Recycleview

Необходимо сделать прокручиваемый список объектов.

Под объектами понимаем рамка, в которой будет распологаться image и name (наименование страны в данном случае) (нарисовать картинку с планом).

На данный момент при запуске приложении на главном экране (activity\_main) будет располагаться кнопка,

при нажатии на которую будет открываться другая активити, в котороу будет отображен прокручиваемый список объектов.

1. Создаем класс ObjectCountry, который будет сохранять 2 переменный: ссылка на картинку и имя страны.
2. Создаем класс **Countries2**, который и будет запускаться при нажатии на кнопку из главной активити.
3. К этому классу создаем layout: **activity\_countries2**, в котором будет только лишь <androidx.recyclerview.widget.RecyclerView>, то есть в ней и будет прокрутка другой активити, которая и будет строить наш необходимый прямоугольник с картинкой и названием страны.
4. Создаем **activity\_list\_countries2**, это как раз активити, которая отображает нужные данные. В нем будет такая структура примерно:

<LianerLayout> - основная обертка

<ImageView> - картинка

<LeanerLayout> - второстеменная обертка

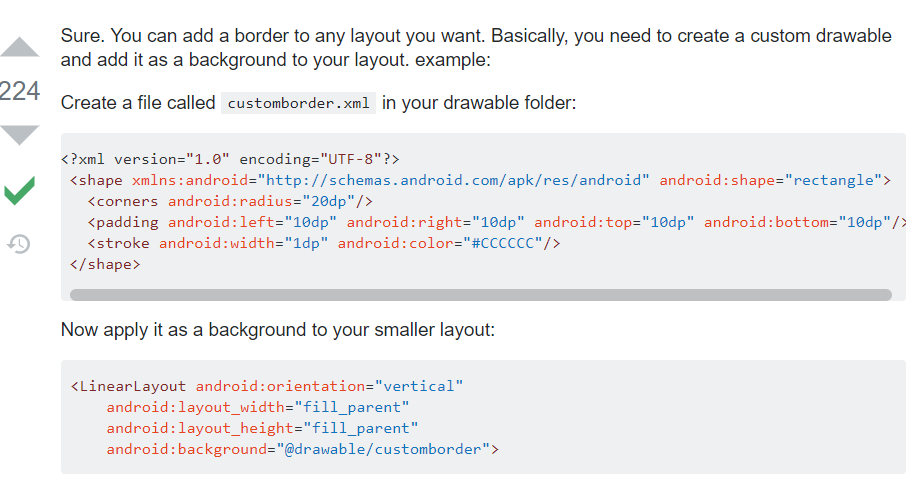
<TextView> - название страны

1. Создаем новый адаптер DataAdapter2

**Как сделать border к LinearLayout**

В папке **res/drawable** необходимо создать файл .xml и назовем его border. Он будет содержать рисунок рамки, который потом подключается к LinearLayout.

Ответ найден с stackoverflow: <https://stackoverflow.com/questions/15111402/how-can-i-create-a-border-around-an-android-linearlayout>



**Делаем переход на новую активити при нажатии на какой-либо объект**

Прям вот то что нужно: <https://codingwithmitch.com/blog/android-recyclerview-onclicklistener/>

Чтобы повесить событие на каждый элемент Recyclerview, то нам необходимо:

* В лайауте **activity\_list\_countries** на родительском LenearLayout проставить id 
* Далее все остальное делаем в классе адаптера **DataAdapter.**

- в методе ViewHolder обьявляем еще одну переменную, которая относится как раз к LinearLayout из шага выше 

- в конструкторе инициализируем его и приводим к LinearLayout 

* Далее идем в метод **onBindViewHolder**. К новой переменной пишем



Это пока что только лишь выводит в консоль “dfer2222”

**Нам необходимо при нажатии на любой item перейти на новую активити с информацией по нажатому item’му.**

1. Создаем новый layout и называем его AudioList. Делаем в нем TextView, который будет отображать имя страны, по которой прошел пользователь.

public class AudioList extends AppCompatActivity {  
  
 TextView t1, t2;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_audio\_list*);  
  
 t1 = (TextView) findViewById(R.id.*textView*);  
 t2 = (TextView) findViewById(R.id.*textView2*);  
  
 Bundle passedValue = getIntent().getExtras(); // получаем переданные значения в переменную типа Bundle  
 String nameCountry = passedValue.get("nameCountry").toString(); // По ключу "nameCountry" получаем значение  
  
 t2.setText(nameCountry);  
  
 }  
}

t2 – переменная типа TextView, которая будет отображать имя страны.

t2 = (TextView) findViewById(R.id.textView2); - этим действием мы инициализируем переменную.

Объявляем объект Bundle c переменной **passedValue**, которая принимает переданный объект.

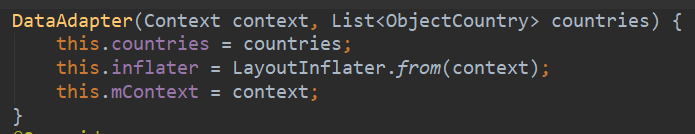
**Bundle passedValue = getIntent().getExtras();** // получаем переданные значения в переменную типа Bundle

**String nameCountry = passedValue.get("nameCountry").toString();** // По ключу "nameCountry" получаем значение

t2.setText(nameCountry); // устанавливаем значение необходимой страной

1. В классе DataAdapter первое, что мы сделаем, так это объявим новую перменную класса .

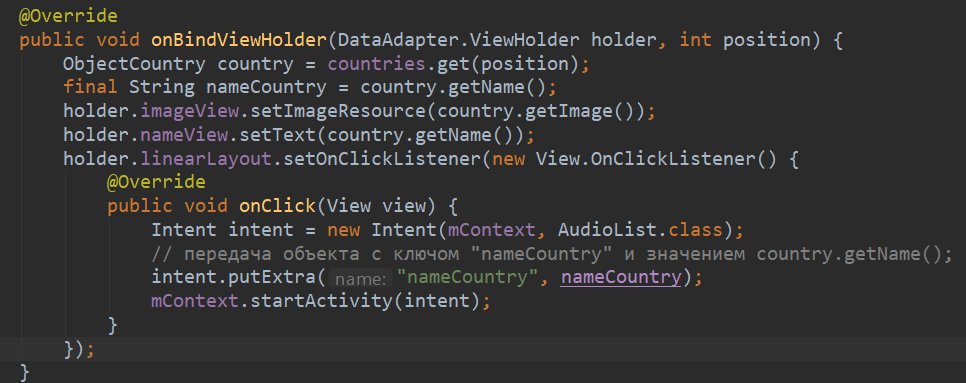
В конструкторе инициализируем ее.



Эта переменная необходима, для того чтобы брать значение текущего контекта в внутреннем классе, когда будем названачать на каждый итем

событие.

Далее, немного переделаем метод onClick, который выводил нам в консоль «dfer2222» таким образом:



Тут нужно обратить внимание, что мы добавили в методе новую переменную **final String nameCountry**, в ней мы сохраняем поле name

класса ObjectCountry, чтобы без проблем передать в метод onClick внутреннего (или ананимного? изучить) класса, а именно,

в строку intent.putExtra("nameCountry", nameCountry);

Кстати, вот в этом методе onClick нам и требуется новая добавленная отдельная переменная **mContext**, так как **this** тут не работает.