Министерство науки и высшего образования Республики Казахстана

Некоммерческое акционерное общество

«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ имени Гумарбека Даукеева»

Институт автоматизации и информационных технологий

Кафедра IT-инженерии и искусственного интеллекта

# **ОТЧЕТ**

# **по лабораторной работе №3**

# По дисциплине: Разработка программных приложений на базе Android

На тему: Создание View-компонент в рабочем приложении

Образовательная программа: 6B06103 – Вычислительная техника и программное обеспечение

Выполнил: Кенес А. Ж. Группа ВТ(ПИ)у-23-4

Проверила: Старший преподаватель Ахсутова А.А.

\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_\_» \_\_\_\_\_\_\_\_2024 г.

Алматы 2024

**Цель работы:** Добавляем компоненты на экран во время работы приложения

**Задание**

Создать три приложения, в первом приложений нужно выровнять текст по трём сторонам: слева, справа и по середине, во втором приложений сделать ползунок, который при перетаскивании влево или вправо уменьшает кнопки, третье приложение, сделать простой калькулятор.

**Ход работы**

**Задание 1,2,3**

Создать три приложения, в первом приложений нужно выровнять текст по трём сторонам: слева, справа и по середине, во втором приложений сделать ползунок, который при перетаскивании влево или вправо уменьшает кнопки, третье приложение, сделать простой калькулятор.

**Листинг программы**

package com.github.shortmikelon.lab3

import android.os.Bundle

import android.util.Log

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.activity.enableEdgeToEdge

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.Spacer

import androidx.compose.foundation.layout.fillMaxSize

import androidx.compose.foundation.layout.fillMaxWidth

import androidx.compose.foundation.layout.height

import androidx.compose.foundation.layout.padding

import androidx.compose.foundation.layout.size

import androidx.compose.foundation.layout.width

import androidx.compose.foundation.text.KeyboardOptions

import androidx.compose.material.icons.Icons

import androidx.compose.material.icons.filled.ArrowBack

import androidx.compose.material3.Button

import androidx.compose.material3.Icon

import androidx.compose.material3.IconButton

import androidx.compose.material3.MaterialTheme

import androidx.compose.material3.RadioButton

import androidx.compose.material3.Scaffold

import androidx.compose.material3.Slider

import androidx.compose.material3.SliderDefaults

import androidx.compose.material3.Text

import androidx.compose.material3.TextField

import androidx.compose.material3.TopAppBar

import androidx.compose.runtime.Composable

import androidx.compose.runtime.getValue

import androidx.compose.runtime.mutableIntStateOf

import androidx.compose.runtime.mutableStateOf

import androidx.compose.runtime.remember

import androidx.compose.runtime.setValue

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.text.input.KeyboardType

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import com.github.shortmikelon.lab3.ui.theme.Lab3Theme

private const val HOME\_SCREEN = 1

private const val RADIO\_BUTTON\_SCREEN = 2

private const val SCROLL\_SCREEN = 3

private const val CALCULATOR\_SCREEN = 4

class MainActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

enableEdgeToEdge()

setContent {

Lab3Theme {

App()

}

}

}

}

@Composable

private fun App() {

var screenIndex by remember { mutableIntStateOf(1) }

var selectedRadioButton by remember { mutableIntStateOf(3) }

Scaffold {

Box(

modifier = Modifier.padding(it)

) {

when (screenIndex) {

HOME\_SCREEN -> HomeScreen(

moveToRadioButtonScreen = {

screenIndex = RADIO\_BUTTON\_SCREEN

},

moveToCalculatorScreen = {

screenIndex = CALCULATOR\_SCREEN

},

moveToScrollScreen = {

screenIndex = SCROLL\_SCREEN

}

)

RADIO\_BUTTON\_SCREEN -> CreateButtonScreen(

onBackNavigate = { screenIndex = HOME\_SCREEN },

selectedRadioButton = selectedRadioButton,

onRadioButtonClicked = { selectedRadioButton = it }

)

SCROLL\_SCREEN -> ScrollScreen(onBackNavigate = { screenIndex = HOME\_SCREEN })

CALCULATOR\_SCREEN -> CalculatorScreen(onBackNavigate = { screenIndex = HOME\_SCREEN })

}

}

}

}

@Composable

private fun HomeScreen(

moveToRadioButtonScreen: () -> Unit,

moveToScrollScreen: () -> Unit = { },

moveToCalculatorScreen: () -> Unit = {}

) {

Column(

horizontalAlignment = Alignment.CenterHorizontally,

verticalArrangement = Arrangement.Center,

modifier = Modifier.fillMaxSize()

) {

Button(onClick = moveToRadioButtonScreen) {

Text(text = "Radio Button Screen")

}

Button(onClick = moveToScrollScreen) {

Text(text = "Scroll Screen")

}

Button(onClick = moveToCalculatorScreen) {

Text(text = "Calculation screen")

}

}

}

@Composable

private fun CreateButtonScreen(

onBackNavigate: () -> Unit,

selectedRadioButton: Int,

onRadioButtonClicked: (Int) -> Unit,

) {

var text by remember { mutableStateOf("") }

var buttons by remember { mutableStateOf(mapOf<String, Int>()) }

Column(

modifier = Modifier.fillMaxSize()

) {

Row(

horizontalArrangement = Arrangement.SpaceBetween,

modifier = Modifier

.fillMaxWidth()

.background(MaterialTheme.colorScheme.primary)

) {

IconButton(onClick = { onBackNavigate() }) {

Icon(

imageVector = Icons.Filled.ArrowBack,

contentDescription = null,

tint = MaterialTheme.colorScheme.onPrimary

)

}

Text(

text = "Create button screen",

color = MaterialTheme.colorScheme.onPrimary,

modifier = Modifier.padding(top = 15.dp, end = 15.dp)

)

}

Row {

Row {

RadioButton(

selected = selectedRadioButton == 1,

onClick = { onRadioButtonClicked(1) }

)

Text(

text = "left",

modifier = Modifier.padding(top = 12.dp)

)

}

Row {

RadioButton(

selected = selectedRadioButton == 2,

onClick = { onRadioButtonClicked(2) }

)

Text(

text = "center",

modifier = Modifier.padding(top = 12.dp)

)

}

Row {

RadioButton(

selected = selectedRadioButton == 3,

onClick = { onRadioButtonClicked(3) }

)

Text(text = "right", modifier = Modifier.padding(top = 12.dp))

}

}

Row {

TextField(

value = text,

onValueChange = { text = it },

modifier = Modifier.size(width = 160.dp, height = 50.dp)

)

Button(

modifier = Modifier.size(width = 90.dp, height = 50.dp),

onClick = {

val copy = buttons.toMutableMap()

copy[text] = selectedRadioButton

text = ""

buttons = copy

Log.d("flkdsjafl", copy.toString())

}

) {

Text(text = "Create")

}

Button(

modifier = Modifier.size(width = 90.dp, height = 50.dp),

onClick = {

val copy = buttons.toMutableMap()

copy.clear()

buttons = copy

Log.d("flkdsjafl", copy.toString())

}

) {

Text(text = "Clear")

}

}

Column(

modifier = Modifier.fillMaxSize()

) {

buttons.forEach {

Row(

horizontalArrangement = getArrangement(it.value),

modifier = Modifier.fillMaxWidth()

) {

Button(onClick = {}) {

Text(it.key)

}

}

}

}

}

}

@Composable

private fun getArrangement(a: Int): Arrangement.Horizontal {

return when (a) {

1 -> Arrangement.Start

2 -> Arrangement.Center

3 -> Arrangement.End

else -> throw IllegalArgumentException()

}

}

@Composable

private fun ScrollScreen(

onBackNavigate: () -> Unit

) {

Column(

modifier = Modifier.fillMaxSize()

) {

Row(

horizontalArrangement = Arrangement.SpaceBetween,

modifier = Modifier

.fillMaxWidth()

.background(MaterialTheme.colorScheme.primary)

) {

IconButton(onClick = { onBackNavigate() }) {

Icon(

imageVector = Icons.Filled.ArrowBack,

contentDescription = null,

tint = MaterialTheme.colorScheme.onPrimary

)

}

Text(

text = "Resize button screen",

color = MaterialTheme.colorScheme.onPrimary,

modifier = Modifier.padding(top = 15.dp, end = 15.dp)

)

}

Column(

modifier = Modifier

.fillMaxSize()

.padding(16.dp),

verticalArrangement = Arrangement.Center,

horizontalAlignment = Alignment.CenterHorizontally

) {

// Slider value state

val sliderPosition = remember { mutableStateOf(0.5f) }

// Slider component

Slider(

value = sliderPosition.value,

onValueChange = { sliderPosition.value = it },

modifier = Modifier

.fillMaxWidth()

.padding(horizontal = 16.dp),

colors = SliderDefaults.colors(

thumbColor = Color.Gray,

activeTrackColor = Color.Yellow,

inactiveTrackColor = Color.Gray

)

)

Spacer(modifier = Modifier.height(16.dp))

Row(

modifier = Modifier.fillMaxWidth(),

horizontalArrangement = Arrangement.SpaceBetween

) {

Button(

onClick = { },

modifier = Modifier.weight(sliderPosition.value)

) {

Text(text = "Button1", fontSize = 16.sp)

}

Spacer(modifier = Modifier.width(8.dp))

Button(

onClick = { },

modifier = Modifier.weight(1f - sliderPosition.value)

) {

Text(text = "Button2", fontSize = 16.sp)

}

}

}

}

}

@Composable

private fun CalculatorScreen(

onBackNavigate: () -> Unit

) {

var rightText by remember { mutableStateOf("") }

var leftText by remember { mutableStateOf("") }

var result by remember { mutableStateOf("") }

Column(

modifier = Modifier.fillMaxSize()

) {

Row(

horizontalArrangement = Arrangement.SpaceBetween,

modifier = Modifier

.fillMaxWidth()

.background(MaterialTheme.colorScheme.primary)

) {

IconButton(onClick = { onBackNavigate() }) {

Icon(

imageVector = Icons.Filled.ArrowBack,

contentDescription = null,

tint = MaterialTheme.colorScheme.onPrimary

)

}

Text(

text = "Calculator screen",

color = MaterialTheme.colorScheme.onPrimary,

modifier = Modifier.padding(top = 15.dp, end = 15.dp)

)

}

Row(

horizontalArrangement = Arrangement.SpaceAround,

modifier = Modifier.fillMaxWidth()

) {

TextField(

value = leftText,

onValueChange = { leftText = it },

keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Number),

modifier = Modifier.width(100.dp)

)

TextField(

value = rightText,

onValueChange = { rightText = it },

keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Number),

modifier = Modifier.width(100.dp)

)

}

Row(

horizontalArrangement = Arrangement.SpaceAround,

modifier = Modifier.fillMaxWidth()

) {

Button(onClick = {

if (leftText.isEmpty()) leftText = "0.0"

if (rightText.isEmpty()) rightText = "0.0"

result = "$leftText + $rightText = ${leftText.toDouble() + rightText.toDouble()}" }

) {

Text("+")

}

Button(onClick = {

if (leftText.isEmpty()) leftText = "0.0"

if (rightText.isEmpty()) rightText = "0.0"

result = "$leftText - $rightText = ${leftText.toDouble() - rightText.toDouble()}"

}) {

Text("-")

}

Button(onClick = {

if (leftText.isEmpty()) leftText = "0.0"

if (rightText.isEmpty()) rightText = "0.0"

result = "$leftText \* $rightText = ${leftText.toDouble() \* rightText.toDouble()}"

}) {

Text("\*")

}

Button(onClick = {

if (leftText.isEmpty()) leftText = "0.0"

if (rightText.isEmpty()) rightText = "0.0"

result = "$leftText / $rightText = ${leftText.toDouble() / rightText.toDouble()}"

}) {

Text("/")

}

}

Row(

horizontalArrangement = Arrangement.Center,

modifier = Modifier.fillMaxWidth()

) {

Text(result)

}

}

}

**Входные/выходные данные**

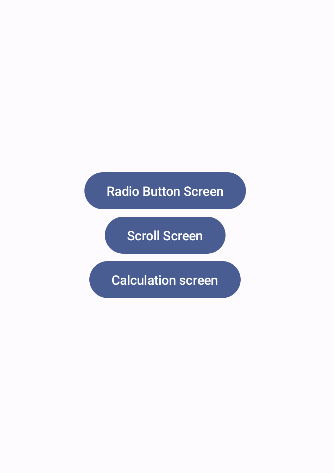


Рисунок 1

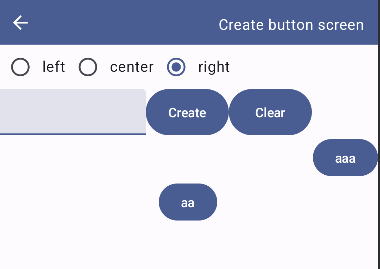


Рисунок 2

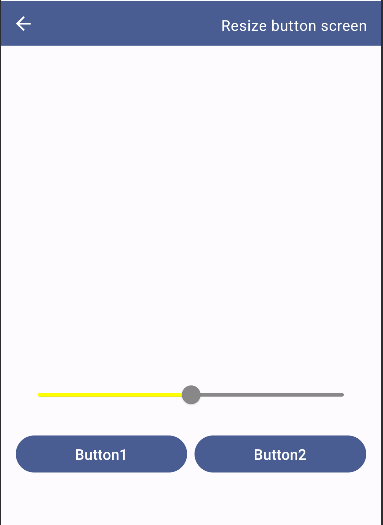


Рисунок 3

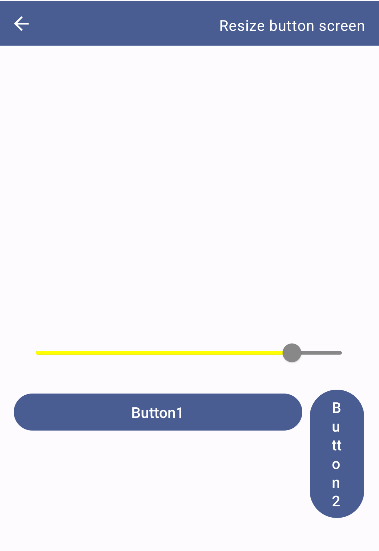


Рисунок 4

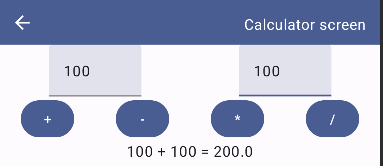


Рисунок 5

**Заключение**

Этот код представляет собой простое Android-приложение с использованием Jetpack Compose для построения пользовательского интерфейса. Приложение включает несколько экранов, реализованных через систему навигации с переключением между ними на основе состояния.

Основные функции приложения:

1. Главный экран (HomeScreen): содержит кнопки для перехода на другие экраны, такие как экран с радиокнопками, калькулятор и экран прокрутки.
2. Экран с радиокнопками (CreateButtonScreen): позволяет пользователю выбрать расположение кнопки (лево, центр или право) с помощью радиокнопок, а также добавлять или удалять динамически создаваемые кнопки. Созданные кнопки отображаются на экране в зависимости от выбранного расположения.
3. Экран с ползунком (ScrollScreen): содержит ползунок (Slider), который изменяет ширину двух кнопок на экране в зависимости от положения ползунка.
4. Калькулятор (CalculatorScreen): предоставляет базовые арифметические операции (сложение, вычитание, умножение и деление) над двумя введенными пользователем числами.

Приложение использует компоненты Material Design 3 (MaterialTheme, Button, TextField и другие), что обеспечивает современный и удобный интерфейс. Логика навигации между экранами выполнена с использованием состояния (screenIndex), что позволяет легко добавлять и переключаться между экранами.

**Список литературы**

1. Динамически меняем параметры элементов экрана в андроид-приложении | Android Studio. URL: <https://www.fandroid.info/urok-18-dinamicheski-menyaem-parametry-elementov-ekrana-v-android-prilozhenii-android-studio/>

2. Делаем приложение: калькулятор для андроид | Android Studio. URL: <https://www.fandroid.info/urok-19-delaem-prilozhenie-kalkulyator-dlya-android-android-studio/>