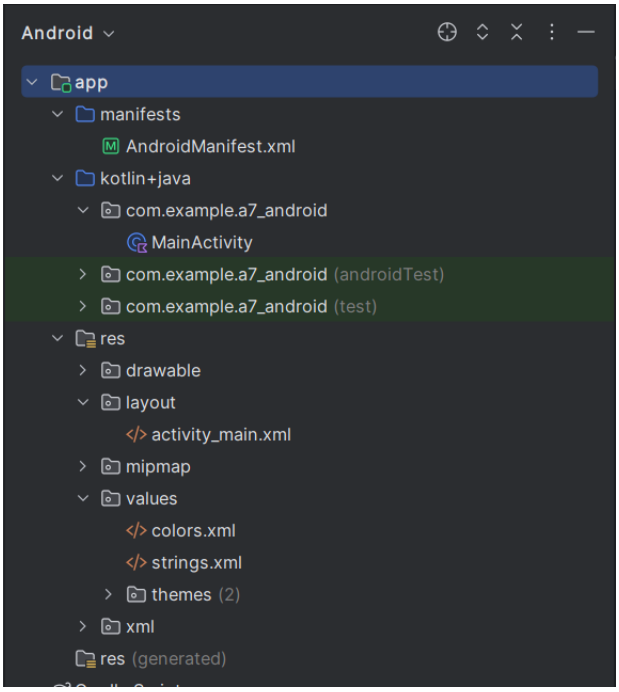
Практика 7:

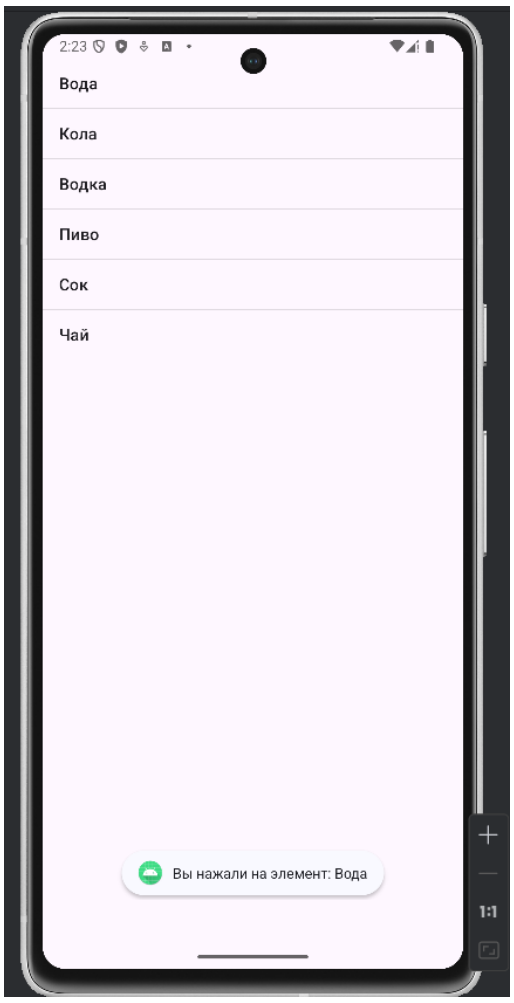
Структура:



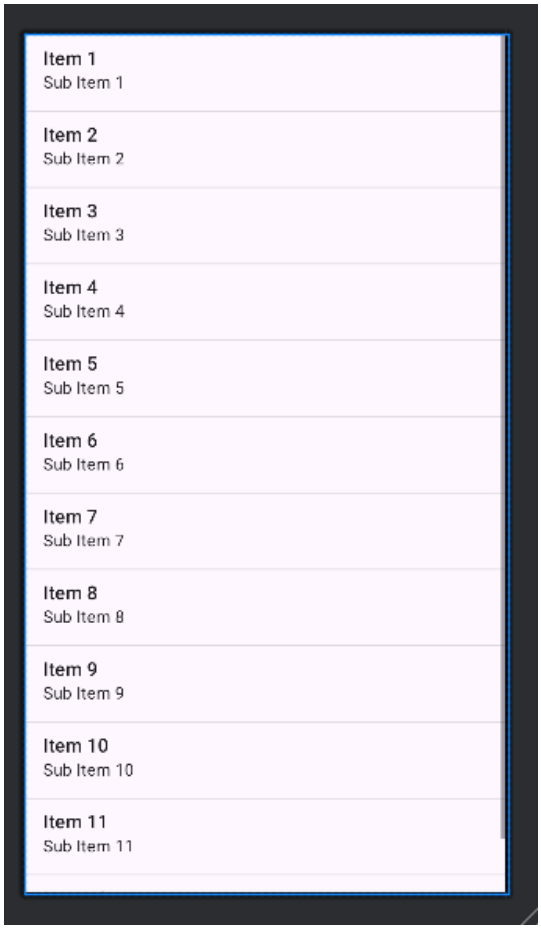
Логика:

```
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:

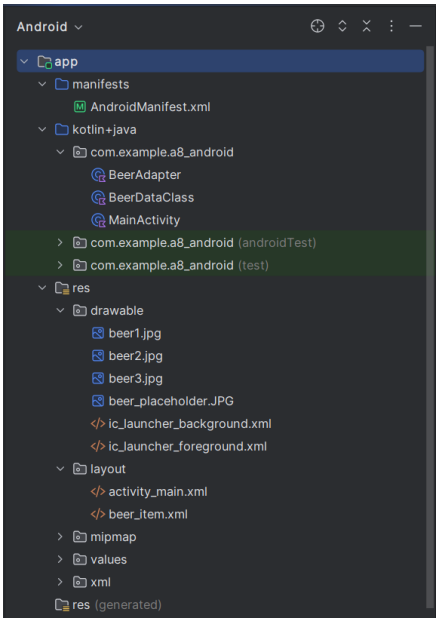


Дизайн:



Практика 8:

Структура:



MainActivity:

```
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)
        }
    }
}
```

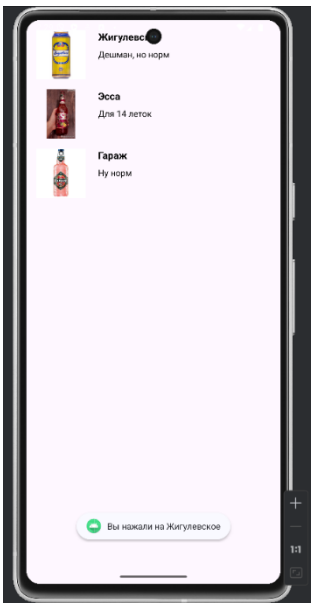
BeerAdapter:

```
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:

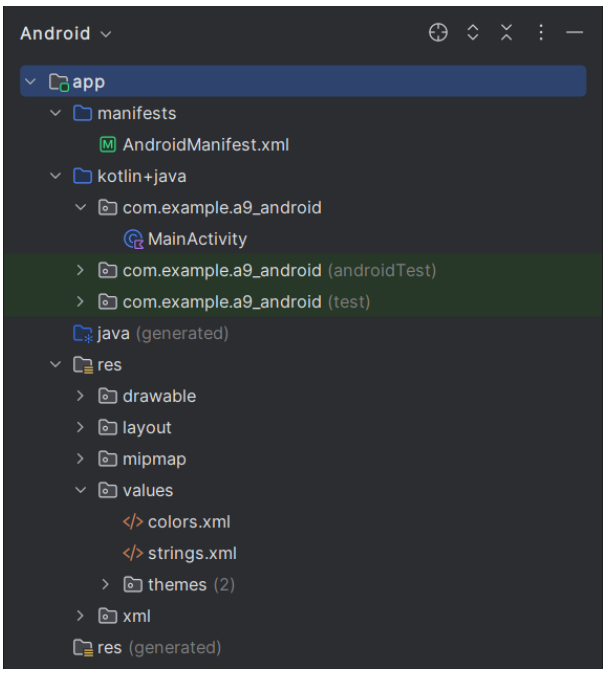
```
<resources>
    <string name="app_name">8_android</string>
    <string-array name="beer_titles">
        <item>Жигулевское</item>
        <item>Эсса</item>
        <item>Гараж</item>
    </string-array>
    <string-array name="beer_descriptions">
        <item>Дешман, но норм</item>
        <item>Для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

Итог:



Практика 9:

Структура:



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:

```
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
        }
        binding.buttonClear.setOnClickListener{
            binding.input.text = " "
            binding.output.text = " "
        }
        binding.buttonBracketLeft.setOnClickListener {
            addToInputText("(")
        }
        binding.buttonBracketLeft.setOnClickListener {
            addToInputText(")")
        }
        binding.button0.setOnClickListener {
            addToInputText("0")
        }
        binding.button1.setOnClickListener {
            addToInputText("1")
        }
        binding.button2.setOnClickListener {
            addToInputText("2")
        }
```

```
        }
        binding.buttonSubtraction.setOnClickListener {
            addToInputText("-")
        }
        binding.buttonAddition.setOnClickListener {
            addToInputText("+")
        }
        binding.buttonEquals.setOnClickListener {
            showResult()
        }
        binding.buttonPercent.setOnClickListener {
            addToInputText("%")
        }
    }
    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)
```

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

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

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

Итог: