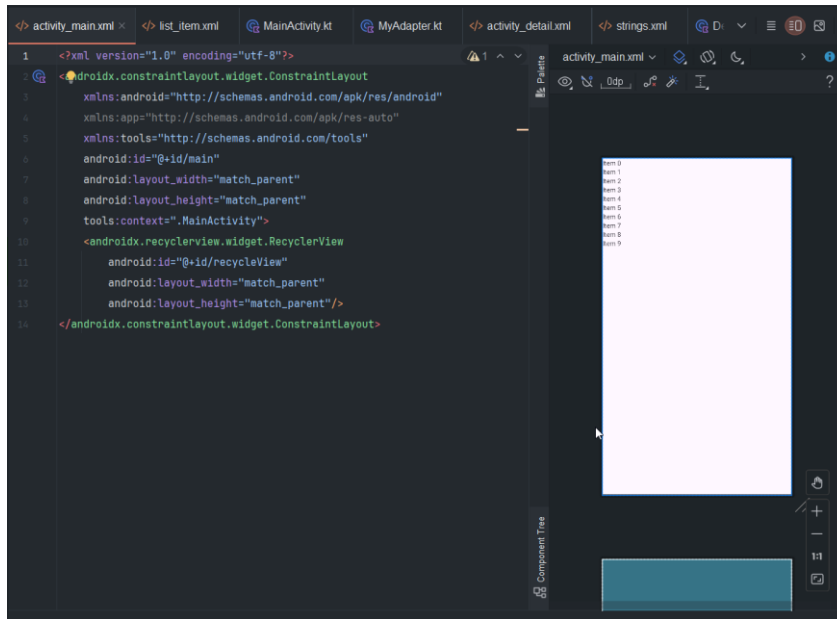
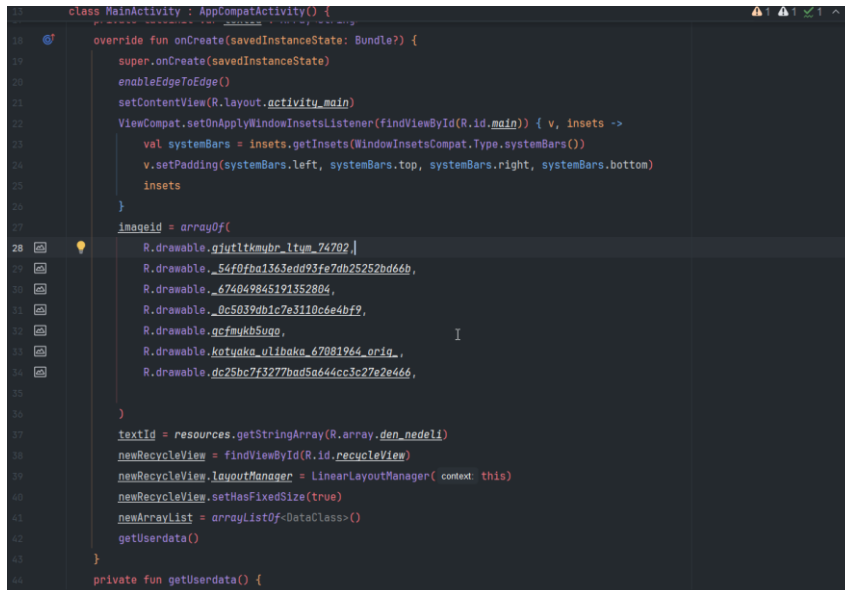


Выполнил: Войтенко Данил

Сначала сделал xml mainactivity



Далее скопировал и модифицировал код предоставленный в практике



```

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        textId = resources.getStringArray(R.array.den_nedeli)
        newRecyclerView = findViewById(R.id.recycleView)
        newRecyclerView.layoutManager = LinearLayoutManager(context, this)
        newRecyclerView.setHasFixedSize(true)
        newArrayList = arrayListOf<DataClass>()
        getUserdata()
    }

    private fun getUserdata() {
        val size = minOf(imageId.size, textId.size)
        for (i in 0 until size) {
            val data = DataClass(imageId[i], textId[i])
            newArrayList.add(data)
        }

        val adapter = MyAdapter(newArrayList) { item ->
            val intent = Intent(packageContext, DetailActivity::class.java).apply {
                putExtra(name, "IMAGE_ID", item.titleImage)
            }
            startActivity(intent)
        }

        newRecyclerView.adapter = adapter
    }
}

```

Здесь я передаю изображение и открываю новую activity при клике

```

import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView
import com.google.android.material.imageview.ShapeableImageView

class MyAdapter(private val newList : ArrayList<DataClass>, private val onItemClick: (DataClass) -> Unit) :
    RecyclerView.Adapter<MyAdapter.MyViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
        MyViewHolder {
        val itemView = LayoutInflater.from(parent.context).inflate(R.layout.list_item, parent, attachToRoot: false)
        return MyViewHolder(itemView)
    }

    override fun getItemCount(): Int {
        return newList.size
    }

    override fun onBindViewHolder(holder: MyViewHolder, position: Int) {
        val currentItem = newList[position]
        holder.imageView.setImageResource(currentItem.titleImage)
        holder.txtHead.text = currentItem.heading
        holder.itemView.setOnClickListener {
            onItemClick(currentItem)
        }
    }
}

class MyViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
    val imageView : ShapeableImageView =
        itemView.findViewById(R.id.imageItem)
    val txtHead : TextView = itemView.findViewById(R.id.txtItem)
}

```

Здесь добавил лишь новую переменную она и принимает действия при нажатии на элемент ну и убрал не нужный текст так выглядит xml

```

<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"
    tools:context=".MainActivity">

    <com.google.android.material.imageview.ShapeableImageView
        android:id="@+id/imageItem"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:src="@drawable/ic_launcher_background"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtItem"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginTop="20dp"
        android:text="tit n h"
        android:textColor="@color/black"
        android:textSize="16sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toEndOf="@+id/imageItem"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

```

package com.example.practice8

import ...

class DetailActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_detail)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }

        val imageId = intent.getIntExtra(name: "IMAGE_ID", defaultValue: -1)
        val image = findViewById<ImageView>(R.id.viewimage)
        image.setImageResource(imageId)
    }
}

```

В новой activity я принимаю изображение и вывожу её на экран вот xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".DetailActivity">

    <ImageView
        android:id="@+id/viewimage"
        android:layout_width="412dp"
        android:layout_height="546dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:srcCompat="@tools:sample/avatars" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Пример работы

