

## Задание 14

activity\_main

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/main">
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_margin="16dp"
        android:src="@drawable/baseline_add_24"/>
</RelativeLayout>
```

item:

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

item\_rec

```
<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="wrap_content">
    <TextView
        android:id="@+id/txt1"
        android:layout_width="0dp"
        android:layout_height="64dp"
        android:layout_marginTop="16dp"
        android:text="test"
        android:textSize="40sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

## ItemAdapter

```
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, false)  
        return ItemViewHolder(view)  
    }  
    override fun onBindViewHolder(holder: ItemViewHolder, position: Int) {  
        holder.textView.text = item[position].text  
    }  
    override fun getItemCount(): Int {  
        return item.size  
    }  
}
```

## MainActivity

```
class MainActivity : AppCompatActivity() {  
    private lateinit var fab: FloatingActionButton  
    private val items: MutableList<Item> = mutableListOf()  
    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(this)  
        recyclerView.adapter = adapter  
        adapter = ItemAdapter(items)  
        recyclerView.layoutManager = LinearLayoutManager(this)  
        recyclerView.adapter = adapter  
        fab.setOnClickListener {  
            showInputDialog()  
        }  
    }  
    private fun showInputDialog() {  
        val builder = AlertDialog.Builder(this)  
        builder.setTitle("Добавить элемент")  
        val input = EditText(this)  
        builder.setView(input)  
        builder.setPositiveButton("Добавить") { dialog, _ ->  
            val text = input.text.toString()  
            if (text.isNotEmpty()) {  
                items.add(Item(text))  
                adapter.notifyItemInserted(items.size - 1)  
            }  
            dialog.dismiss()  
        }  
        builder.setNegativeButton("Отмена") { dialog, _ -> dialog.cancel() }  
    }  
}
```

Результат

