

## Recycle View

Выполнили: Бурова Екатерина

Кахнович Мария

### activity\_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/main"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9
10    <TextView
11        android:layout_width="wrap_content"
12        android:layout_height="wrap_content"
13        app:layout_constraintBottom_toBottomOf="parent"
14        app:layout_constraintEnd_toEndOf="parent"
15        app:layout_constraintStart_toStartOf="parent"
16        app:layout_constraintTop_toTopOf="parent" />
17
18    <androidx.recyclerview.widget.RecyclerView
19        android:id="@+id/recyclerView"
20        android:layout_width="match_parent"
21        android:layout_height="match_parent"/>
22
23 </androidx.constraintlayout.widget.ConstraintLayout>
```

### MainActivity

```
1 package com.example.lb8
2
3 import android.os.Bundle
4 import android.view.LayoutInflater
5 import android.view.View
6 import android.view.ViewGroup
7 import androidx.activity.enableEdgeToEdge
8 import androidx.appcompat.app.AppCompatActivity
9 import androidx.core.view.ViewCompat
10 import androidx.core.view.WindowInsetsCompat
11 import androidx.recyclerview.widget.RecyclerView
12 import androidx.recyclerview.widget.LinearLayoutManager
13 import com.google.android.material.imageview.ShapeableImageView
14 import android.widget.TextView
15
16 class MainActivity : AppCompatActivity() {
17     private lateinit var recyclerView: RecyclerView
18     private lateinit var arrayList: ArrayList<DataClass>
19     private lateinit var imageId: Array<Int>
20     private lateinit var textId: Array<String>
21     private lateinit var textBonkId: Array<String>
22
23     override fun onCreate(savedInstanceState: Bundle?) {
24         super.onCreate(savedInstanceState)
25
26         enableEdgeToEdge()
27         setContentView(R.layout.activity_main)
28
29         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
30             val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
31             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
32             insets
33         }
```

```

33     }
34
35     // Инициализация массива с ресурсами изображений
36     imageId = arrayOf(
37         R.drawable.cta,
38         R.drawable.cta,
39         R.drawable.cta,
40         R.drawable.cta,
41         R.drawable.cta,
42         R.drawable.cta,
43         R.drawable.cta,
44         R.drawable.cta,
45         R.drawable.cta,
46         R.drawable.cta
47     )
48
49     // Получение строковых ресурсов и сохранение их в массивах
50     textId = resources.getStringArray(R.array.porodi_cat)
51     textBonkId = resources.getStringArray(R.array.opisanie_cat)
52
53     // Создание списка данных для адаптера
54     newArrayList = ArrayList()
55     for (i in imageId.indices) {
56         newArrayList.add(DataClass(imageId[i], textId[i], textBonkId[i]))
57     }
58
59     // Инициализация и установка адаптера RecyclerView
60     newRecyclerView = findViewById(R.id.recyclerView) // Убедитесь, что ID соответствует вашему RecyclerView в layout
61     val adapter = MyAdapter(newArrayList)

```

```

val adapter = MyAdapter(newArrayList)
newRecyclerView.adapter = adapter
newRecyclerView = findViewById(R.id.recyclerView)
// Установка менеджера компоновки для RecyclerView
newRecyclerView.layoutManager = LinearLayoutManager(this)

newRecyclerView.setHasFixedSize(true)
newArrayList = arrayOf<DataClass>()
getUserdata()

}

private fun getUserdata() {
    // Проход по индексам изображений
    for (i in imageId.indices)
    {
        // Создание нового экземпляра DataClass для каждого элемента
        val data = DataClass(imageId[i],textId[i],textBonkId[i])
        // Добавление созданного экземпляра в список newArrayList
        newArrayList.add(data)
    }
    // Установка адаптера для RecyclerView с заполненным списком данных
    newRecyclerView.adapter = MyAdapter(newArrayList)
}

}

data class DataClass(var titleImage: Int, var heading: String, var bonktxt: String)

```

```

<? activity_main.xml <? list_item.xml <? MainActivity.kt x <? strings.xml
90 class MyAdapter(private val newList: ArrayList<DataClass>) : RecyclerView.Adapter<MyAdapter.MyViewHolder>() {
91     // Метод, вызываемый для создания нового ViewHolder
92     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyViewHolder {
93         // Создаем View для элемента списка, используя разметку list_item
94         val itemView = LayoutInflater.from(parent.context).inflate(R.layout.list_item, parent, false)
95         // Возвращаем новый экземпляр MyViewHolder с созданным View
96         return MyViewHolder(itemView)
97     }
98
99     // Метод для получения общего количества элементов в списке
100    override fun getItemCount(): Int {
101        return newList.size
102    }
103
104    // Метод для привязки данных к Views внутри ViewHolder
105    override fun onBindViewHolder(holder: MyViewHolder, position: Int) {
106        // Получаем текущий элемент списка по позиции
107        val currentItem = newList[position]
108        // Устанавливаем изображение для ImageView из ресурсов по id и 2 TextView
109        holder.imageView.setImageResource(currentItem.titleImage)
110        holder.txtHead.text = currentItem.heading
111        holder.txtBonk.text = currentItem.bonktxt
112    }
113
114    // Вложенный класс ViewHolder для управления представлением элемента
115    class MyViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
116        // Инициализируем Views для элемента списка
117        val imageView: ShapeableImageView = itemView.findViewById(R.id.imageItem)
118        val txtHead: TextView = itemView.findViewById(R.id.txtItem)
119        val txtBonk: TextView = itemView.findViewById(R.id.txtBonk)
120    }
121
122

```

## list\_item.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="wrap_content"
    android:padding="8dp">
    <com.google.android.material.imageview.ShapeableImageView
        android:id="@+id/imageItem"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@drawable/cta"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <TextView
        android:id="@+id/txtItem"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Cat u r so funny ."
        android:layout_marginStart="20dp"
        android:textColor="@color/black"
        android:textSize="16sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/imageItem"
        app:layout_constraintTop_toTopOf="parent" />
    <TextView
        android:id="@+id/txtBonk"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/txtItem"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

## Strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">lb8</string>
    <string-array name="porodi_cat">
        <item>Кот</item>
        <item>Котик</item>
        <item>Кошка</item>
        <item>Котище</item>
        <item>Котёнок</item>
        <item>Котэ</item>
        <item>Котёныш</item>
        <item>Котя</item>
        <item>Котяндра</item>
        <item>Кошак</item>
    </string-array>
    <string-array name="opisanie_cat">
        <item>Расстроен</item>
        <item>Удручён</item>
        <item>В депрессии</item>
        <item>В апатии</item>
        <item>Разочарован</item>
        <item>Недоволен</item>
        <item>В меланхолии</item>
        <item>Грустный</item>
        <item>Угрюмый</item>
        <item>Интроверт</item>
    </string-array>
</resources>
```

## Запускаем

