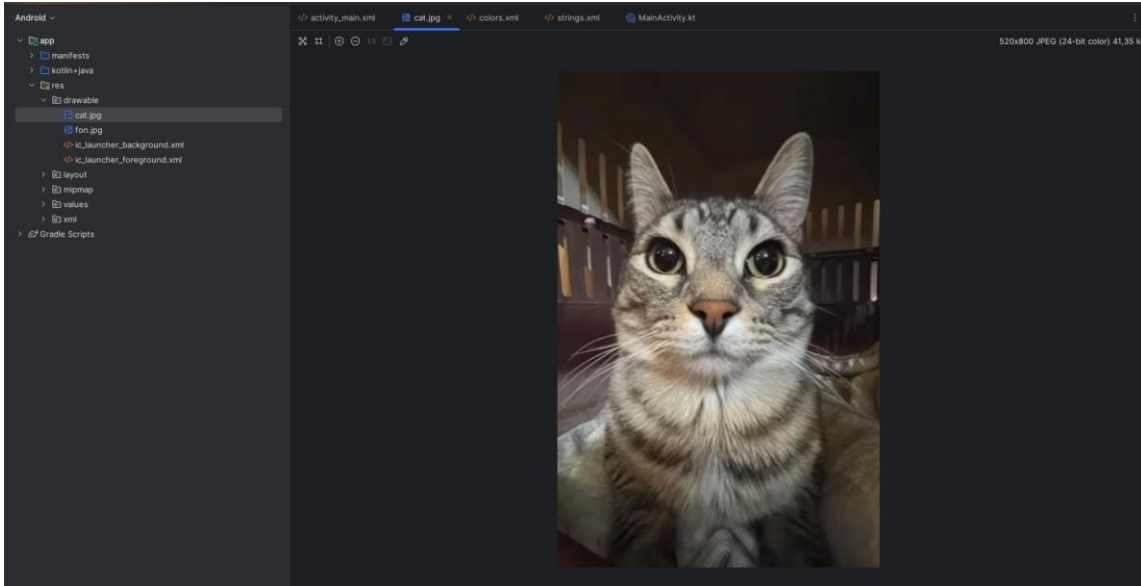


Практика 2

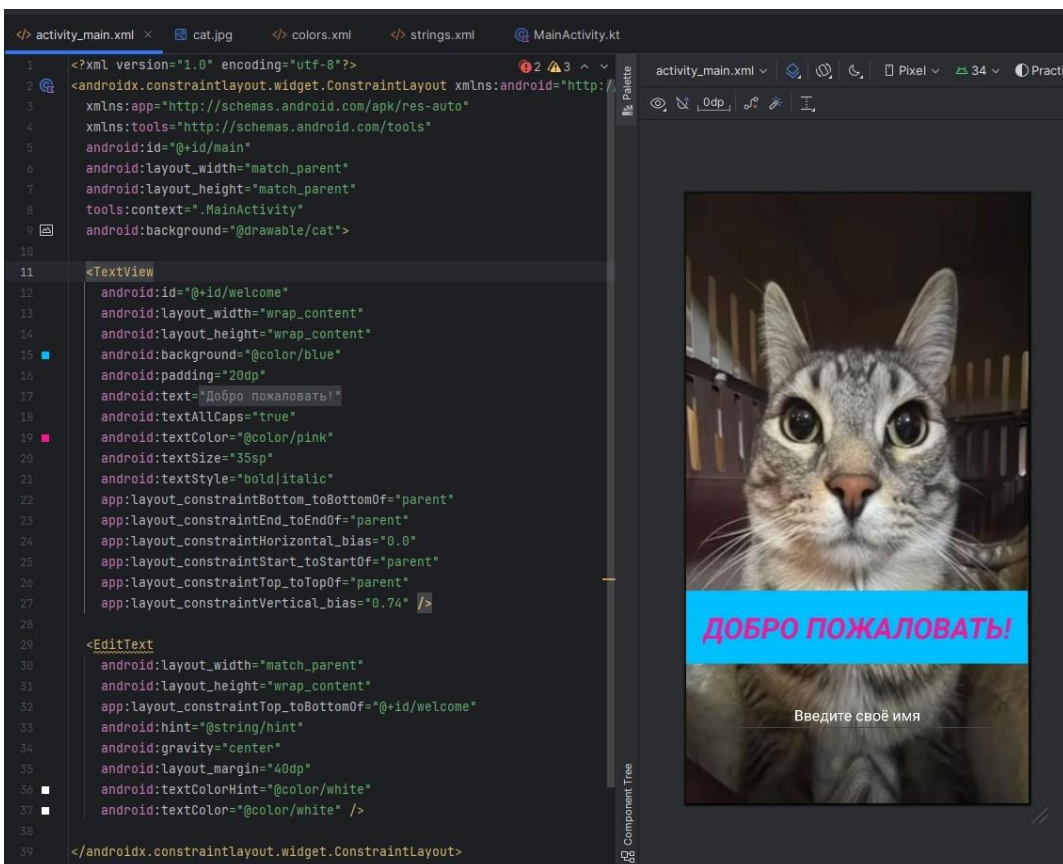
Задание 1.

Что находится в папке res? Там есть:

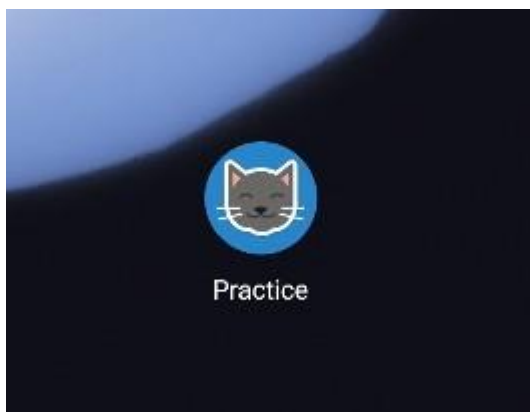
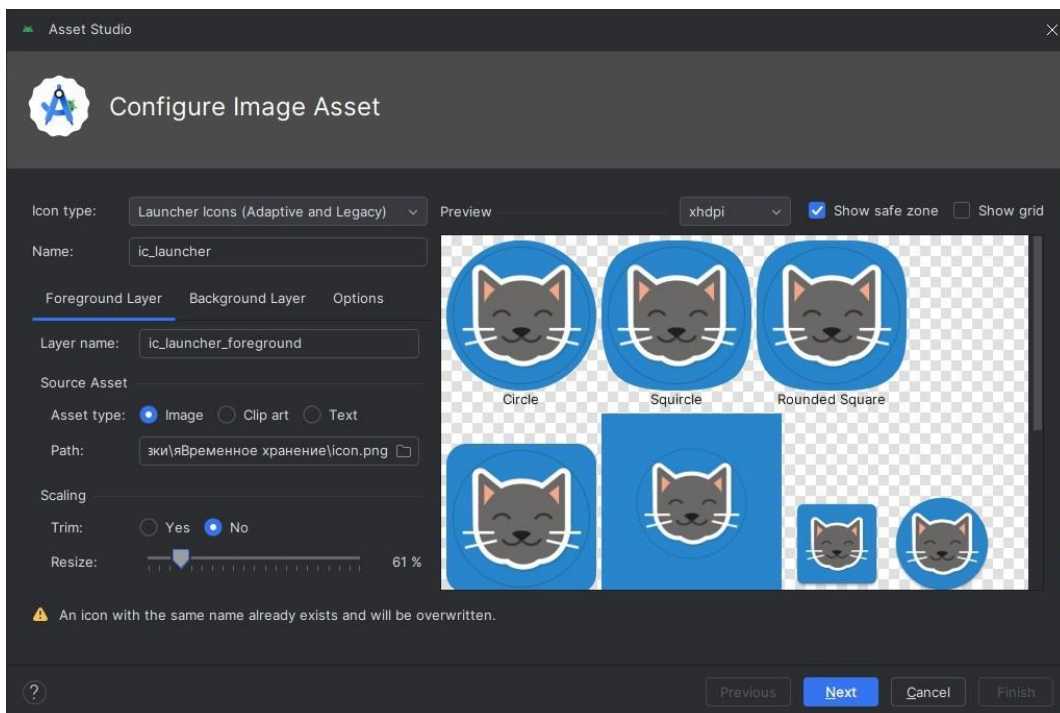
- drawable хранит в себе графические файлы, такие как jpg, png, gif, который будут использоваться в проекте. Для примера закинем туда котика:



- layout хранит xml файлы, определяющие пользовательский интерфейс приложения
- Добавим ранее добавленного кота:



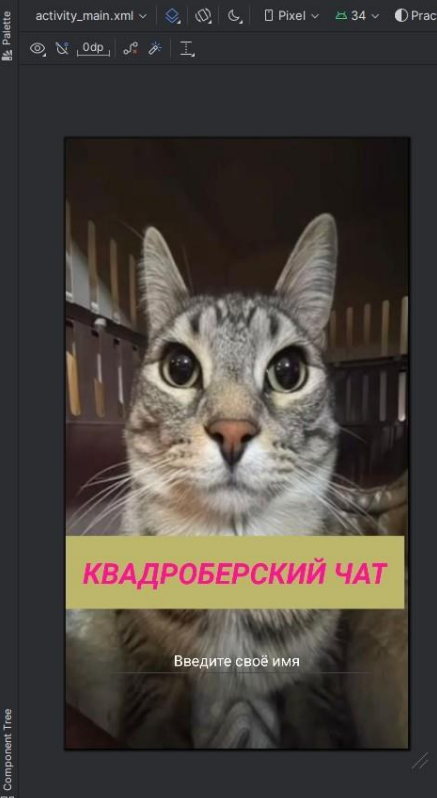
- minipar хранит графические файлы, используемые для иконок приложения под различные разрешения экранов



- values хранит xml файлы, которые содержат различные используемые в приложении значения, например, ресурсы строк или кодировки цветов

```
androidx.constraintlayout.widget.ConstraintLayout → TextView
</> colors.xml ×
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3  ■  <color name="black">#FF000000</color>
4  ■  <color name="white">#FFFFFFFF</color>
5  ■  <color name="pink">#FF1493</color>
6  ■  <color name="blue">#00BFFF</color>
7  ■  <color name="dark_khaki">#BDB76B</color>
8  </resources>
resources
</> strings.xml ×
i Edit translations for all locales in the translations editor.
1  <resources>
2  <string name="app_name">Practice</string>
3  <string name="welcome">Квадроберский чат</string>
4  <string name="hint">Введите своё имя</string>
5  </resources>
```

```
</> activity_main.xml × </> colors.xml </> strings.xml MainActivity.kt
2  <androidx.constraintlayout.widget.ConstraintLayout
8  tools:context=".MainActivity"
9  android:background="@drawable/cat">
10
11  <TextView
12  android:id="@+id/welcome"
13  android:layout_width="wrap_content"
14  android:layout_height="wrap_content"
15  android:gravity="center"
16  ■  android:background="@color/dark_khaki"
17  android:padding="20dp"
18  android:text="Квадроберский чат"
19  ⚡  android:textAllCaps="true"
20  ■  android:textColor="@color/pink"
21  android:textSize="35sp"
22  android:textStyle="bold|italic"
23  app:layout_constraintBottom_toBottomOf="parent"
24  app:layout_constraintEnd_toEndOf="parent"
25  app:layout_constraintHorizontal_bias="0.0"
26  app:layout_constraintStart_toStartOf="parent"
27  app:layout_constraintTop_toTopOf="parent"
28  app:layout_constraintVertical_bias="0.74" />
29
30  <EditText
31  android:layout_width="match_parent"
32  android:layout_height="wrap_content"
33  app:layout_constraintTop_toBottomOf="@+id/welcome"
34  android:hint="Введите своё имя"
35  android:gravity="center"
36  android:layout_margin="40dp"
37  ■  android:textColorHint="@color/white"
38  ■  android:textColor="@color/white" />
39
40  </androidx.constraintlayout.widget.ConstraintLayout>
```



И для красоты уберем строку состояния в файле themes.xml, включив полноэкранный режим:

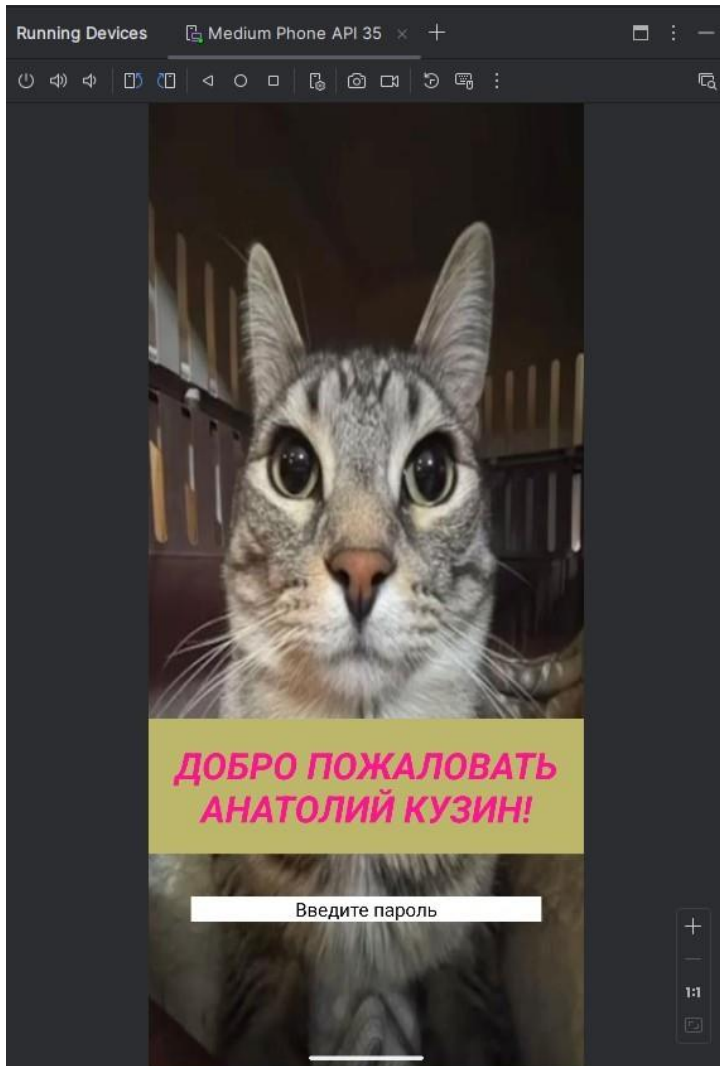
```
MainActivity.kt x themes.xml x activity_main.xml x colors.xml x strings.xml
1 <resources xmlns:tools="http://schemas.android.com/tools">
2   <!-- Base application theme. -->
3   <style name="Base.Theme.Practice" parent="Theme.Material3.DayNight.NoActionBar">
4     <!-- Customize your light theme here. -->
5     <!-- <item name="colorPrimary">@color/my_light_primary</item> -->
6     <item name="android:windowFullscreen">true</item>
7   </style>
8
9   <style name="Theme.Practice" parent="Base.Theme.Practice" />
10 </resources>
```

- xml хранит различные xml файлы. Попробуем создать файл с данными пользователя, затем считаем их и выведем на экран. Добавим в xml файл user.xml, и сделаем код, который считывает имя и фамилию:

```
MainActivity.kt x user.xml x activity_main.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <user>
3   <name>Анатолий</name>
4   <surname>Кузин</surname>
5 </user>
```

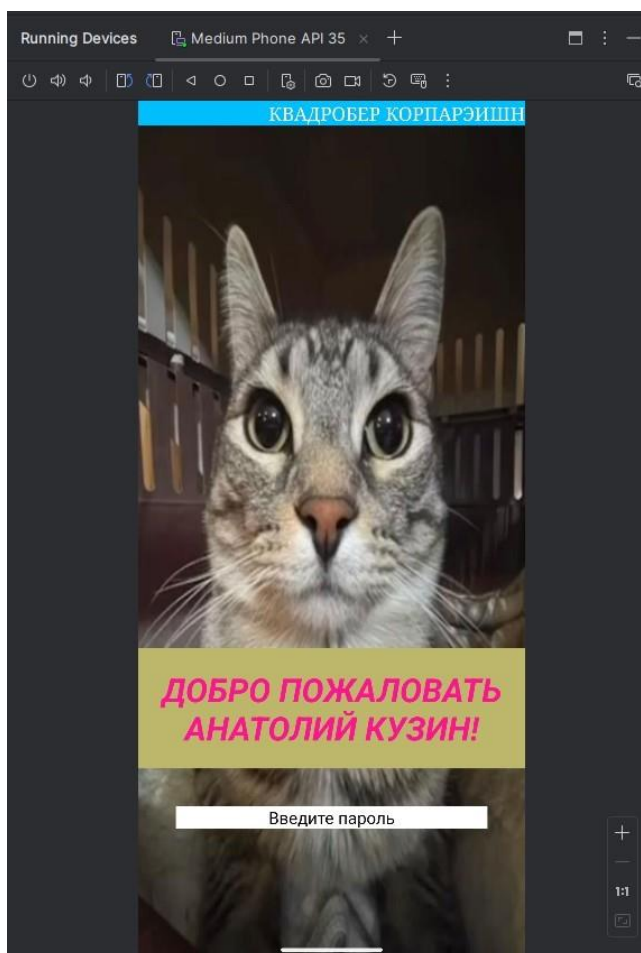
```
MainActivity.kt x user.xml x activity_main.xml
1 package com.example.practice
2
3 > import ...
4
5
6
7
8
9 class MainActivity : AppCompatActivity() {
10
11   override fun onCreate(savedInstanceState: Bundle?) {
12     super.onCreate(savedInstanceState)
13     enableEdgeToEdge()
14     setContentView(R.layout.activity_main)
15
16     val xpp = resources.xml(R.xml.user)
17     var eventType = xpp.eventType
18     var curValue = ""
19     var result = "Добро пожаловать"
20     while(eventType != XmlPullParser.END_DOCUMENT){
21       val name = xpp.name
22       when(eventType){
23         XmlPullParser.TEXT -> curValue = xpp.text
24         XmlPullParser.END_TAG ->
25           if((name == "name") || (name == "surname")) result += " $curValue"
26       }
27       eventType = xpp.next()
28     }
29     result += '!'
30     findViewById<TextView>(R.id.welcome).text = result
31   }
32 }
```

И получаем следующее



Задание 2. Протестировать основные атрибуты TextView. Для тестирования добавим новый текст, поставим ему атрибуты:

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/company"
    android:textSize="20sp"
    android:background="@color/blue"
    android:textColor="@color/white"
    android:textAllCaps="true"
    android:textDirection="rtl"
    android:textAlignment="textStart"
    android:fontFamily="serif"
/>
```

Сверху расположен текст на голубом фоне. Атрибуты `android:textDirection` выравнивает текст справа налево, а атрибут `android:textAligment` выравнивает его слева направо. Но как видно на скриншоте: текст остался справа.