

```
package com.example.l7
```

```
class Item(var img: Int, var name: String, var author: String)
```

```
package com.example.l7
```

```
import ...
```

```
class ItemAdapter(private val context: Context, private val items: List<Item>) :
```

```
    ArrayAdapter<Item?>(
        context, R.layout.item, items
```

```
    ) {
```

```
        override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {
```

```
            val inflater = context.getSystemService(Context.LAYOUT_INFLATER_SERVICE) as LayoutInflater
```

```
            val view = inflater.inflate(R.layout.item, parent, attachToRoot: false)
```

```
            val name = view.findViewById<TextView>(R.id.name)
```

```
            name.text = items[position].name
```

```
            val author = view.findViewById<TextView>(R.id.author)
```

```
            author.text = items[position].author
```

```
            val image = view.findViewById<ImageView>(R.id.image)
```

```
            image.setImageResource(items[position].img)
```

```
            return view
```

```
        }
```

```
    }
```

```

package com.example.l7

import ...

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

        val listView = findViewById<ListView>(R.id.listView)

        val items: MutableList<Item> = ArrayList()
        items.add(Item(R.drawable.img, name: "Музыка 1990-х", author: ""))
        items.add(Item(R.drawable.img_5, name: "Лучшие песни русского рока", author: ""))
        items.add(Item(R.drawable.img_10, name: "Шедевры мировой классики", author: ""))
        for (i in 0 ≤ .. ≤ 9) {
            items.add(Item(R.drawable.ic_launcher_foreground, name: "Название", author: ""))
        }

        val itemAdapter = ItemAdapter(context: this, items)
        listView.adapter = itemAdapter

        val intent = Intent(packageContext: this, MainActivity2::class.java)

        listView.setOnItemClickListener = OnItemClickListener { parent, view, position, id ->
            intent.putExtra(name: "pos", position)
            startActivity(intent)
        }
    }
}

```

```

        items3!!.add(
            Item(
                R.drawable.img_13,
                name: "Richter: November",
                author: "Mari Samuelsen, Konzerthausorchester Berlin, Jonathan Stockhammer"
            )
        )
    }
    items3!!.add(
        Item(
            R.drawable.img_14,
            name: "Concerto No. 4 in F minor (L'inverno/ Winter) RV297 (Op. 8 No. 4): I. Allegro non
            author: "Anne-Sophie Mutter, Антонио Вивальди"
        )
    )
}

for (i in 0 ≤ .. ≤ 9) {
    items3!!.add(Item(R.drawable.ic_launcher_foreground, name: "Name", author: "Author"))
}

items = ArrayList()
items!!.add(Item(R.drawable.ic_launcher_foreground, name: "Name1", author: "Author1"))
items!!.add(Item(R.drawable.ic_launcher_foreground, name: "Name2", author: "Author2"))
items!!.add(Item(R.drawable.ic_launcher_foreground, name: "Name3", author: "Author3"))
items!!.add(Item(R.drawable.ic_launcher_foreground, name: "Name4", author: "Author4"))
for (i in 0 ≤ .. ≤ 9) {
    items!!.add(Item(R.drawable.ic_launcher_foreground, name: "Name", author: "Author"))
}

val id = intent.getIntExtra( name: "pos", defaultValue: 0)

val playlistName = findViewById<TextView>(R.id.playlistName)
playlistName.text = getNameById(id)


val currentItem = getItemById(id)
val itemAdapter = ItemAdapter( context: this, currentItem!!)
val listView = findViewById<ListView>(R.id.listView)
listView.adapter = itemAdapter
}

private fun getNameById(id: Int): String {
    if (id > 2) return playlistNames[3]
    return playlistNames[id]
}

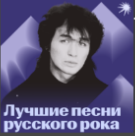
private fun getItemById(id: Int): Item? {
    if (id == 0) return item1
    if (id == 1) return item2
    if (id == 2) return item3
    return item
}

```


Список плейлистов



Музыка 1990-х




Лучшие песни русского рока




Шедевры мировой классики


Название




Название




Название




Название




Название



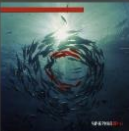
Лучшие песни русского рока




Группа крови  
КИНО



Шнурок  
Пилот




Жара  
Чичерина




2007  
Йорш


Name  
Author



Name  
Author



Name  
Author



Name

