

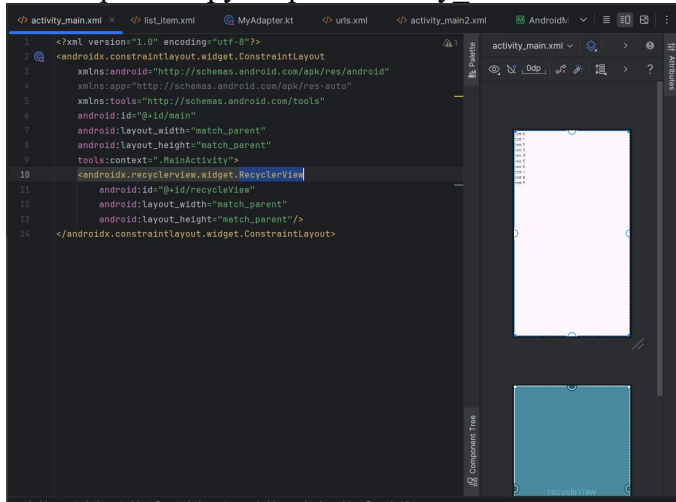
Практика: Практика 8. Практическая работа Recycle View

Кто работал: Наглов Виктор Ис221

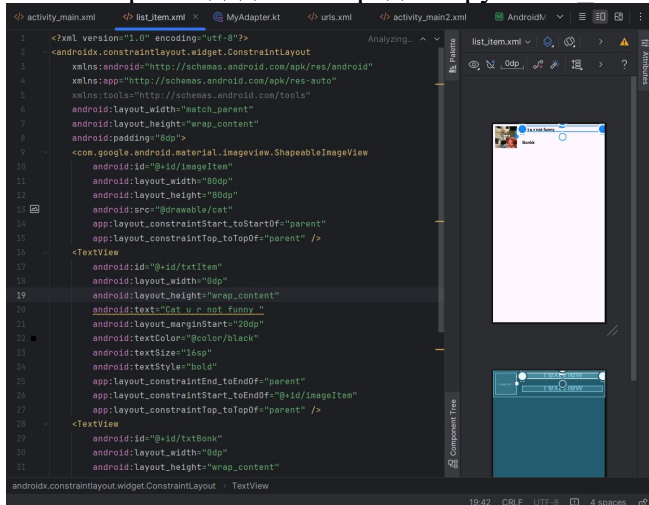
Ход работы:

1. Открываем документ и приступаем к выполнению.

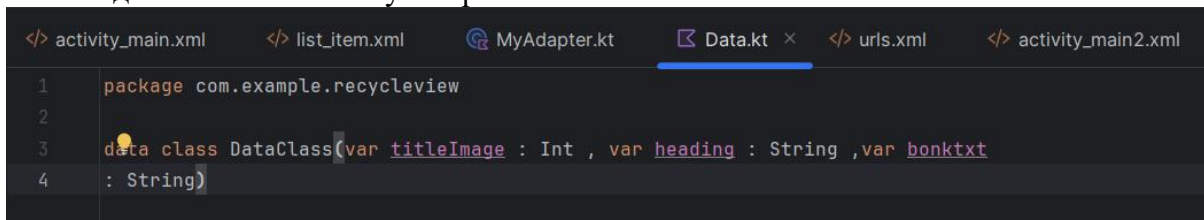
2. Отредактируем файл activity_main.xml



3. Теперь создадим и отредактируем list_item.xml



4. Создаем Data Class и My Adapter:



```

12 class MyAdapter(private val newList : ArrayList<DataClass>) : RecyclerView.Adapter<MyAdapter.MyViewHolder>() {
13     override fun onBindViewHolder(holder: MyViewHolder, position: Int) {
14         holder.itemView.setOnClickListener {
15             val url_conf = holder.itemView.context.getString(R.string.url1)
16             val intent = Intent(holder.itemView.context, MainActivity2::class.java)
17             intent.putExtra(name="url", url_conf)
18             holder.itemView.context.startActivity(intent)
19         }
20     } else if (position == 0) {
21         // Показываем Toast сообщение
22         Toast.makeText(holder.itemView.context, text: "Мы нашли на 9-м элемент!",
23             Toast.LENGTH_SHORT).show()
24         val url_conf = holder.itemView.context.getString(R.string.url2)
25         val intent = Intent(holder.itemView.context, MainActivity2::class.java)
26         intent.putExtra(name="url", url_conf)
27         holder.itemView.context.startActivity(intent)
28     }
29 }
30 // Вспомогательный класс ViewHolder для управления представлением элемента
31 class MyViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
32     // Инициализируем Views для элемента списка
33     val imageView : ShoppableImageView =
34         itemView.findViewById(R.id.imageView)
35     val txtHead : TextView = itemView.findViewById(R.id.txtItem)
36     val txtBook : TextView = itemView.findViewById(R.id.txtBook)
37 }
38 }
39 }
40 }

```

5. Редактируем MainActivity

```

11 class MainActivity : AppCompatActivity() {
12     // Объявляем переменные для RecyclerView и списков данных
13     // lateinit var говорит о том, что переменная будет инициализирована позже.
14     // При этом инициализировать свойство можно во любом месте, главное оно будет.
15     private lateinit var newRecyclerView : RecyclerView
16     private lateinit var newList : ArrayList<DataClass>
17     private lateinit var imageId : Array<Int>
18     private lateinit var textId : Array<String>
19     private lateinit var textBookId : Array<String>
20
21     override fun onCreate(savedInstanceState: Bundle?) {
22         super.onCreate(savedInstanceState)
23         enableEdgeToEdge()
24         setContentView(R.layout.activity_main)
25         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
26             val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
27             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
28             insets.setOnApplyWindowInsetsListener
29         }
30
31         // Инициализация массива с ресурсами изображений
32         imageId = arrayOf(
33             R.drawable.cat,
34             R.drawable.moj_kun,
35             R.drawable.scotland,
36             R.drawable.britanec_2lai,
37             R.drawable.siamski,
38             R.drawable.abbis,
39             R.drawable.russ_blue,
40             R.drawable.oriental,
41             R.drawable.sphynx,

```

6. Добавляем переменные в strings.xml

```

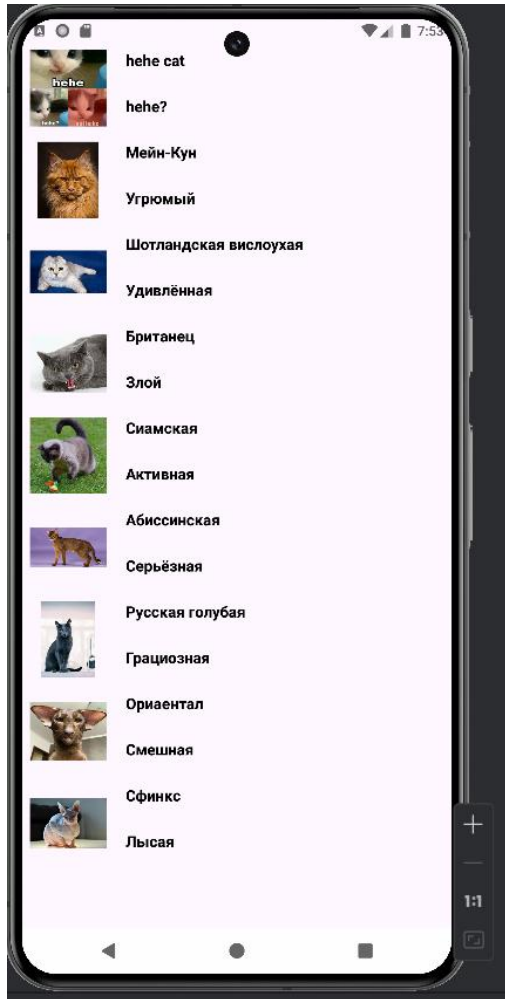
1 <resources>
2 <string name="app_name">Recycle View</string>
3 <string-array name="porodi_cat">
4     <item>hehe cat</item>
5     <item>Мейн-Кун</item>
6     <item>Шотландская вислоухая</item>
7     <item>Британец</item>
8     <item>Сиамская</item>
9     <item>Абиссинская</item>
10    <item>Русская голубая</item>
11    <item>Ориентал</item>
12    <item>Сфинкс</item>
13
14 </string-array>
15
16 <string-array name="pisane_cat">
17     <item>hehe?</item>
18     <item>Угрюмый</item>
19     <item>Удивительная</item>
20     <item>Злой</item>
21     <item>Активная</item>
22     <item>Серьёзная</item>
23     <item>Грациозная</item>
24     <item>Снежная</item>
25     <item>Лиса</item>
26
27 </string-array>
28
29 </resources>

```

Как уже можно было заметить из скриншотов, листинг значительно отличается, это сделано для творческого задания, которое расположено в конце работы.

Творческая работа непосредственно у меня заключается в следующем:

1. Для разных пород кошек добавил разные картинки.
2. Первый элемент - “мемный” котенок.
3. Добавил второй активити с WebView.
4. При нажатии на любой из элементов открывается второй активити, где перед нами предстаёт страница на википедии с выбранной породой кошки.
5. Немного скриншотов работы приложения:



Выберем например Мейн-Куна:

