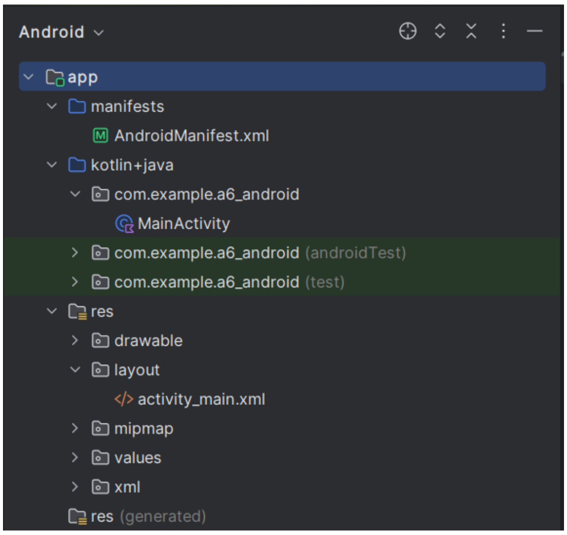
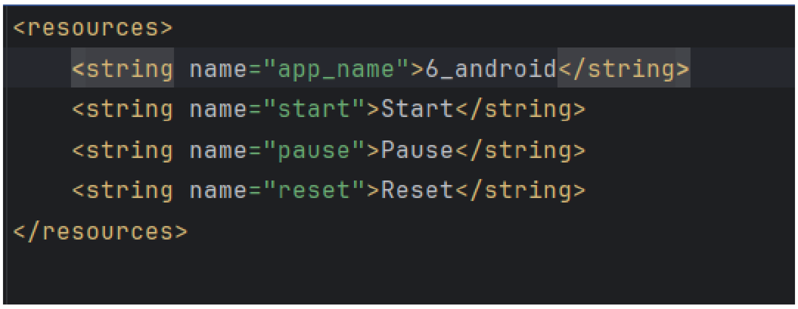
Практика 6

Структура:



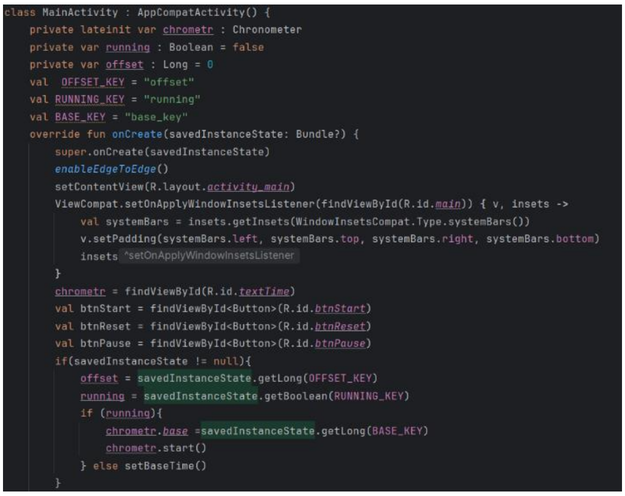
```
Android
app
  manifests
    AndroidManifest.xml
  kotlin+java
    com.example.a6_android
      MainActivity
    com.example.a6_android (androidTest)
    com.example.a6_android (test)
  res
    drawable
    layout
      activity_main.xml
    mipmap
    values
    xml
  res (generated)
```

String:

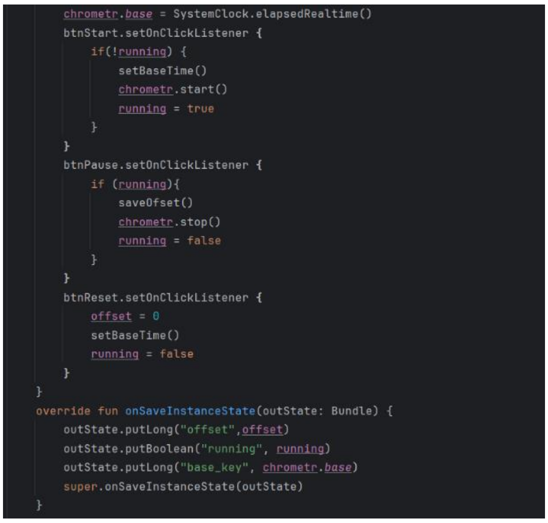


```
<resources>
    <string name="app_name">6_android</string>
    <string name="start">Start</string>
    <string name="pause">Pause</string>
    <string name="reset">Reset</string>
</resources>
```

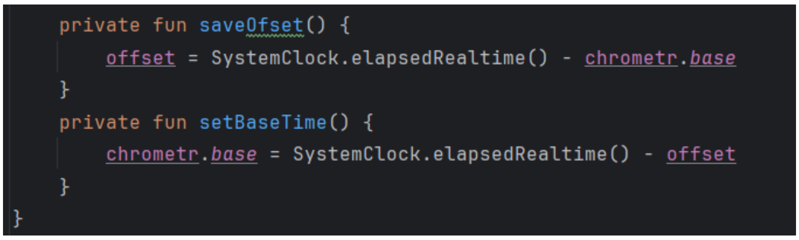
Логика:



```
class MainActivity : AppCompatActivity() {
    private lateinit var chrometr : Chronometer
    private var running : Boolean = false
    private var offset : Long = 0
    val  OFFSET_KEY = "offset"
    val RUNNING_KEY = "running"
    val BASE_KEY = "base_key"
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        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 ^setOnApplyWindowInsetsListener
        }
        chrometr = findViewById(R.id.textTime)
        val btnStart = findViewById<Button>(R.id.btnStart)
        val btnReset = findViewById<Button>(R.id.btnReset)
        val btnPause = findViewById<Button>(R.id.btnPause)
        if(savedInstanceState != null){
            offset = savedInstanceState.getLong(OFFSET_KEY)
            running = savedInstanceState.getBoolean(RUNNING_KEY)
            if (running){
                chrometr.base =savedInstanceState.getLong(BASE_KEY)
                chrometr.start()
            } else setBaseTime()
        }
```

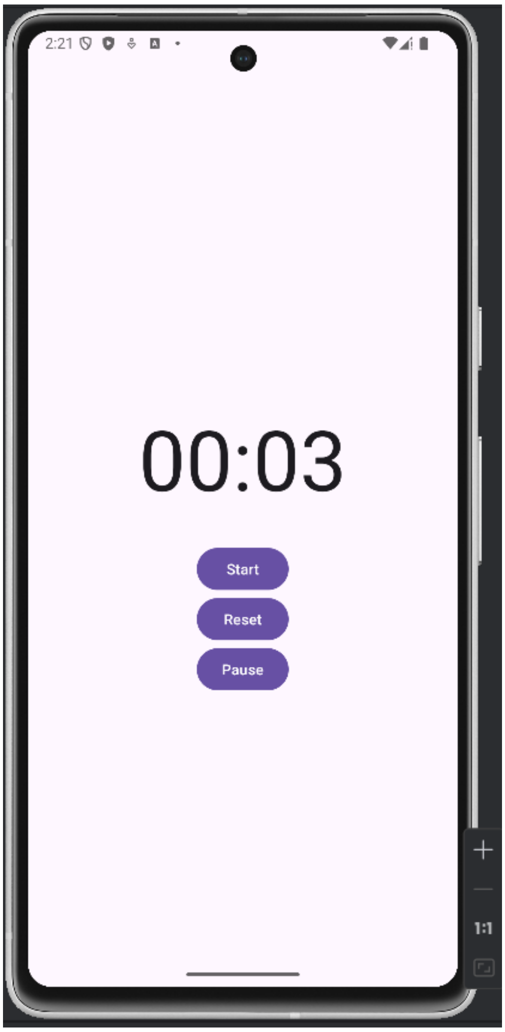


```
        chrometr.base = SystemClock.elapsedRealtime()
        btnStart.setOnClickListener {
            if(!running) {
                setBaseTime()
                chrometr.start()
                running = true
            }
        }
        btnPause.setOnClickListener {
            if (running){
                saveOfset()
                chrometr.stop()
                running = false
            }
        }
        btnReset.setOnClickListener {
            offset = 0
            setBaseTime()
            running = false
        }
    }
    override fun onSaveInstanceState(outState: Bundle) {
        outState.putLong("offset",offset)
        outState.putBoolean("running", running)
        outState.putLong("base_key", chrometr.base)
        super.onSaveInstanceState(outState)
    }
```



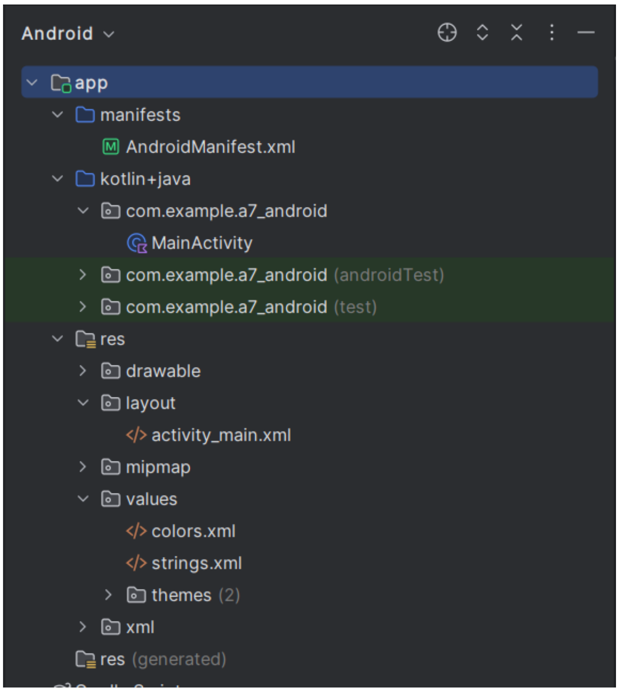
```
    private fun saveOfset() {
        offset = SystemClock.elapsedRealtime() - chrometr.base
    }
    private fun setBaseTime() {
        chrometr.base = SystemClock.elapsedRealtime() - offset
    }
}
```

Итог:



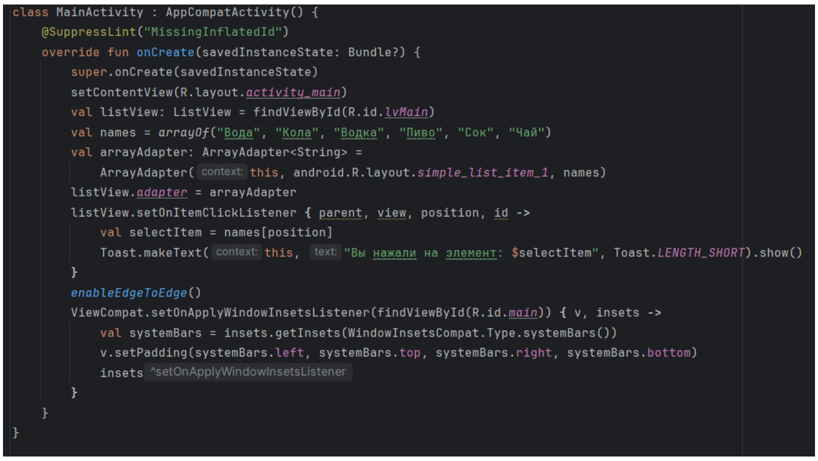
Практика 7:

Структура:



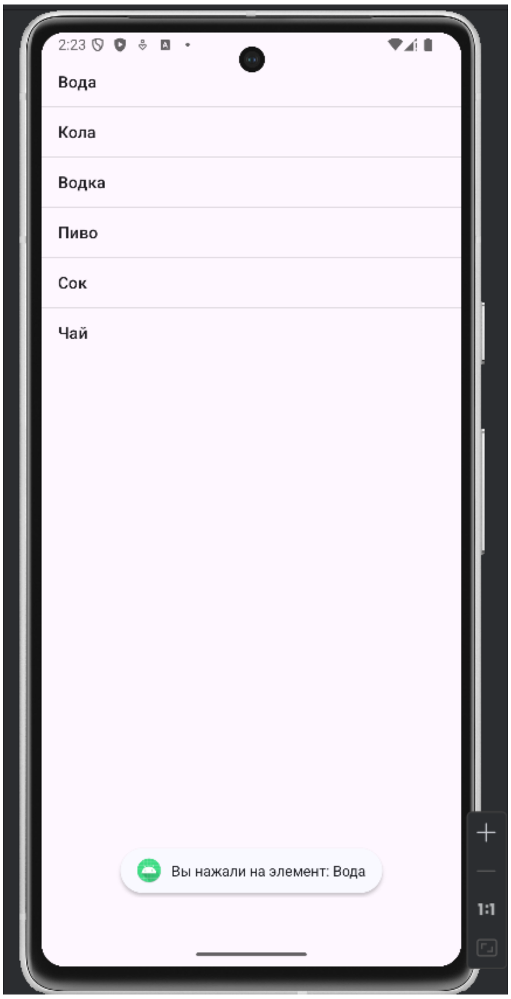
```
Android
app
  manifests
    AndroidManifest.xml
  kotlin+java
    com.example.a7_android
      MainActivity
    com.example.a7_android (androidTest)
    com.example.a7_android (test)
  res
    drawable
    layout
      activity_main.xml
    mipmap
    values
      colors.xml
      strings.xml
      themes (2)
    xml
  res (generated)
```

Логика:



```kotlin
class MainActivity : AppCompatActivity() {
    @SuppressLint("MissingInflatedId")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val listView: ListView = findViewById(R.id.lvMain)
        val names = arrayOf("Вода", "Кола", "Водка", "Пиво", "Сок", "Чай")
        val arrayAdapter: ArrayAdapter<String> =
            ArrayAdapter( context: this, android.R.layout.simple_list_item_1, names)
        listView.adapter = arrayAdapter
        listView.setOnItemClickListener { parent, view, position, id ->
            val selectItem = names[position]
            Toast.makeText( context: this,  text: "Вы нажали на элемент: $selectItem", Toast.LENGTH_SHORT).show()
        }
        enableEdgeToEdge()
        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 ^setOnApplyWindowInsetsListener
        }
    }
}
```

Итог:



2:23

Вода

Кола

Водка

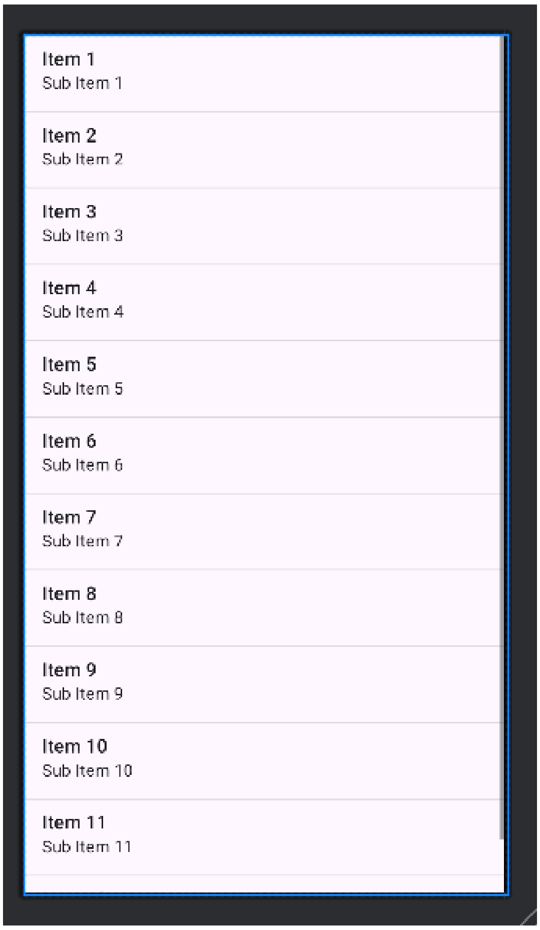
Пиво

Сок

Чай

Вы нажали на элемент: Вода

Дизайн:



Item 1
Sub Item 1

Item 2
Sub Item 2

Item 3
Sub Item 3

Item 4
Sub Item 4

Item 5
Sub Item 5

Item 6
Sub Item 6

Item 7
Sub Item 7

Item 8
Sub Item 8

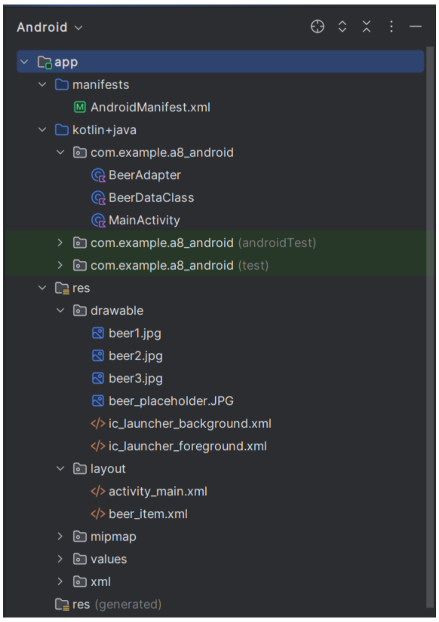
Item 9
Sub Item 9

Item 10
Sub Item 10

Item 11
Sub Item 11

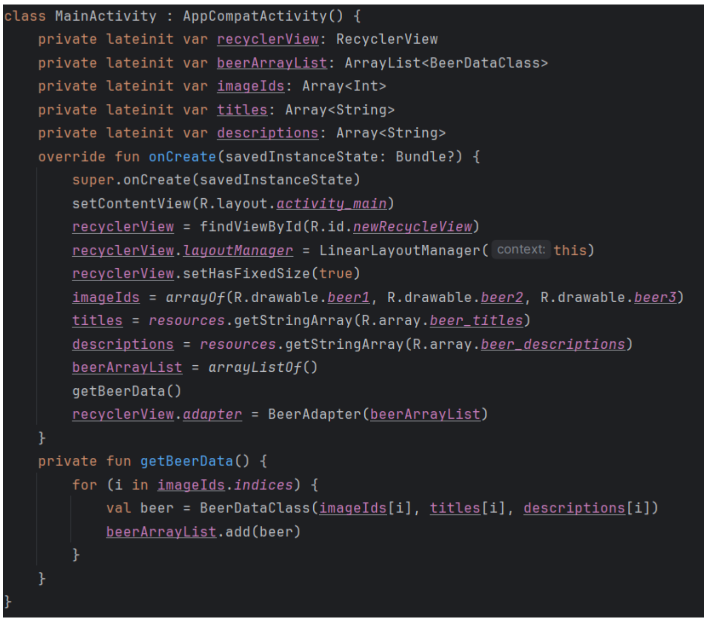
Практика 8:

Структура:



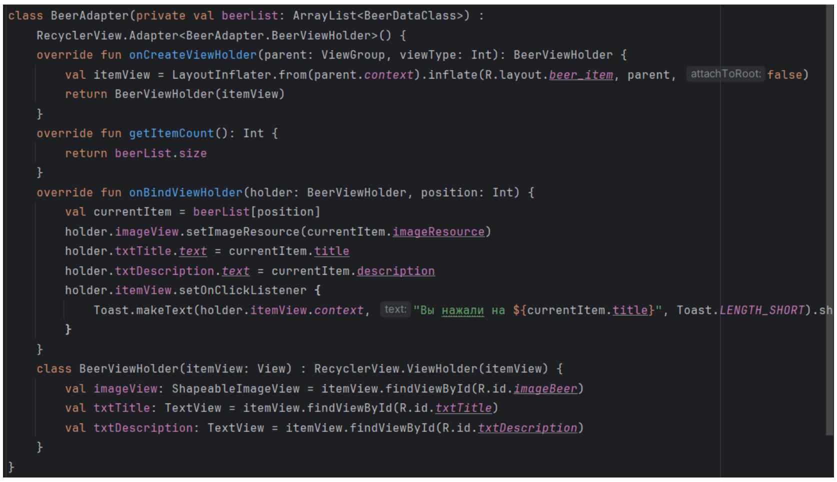
```
Android ⌄
⌄ app
  ⌄ manifests
      AndroidManifest.xml
  ⌄ kotlin+java
    ⌄ com.example.a8_android
        BeerAdapter
        BeerDataClass
        MainActivity
    › com.example.a8_android (androidTest)
    › com.example.a8_android (test)
  ⌄ res
    ⌄ drawable
        beer1.jpg
        beer2.jpg
        beer3.jpg
        beer_placeholder.JPG
        ic_launcher_background.xml
        ic_launcher_foreground.xml
    ⌄ layout
        activity_main.xml
        beer_item.xml
    › mipmap
    › values
    › xml
  res (generated)
```

MainActivity:



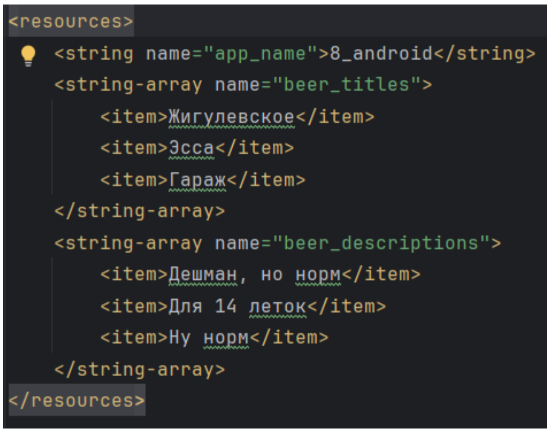
```kotlin
class MainActivity : AppCompatActivity() {
    private lateinit var recyclerView: RecyclerView
    private lateinit var beerArrayList: ArrayList<BeerDataClass>
    private lateinit var imageIds: Array<Int>
    private lateinit var titles: Array<String>
    private lateinit var descriptions: Array<String>
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        recyclerView = findViewById(R.id.newRecycleView)
        recyclerView.layoutManager = LinearLayoutManager( context: this)
        recyclerView.setHasFixedSize(true)
        imageIds = arrayOf(R.drawable.beer1, R.drawable.beer2, R.drawable.beer3)
        titles = resources.getStringArray(R.array.beer_titles)
        descriptions = resources.getStringArray(R.array.beer_descriptions)
        beerArrayList = arrayListOf()
        getBeerData()
        recyclerView.adapter = BeerAdapter(beerArrayList)
    }
    private fun getBeerData() {
        for (i in imageIds.indices) {
            val beer = BeerDataClass(imageIds[i], titles[i], descriptions[i])
            beerArrayList.add(beer)
        }
    }
}
```

BeerAdapter:



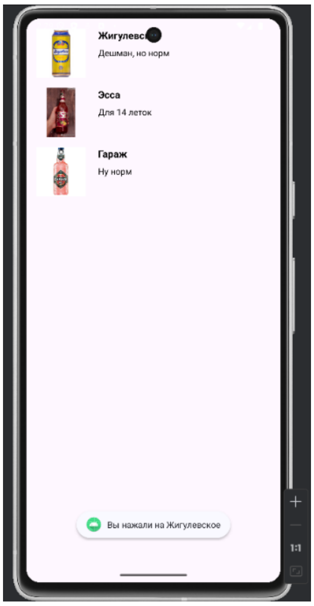
```kotlin
class BeerAdapter(private val beerList: ArrayList<BeerDataClass>) :
    RecyclerView.Adapter<BeerAdapter.BeerViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): BeerViewHolder {
        val itemView = LayoutInflater.from(parent.context).inflate(R.layout.beer_item, parent,  attachToRoot: false)
        return BeerViewHolder(itemView)
    }
    override fun getItemCount(): Int {
        return beerList.size
    }
    override fun onBindViewHolder(holder: BeerViewHolder, position: Int) {
        val currentItem = beerList[position]
        holder.imageView.setImageResource(currentItem.imageResource)
        holder.txtTitle.text = currentItem.title
        holder.txtDescription.text = currentItem.description
        holder.itemView.setOnClickListener {
            Toast.makeText(holder.itemView.context,  text: "Вы нажали на ${currentItem.title}", Toast.LENGTH_SHORT).sh
        }
    }
    class BeerViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        val imageView: ShapeableImageView = itemView.findViewById(R.id.imageBeer)
        val txtTitle: TextView = itemView.findViewById(R.id.txtTitle)
        val txtDescription: TextView = itemView.findViewById(R.id.txtDescription)
    }
}
```

String:



```xml
<resources>
    <string name="app_name">8_android</string>
    <string-array name="beer_titles">
        <item>Жигулевское</item>
        <item>Эсса</item>
        <item>Гараж</item>
    </string-array>
    <string-array name="beer_descriptions">
        <item>Дешман, но норм</item>
        <item>Для 14 леток</item>
        <item>Ну норм</item>
    </string-array>
</resources>
```

Итог:



Жигулевс

Дешман, но норм

Эсса

Для 14 леток

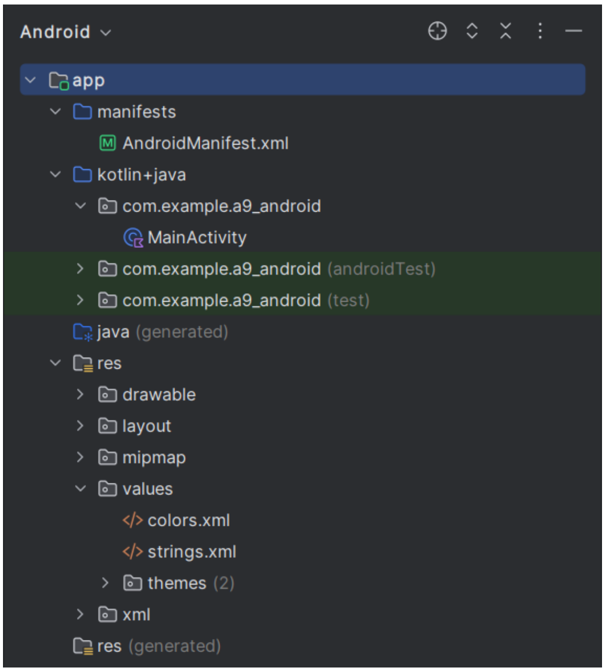
Гараж

Ну норм

Вы нажали на Жигулевское

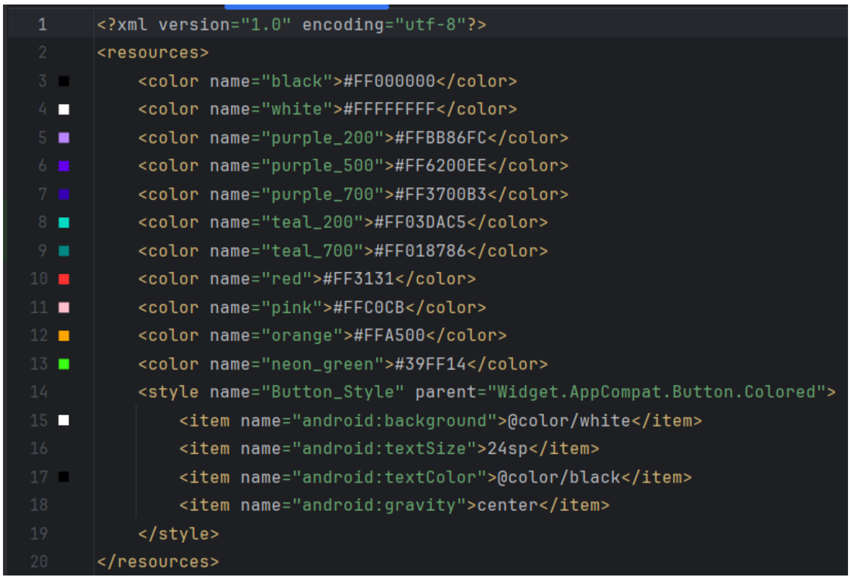
Практика 9:

Структура:



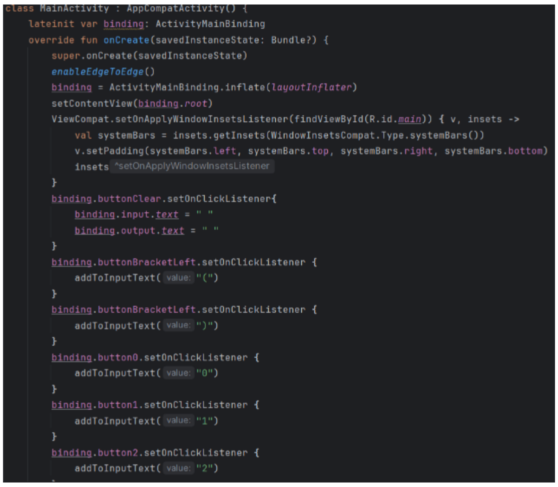
```
Android
app
    manifests
        AndroidManifest.xml
    kotlin+java
        com.example.a9_android
            MainActivity
        com.example.a9_android (androidTest)
        com.example.a9_android (test)
    java (generated)
    res
        drawable
        layout
        mipmap
        values
            colors.xml
            strings.xml
            themes (2)
        xml
    res (generated)
```

Colors:

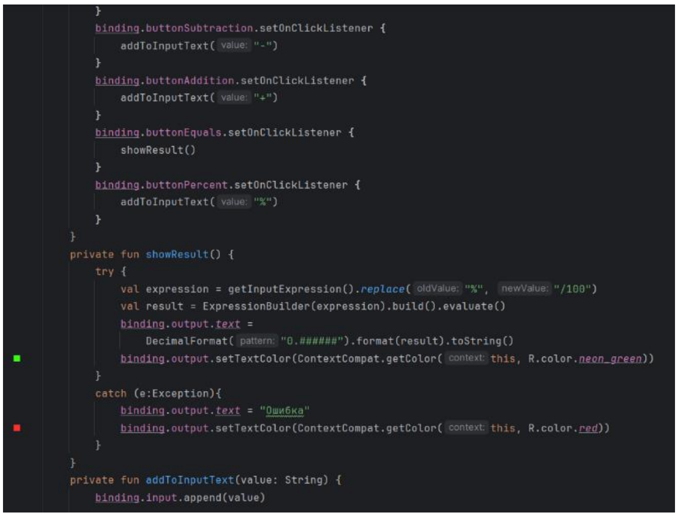


```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="red">#FF3131</color>
    <color name="pink">#FFC0CB</color>
    <color name="orange">#FFA500</color>
    <color name="neon_green">#39FF14</color>
    <style name="Button_Style" parent="Widget.AppCompat.Button.Colored">
        <item name="android:background">@color/white</item>
        <item name="android:textSize">24sp</item>
        <item name="android:textColor">@color/black</item>
        <item name="android:gravity">center</item>
    </style>
</resources>
```

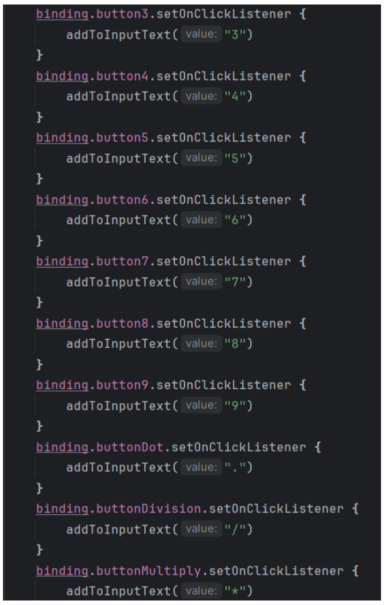
Логика:



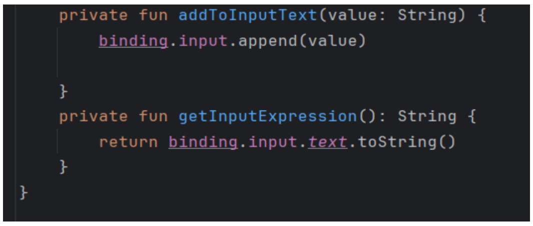
```kotlin
class MainActivity : AppCompatActivity() {
    lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        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 ^setOnApplyWindowInsetsListener
        }
        binding.buttonClear.setOnClickListener{
            binding.input.text = " "
            binding.output.text = " "
        }
        binding.buttonBracketLeft.setOnClickListener {
            addToInputText( value: "(")
        }
        binding.buttonBracketLeft.setOnClickListener {
            addToInputText( value: ")")
        }
        binding.button0.setOnClickListener {
            addToInputText( value: "0")
        }
        binding.button1.setOnClickListener {
            addToInputText( value: "1")
        }
        binding.button2.setOnClickListener {
            addToInputText( value: "2")
```



```kotlin
        }
        binding.buttonSubtraction.setOnClickListener {
            addToInputText( value: "-")
        }
        binding.buttonAddition.setOnClickListener {
            addToInputText( value: "+")
        }
        binding.buttonEquals.setOnClickListener {
            showResult()
        }
        binding.buttonPercent.setOnClickListener {
            addToInputText( value: "%")
        }
    }
    private fun showResult() {
        try {
            val expression = getInputExpression().replace( oldValue: "%",  newValue: "/100")
            val result = ExpressionBuilder(expression).build().evaluate()
            binding.output.text =
                DecimalFormat( pattern: "0.######").format(result).toString()
            binding.output.setTextColor(ContextCompat.getColor( context: this, R.color.neon_green))
        }
        catch (e:Exception){
            binding.output.text = "Ошибка"
            binding.output.setTextColor(ContextCompat.getColor( context: this, R.color.red))
        }
    }
    private fun addToInputText(value: String) {
        binding.input.append(value)
```



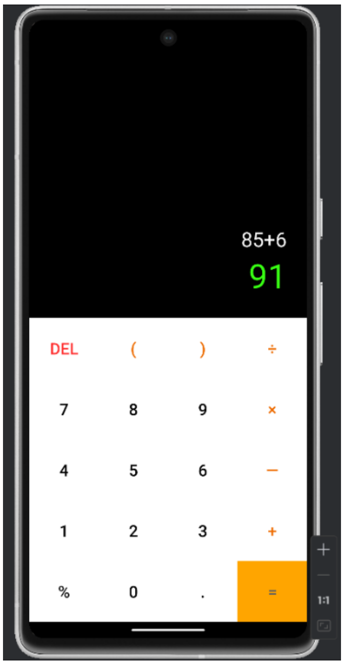
```kotlin
        binding.button3.setOnClickListener {
            addToInputText( value: "3")
        }
        binding.button4.setOnClickListener {
            addToInputText( value: "4")
        }
        binding.button5.setOnClickListener {
            addToInputText( value: "5")
        }
        binding.button6.setOnClickListener {
            addToInputText( value: "6")
        }
        binding.button7.setOnClickListener {
            addToInputText( value: "7")
        }
        binding.button8.setOnClickListener {
            addToInputText( value: "8")
        }
        binding.button9.setOnClickListener {
            addToInputText( value: "9")
        }
        binding.buttonDot.setOnClickListener {
            addToInputText( value: ".")
        }
        binding.buttonDivision.setOnClickListener {
            addToInputText( value: "/")
        }
        binding.buttonMultiply.setOnClickListener {
            addToInputText( value: "*")
```



```kotlin
    private fun addToInputText(value: String) {
        binding.input.append(value)

    }
    private fun getInputExpression(): String {
        return binding.input.text.toString()
    }
}
```

Итог:


85+6
91
DEL
(
)
÷
7
8
9
×
4
5
6
−
1
2
3
+
%
0
.
=