавать при создании неявного интента?

Можно указать пары ключ-значения для дополнительной информации так же, как это делается при вызове явного интента с помощью ряда методов Intent.put...(). Например, чтобы сохранить фотографию, сделанную пользователем из нашего приложения, в файл, нужно указать ключ MediaStore.EXTRA\_OUTPUT и Uri, сооответсвующее файлу:

Intent takePictureIntent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);

takePictureIntent.putExtra(MediaStore.EXTRA\_OUTPUT, photoURI);

1. Зачем нужна категория в интент-фильтрах? Какие существуют категории?

Категории нужны, чтобы указать, при каких обстоятельствах действие должно обслуживаться. Можно задать собственные категории или же брать стандартные значения, предоставляемые системой:

* ALTERNATIVE — действие должно быть доступно в качестве альтернативного тому, которое выполняется по умолчанию для элемента этого типа данных.
* BROWSABLE — говорит о том, что действие доступно из браузера.
* DEFAULT — категория, позволяющая сделать компонент обработчиком по умолчанию для действия, выполняемого с указанным типом.
* GADGET — наличие этой категории указывает на то, что данная активность может запускаться внутри другой активности.
* LAUNCHER — используя эту категорию, мы помещаем Activity в окно для запуска приложений.
* и др.

1. Зачем нужен элемент <requestFocus>?

Элемент <requestFocus> позволяет установить фокус на нужном компоненте. Предположим, что у нас имеется три текстовых поля, и нужно, чтобы фокус был на втором поле. В этом случае достаточно добавить тег <requestFocus> внутри второго элемента EditText.

1. Зачем нужны аргументы requestCode и resultCode в обратном интенте?

requestCode - это неотрицательное целое число, которое запоминается (обычно записывается в поле класса активности, вызывающей другое действие, как константа) и передается в функцию StartActivityForResult() вторым аргументом. Это число позволяет при завершении одной из активности в методе onActivityResult() различить по этому числу, какая именно Activity была завершена. А resultCode позволяет определить успешно прошел вызов или нет.

### 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.

AndoidManifest.xml

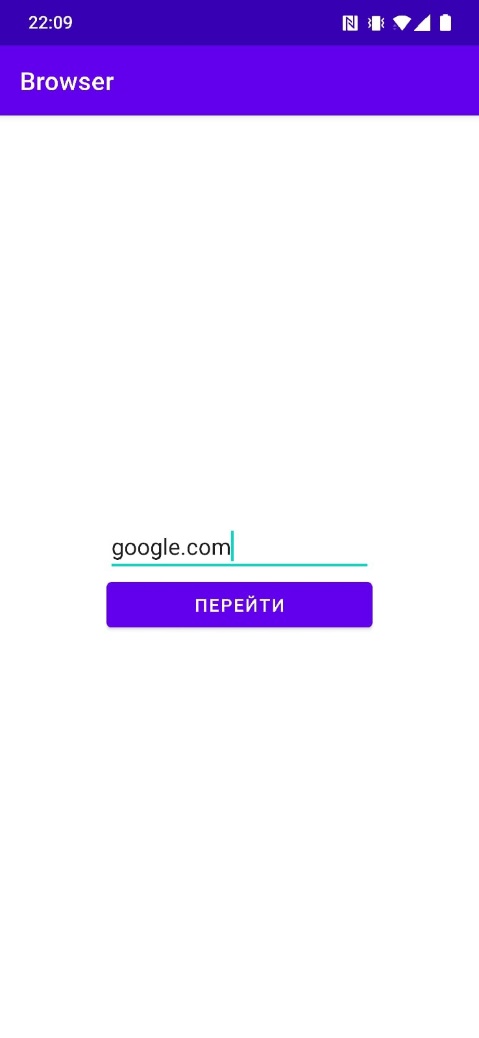
*<?*xml version="1.0" encoding="utf-8"*?>*<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.browser">  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Browser">  
 <activity  
 android:name=".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>

MainActivity.xml

package com.example.browser;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
  
public class MainActivity extends AppCompatActivity {  
  
 EditText editText;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 editText = findViewById(R.id.editText);  
 }  
  
 public void onClick(View v) {  
 String url = editText.getText().toString();  
 if (!url.startsWith("http://") && !url.startsWith("https://"))  
 url = "http://" + url;  
 Intent browserIntent = new Intent(Intent.ACTION\_VIEW, Uri.parse(url));  
 startActivity(browserIntent);  
 }  
}

activity\_main.xml

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <EditText  
 android:id="@+id/editText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="text" />  
  
 <Button  
 android:id="@+id/button2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Перейти"  
 android:onClick="onClick"/>  
 </LinearLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>



Изображение выглядит как текст

Автоматически созданное описание

1. Создайте приложение, отвечающее на какое-либо стандартное системное действие. Проверьте его работоспособность.

AndroidManifest.xml

*<?*xml version="1.0" encoding="utf-8"*?>*<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.bluetoothenabler">  
  
 <uses-permission android:name="android.permission.BLUETOOTH\_CONNECT" />  
 <uses-permission android:name="android.permission.BLUETOOTH" />  
 <uses-permission android:name="android.permission.BLUETOOTH\_ADMIN" />  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.BluetoothEnabler">  
 <activity  
 android:name=".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>

MainActivity.java

package com.example.bluetoothenabler;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.bluetooth.BluetoothAdapter;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 findViewById(R.id.*button*).setOnClickListener(  
 new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent enableBtIntent = new Intent(BluetoothAdapter.*ACTION\_REQUEST\_ENABLE*);  
 startActivity(enableBtIntent);  
 }  
 }  
 );  
 }  
}

activity\_main.xml

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Включить Bluetooth"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>\

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

1. 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.

AndroidManifest.xml

*<?*xml version="1.0" encoding="utf-8"*?>*<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.colorandalignment">  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.ColorAndAlignment">  
 <activity  
 android:name=".AlignmentActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="com.example.intent.action.alignment" />  
  
 <category android:name="android.intent.category.DEFAULT" />  
 </intent-filter>  
 </activity>  
 <activity  
 android:name=".ColorActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="com.example.intent.action.color" />  
  
 <category android:name="android.intent.category.DEFAULT" />  
 </intent-filter>  
 </activity>  
 <activity  
 android:name=".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>

MainActivity.java

package com.example.colorandalignment;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.Gravity;  
import android.view.View;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity {  
  
 TextView textView;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 textView = findViewById(R.id.*textView*);  
 }  
  
 public void onClick(View v) {  
 Intent intent;  
  
 switch(v.getId()) {  
 case R.id.*colorButton*:  
 intent = new Intent("com.example.intent.action.color");  
 startActivityForResult(intent, 1);  
 break;  
 case R.id.*alignmentButton*:  
 intent = new Intent("com.example.intent.action.alignment");  
 startActivityForResult(intent, 1);  
 break;  
 }  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent intent) {  
 super.onActivityResult(requestCode, resultCode, intent);  
 if (intent == null) {  
 return;  
 }  
 int color = intent.getIntExtra("color", 0);  
 int alignment = intent.getIntExtra("alignment", 0);  
 if (color != 0) {  
 textView.setTextColor(color);  
 }  
 if (alignment != 0) {  
 textView.setGravity(alignment);  
 }  
 }  
  
}

ColorActivity.java

package com.example.colorandalignment;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.graphics.Color;  
import android.os.Bundle;  
import android.view.View;  
  
public class ColorActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_color*);  
 }  
  
 public void onClick(View v) {  
 Intent intent = new Intent();  
 String attr = "color";  
  
 switch(v.getId()) {  
 case R.id.*redButton*:  
 intent.putExtra(attr, Color.*RED*);  
 break;  
 case R.id.*greenButton*:  
 intent.putExtra(attr, Color.*GREEN*);  
 break;  
 case R.id.*blueButton*:  
 intent.putExtra(attr, Color.*BLUE*);  
 break;  
 }  
 setResult(*RESULT\_OK*, intent);  
 finish();  
 }  
}

AlignmentActivity.java

package com.example.colorandalignment;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.graphics.Color;  
import android.os.Bundle;  
import android.view.Gravity;  
import android.view.View;  
  
public class AlignmentActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_alignment*);  
 }  
  
 public void onClick(View v) {  
 Intent intent = new Intent();  
 String attr = "alignment";  
  
 switch(v.getId()) {  
 case R.id.*leftButton*:  
 intent.putExtra(attr, Gravity.*START*);  
 break;  
 case R.id.*rightButton*:  
 intent.putExtra(attr, Gravity.*END*);  
 break;  
 case R.id.*centerButton*:  
 intent.putExtra(attr, Gravity.*CENTER*);  
 break;  
 }  
 setResult(*RESULT\_OK*, intent);  
 finish();  
 }  
}

activity\_main.xml

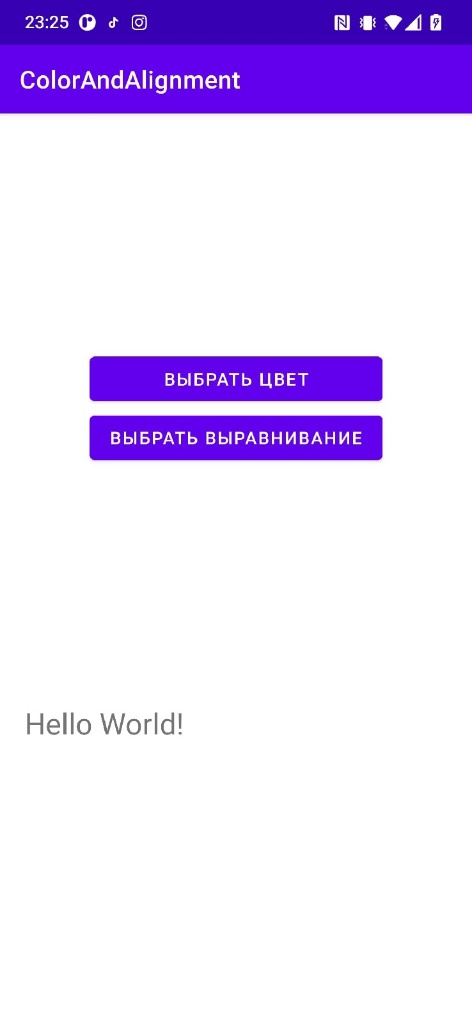
*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 app:layout\_constraintBottom\_toTopOf="@+id/textView"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <Button  
 android:id="@+id/colorButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Выбрать цвет"  
 android:onClick="onClick" />  
  
 <Button  
 android:id="@+id/alignmentButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Выбрать выравнивание"  
 android:onClick="onClick" />  
 </LinearLayout>  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.691"  
 android:layout\_margin="20dp" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_color.xml

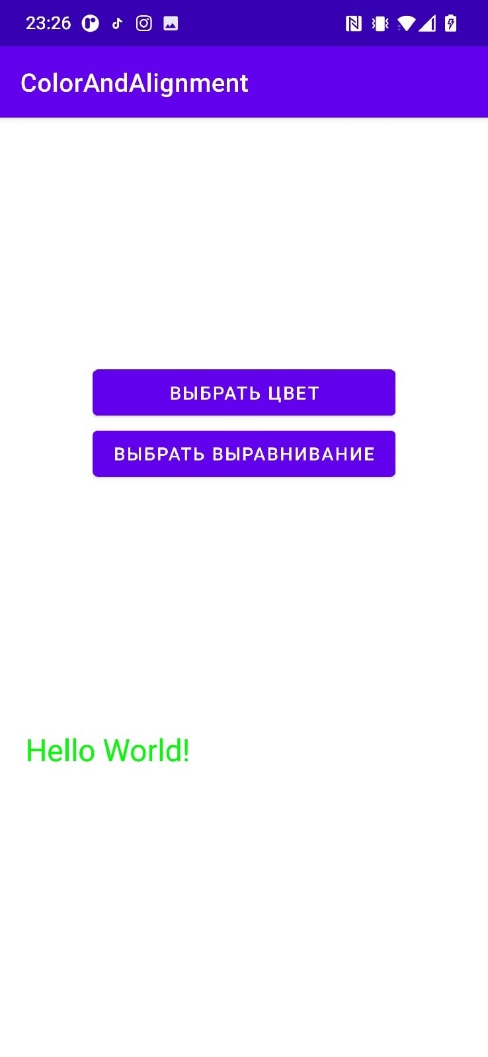
*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ColorActivity">  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <Button  
 android:id="@+id/redButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Красный"  
 android:onClick="onClick" />  
  
  
 <Button  
 android:id="@+id/greenButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Зеленый"  
 android:onClick="onClick" />  
  
 <Button  
 android:id="@+id/blueButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Синий"  
 android:onClick="onClick"/>  
  
 </LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_alignment.xml

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".AlignmentActivity">  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <Button  
 android:id="@+id/leftButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="По левому краю"  
 android:onClick="onClick" />  
  
 <Button  
 android:id="@+id/centerButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="По центру"  
 android:onClick="onClick" />  
  
 <Button  
 android:id="@+id/rightButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="По правому краю"  
 android:onClick="onClick" />  
 </LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>







Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

1. (\*) 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.

AndoroidManifest.xml

*<?*xml version="1.0" encoding="utf-8"*?>*<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.thumbnail">  
  
 <uses-feature android:name="android.hardware.camera"  
 android:required="true" />  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Thumbnail">  
 <activity  
 android:name=".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>

MainActivity.xml

package com.example.thumbnail;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.os.Bundle;  
import android.provider.MediaStore;  
import android.view.View;  
import android.widget.ImageView;  
  
public class MainActivity extends AppCompatActivity {  
  
 private ImageView imageView;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 imageView = findViewById(R.id.*imageView*);  
 }  
  
 public void onClick(View view) {  
 Intent takePictureIntent = new Intent(MediaStore.*ACTION\_IMAGE\_CAPTURE*);  
 startActivityForResult(takePictureIntent, 0);  
 }  
  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (resultCode == *RESULT\_OK*) {  
 Bundle extras = data.getExtras();  
 Bitmap imageBitmap = (Bitmap) extras.get("data");  
 imageView.setImageBitmap(imageBitmap);  
 }  
 }  
}

activity\_main.xml

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:srcCompat="@tools:sample/avatars" />  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="Сделать фото"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView"  
 android:onClick="onClick" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

