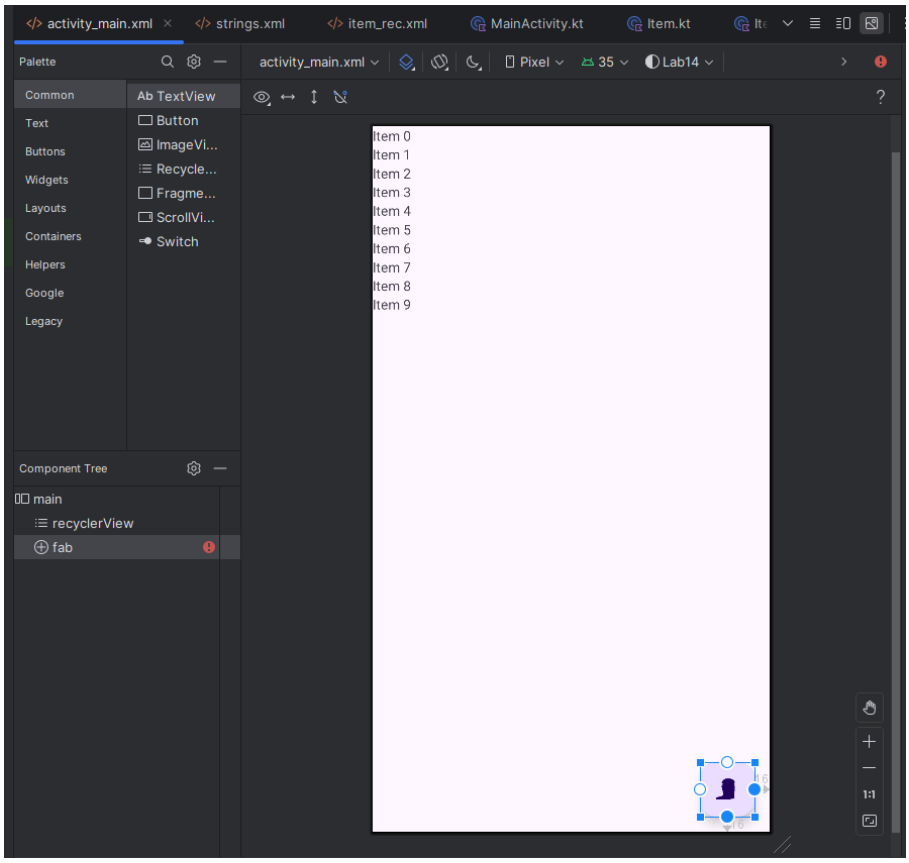
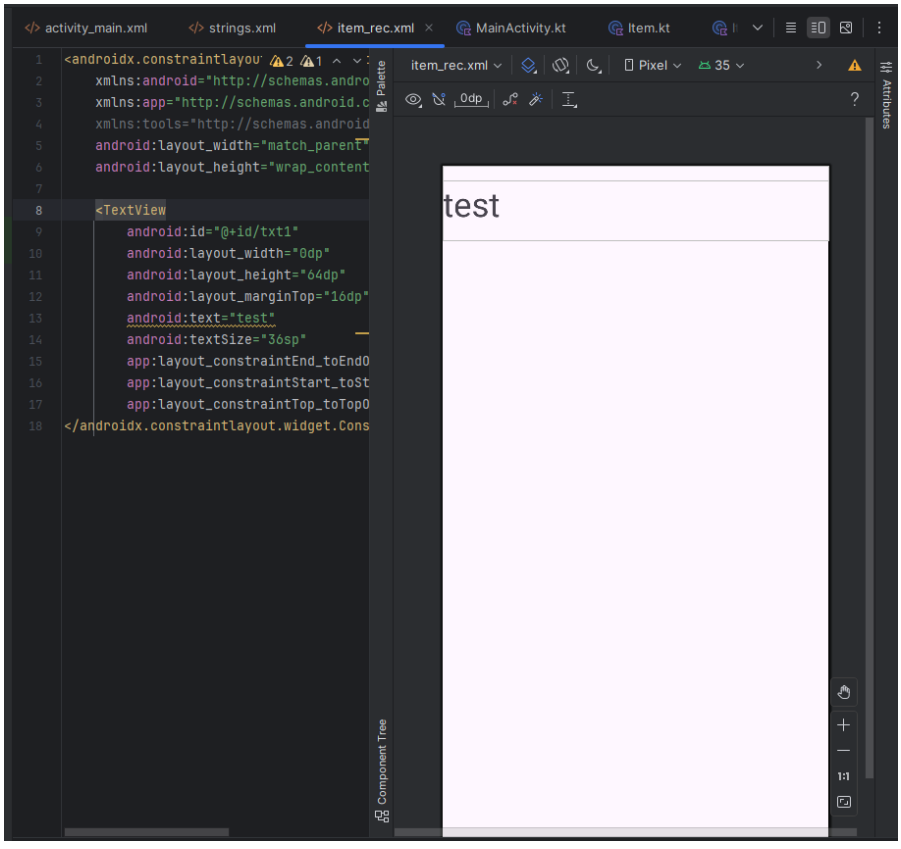
Лабораторная работа 14

1. Главная активность



2. Шаблон элемента RecycleView:



3. Дата класс, который будет хранить информацию о каждом элементе

```
data class Item(
    val text : String
)
```

4. Адаптер

```
class ItemAdapter(private val item: List<Item>) :
    RecyclerView.Adapter<ItemAdapter.ItemViewHolder>(){

    class ItemViewHolder(view: View) : RecyclerView.ViewHolder(view) {
        val textView : TextView = view.findViewById(R.id.txt1)
    }
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
            ItemViewHolder {
        val view =
            LayoutInflater.from(parent.context).inflate(R.layout.item_rec, parent, attachToRoot: false)
        return ItemViewHolder(view)
    }

    override fun onBindViewHolder(holder: ItemViewHolder, position: Int) {
        holder.textView.text = item[position].text
    }

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

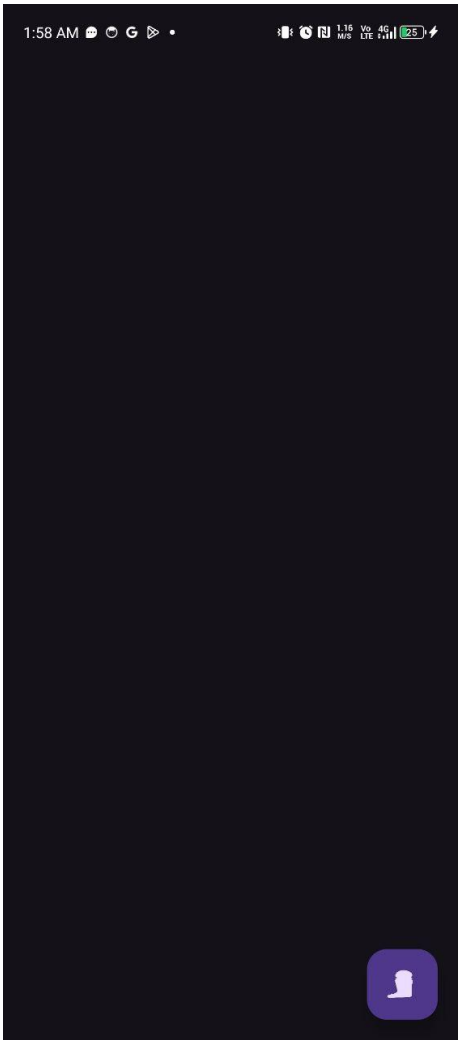
5. Код MainActivity, обработка ввода текста и добавления его как элемента RecycleView

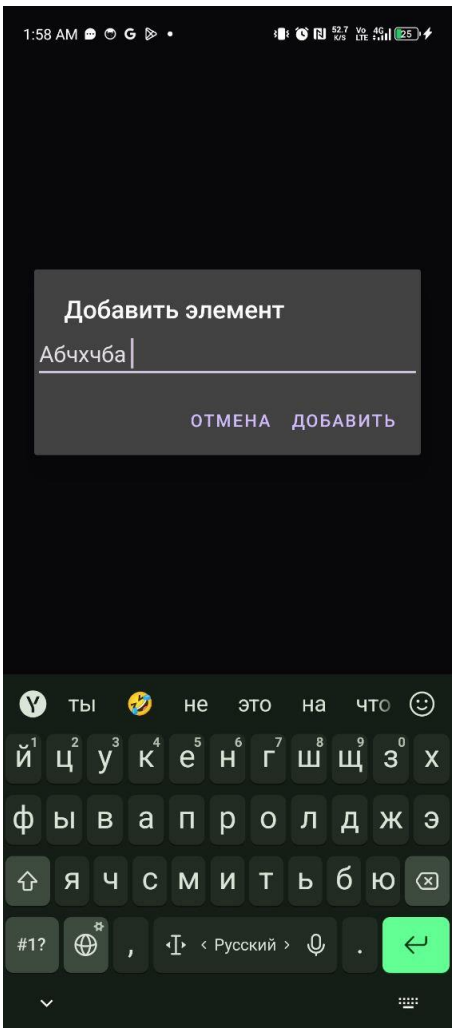
```
class MainActivity : AppCompatActivity() {
    // Входные компоненты для RecyclerView и FloatingActionButton
    private lateinit var recyclerView: RecyclerView
    private lateinit var fab: FloatingActionButton

    // Список элементов, которые будут отображаться в RecyclerView
    private val items: MutableList<Item> = mutableListOf()

    // Адаптер для управления отображением элементов в RecyclerView
    private lateinit var adapter: ItemAdapter


    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
        }
```

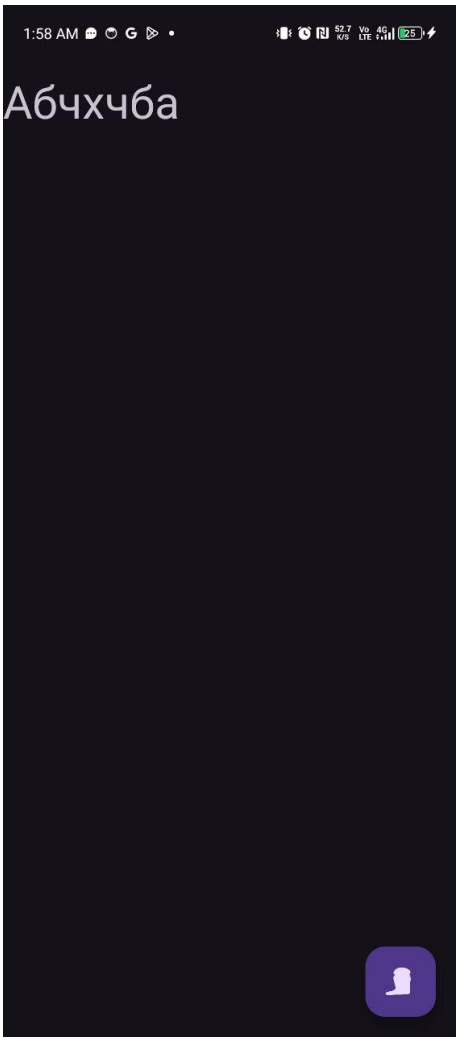
```
        recyclerView = findViewById(R.id.recyclerView)
        fab = findViewById(R.id.fab)

        adapter = ItemAdapter(items)
        recyclerView.layoutManager = LinearLayoutManager( context: this)
        recyclerView.adapter = adapter

        // Создаем адаптер, передавая ему список элементов
        adapter = ItemAdapter(items)
        recyclerView.layoutManager = LinearLayoutManager( context: this) // Устанавливаем LinearLayoutM
                recyclerView.adapter = adapter // Устанавливаем адаптер для

        // Устанавливаем обработчик нажатия на FloatingActionButton
        fab.setOnClickListener {
            showInputDialog() // Вызываем метод для отображения диалогового окна
        }
    }
```

```
    // Метод для отображения диалогового окна для ввода текста
    private fun showInputDialog() {
        val builder = AlertDialog.Builder( context: this) // Создаем экземпляр
        builder.setTitle("Добавить элемент") // Устанавливаем заголовок диалогового окна

        val input = EditText( context: this) // Создаем EditText для ввода текста
        builder.setView(input) // Устанавливаем созданный EditText в диалог

        // Устанавливаем кнопку "Добавить" в диалоговом окне
        builder.setPositiveButton( text: "Добавить") { dialog, _ ->
            val text = input.text.toString() // Получаем текст из EditText
            if (text.isNotEmpty()) { // Проверяем, не пустой ли он
                items.add(Item(text)) // Добавляем новый элемент в список
                adapter.notifyItemInserted( position: items.size - 1) // Уведомляем адаптер о добавлени
            }
            dialog.dismiss() // Закрываем диалог
        }
        // Устанавливаем кнопку "Отмена" в диалоговом окне
        builder.setNegativeButton( text: "Отмена") { dialog, _ -> dialog.cancel() }

        builder.show() // Показываем диалоговое окно
    }
}
```

6. Результат:




Добавить элемент
Абчхчба
ОТМЕНА
ДОБАВИТЬ
ты
не
это
на
что
Русский


Абчхчба