

**Московский государственный технический
университет им. Н.Э. Баумана**

**Факультет «Радиотехнический»
Кафедра «Системы обработки информации и управления»**

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №4

«Работа со списком на языке kotlin»

Выполнил:

студент группы РТ5-31Б:

Паншин М.В.

Проверил:

преподаватель кафедры ИУ5

Гапанюк Ю.Е.

Москва, 2024 г.

Постановка задачи

Требуется разработать небольшое приложение, которое отображает прямоугольники с числами в несколько колонок. Для портретной ориентации - требуется отображать прямоугольники(или карточки) в 3 колонки, для горизонтальной ориентации требуется отображать прямоугольники в 4 колонки. Отображаемое число - индекс элемента (считать можно как от 0 так и от 1). Четные элементы должны иметь красный фон, нечетные - синий. Под списком должна находиться кнопка (если используется `ActionButton`, то она может находиться в нижней части списка), тап по которой добавляет новый прямоугольник к списку. Кнопка (даже `Action Button`) не должна перекрывать прямоугольники! Переворот экрана не должен сбивать количество прямоугольников. Так же реализовать кнопку удаления элемента из начала списка

Текст программы

Файл `MainActivity.kt`

```
package com.example.panshin_homework

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.runtime.mutableStateListOf

class MainActivity : ComponentActivity() {
    private var listSize = 0
    private val dataList = mutableStateListOf<String>()
    private var itemsNumber = 1
    override fun onCreate(savedInstanceState: Bundle?) {

        if (savedInstanceState != null){
            listSize = savedInstanceState.getInt("key_size")
            itemsNumber = savedInstanceState.getInt("items_number")
            for(i in itemsNumber ..listSize + itemsNumber - 1){
                dataList.add("$i")
            }
        }

        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
    }
}
```

```

        setContent {
            ListScreen(dataList, addItem = {addItem()}, deleteItem =
{deleteItem() })
        }

        override fun onSaveInstanceState(outState: Bundle) {
            super.onSaveInstanceState(outState)
            outState.putInt("items_number", itemsNumber)
            outState.putInt("key_size",dataList.size)

        }

        private fun addItem(){
            listSize++
            val gg = listSize + itemsNumber - 1
            dataList.add("$gg")
        }
        private fun deleteItem(){
            dataList.removeAt(0)
            itemsNumber++
            listSize--

        }

    }
}

```

Файл ListScreen.kt

```

package com.example.panshin_homework

import android.content.res.Configuration
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row

import androidx.compose.foundation.layout.aspectRatio
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.lazy.grid.GridCells
import androidx.compose.foundation.lazy.grid.LazyVerticalGrid
import androidx.compose.foundation.lazy.grid.items
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.Add
import androidx.compose.material.icons.filled.Clear
import androidx.compose.material.icons.filled.Delete

import androidx.compose.material3.FloatingActionButton
import androidx.compose.material3.Icon

import androidx.compose.material3.Text
import androidx.compose.runtime.Composable

import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip

```

```

import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.LocalConfiguration
import androidx.compose.ui.res.colorResource
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp

@Composable
fun ListScreen(dataList: List<String>, addItem: () -> Unit, deleteItem: () ->
Unit) {

    val currentOrientationLandscape =
        (LocalConfiguration.current.orientation ==
Configuration.ORIENTATION_LANDSCAPE)
    Column(modifier = Modifier.fillMaxSize()) {
        LazyVerticalGrid(
            columns = GridCells.Fixed(if (currentOrientationLandscape) 4
else 3),
            modifier = Modifier
                .fillMaxWidth()
                .weight(1f)

        ) {
            items(dataList) { item ->
                ListItem(item)
            }
        }

        Row(modifier = Modifier
            .fillMaxWidth()
            , horizontalArrangement = Arrangement.Center,
        ) {
            FloatingActionButton(
                onClick = deleteItem,
                modifier = Modifier
                    .padding(10.dp),
                containerColor = colorResource(R.color.light_purple)
            ) {
                Icon(Icons.Filled.Delete, "Удалить")
            }
            FloatingActionButton(
                onClick = addItem,
                modifier = Modifier
                    .padding(10.dp),
                containerColor = colorResource(R.color.light_purple)
            ) {
                Icon(Icons.Filled.Add, "Добавить")
            }
        }
    }
}

fun Modifier.conditional(condition: Boolean, modifier: Modifier.() ->
Modifier) : Modifier {
    return if (condition){
        then(modifier(Modifier))
    }
    else{
        this
    }
}

```

```

@Composable
fun ListItem(number: String) {

    Box(contentAlignment = Alignment.Center,
        modifier = Modifier
            .fillMaxSize()
            .padding(8.dp)
            .aspectRatio(1f)
            .clip(shape = RoundedCornerShape(12.dp))
            .background(Color.Red)
            .conditional(number.toInt() % 2 == 1) {
                background(Color.Blue)
            }

    ) {
        Text(
            text = number,
            color = Color.White,
            fontSize = 32.sp
        )
    }
}

```

Результат работы программы



