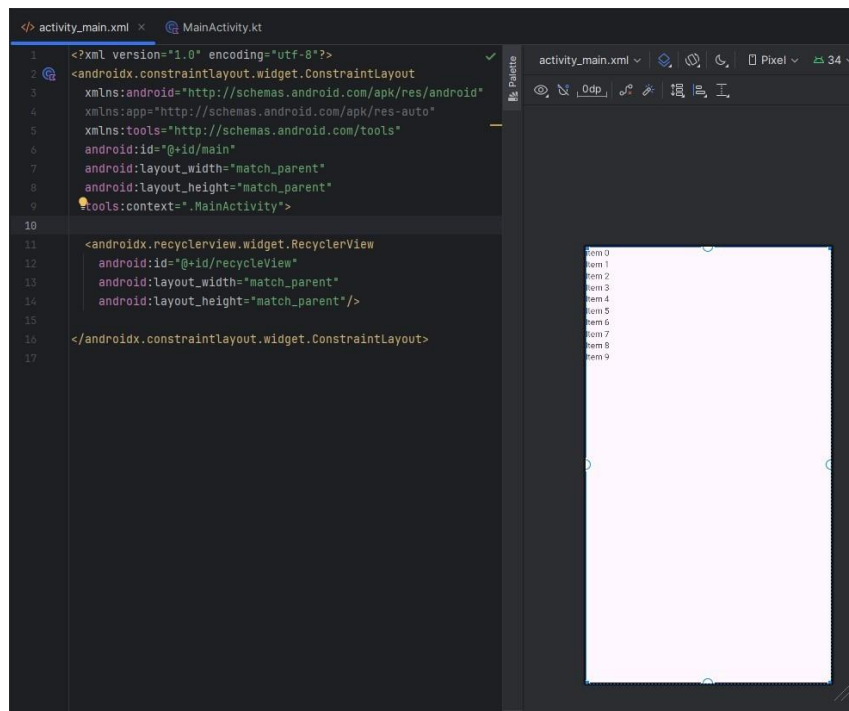
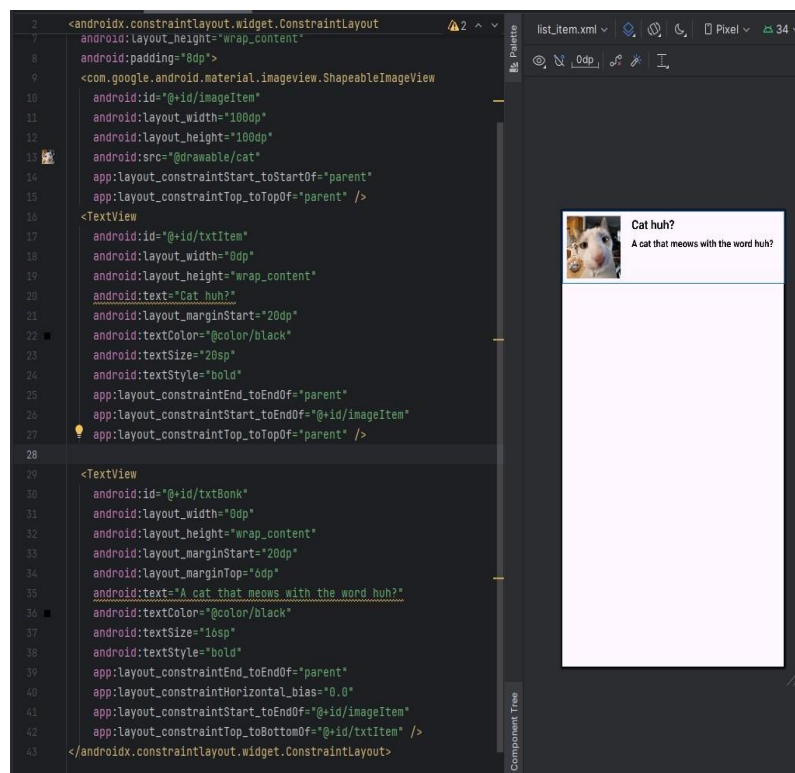


Практика 7: Работа с RecyclerView

В пустом проекте добавим RecyclerView на главную активность



Создадим элемент списка



В теле класса MainActivity создаем дата класс для хранения информации для каждого элемента списка

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

И класс MyAdapter для формирования элементов по данным

```
class MyAdapter(private val newList: ArrayList<DataClass>):
    RecyclerView.Adapter<MyAdapter.MyViewHolder>() {
    // Метод, вызываемый для создания нового ViewHolder
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
        MyViewHolder {
        // Создаем View для элемента списка, используя разметку list_item
        val itemView = LayoutInflater.from(parent.context).inflate(R.layout.list_item, parent, attachToRoot: false)
        // Возвращаем новый экземпляр MyViewHolder с созданным View
        return MyViewHolder(itemView)
    }
    // Метод для получения общего количества элементов в списке
    override fun getItemCount(): Int {
        return newList.size
    }
    // Метод для привязки данных к Views внутри ViewHolder
    override fun onBindViewHolder(holder: MyViewHolder, position: Int) {
        // Получаем текущий элемент списка по позиции
        val currentItem = newList[position]
        // Устанавливаем изображение для ImageView из ресурсов по id и 2 TextView
        holder.imageView.setImageResource(currentItem.titleImage)
        holder.txtHead.text = currentItem.heading
        holder.txtBonk.text = currentItem.bonktxt
    }
}
// Вложенный класс ViewHolder для управления представлением элемента
class MyViewHolder(itemView: View): RecyclerView.ViewHolder(itemView) {
    // Инициализируем Views для элемента списка
    val imageView: ShapeableImageView = itemView.findViewById(R.id.imageItem)
    val txtHead: TextView = itemView.findViewById(R.id.txtItem)
    val txtBonk: TextView = itemView.findViewById(R.id.txtBonk)
}
}
```

Далее в классе MainActivity объявим переменные для RecyclerView и списков данных

```
// Объявление переменных для RecyclerView и списков данных
private lateinit var newRecyclerView : RecyclerView
private lateinit var newList : ArrayList<DataClass>
private lateinit var imageid : Array<Int>
private lateinit var textid : Array<String>
private lateinit var textBonkId : Array<String>
```

Названия и описания котов сохраним в файл strings

```
1 <resources>
2 <string name="app_name">Recycle View</string>
3 <string-array name="porodi_cat">
4 <item>Мейн-Кун</item>
5 <item>Шотландская вислouxая</item>
6 <item>Британец</item>
7 <item>Сиамская</item>
8 <item>Абиссинская</item>
9 <item>Русская голубая</item>
10 <item>Ориентал</item>
11 <item>Сфинкс</item>
12 <item>Рэгдолл</item>
13 <item>Тойгер</item>
14 </string-array>
15 <string-array name="opisanie_cat">
16 <item>Угрюмый</item>
17 <item>Удивлённая</item>
18 <item>Злой</item>
19 <item>Активная</item>
20 <item>Серьёзная</item>
21 <item>Грациозная</item>
22 <item>Смешная</item>
23 <item>Лысая</item>
24 <item>Пушистая</item>
25 <item>Тигр</item>
26 </string-array>
27 </resources>
```

Создадим функции для заполнения списка данными и создание адаптера

```
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)  
}
```

Код в onCreate

```
// Инициализация массива с ресурсами изображений  
imageid = arrayOf(  
    R.drawable.maine_coon,  
    R.drawable.scottish_fold,  
    R.drawable.british,  
    R.drawable.siamese,  
    R.drawable.abyssinian,  
    R.drawable.russian_blue,  
    R.drawable.oriental,  
    R.drawable.sphinx,  
    R.drawable.ragdoll,  
    R.drawable.touger,  
)  
Получение строковых ресурсов и сохранение их в массивах  
textId = resources.getStringArray(R.array.porodi_cat)  
textBonkId = resources.getStringArray(R.array.opisanie_cat)  
// Находим RecyclerView в макете по его ID  
newRecyclerView = findViewById(R.id.recyclerView)  
// Установка менеджера компоновки для RecyclerView  
newRecyclerView.layoutManager = LinearLayoutManager( context: this)  
//recyclerView не планирует изменять размеры своих дочерних элементов динамически.  
newRecyclerView.setHasFixedSize(true)  
// Инициализация нового списка для хранения данных  
newArrayList = arrayListOf<DataClass>()  
getUserdata()
```

Результат:



Творческое задание

Сделаем вики по кошкам, при нажатии на кошку нас перекинет на активность с подробным описанием. Начнем с новой активности

```
<?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"
    android:background="@drawable/fon"
    android:padding="8dp">

    <com.google.android.material.imageview.ShapeableImageView
        android:id="@+id/imageDescription"
        android:src="@drawable/cat"
        android:layout_width="200dp"
        android:layout_height="200dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
    />

    <TextView
        android:id="@+id/headerDescription"
        android:textColor="@color/white"
        android:background="@color/blackCC"
```

```

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Title"
    android:gravity="center"
    android:textStyle="bold"
    android:textSize="20sp"
    app:layout_constraintTop_toBottomOf="@+id/imageDescription"
    android:layout_marginTop="10dp"
/>

```

```

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_marginTop="10dp"
    app:layout_constraintTop_toBottomOf="@+id/headerDescription"
    android:background="@color/blackCC"
    app:layout_constraintBottom_toBottomOf="parent">
    <TextView
        android:id="@+id/description"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ellipsize="none"
        android:scrollbars="vertical"
        android:text="Description"
        android:textColor="@color/white"
        android:textSize="18sp"
        android:lineSpacingMultiplier="1.4"
    />
</ScrollView>

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Описание помещено в ScrollView, что позволяет его прокручивать

(Да оказывается код можно вставлять вот так `_(\`)/\` `) Результат:



Я решил вывести imageid в переменные класса и сделать компаньоном, чтобы потом обратиться к ней из другой активности

```
companion object {  
    // Инициализация массива с ресурсами изображений  
    val imageid = arrayOf(  
        R.drawable.maine_coon,  
        R.drawable.scottish_fold,  
        R.drawable.british,  
        R.drawable.siamese,  
        R.drawable.abbyssinian,  
        R.drawable.russian_blue,  
        R.drawable.oriental,  
        R.drawable.sphinx,  
        R.drawable.ragdoll,  
        R.drawable.toyger,  
    )  
}
```

Для обработки нажатия можно передать в конструктор адаптера функцию обработки

```
class MyAdapter(private val newList: ArrayList<DataClass>, private val onItemClick: (index: Int) -> Unit):
```

В данном случае onItemClick принимает индекс кота

Далее в функцию onBindViewHolder добавим

```
// Переход к активности с описанием  
holder.itemView.setOnClickListener {  
    onItemClick(position)  
}
```

И в функции getUserData определим переход к новой активности

```
// Установка адаптера для RecyclerView с заполненным списком данных и биндом на нажатие  
newRecyclerView.adapter = MyAdapter(newArrayList) { index: Int ->  
    val intent = Intent(this, CatDescription::class.java)  
    intent.putExtra("index", index)  
    startActivity(intent)  
}
```

В коде новой активности считаем индекс и подогоним нужные данные

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_cat_description)  
    // Инициализация переменных  
    val index = intent.getIntExtra("index", 0)  
    val textId = resources.getStringArray(R.array.porodi_cat)  
    val descriptionId = resources.getStringArray(R.array.description)  
    val image: ShapeableImageView = findViewById(R.id.imageDescription)  
    val header: TextView = findViewById(R.id.headerDescription)  
    val description: TextView = findViewById(R.id.description)  
    // Определение атрибутов  
    image.setImageResource(MainActivity.imageid[index])  
    header.text = textId[index]  
    description.text = descriptionId[index]  
}
```

Результат:



Мейн-Кун
Угрюмый



Шотландская вислоухая
Удивлённая



Британец
Злой



Сиамская
Активная



Абиссинская
Серьёзная



Русская голубая
Грациозная



Ориентал
Смешная



Сфинкс
Лысая



Британец

Британская короткошёрстная кошка — крупная порода, имеющая массивное, коренастое телосложение. У представителей этой породы широкая грудная клетка и толстый хвост. Морда, уши, глаза и голова имеют округлую форму. У этих кошек короткий широкий нос и массивная нижняя челюсть. Шерсть — короткая и густая, без пушистости, бывает более ста окрасов и множество комбинаций пятен.

Описание породы британская короткошёрстная

Главная черта британской породы кошек — это их милые щёчки, от которых люди приходят в восторг. Именно округлая форма щёк, а также приземистое массивное тело с толстым хвостом покорило сердца окружающих и сделали британцев популярными не только в Англии, но и во всём мире.



Русская голубая

серебристым отливом.

Характер

Русские голубые кошки настоящие эмпаты: тонко чувствуют настроение хозяина и подстраиваются под него. Это тихие, сдержанные интеллектуалы. С настороженностью относятся к новым людям, не демонстрируют чувства первому встречному. У русской голубой бывают периоды активности, во время которых кошка совершает пробежки, прыгает с высоты и энергично требует внимания.

Происхождение

Страна происхождения: Россия

Русская голубая кошка была изначально известна как архангельская, поскольку, как утверждается, она прибыла в Европу на кораблях из российского одноименного порта (Архангельск). Она также известна как испанская кошка и мальтийская. В США до



Абиссинская

Абиссинская кошка

Абиссинская порода — это элегантные кошки средних размеров с сильными грациозными телами и длинными стройными лапами. Для этой породы характерны округлая клиновидная форма головы с большими миндалевидными глазами и уши с небольшими кисточками на кончиках.

Короткая прилегающая шерсть абиссинской кошки отличается тикингом — смешением цветов на каждом из волосков. Наиболее популярный окрас — «дикий» (ruddy), но также встречаются и другие виды.

Характер

Абиссинская кошка достаточно уравновешенное по характеру животное. Это активная, умная и очень любопытная порода. Ей нравится общество человека, она сильно привязывается к семье. Кошки этой породы любят свободное пространство, подвижные игры и ползать на высоте. Например, им



Шотландская вислоухая

Шотландский вислоухий кот или скоттиш-фолд (англ. Scottish Fold) — порода домашних кошек с характерными загнутыми вперёд и вниз ушами, что вызвано действием доминантного гена, который возник в результате естественной мутации и оказывающий влияние на хрящи во всём организме.

Внешность

Уши

Все вислоухие котята рождаются с прямыми ушами, которые под действием гена Fd начинают складываться приблизительно в течение 21 дня. Котята, у которых не развивается вислоухость, называются шотландскими прямоухими или скоттиш-страйт (англ. Scottish Straight). У первых представителей породы на ушах образовывалась одна складка, однако, благодаря селекционному отбору, заводчикам удалось увеличить количество



Сиамская

Сиамская кошка — это кошка среднего размера с длинным, стройным и грациозным телом. Тело — хорошо сбалансированное, атлетического строения, лапы — стройные, аккуратные, небольшого размера. Спереди голова этой кошки напоминает треугольник, сужающийся прямыми линиями к утонченной морде. Уши — крупные и заостренные. Если мысленно провести линию между крыльями носа и кончиками ушей, получится равносторонний треугольник. Форма глаз — восточного типа, с наклоном по направлению к носу. Длинный, сужающийся к кончику хвост не имеет завитков. Короткая, красивая, прилегающая к телу шерсть имеет бледный тон с более насыщенными темными пятнами в зоне маски морды, ушей, лап и хвоста. Она бывает различных окрасов. Цвет глаз при любых окрасах и вариантах пятнистости всегда остается ярким, насыщенно-голубым.

Характер