МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

Отчет по Модулю 2 – Основы языка Котлин

по дисциплине «Проектирования и разработка мобильных приложений на языке Котлин»

Выполнили:
Студенты группы ИКБО-65-23 Олефиров Г.Г.

Проверил: Егоров Н.С.

2025 г.

Практическое задание

Задание 1

1. Напишите compose-функцию Hello(name: String?), куда в качестве параметра можно было бы передавать имя для приветственной надписи. Если в качестве параметра передано значение null, должна выводиться надпись «Имя не задано». 2. Вызовите функцию в активити. 3. Отдельно скомпилируйте проект, передав в функцию конкретное имя, и отдельно — передав значение null.

```
package com.example.a2modul

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.a2modul.ui.theme._2ModulTheme
import androidx.compose.ui.graphics.Color

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        // Вариант 1: с именем
        setContent {
            _2ModulTheme {
                Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->
                    Hello(
                        name = "Анна",
                        modifier = Modifier
                            .padding(innerPadding)
                            .fillMaxSize()
                    )
                }
            }
        }
    }
}

// ЗАДАНИЕ 1: Простая функция Hello
@Composable
fun Hello(name: String?, modifier: Modifier = Modifier) {
    Surface(
        color = Color.Yellow,
        modifier = modifier
    ) {
        Text(
            text = if (name != null) "Привет, $name!" else "Имя не задано",
```

```
            modifier = Modifier.padding(16.dp)
        )
    }
}

// Preview для проверки
@Preview(showBackground = true)
@Composable
fun HelloPreview() {
    _2ModulTheme {
        Hello(name = "Иван")
    }
}

@Preview(showBackground = true)
@Composable
fun HelloPreviewNull() {
    _2ModulTheme {
        Hello(name = null)
    }
}
```

Листинг 1 – MainActivity

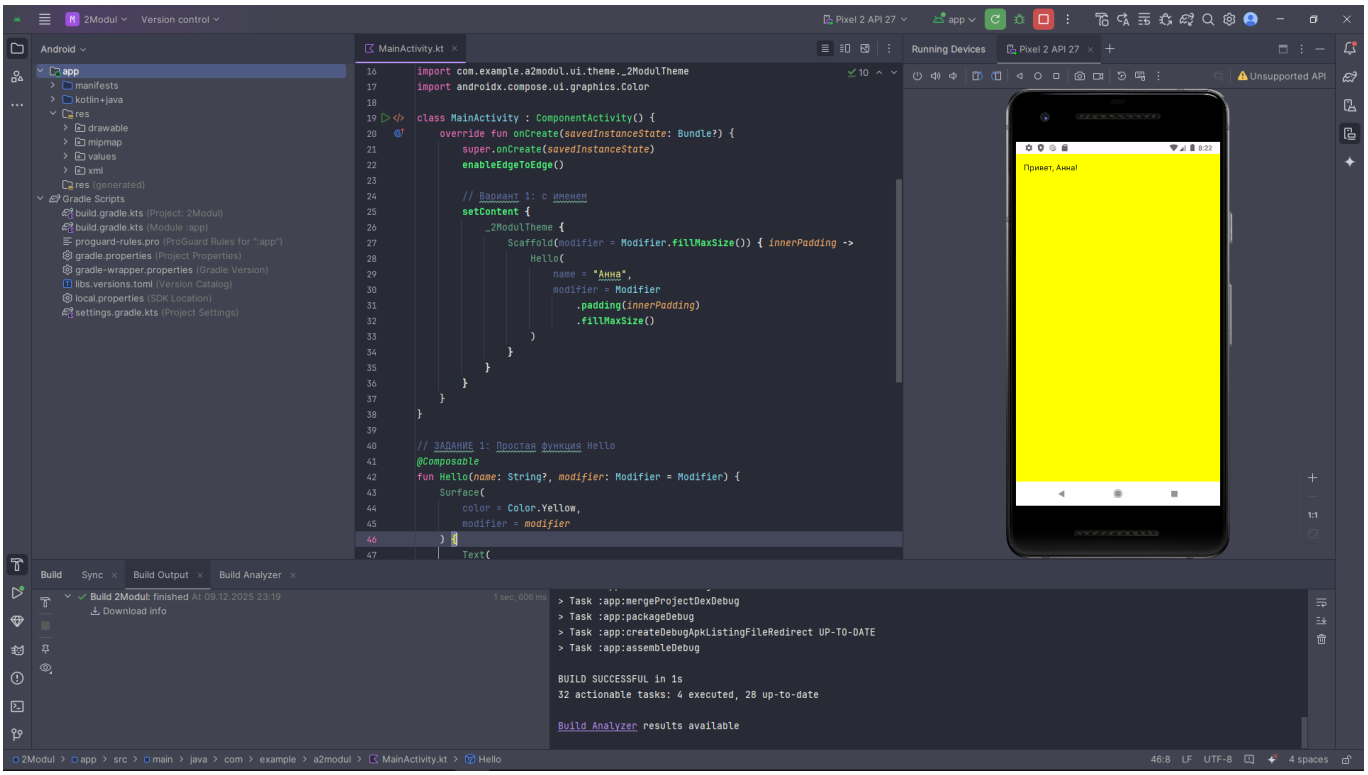


Рисунок 1 – MainActivity

Задание 2

Создайте для функции HelloPreview из предыдущего урока три макета предпросмотра с показом системного UI: 1. portrait — с портретной ориентацией экрана. 2. landscape — с альбомной ориентацией экрана. 3. round — с круглым дисплеем размером 200dpx200dp, а также жёлтым задним фоном (dp — единица измерения размера экрана, про неё будет отдельный урок).

```
package com.example.a2modul

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.size
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Devices
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.a2modul.ui.theme._2ModulTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        setContent {
            _2ModulTheme {
                Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->
                    Hello(
                        name = "Анна",
                        modifier = Modifier
                            .padding(innerPadding)
                            .fillMaxSize()
                    )
                }
            }
        }
    }
}

@Composable
fun Hello(name: String?, modifier: Modifier = Modifier) {
    Surface(
        color = Color.Yellow,
        modifier = modifier
    ) {
        Text(
            text = if (name != null) "Привет, $name!" else "Имя не задано",
            modifier = Modifier.padding(16.dp)
        )
    }
}

// ЗАДАНИЕ 2: Три макета предпросмотра

// 1. Portrait (портретная ориентация)
@Preview(
    device = Devices.PIXEL_4,
    showSystemUi = true,
```

```
    name = "Portrait Preview"
)
@Composable
fun HelloPreviewPortrait() {
    _2ModulTheme {
        Hello(name = "Иван (Portrait)")
    }
}

// 2. Landscape (альбомная ориентация)
@Preview(
    device = Devices.PIXEL_4,
    showSystemUi = true,
    name = "Landscape Preview"
)
@Composable
fun HelloPreviewLandscape() {
    _2ModulTheme {
        Hello(name = "Мария (Landscape)")
    }
}

// 3. Round (круглый дисплей 200dp x 200dp с желтым фоном)
@Preview(
    widthDp = 200,
    heightDp = 200,
    name = "Round 200dp Preview"
)
@Composable
fun HelloPreviewRound() {
    // Желтый задний фон
    Box(
        modifier = Modifier
            .size(200.dp)
            .background(Color.Yellow),
        contentAlignment = Alignment.Center
    ) {
        // Сама функция Hello
        Hello(
            name = "Алексей (Round)",
            modifier = Modifier.fillMaxSize()
        )
    }
}


@Composable
fun HelloPreviewWearRound() {
    _2ModulTheme {
        Box(
            modifier = Modifier
                .fillMaxSize()
                .background(Color.Yellow)
        ) {
            Hello(name = "Олег (Wear)")
        }
    }
}
```

Листинг 2 – MainActivity

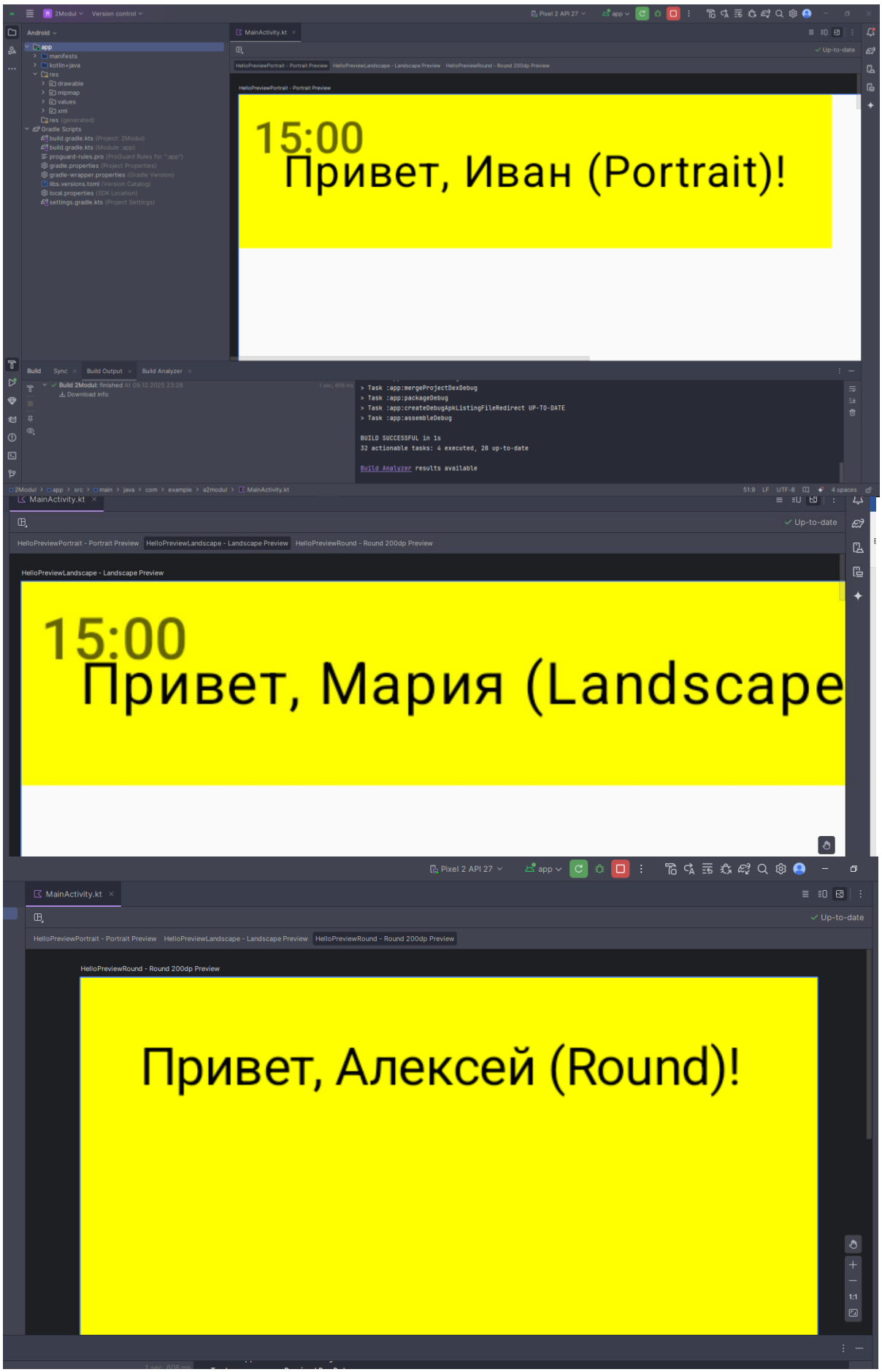


Рисунок 2 – MainActivity

Задание 3

Добавьте в ресурсы строку «Jetpack Compose — это современный набор инструментов для создания пользовательского интерфейса (UI) на Android, разработанный Google». 32 Разместите в режиме предпросмотра текст с этой строкой в нескольких вариантах: 1. Зелёный цвет шрифта, размер шрифта — 16.sp, стиль шрифта — курсив, выравнивание — по центру. 2. Для текста отводится одна строка, в завершении которой — многоточие (как признак того, что текст не умещается). 3. Перед первой строкой — отступ 48.sp, размер шрифта — 24.sp, чёрный цвет шрифта, зелёный цвет фона, текст подчёркнут.

```
package com.example.a2modul

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontStyle
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.text.style.TextDecoration
import androidx.compose.ui.tooling.preview.Devices
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.a2modul.ui.theme._2ModulTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        setContent {
            _2ModulTheme {
                Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->
                    Column(
                        modifier = Modifier
                            .padding(innerPadding)
                            .fillMaxSize()
                            .padding(16.dp)
                    ) {
                        // Задание 1
                        Hello(
                            name = "Анна",
                            modifier = Modifier.fillMaxWidth()
                        )

                        Spacer(modifier = Modifier.height(32.dp))

                        // Задание 3 - вариант 1
                        TextVariant1()
```

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

                        // Задание 3 - вариант 2
                        TextVariant2()

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

                        // Задание 3 - вариант 3
                        TextVariant3()
                    }
                }
            }
        }
    }
}

@Composable
fun Hello(name: String?, modifier: Modifier = Modifier) {
    Surface(
        color = Color.Yellow,
        modifier = modifier
    ) {
        Text(
            text = if (name != null) "Привет, $name!" else "Имя не задано",
            modifier = Modifier.padding(16.dp)
        )
    }
}

// ЗАДАНИЕ 3: Вариант 1
@Composable
fun TextVariant1() {
    Surface(
        color = Color.White,
        modifier = Modifier.fillMaxWidth()
    ) {
        Column(
            modifier = Modifier.padding(16.dp),
            horizontalAlignment = Alignment.CenterHorizontally
        ) {
            Text(
                text = stringResource(R.string.compose_description),
                color = Color.Green,
                fontSize = 16.sp,
                fontStyle = FontStyle.Italic,
                textAlign = TextAlign.Center,
                modifier = Modifier.fillMaxWidth()
            )
        }
    }
}

// ЗАДАНИЕ 3: Вариант 2
@Composable
fun TextVariant2() {
    Surface(
        color = Color.White,
        modifier = Modifier.fillMaxWidth()
    ) {
        Column(
            modifier = Modifier.padding(16.dp)
        ) {
            Text(
                text = stringResource(R.string.compose_description),
```

```
                color = Color.Black,
                fontSize = 16.sp,
                maxLines = 1,
                overflow = androidx.compose.ui.text.style.TextOverflow.Ellipsis,
                modifier = Modifier.fillMaxWidth()
            )
        }
    }
}

// ЗАДАНИЕ 3: Вариант 3
@Composable
fun TextVariant3() {
    Surface(
        color = Color.White,
        modifier = Modifier.fillMaxWidth()
    ) {
        Box(
            modifier = Modifier
                .fillMaxWidth()
                .background(Color.Green)
                .padding(vertical = 8.dp)
        ) {
            Text(
                text = stringResource(R.string.compose_description),
                fontSize = 24.sp,
                color = Color.Black,
                textDecoration = TextDecoration.Underline,
                modifier = Modifier.padding(start = 48.dp, top = 16.dp, bottom = 16.dp)
            )
        }
    }
}

// ЗАДАНИЕ 2: Preview
@Preview(
    device = Devices.PIXEL_4,
    showSystemUi = true,
    name = "Portrait Preview"
)
@Composable
fun HelloPreviewPortrait() {
    _2ModulTheme {
        Column(
            modifier = Modifier
                .fillMaxSize()
                .padding(16.dp)
        ) {
            Hello(name = "Иван (Portrait)")
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant1()
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant2()
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant3()
        }
    }
}

@Preview(
    device = Devices.PIXEL_4,
    showSystemUi = true,
    name = "Landscape Preview"
)
```

```
@Composable
fun HelloPreviewLandscape() {
    _2ModulTheme {
        Column(
            modifier = Modifier
                .fillMaxSize()
                .padding(16.dp)
        ) {
            Hello(name = "Мария (Landscape)")
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant1()
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant2()
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant3()
        }
    }
}

// Preview для задания 3
@Preview(showBackground = true)
@Composable
fun TextVariantsPreview() {
    _2ModulTheme {
        Column(
            modifier = Modifier
                .fillMaxSize()
                .padding(16.dp)
        ) {
            TextVariant1()
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant2()
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant3()
        }
    }
}
```

Листинг 3 – MainActivity

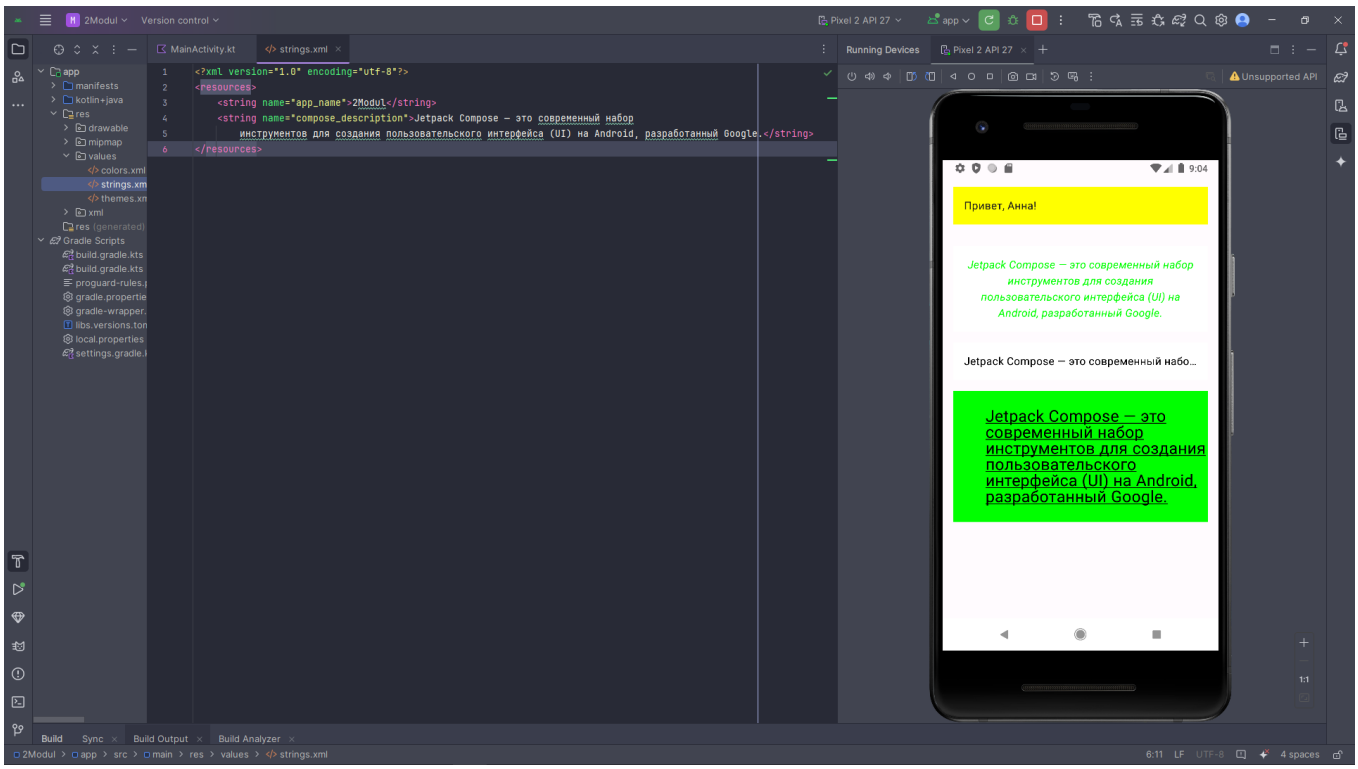
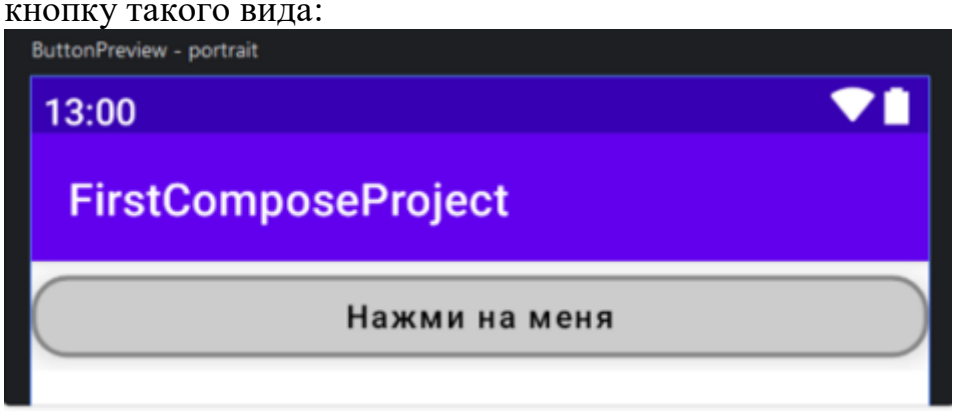


Рисунок 3 – MainActivity и Strings.xml

Задание 4

Попробуйте самостоятельно определить значения параметров и вывести на экран кнопку такого вида:



1. Для растягивания кнопки по ширине экрана задайте параметр modifier. 2. Для установки формы прямоугольника со скруглёнными углами используйте параметр shape со значением RoundedCornerShape(16.dp). 3. Для границы кнопки задайте значение BorderStroke(2.dp, Color.Gray). 4. Установите тень глубиной 4.dp. 5. Для фона кнопки выберите светло-серый цвет(Color.LightGray), а для цвета текста — чёрный (Color.Black).

```
package com.example.a2modul

import android.os.Bundle
import androidx.activity.ComponentActivity
```

```
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.background
import androidx.compose.foundation.BorderStroke
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material3.Card
import androidx.compose.material3.CardDefaults
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontStyle
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.text.style.TextDecoration
import androidx.compose.ui.tooling.preview.Devices
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.a2modul.ui.theme._2ModulTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        setContent {
            _2ModulTheme {
                Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->
                    Column(
                        modifier = Modifier
                            .padding(innerPadding)
                            .fillMaxSize()
                            .padding(16.dp)
                    ) {
                        // Задание 1
                        Hello(
                            name = "Анна",
                            modifier = Modifier.fillMaxWidth()
                        )

                        Spacer(modifier = Modifier.height(32.dp))

                        // Задание 3 - вариант 1
                        TextVariant1()

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

                        // Задание 3 - вариант 2
                        TextVariant2()

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

                        // Задание 3 - вариант 3
                        TextVariant3()

                        Spacer(modifier = Modifier.height(32.dp))
```

```
                        // Задание 4: Кастомная кнопка
                        CustomButton()
                    }
                }
            }
        }
    }
}

@Composable
fun Hello(name: String?, modifier: Modifier = Modifier) {
    Surface(
        color = Color.Yellow,
        modifier = modifier
    ) {
        Text(
            text = if (name != null) "Привет, $name!" else "Имя не задано",
            modifier = Modifier.padding(16.dp)
        )
    }
}

// ЗАДАНИЕ 3: Вариант 1
@Composable
fun TextVariant1() {
    Surface(
        color = Color.White,
        modifier = Modifier.fillMaxWidth()
    ) {
        Column(
            modifier = Modifier.padding(16.dp),
            horizontalAlignment = Alignment.CenterHorizontally
        ) {
            Text(
                text = stringResource(R.string.compose_description),
                color = Color.Green,
                fontSize = 16.sp,
                fontStyle = FontStyle.Italic,
                textAlign = TextAlign.Center,
                modifier = Modifier.fillMaxWidth()
            )
        }
    }
}

// ЗАДАНИЕ 3: Вариант 2
@Composable
fun TextVariant2() {
    Surface(
        color = Color.White,
        modifier = Modifier.fillMaxWidth()
    ) {
        Column(
            modifier = Modifier.padding(16.dp)
        ) {
            Text(
                text = stringResource(R.string.compose_description),
                color = Color.Black,
                fontSize = 16.sp,
                maxLines = 1,
                overflow = androidx.compose.ui.text.style.TextOverflow.Ellipsis,
                modifier = Modifier.fillMaxWidth()
            )
        }
```

```
    }
}

// ЗАДАНИЕ 3: Вариант 3
@Composable
fun TextVariant3() {
    Surface(
        color = Color.White,
        modifier = Modifier.fillMaxWidth()
    ) {
        Box(
            modifier = Modifier
                .fillMaxWidth()
                .background(Color.Green)
                .padding(vertical = 8.dp)
        ) {
            Text(
                text = stringResource(R.string.compose_description),
                fontSize = 24.sp,
                color = Color.Black,
                textDecoration = TextDecoration.Underline,
                modifier = Modifier.padding(start = 48.dp, top = 16.dp, bottom = 16.dp)
            )
        }
    }
}

// Задание 4: Кастомная кнопка
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun CustomButton() {
    Card(
        modifier = Modifier
            .fillMaxWidth()
            .padding(horizontal = 16.dp),
        shape = RoundedCornerShape(16.dp),
        border = BorderStroke(2.dp, Color.Gray),
        elevation = CardDefaults.cardElevation(defaultElevation = 4.dp),
        colors = CardDefaults.cardColors(containerColor = Color.LightGray),
        onClick = {
            // Действие при нажатии кнопки
        }
    ) {
        Text(
            text = "Нажми на меня",
            color = Color.Black,
            fontSize = 18.sp,
            fontWeight = FontWeight.Medium,
            textAlign = TextAlign.Center,
            modifier = Modifier
                .fillMaxWidth()
                .padding(vertical = 16.dp)
        )
    }
}

// ЗАДАНИЕ 2: Preview
@Preview(
    device = Devices.PIXEL_4,
    showSystemUi = true,
    name = "Portrait Preview"
)
@Composable
fun HelloPreviewPortrait() {
```

```
    _2ModulTheme {
        Column(
            modifier = Modifier
                .fillMaxSize()
                .padding(16.dp)
        ) {
            Hello(name = "Иван (Portrait)")
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant1()
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant2()
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant3()
            Spacer(modifier = Modifier.height(16.dp))
            CustomButton()
        }
    }
}

@Preview(
    device = Devices.PIXEL_4,
    showSystemUi = true,
    name = "Landscape Preview"
)
@Composable
fun HelloPreviewLandscape() {
    _2ModulTheme {
        Column(
            modifier = Modifier
                .fillMaxSize()
                .padding(16.dp)
        ) {
            Hello(name = "Мария (Landscape)")
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant1()
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant2()
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant3()
            Spacer(modifier = Modifier.height(16.dp))
            CustomButton()
        }
    }
}



// Preview для задания 3
@Preview(showBackground = true)
@Composable
fun TextVariantsPreview() {
    _2ModulTheme {
        Column(
            modifier = Modifier
                .fillMaxSize()
                .padding(16.dp)
        ) {
            TextVariant1()
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant2()
            Spacer(modifier = Modifier.height(16.dp))
            TextVariant3()
            Spacer(modifier = Modifier.height(16.dp))
            CustomButton()
```

```
            }
        }
}
```

Листинг 4 – MainActivity

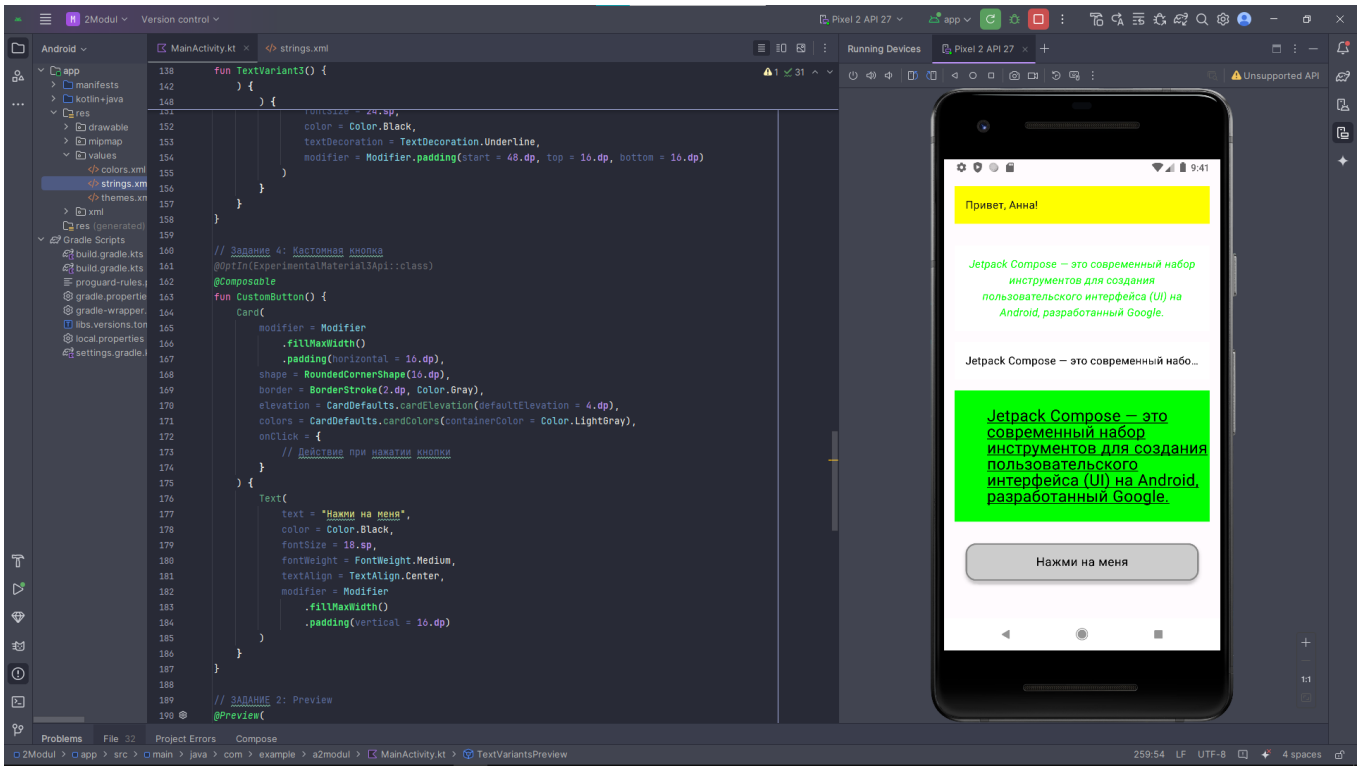
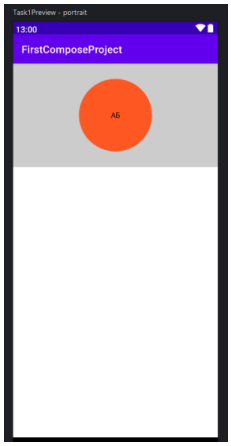


Рисунок 4 – MainActivity

Задание 5

Выполнить задание можно в отдельном проекте или в новом файле в уже существующем проекте. Итак, задача такая: 1. Разместите во всю ширину экрана серый контейнер высотой 100.dp. 2. Расположите посередине компонент с инициалами с вертикальными отступами 16.dp. 3. Круг должен занимать всю доступную высоту.




```
package com.example.a2modul

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
```

```
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.CircleShape
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.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.a2modul.ui.theme._2ModulTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        setContent {
            _2ModulTheme {
                // Задание 5: Серый контейнер с красным кругом и инициалами
                Assignment5()
            }
        }
    }
}

// Задание 5: Серый контейнер с красным кругом и инициалами
@Composable
fun Assignment5() {
    Box(
        modifier = Modifier
            .fillMaxWidth()
            .height(100.dp)
            .background(Color.Gray),
        contentAlignment = Alignment.Center
    ) {
        Box(
            modifier = Modifier
                .fillMaxHeight()
                .padding(vertical = 16.dp),
            contentAlignment = Alignment.Center
        ) {
            Box(
                modifier = Modifier
                    .fillMaxHeight()
                    .aspectRatio(1f)
                    .clip(CircleShape)
                    .background(Color.Red),
                contentAlignment = Alignment.Center
            ) {
                Text(
                    text = "A5",
                    color = Color.Black,
                    fontSize = 24.sp,
                    fontWeight = FontWeight.Bold
                )
            }
        }
    }
}
```

```
@Preview(showBackground = true)
@Composable
fun Assignment5Preview() {
    _2ModulTheme {
        Assignment5()
    }
}
```

Листинг 5 – MainActivity

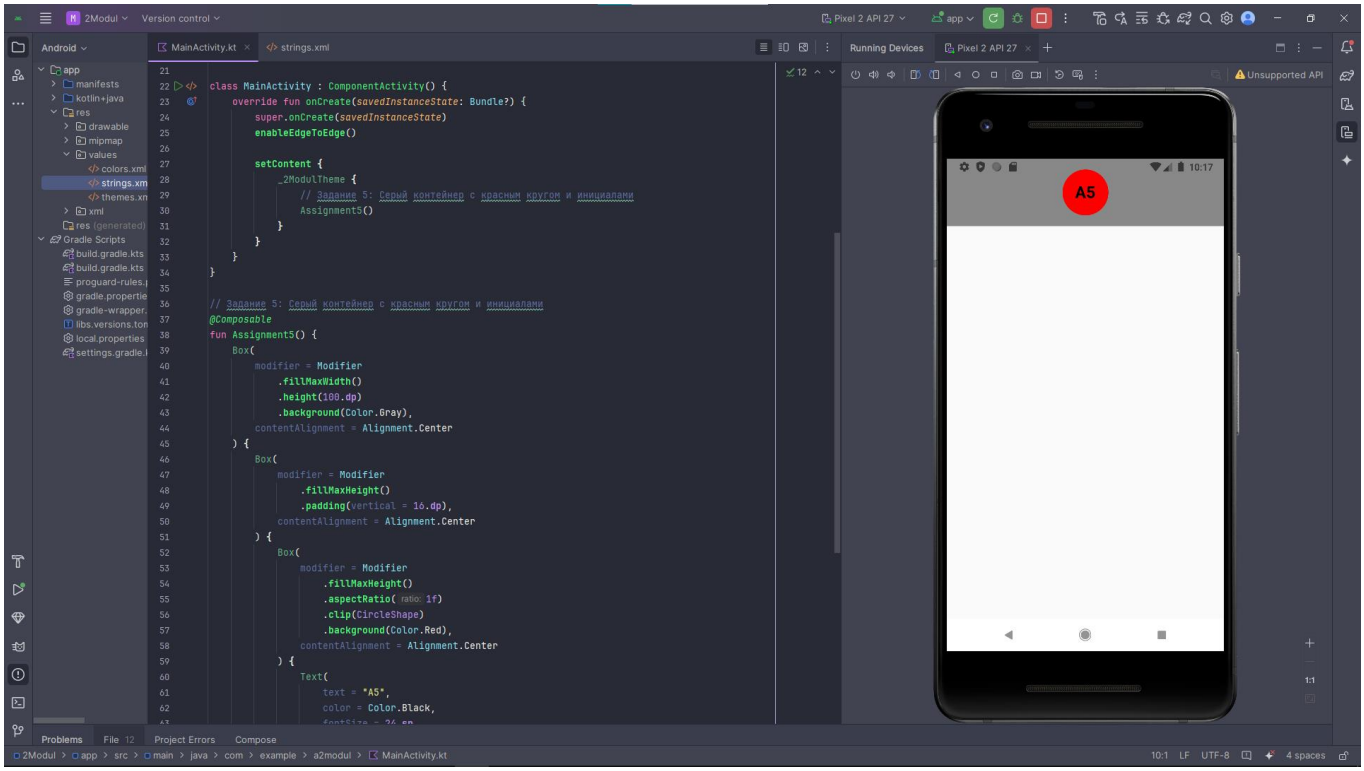


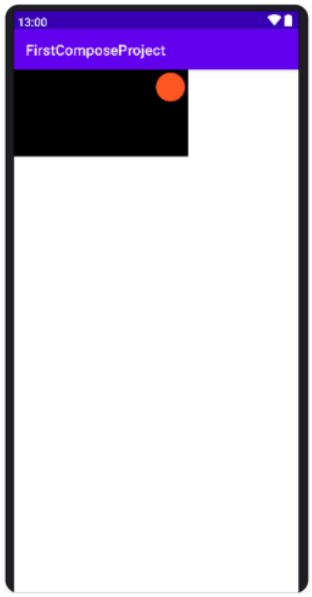
Рисунок 5 – MainActivity

Задание 6

Создайте в ресурсах файл circle.xml с векторным изображением закрашенного круга:

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
   android:width="48dp"
   android:height="48dp"
   android:viewportWidth="24"
   android:viewportHeight="24">
   <path
      android:fillColor="#FF5722"
      android:pathData="M12,2a10,10 0,1,1 0,20a10,10 0,1,1 0,-20"/>
</vector>
```

Напишите две функции: 1. Расположите круг исходного размера в правом верхнем углу в компоненте с чёрным фоном, размер — 240.dp x 120.dp:



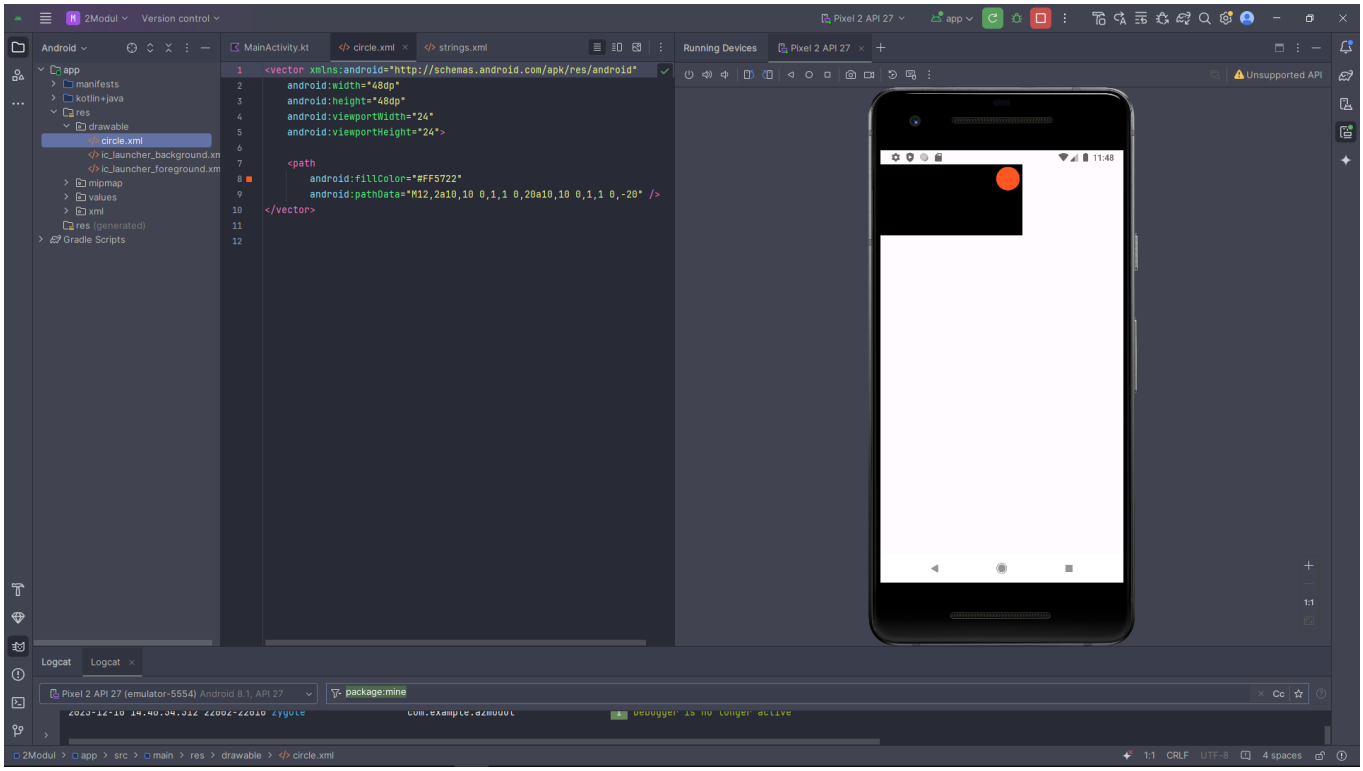


Рисунок 6.1 – MainActivity и circle.xml

Листинг 6.1 – MainActivity

```
package com.example.a2modul

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.material3.FilledTonalButton
import androidx.compose.material3.Scaffold
```

```
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.a2modul.ui.theme._2ModulTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        setContent {
            _2ModulTheme {
                // Задание 5: Серый контейнер с красным кругом и инициалами
                MainScreen()
            }
        }
    }
}

@Composable
fun MainScreen(
    modifier: Modifier = Modifier
){
    // Создание родительского контейнера
    Scaffold (
        // Установка Modifier с наслаиванием аттрибутов
        // те что были переданы при вызове MainScreen и записаны в переменную
                modifier = modifier
                .fillMaxSize(),
        // Заполнение контентом
        content = {
            Column(
                // Заполнение "чистого" Modifier
                modifier = Modifier
                    .padding(it)
                    .wrapContentSize(align = Alignment.TopStart),
                // Для Column базовой осью является вертикаль
                // Центровка вертикали с левого края будет выглядеть так:
                verticalArrangement = Arrangement.Center,
                horizontalAlignment = Alignment.Start
            ) {
                CircleInTopRight()
            }
        }
    )
}

@Composable
fun CircleInTopRight() {
    Box(
        modifier = Modifier
            .size(width = 240.dp, height = 120.dp)
            .background(Color.Black)
    ) {
        Image(
            painter = painterResource(id = R.drawable.circle),
            contentDescription = "Красный круг",
            modifier = Modifier
                .align(Alignment.TopEnd)
```

```
        )
    }
}

@Composable
fun FilledTonalButtonExample(onClick: () -> Unit){
    FilledTonalButton(onClick = {onClick() }) {
        Text("Tonal")
    }
}



@Preview(showBackground = true)
@Composable
fun Assignment5Preview() {
    _2ModulTheme {
        MainScreen()
    }
}
```

2. Задайте синий цвет фона компонента. Расположите круг по центру и растяните по размерам. Используйте один из параметров компонента, чтобы круг был фиолетового цвета:

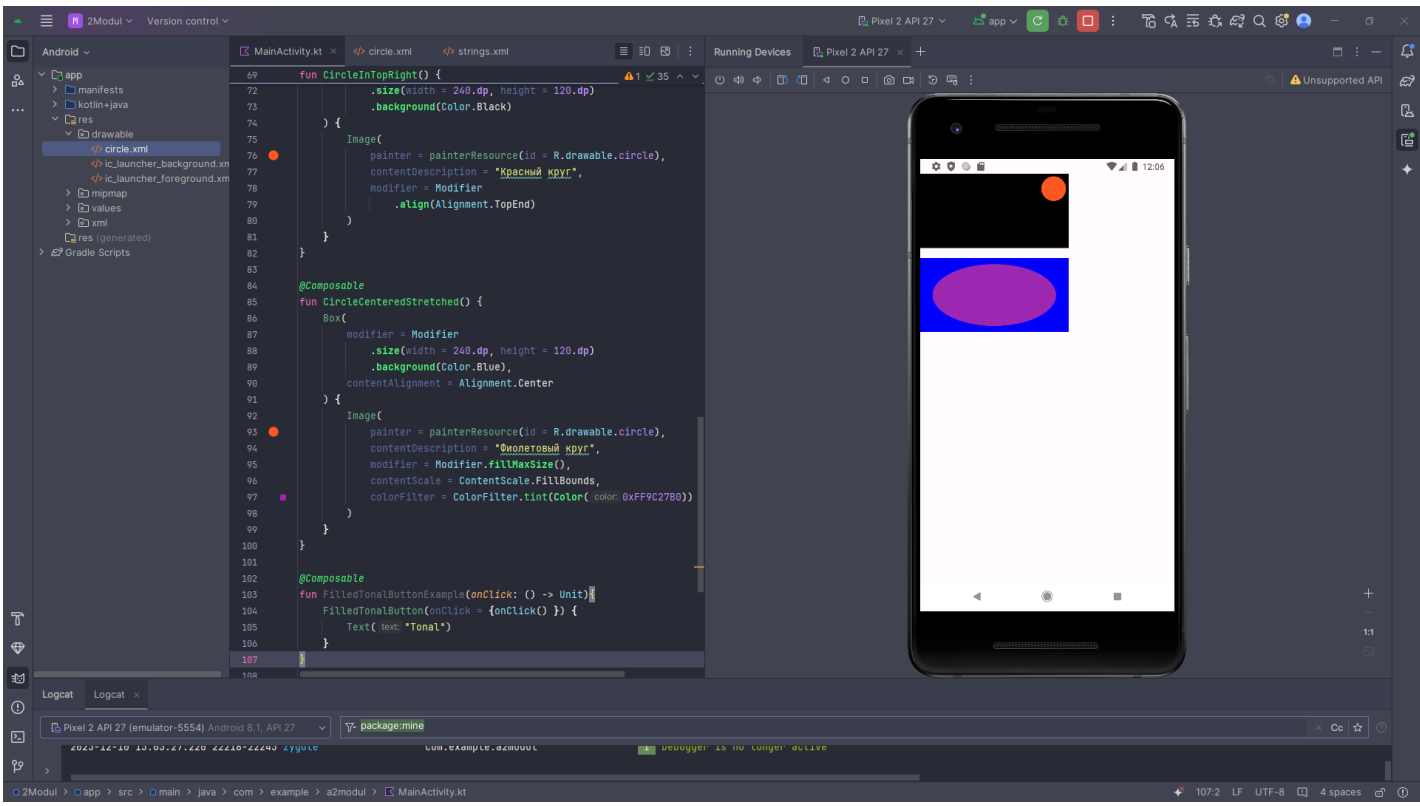


Рисунок 6.2 – MainActivity

Листинг 6.2 – MainActivity

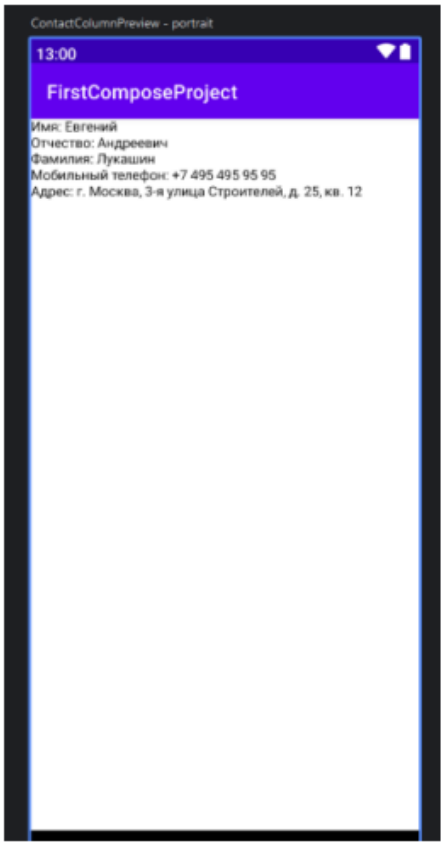
```
package com.example.a2modul

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.material3.FilledTonalButton
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.graphics.ColorFilter
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.a2modul.ui.theme._2ModulTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        setContent {
            _2ModulTheme {
                // Задание 5: Серый контейнер с красным кругом и инициалами
```

```
                MainScreen()
            }
        }
    }
}

@Composable
fun MainScreen(
    modifier: Modifier = Modifier
){
    // Создание родительского контейнера
    Scaffold (
        // Установка Modifier с наслаиванием аттрибутов
        // те что были переданы при вызове MainScreen и записаны в переменную
                modifier = modifier
                .fillMaxSize(),
        // Заполнение контентом
        content = {
            Column(
                // Заполнение "чистого" Modifier
                modifier = Modifier
                    .padding(it)
                    .wrapContentSize(align = Alignment.TopStart),
                // Для Column базовой осью является вертикаль
                // Центровка вертикали с левого края будет выглядеть так:
                verticalArrangement = Arrangement.Center,
                horizontalAlignment = Alignment.Start
            ) {
                CircleInTopRight()
                Spacer(modifier = Modifier.height(16.dp))
                CircleCenteredStretched()
            }
        }
    )
}

@Composable
fun CircleInTopRight() {
    Box(
        modifier = Modifier
            .size(width = 240.dp, height = 120.dp)
            .background(Color.Black)
    ) {
        Image(
            painter = painterResource(id = R.drawable.circle),
            contentDescription = "Красный круг",
            modifier = Modifier
                .align(Alignment.TopEnd)
        )
    }
}

@Composable
fun CircleCenteredStretched() {
    Box(
        modifier = Modifier
            .size(width = 240.dp, height = 120.dp)
            .background(Color.Blue),
        contentAlignment = Alignment.Center
    ) {
        Image(
            painter = painterResource(id = R.drawable.circle),
            contentDescription = "Фиолетовый круг",
            modifier = Modifier.fillMaxSize(),
```

```
            contentScale = ContentScale.FillBounds,
            colorFilter = ColorFilter.tint(Color(0xFF9C27B0))
        )
    }
}

@Composable
fun FilledTonalButtonExample(onClick: () -> Unit){
    FilledTonalButton(onClick = {onClick() }) {
        Text("Tonal")
    }
}



@Preview(showBackground = true)
@Composable
fun Assignment5Preview() {
    _2ModulTheme {
        MainScreen()
    }
}
```

Задание 7

Реализуйте Composable функции, используя контейнер Column, для изображений, представленных ниже.



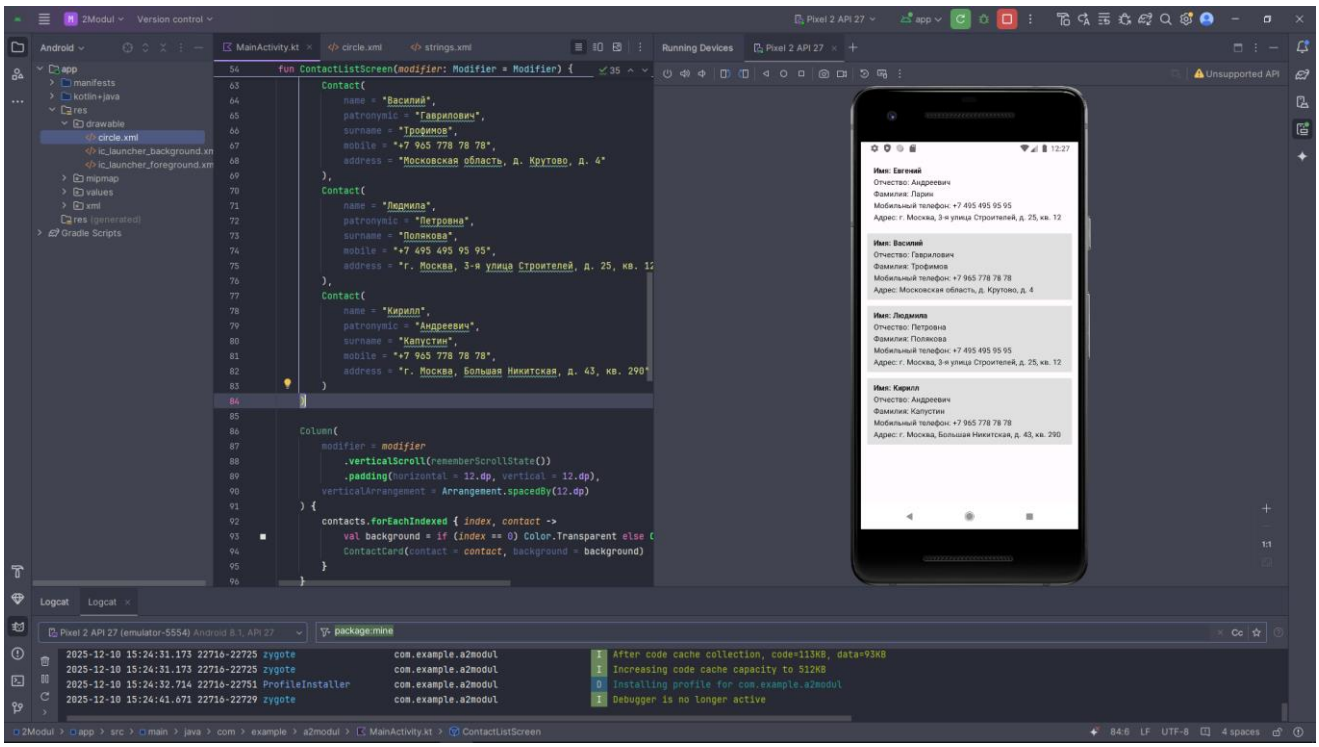


Рисунок 7.1– MainActivity

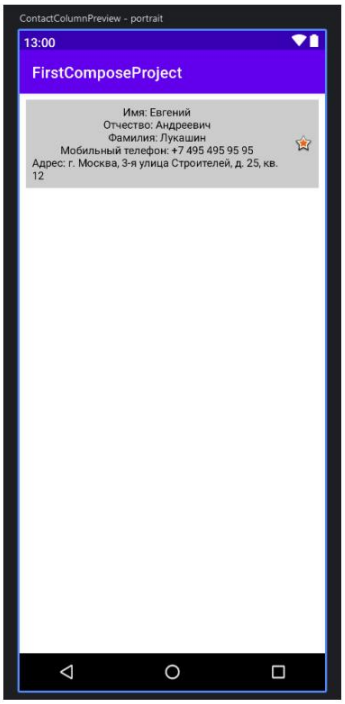
Листинг 7.1 – MainActivity

```
package com.example.a2modul

import android.os.Bundle
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.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.a2modul.ui.theme._2ModulTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        setContent {
            _2ModulTheme {
                MainScreen()
            }
```

```
        }
    }
}

@Composable
fun MainScreen(modifier: Modifier = Modifier) {
    Scaffold(
        modifier = modifier.fillMaxSize()
    ) { padding ->
        ContactListScreen(
            modifier = Modifier
                .padding(padding)
                .fillMaxSize()
        )
    }
}

@Composable
fun ContactListScreen(modifier: Modifier = Modifier) {
    val contacts = listOf(
        Contact(
            name = "Евгений",
            patronymic = "Андреевич",
            surname = "Ларин",
            mobile = "+7 495 495 95 95",
            address = "г. Москва, 3-я улица Строителей, д. 25, кв. 12"
        ),
        Contact(
            name = "Василий",
            patronymic = "Гаврилович",
            surname = "Трофимов",
            mobile = "+7 965 778 78 78",
            address = "Московская область, д. Крутово, д. 4"
        ),
        Contact(
            name = "Людмила",
            patronymic = "Петровна",
            surname = "Полякова",
            mobile = "+7 495 495 95 95",
            address = "г. Москва, 3-я улица Строителей, д. 25, кв. 12"
        ),
        Contact(
            name = "Кирилл",
            patronymic = "Андреевич",
            surname = "Капустин",
            mobile = "+7 965 778 78 78",
            address = "г. Москва, Большая Никитская, д. 43, кв. 290"
        )
    )

    Column(
        modifier = modifier
            .verticalScroll(rememberScrollState())
            .padding(horizontal = 12.dp, vertical = 12.dp),
        verticalArrangement = Arrangement.spacedBy(12.dp)
    ) {
        contacts.forEachIndexed { index, contact ->
            val background = if (index == 0) Color.Transparent else Color(0xFFE0E0E0)
            ContactCard(contact = contact, background = background)
        }
    }
}

@Composable
```

```
fun ContactCard(
    contact: Contact,
    background: Color
) {
    Column(
        modifier = Modifier
            .fillMaxWidth()
            .background(background)
            .padding(horizontal = 12.dp, vertical = 8.dp),
        verticalArrangement = Arrangement.spacedBy(2.dp),
        horizontalAlignment = Alignment.Start
    ) {
        Text(
            text = "Имя: ${contact.name}",
            style = MaterialTheme.typography.bodyMedium,
            fontWeight = FontWeight.SemiBold
        )
        Text("Отчество: ${contact.patronymic}", style =
MaterialTheme.typography.bodyMedium)
        Text("Фамилия: ${contact.surname}", style = MaterialTheme.typography.bodyMedium)
        Text("Мобильный телефон: ${contact.mobile}", style =
MaterialTheme.typography.bodyMedium)
        Text("Адрес: ${contact.address}", style = MaterialTheme.typography.bodyMedium)
    }
}

data class Contact(
    val name: String,
    val patronymic: String,
    val surname: String,
    val mobile: String,
    val address: String
)

@Preview(showBackground = true, name = "ContactColumnPreview")
@Composable
fun ContactColumnPreview() {
    _2ModulTheme {
        ContactCard(
            contact = Contact(
                name = "Евгений",
                patronymic = "Андреевич",
                surname = "Ларин",
                mobile = "+7 495 495 95 95",
                address = "г. Москва, 3-я улица Строителей, д. 25, кв. 12"
            ),
            background = Color.Transparent
        )
    }
}

@Preview(showBackground = true, name = "ListPreview")
@Composable
fun ListPreview() {
    _2ModulTheme {
        ContactListScreen()
    }
}
```

А также Composable функцию, применяя контейнер Row (изображение справа можно взять произвольное):



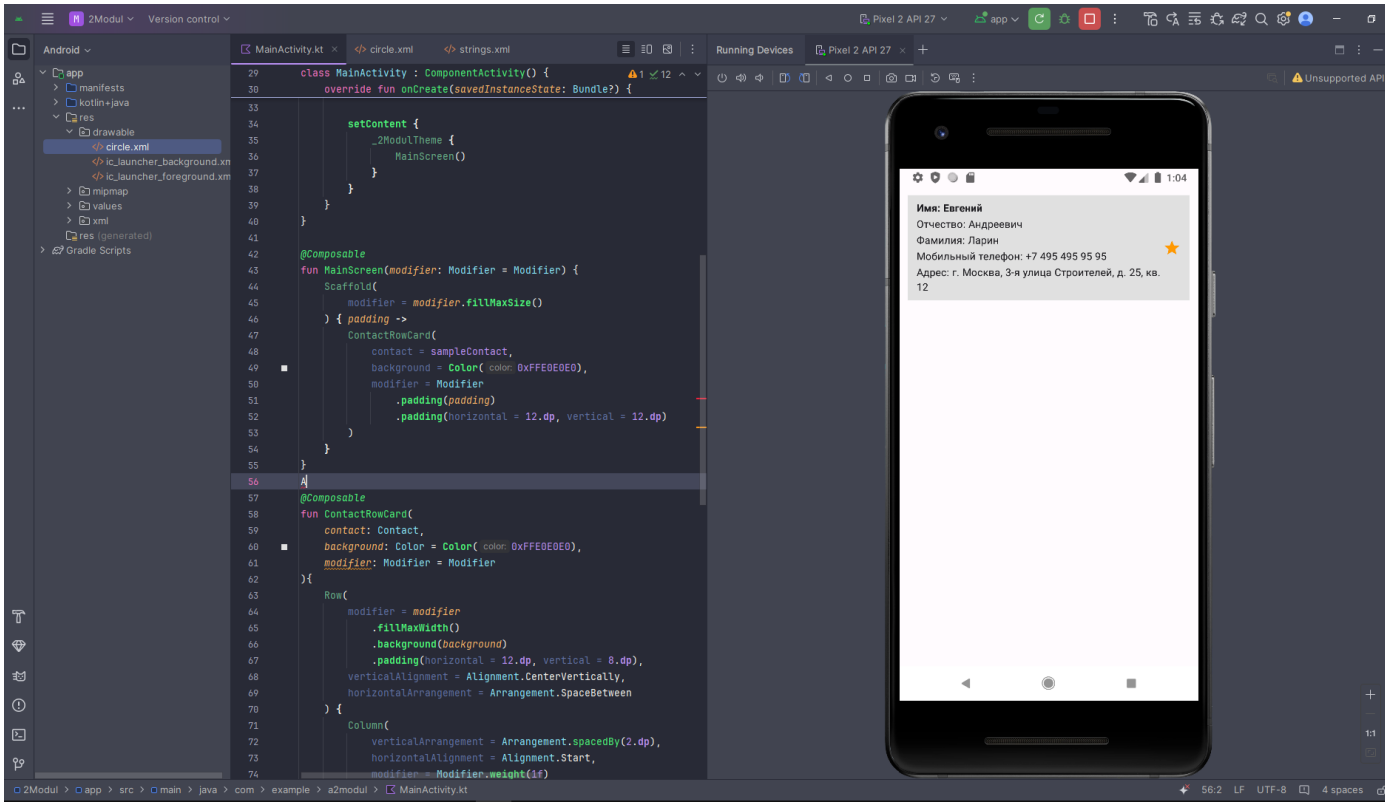


Рисунок 7.2– MainActivity

Листинг 7.2 – MainActivity

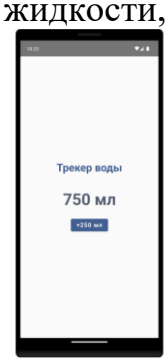
```
package com.example.a2modul

import android.os.Bundle
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.Column
```

```
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.Star
import androidx.compose.material3.Icon
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.a2modul.ui.theme._2ModulTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        setContent {
            _2ModulTheme {
                MainScreen()
            }
        }
    }
}

@Composable
fun MainScreen(modifier: Modifier = Modifier) {
    Scaffold(
        modifier = modifier.fillMaxSize()
    ) { padding ->
        ContactRowCard(
            contact = sampleContact,
            background = Color(0xFFE0E0E0),
            modifier = Modifier
                .padding(padding)
                .padding(horizontal = 12.dp, vertical = 12.dp)
        )
    }
}
A
@Composable
fun ContactRowCard(
    contact: Contact,
    background: Color = Color(0xFFE0E0E0),
    modifier: Modifier = Modifier
){
    Row(
        modifier = modifier
            .fillMaxWidth()
            .background(background)
            .padding(horizontal = 12.dp, vertical = 8.dp),
        verticalAlignment = Alignment.CenterVertically,
        horizontalArrangement = Arrangement.SpaceBetween
    ) {
        Column(
            verticalArrangement = Arrangement.spacedBy(2.dp),
            horizontalAlignment = Alignment.Start,
```

```
            modifier = Modifier.weight(1f)
        ) {
            Text(
                text = "Имя: ${contact.name}",
                style = MaterialTheme.typography.bodyMedium,
                fontWeight = FontWeight.SemiBold
            )
            Text("Отчество: ${contact.patronymic}", style =
MaterialTheme.typography.bodyMedium)
            Text("Фамилия: ${contact.surname}", style =
MaterialTheme.typography.bodyMedium)
            Text("Мобильный телефон: ${contact.mobile}", style =
MaterialTheme.typography.bodyMedium)
            Text("Адрес: ${contact.address}", style =
MaterialTheme.typography.bodyMedium)
        }
        Icon(
            imageVector = Icons.Default.Star,
            contentDescription = "Избранное",
            tint = Color(0xFFFF9800)
        )
    }
}

data class Contact(
    val name: String,
    val patronymic: String,
    val surname: String,
    val mobile: String,
    val address: String
)

private val sampleContact = Contact(
    name = "Евгений",
    patronymic = "Андреевич",
    surname = "Ларин",
    mobile = "+7 495 495 95 95",
    address = "г. Москва, 3-я улица Строителей, д. 25, кв. 12"
)

@Preview(showBackground = true, name = "RowCardPreview")
@Composable
fun RowCardPreview() {
    _2ModulTheme {
        ContactRowCard(contact = sampleContact, background = Color(0xFFE0E0E0))
    }
}
```

Задание 8

Имеется следующее приложение, которое представляет собой счетчик выпитой жидкости, см. скриншот:



А также представлен код Composable функции WaterTracker: @Composable fun WaterTracker() { var waterCount by { mutableStateOf(100) } Column(modifier = Modifier .fillMaxSize() .padding(16.dp),) { Text(text = "Трекер воды", fontSize = 32.sp, 27 fontWeight = FontWeight.Bold, color = primaryColor) Spacer(modifier = Modifier.height(48.dp)) Text(fontSize = 48.sp, fontWeight = FontWeight.Bold, color = secondaryColor) Spacer(modifier = Modifier.height()) Button(colors = ButtonDefaults(backgroundColor = primaryColor), onClick = { waterCount = 250 }) { Text(text = "+250 мл", fontSize = 18.sp, fontWeight = FontWeight.Bold, color = buttonTextColor) } } } 28 Необходимо исправить код в Composable функции WaterTracker, чтобы при нажатии на кнопку происходило увеличение количества выпитой жидкости на 250 мл. А также мы хотим дать пользователям возможность сбрасывать счётчик стаканов по завершении дня, а также выводить количество дней подряд, когда пользователь выпил 1500 мл и больше. Если в один из дней пользователь выпивает меньшее количество воды, счётчик должен обнулиться. Для этого нужно: •Добавить новый стейт с сохранением количества дней, в течение которых пользователь пил 1500 мл воды и больше. •Добавить новую кнопку с текстом «Завершить день». •При нажатии на кнопку сбрасывать счётчик стаканов воды до нуля, а при условии достижения 1500 повышать счётчик дней. В противном случае счётчик дней сбрасывается до нуля. •Добавить вывод текста со значением количества дней, в рамках которых было выпито 1500 мл и больше, подряд.

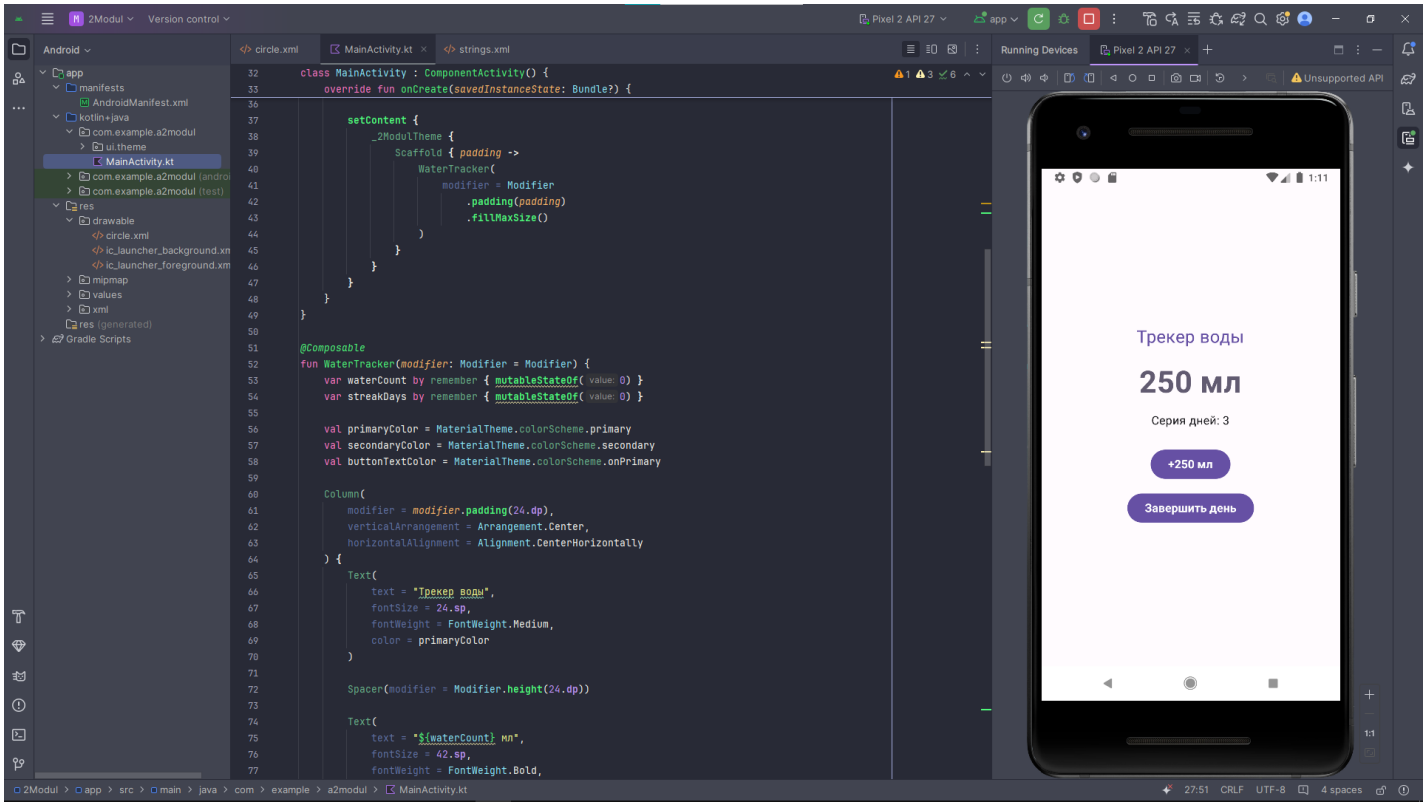


Рисунок 8– MainActivity

Листинг 8 – MainActivity

```
package com.example.a2modul

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Spacer
```

```
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.ButtonDefaults
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
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.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.a2modul.ui.theme._2ModulTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        setContent {
            _2ModulTheme {
                Scaffold { padding ->
                    WaterTracker(
                        modifier = Modifier
                            .padding(padding)
                            .fillMaxSize()
                    )
                }
            }
        }
    }
}

@Composable
fun WaterTracker(modifier: Modifier = Modifier) {
    var waterCount by remember { mutableStateOf(0) }
    var streakDays by remember { mutableStateOf(0) }

    val primaryColor = MaterialTheme.colorScheme.primary
    val secondaryColor = MaterialTheme.colorScheme.secondary
    val buttonTextColor = MaterialTheme.colorScheme.onPrimary

    Column(
        modifier = modifier.padding(24.dp),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text(
            text = "Трекер воды",
            fontSize = 24.sp,
            fontWeight = FontWeight.Medium,
            color = primaryColor
        )

        Spacer(modifier = Modifier.height(24.dp))
```

```
        Text(
            text = "${waterCount} мл",
            fontSize = 42.sp,
            fontWeight = FontWeight.Bold,
            color = secondaryColor
        )

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

        Text(
            text = "Серия дней: $streakDays",
            fontSize = 16.sp,
            fontWeight = FontWeight.Medium,
            color = MaterialTheme.colorScheme.onBackground
        )

        Spacer(modifier = Modifier.height(24.dp))

        Button(
            colors = ButtonDefaults.buttonColors(containerColor = primaryColor),
            onClick = { waterCount += 250 }
        ) {
            Text(
                text = "+250 мл",
                fontSize = 16.sp,
                fontWeight = FontWeight.Bold,
                color = buttonTextColor
            )
        }

        Spacer(modifier = Modifier.height(12.dp))

        Button(
            colors = ButtonDefaults.buttonColors(containerColor = primaryColor),
            onClick = {
                if (waterCount >= 1500) {
                    streakDays += 1
                } else {
                    streakDays = 0
                }
                waterCount = 0
            }
        ) {
            Text(
                text = "Завершить день",
                fontSize = 16.sp,
                fontWeight = FontWeight.Bold,
                color = buttonTextColor
            )
        }
    }
}

@Preview(showBackground = true)
@Composable
fun WaterTrackerPreview() {
    _2ModulTheme {
        WaterTracker()
    }
}
```

Задание 9

Один из джуниор-разработчиков приложения онлайн-магазина решил применить

фреймворк Jetpack Compose, о котором недавно услышал. Но он уделил маловато времени изучению всех возможностей этого фреймворка. И 37 вот пользователи приложения уже жалуются, что за товары с них требуют больше денег, чем должны! Помогите разработчику исправить ошибки. Вот что должно быть в приложении: • При нажатии на кнопку «Добавить товар» добавляем новый товар в корзину. •Если в корзину добавили товаров на сумму больше 500, покажется Toast с сообщением «Доставка бесплатная!» •Кнопка «Удалить товар» показывается только при наличии товаров в корзине. •При нажатии на кнопку «Удалить товар» удаляется последний товар. data class Cart(val products: List<Product>) data class Product(val id: Int, val name: String, val price: Int) class MainActivity : ComponentActivity() { 38 private var totalSum = 0 private var productSize = 0 override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) setContent { ShoppingCartScreen() } } @Composable fun ShoppingCartScreen() { var products by remember { mutableStateOf(listOf(Product(0, "Товар #1", 100), Product(1, "Товар #2", 150), Product(2, "Товар #3", 56)) 39) } Column { for (product in products) { Text(text = product.name + " - " + product.price + " рублей") totalSum += product.price productSize++ } Spacer(modifier = Modifier.height(10.dp)) Text(text = "Товаров на сумму: $totalSum рублей") AddProductSection { products = products + Product(id = products.size, name = "Товар #${products.size + 1}", price = Random.nextInt(0, 100)) } 40 RemoveProductSection { products = products.dropLast(1) } } } @Composable fun AddProductSection(onAdd: () -> Unit) { // Если сумма заказа больше 500, то показываем сообщение о бесплатной доставке if (totalSum > 500) { Toast.makeText(LocalContext.current, "Доставка бесплатная!", Toast.LENGTH_SHORT).show() } Button(onClick = onAdd) { Text(text = "Добавить товар") } 41 } @Composable fun RemoveProductSection(onRemove: () -> Unit) { if (productSize > 0) { Button(onClick = onRemove) { Text(text = "Удалить товар") } } } }

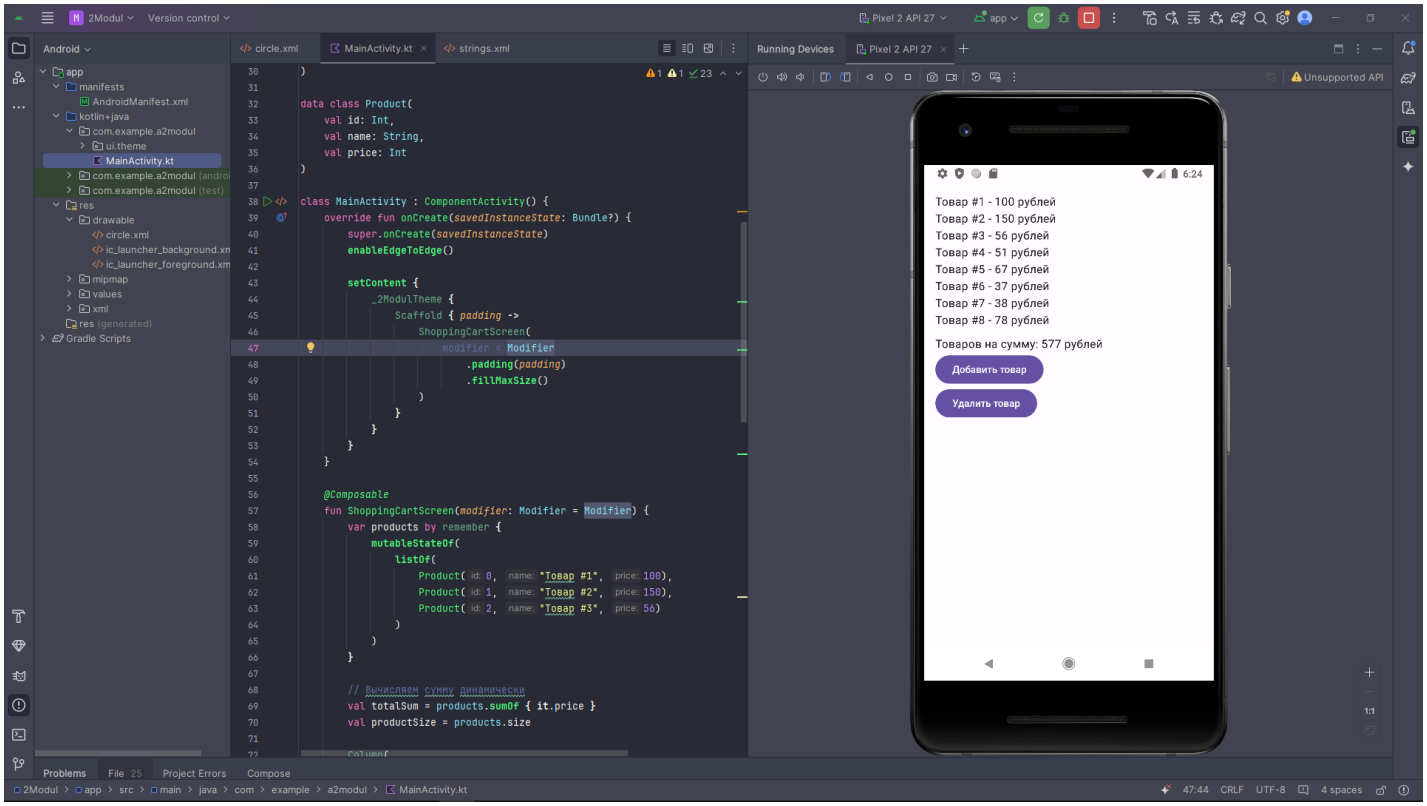


Рисунок 9.1– MainActivity

• При нажатии на кнопку «Добавить товар» добавляем новый товар в корзину

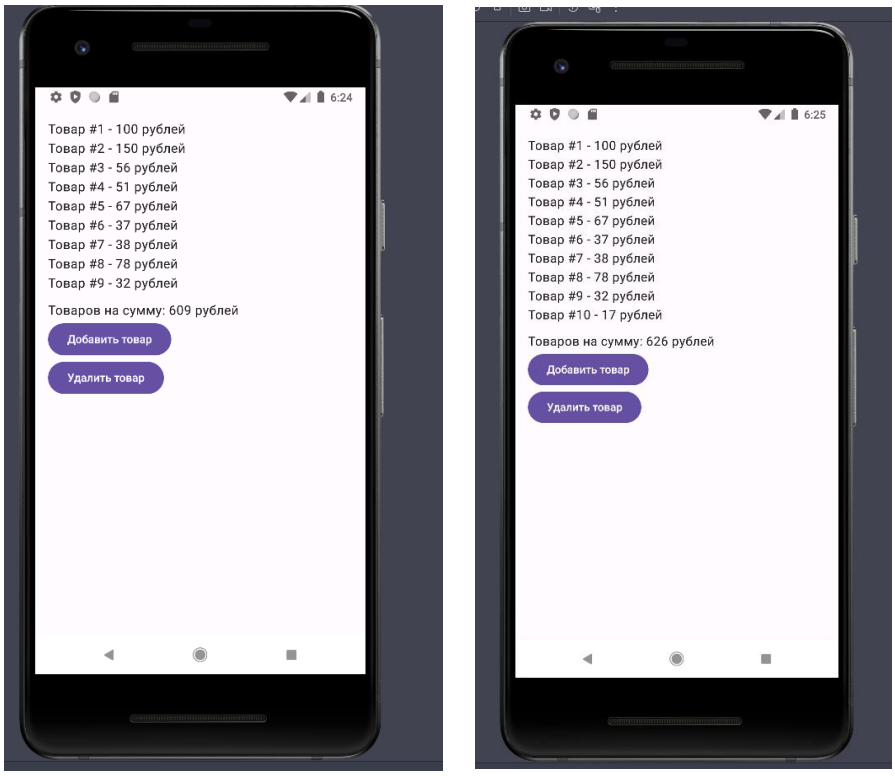


Рисунок 9.2– MainActivity

•Если в корзину добавили товаров на сумму больше 500, покажется Toast с сообщением «Доставка бесплатная!»

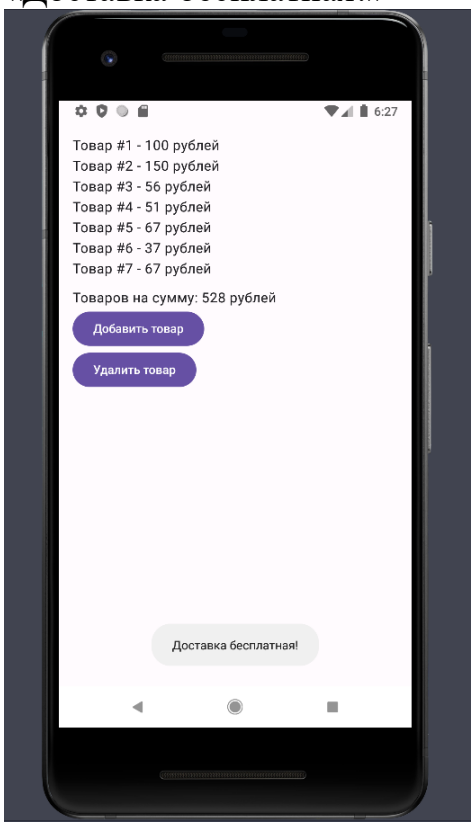


Рисунок 9.3– MainActivity

•При нажатии на кнопку «Удалить товар» удаляется последний товар.

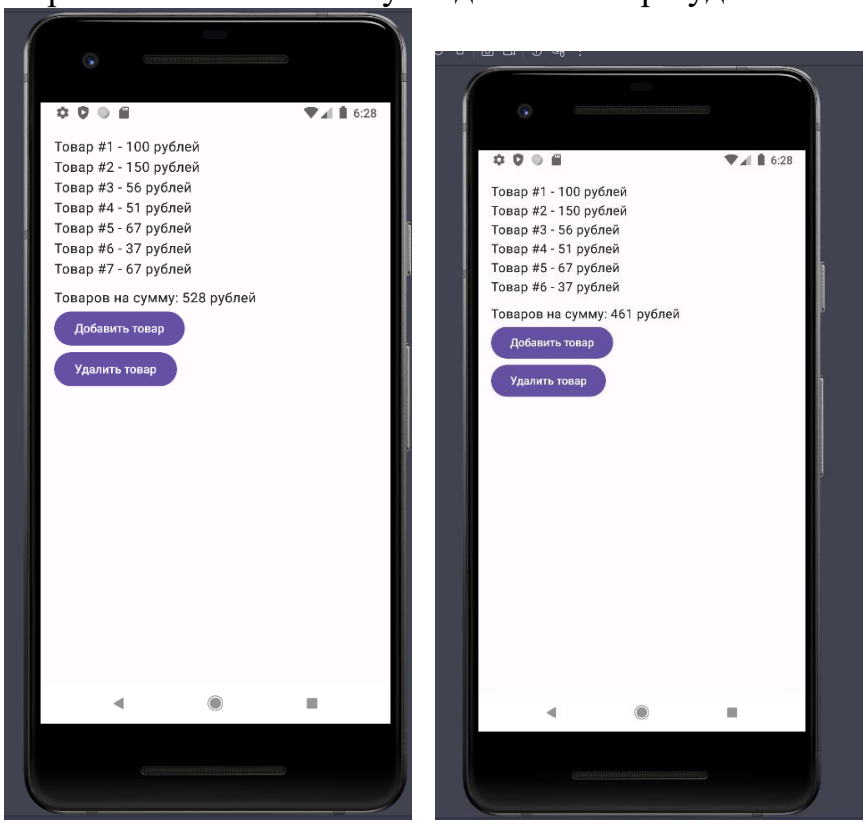


Рисунок 9.4– MainActivity

Листинг 9 – MainActivity

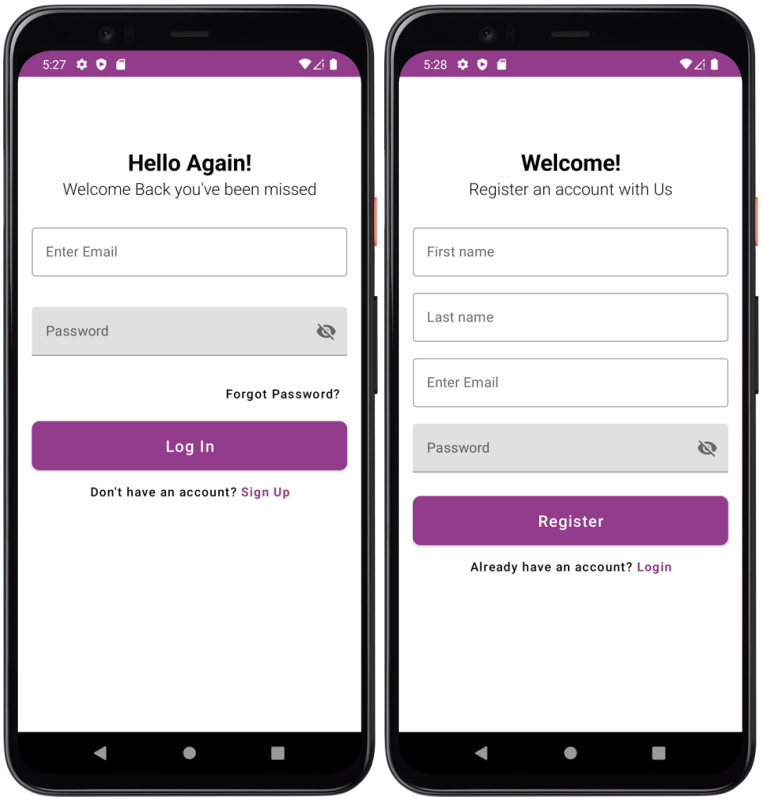
```
package com.example.a2modul

import android.os.Bundle
import android.widget.Toast
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.LaunchedEffect
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.unit.dp
import com.example.a2modul.ui.theme._2ModulTheme
import kotlin.random.Random

data class Cart(
    val products: List<Product>
)
```

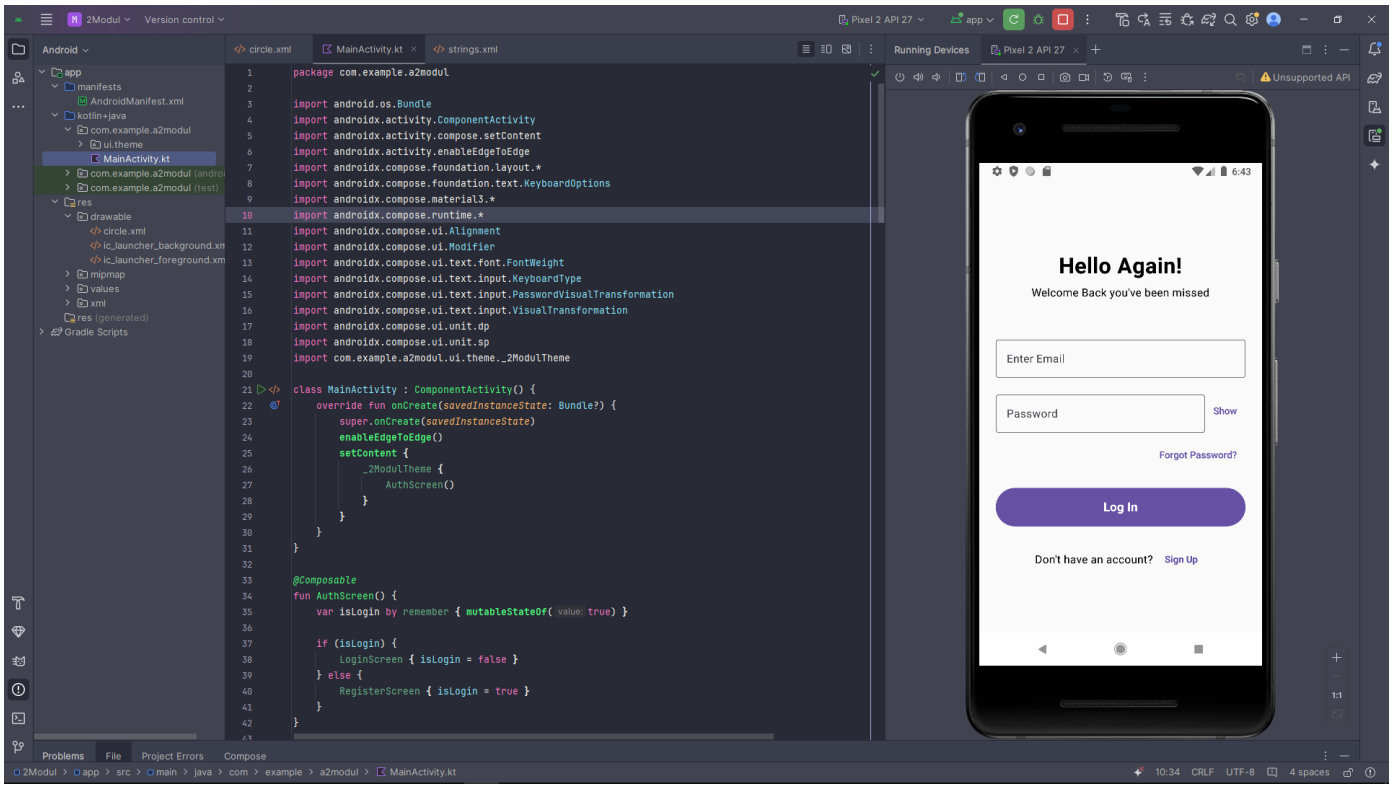
```
data class Product(
    val id: Int,
    val name: String,
    val price: Int
)

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        setContent {
            _2ModulTheme {
                Scaffold { padding ->
                    ShoppingCartScreen(
                        modifier = Modifier
                            .padding(padding)
                            .fillMaxSize()
                    )
                }
            }
        }
    }

    @Composable
    fun ShoppingCartScreen(modifier: Modifier = Modifier) {
        var products by remember {
            mutableStateOf(
                listOf(
                    Product(0, "Товар #1", 100),
                    Product(1, "Товар #2", 150),
                    Product(2, "Товар #3", 56)
                )
            )
        }

        // Вычисляем сумму динамически
        val totalSum = products.sumOf { it.price }
        val productSize = products.size

        Column(
            modifier = modifier.padding(16.dp)
        ) {
            for (product in products) {
                Text(text = "${product.name} - ${product.price} рублей")
            }

            Spacer(modifier = Modifier.height(10.dp))

            Text(text = "Товаров на сумму: $totalSum рублей")

            AddProductSection(
                totalSum = totalSum,
                onAdd = {
                    products = products + Product(
                        id = products.size,
                        name = "Товар #${products.size + 1}",
                        price = Random.nextInt(0, 100)
                    )
                }
            )

            RemoveProductSection(
```

```
                productSize = productSize,
                onRemove = {
                    products = products.dropLast(1)
                }
            )
        }
    }

    @Composable
    fun AddProductSection(
        totalSum: Int,
        onAdd: () -> Unit
    ) {
        val context = LocalContext.current
        var previousSum by remember { mutableStateOf(totalSum) }

        // Показываем Toast только когда сумма переходит через 500
        LaunchedEffect(totalSum) {
            if (totalSum > 500 && previousSum <= 500) {
                Toast.makeText(context, "Доставка бесплатная!",
Toast.LENGTH_SHORT).show()
            }
            previousSum = totalSum
        }

        Button(onClick = onAdd) {
            Text(text = "Добавить товар")
        }
    }

    @Composable
    fun RemoveProductSection(
        productSize: Int,
        onRemove: () -> Unit
    ) {
        if (productSize > 0) {
            Button(onClick = onRemove) {
                Text(text = "Удалить товар")
            }
        }
    }
}
```

Задание 10

Реализовать простое приложение регистрация + авторизация. Примерный вид UI представлен ниже:


5:27
Hello Again!
Welcome Back you've been missed
Enter Email
Password
Forgot Password?
Log In
Don't have an account? Sign Up
5:28
Welcome!
Register an account with Us
First name
Last name
Enter Email
Password
Register
Already have an account? Login


2Modul
Version control
Pixel 2 API 27
app
Android
circle.xml
MainActivity.kt
strings.xml
Running Devices
Pixel 2 API 27
Unsupported API
app
manifests
AndroidManifest.xml
kotlin+java
com.example.a2modul
ui.theme
MainActivity.kt
com.example.a2modul (androi
com.example.a2modul (test)
res
drawable
circle.xml
ic_launcher_background.xm
ic_launcher_foreground.xm
mipmap
values
xml
res (generated)
Gradle Scripts
package com.example.a2modul
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.text.input.VisualTransformation
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.a2modul.ui.theme._2ModulTheme
class MainActivity : ComponentActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
enableEdgeToEdge()
setContent {
_2ModulTheme {
AuthScreen()
}
}
}
}
@Composable
fun AuthScreen() {
var isLogin by remember { mutableStateOf(value: true) }
if (isLogin) {
LoginScreen { isLogin = false }
} else {
RegisterScreen { isLogin = true }
}
}
6:43
Hello Again!
Welcome Back you've been missed
Enter Email
Password
Show
Forgot Password?
Log In
Don't have an account?
Sign Up
Problems
File
Project Errors
Compose
2Modul > app > src > main > java > com > example > a2modul > MainActivity.kt
10:34
CRLF
UTF-8
4 spaces

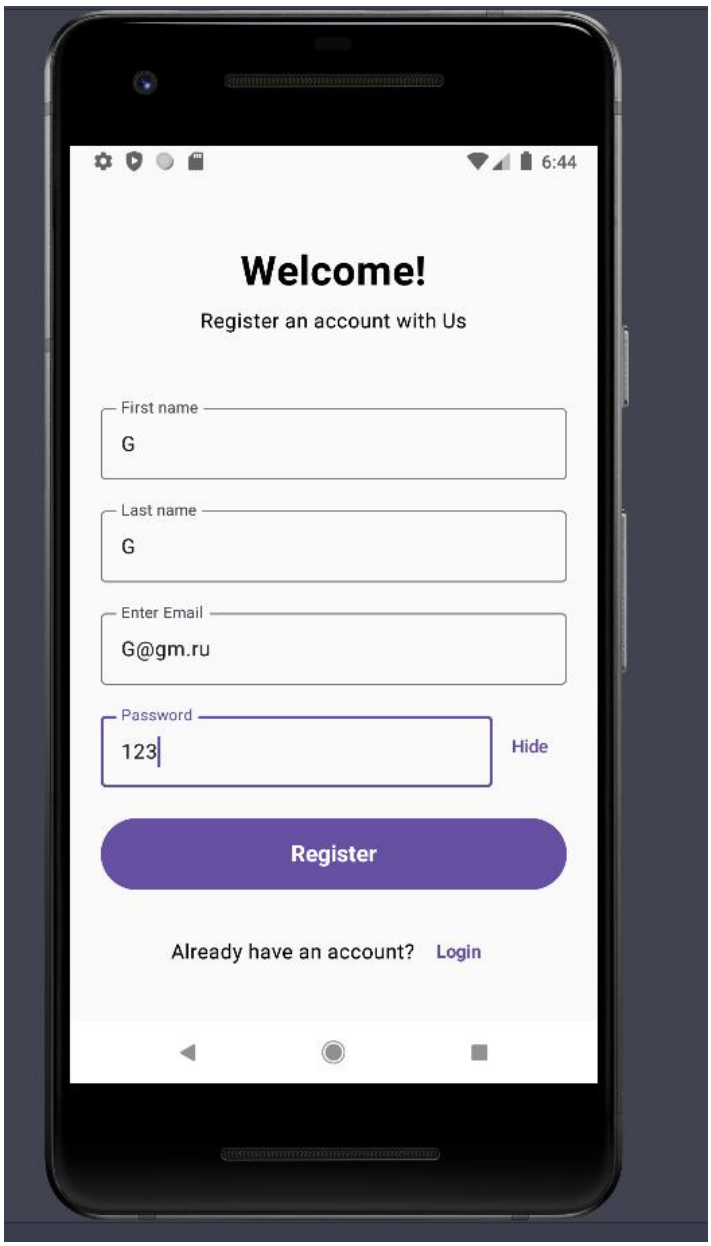


Рисунок 10.1– MainActivity

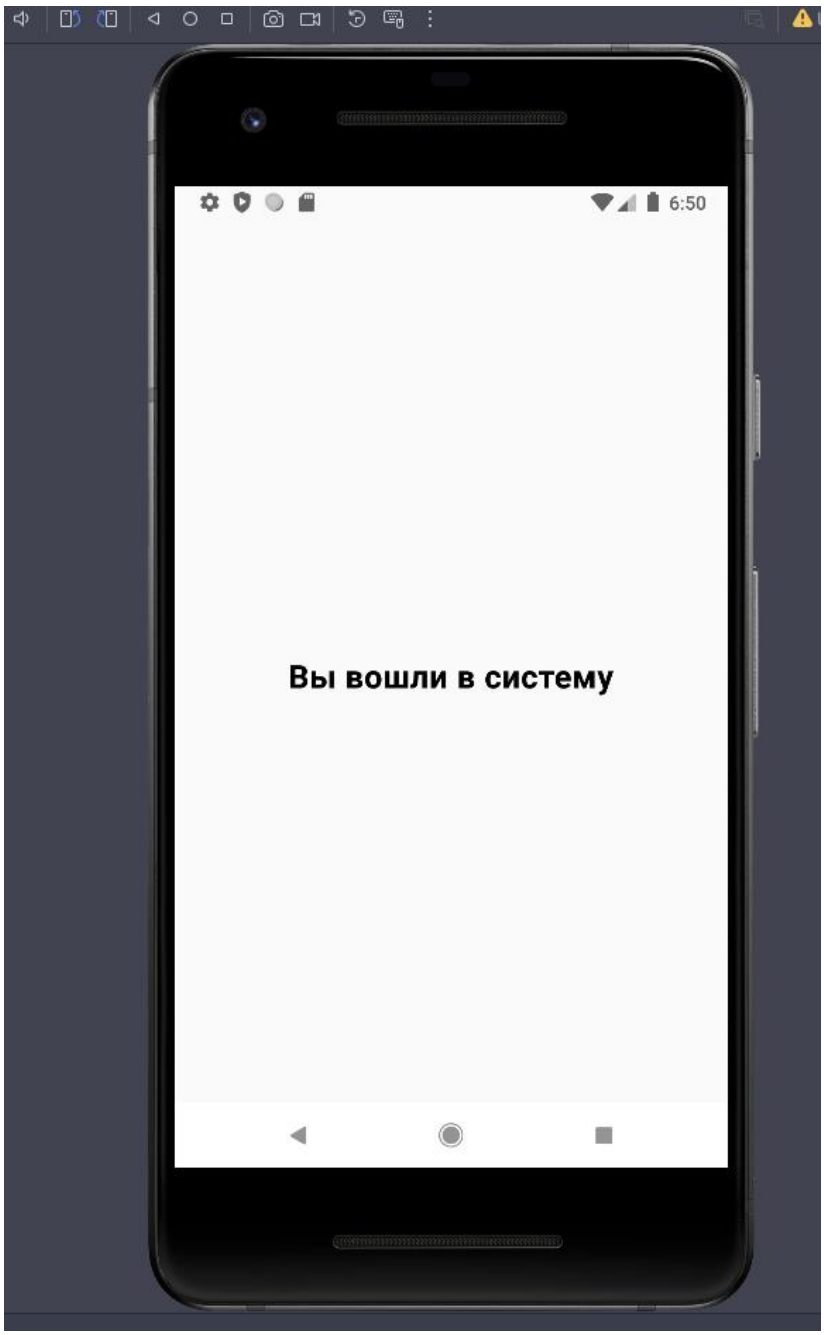


Рисунок 10.2– MainActivity

Листинг 10 – MainActivity

```
package com.example.a2modul

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.text.font.FontWeight
```

```
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.text.input.VisualTransformation
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.a2modul.ui.theme._2ModulTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            _2ModulTheme {
                AuthScreen()
            }
        }
    }
}

data class User(
    val email: String,
    val password: String,
    val firstName: String,
    val lastName: String
)

@Composable
fun AuthScreen() {
    var isLogin by remember { mutableStateOf(true) }
    var isLoggedIn by remember { mutableStateOf(false) }
    var users by remember { mutableStateOf<List<User>>(emptyList()) }

    if (isLoggedIn) {
        SuccessScreen()
    } else if (isLogin) {
        LoginScreen(
            users = users,
            onSignUp = { isLogin = false },
            onLoginSuccess = { isLoggedIn = true }
        )
    } else {
        RegisterScreen(
            onLogin = { isLogin = true },
            onRegister = { user ->
                users = users + user
                isLogin = true
            }
        )
    }
}

@Composable
fun LoginScreen(
    users: List<User>,
    onSignUp: () -> Unit,
    onLoginSuccess: () -> Unit
) {
    var email by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var showPassword by remember { mutableStateOf(false) }
    var errorMessage by remember { mutableStateOf("") }

    Column(
        modifier = Modifier
```

```
            .fillMaxSize()
            .padding(24.dp),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        Text("Hello Again!", fontSize = 32.sp, fontWeight = FontWeight.Bold)
        Spacer(Modifier.height(8.dp))
        Text("Welcome Back you've been missed", fontSize = 16.sp)
        Spacer(Modifier.height(48.dp))

        OutlinedTextField(
            value = email,
            onValueChange = { email = it },
            label = { Text("Enter Email") },
            modifier = Modifier.fillMaxWidth(),
            keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Email)
        )

        Spacer(Modifier.height(16.dp))

        Row(modifier = Modifier.fillMaxWidth(), verticalAlignment =
Alignment.CenterVertically) {
            OutlinedTextField(
                value = password,
                onValueChange = { password = it },
                label = { Text("Password") },
                modifier = Modifier.weight(1f),
                visualTransformation = if (showPassword) VisualTransformation.None else
PasswordVisualTransformation()
            )
            TextButton(onClick = { showPassword = !showPassword }) {
                Text(if (showPassword) "Hide" else "Show")
            }
        }

        Spacer(Modifier.height(8.dp))
        Row(Modifier.fillMaxWidth(), horizontalArrangement = Arrangement.End) {
            TextButton(onClick = {}) {
                Text("Forgot Password?")
            }
        }

        if (errorMessage.isNotEmpty()) {
            Text(
                text = errorMessage,
                color = MaterialTheme.colorScheme.error,
                fontSize = 14.sp
            )
            Spacer(Modifier.height(8.dp))
        }

        Spacer(Modifier.height(24.dp))

        Button(
            onClick = {
                val user = users.find { it.email == email && it.password == password }
                if (user != null) {
                    errorMessage = ""
                    onLoginSuccess()
                } else {
                    errorMessage = "Неверный email или пароль"
                }
            },
            modifier = Modifier.fillMaxWidth().height(56.dp)
```

```
        ) {
            Text("Log In", fontSize = 18.sp, fontWeight = FontWeight.Bold)
        }

        Spacer(Modifier.height(24.dp))

        Row(verticalAlignment = Alignment.CenterVertically) {
            Text("Don't have an account? ")
            TextButton(onClick = onSignUp) {
                Text("Sign Up", fontWeight = FontWeight.Bold)
            }
        }
    }
}

@Composable
fun RegisterScreen(
    onLogin: () -> Unit,
    onRegister: (User) -> Unit
) {
    var firstName by remember { mutableStateOf("") }
    var lastName by remember { mutableStateOf("") }
    var email by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var showPassword by remember { mutableStateOf(false) }
    var errorMessage by remember { mutableStateOf("") }

    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(24.dp),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        Text("Welcome!", fontSize = 32.sp, fontWeight = FontWeight.Bold)
        Spacer(Modifier.height(8.dp))
        Text("Register an account with Us", fontSize = 16.sp)
        Spacer(Modifier.height(48.dp))

        OutlinedTextField(
            value = firstName,
            onValueChange = { firstName = it },
            label = { Text("First name") },
            modifier = Modifier.fillMaxWidth()
        )

        Spacer(Modifier.height(16.dp))

        OutlinedTextField(
            value = lastName,
            onValueChange = { lastName = it },
            label = { Text("Last name") },
            modifier = Modifier.fillMaxWidth()
        )

        Spacer(Modifier.height(16.dp))

        OutlinedTextField(
            value = email,
            onValueChange = { email = it },
            label = { Text("Enter Email") },
            modifier = Modifier.fillMaxWidth(),
            keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Email)
        )
```

```
        Spacer(Modifier.height(16.dp))

        Row(modifier = Modifier.fillMaxWidth(), verticalAlignment =
Alignment.CenterVertically) {
            OutlinedTextField(
                value = password,
                onValueChange = { password = it },
                label = { Text("Password") },
                modifier = Modifier.weight(1f),
                visualTransformation = if (showPassword) VisualTransformation.None else
PasswordVisualTransformation()
            )
            TextButton(onClick = { showPassword = !showPassword }) {
                Text(if (showPassword) "Hide" else "Show")
            }
        }

        if (errorMessage.isNotEmpty()) {
            Text(
                text = errorMessage,
                color = MaterialTheme.colorScheme.error,
                fontSize = 14.sp
            )
            Spacer(Modifier.height(8.dp))
        }

        Spacer(Modifier.height(24.dp))

        Button(
            onClick = {
                if (firstName.isBlank() || lastName.isBlank() || email.isBlank() ||
password.isBlank()) {
                    errorMessage = "Заполните все поля"
                } else if (password.length < 4) {
                    errorMessage = "Пароль должен быть не менее 4 символов"
                } else {
                    errorMessage = ""
                    onRegister(User(email, password, firstName, lastName))
                }
            },
            modifier = Modifier.fillMaxWidth().height(56.dp)
        ) {
            Text("Register", fontSize = 18.sp, fontWeight = FontWeight.Bold)
        }

        Spacer(Modifier.height(24.dp))

        Row(verticalAlignment = Alignment.CenterVertically) {
            Text("Already have an account? ")
            TextButton(onClick = onLogin) {
                Text("Login", fontWeight = FontWeight.Bold)
            }
        }
    }
}

@Composable
fun SuccessScreen() {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(24.dp),
        horizontalAlignment = Alignment.CenterHorizontally,
```

```
        verticalArrangement = Arrangement.Center
    ) {
        Text(
            text = "Вы вошли в систему",
            fontSize = 24.sp,
            fontWeight = FontWeight.Bold
        )
    }
}
```

ВЫВОДЫ

По Заданию 1 :
Освоил принципы ленивой загрузки в Compose — LazyColumn эффективно отображает большие списки, создавая элементы только при их видимости, что экономит ресурсы. Успешно применил композицию UI — использовал Card, Image и Text внутри Row, получив структурированные, визуально приятные карточки с чёткой иерархией информации.

По Заданию 2 :
Настроил кастомную тему через Material Theme Builder — создал уникальную цветовую палитру и типографику, что обеспечило единый и профессиональный вид всему приложению.
Добавил пользовательские шрифты и формы — это повысило визуальную индивидуальность интерфейса и улучшило соответствие дизайн-системе Material Design 3.

По заданию 3:
Освоил продвинутую стилизацию текста в Compose, включая работу с цветами, размерами шрифта, отступами и оформлением. Понял, как управлять поведением текста при переполнении с помощью maxLines и TextOverflow, а также как комбинировать различные параметры для создания разнообразных визуальных эффектов.
По заданию 4:
Научился создавать полностью кастомные интерактивные элементы интерфейса с использованием компонента Card. Применил знания о модификаторах для управления размерами, формой, тенями и границами, что позволило стилизовать кнопку в соответствии с техническим заданием.

По заданию 5:
Освоил работу с компоновкой Box и её параметром contentAlignment, что позволило точно позиционировать элементы. Понял, как использовать комбинацию aspectRatio и fillMaxHeight для создания идеального круга, который адаптируется к высоте контейнера, а также как применять clip для обрезки фигур.

По заданию 6:
Освоил работу с векторными изображениями и их стилизацией в Compose. Научился

позиционировать элементы в углах контейнера с помощью Alignment, а также растягивать изображения с сохранением пропорций через ContentScale и менять их цвет через ColorFilter.

По заданию 7:
Понял разницу между вертикальной (Column) и горизонтальной (Row) компоновками, их параметрами Arrangement и Alignment. Научился создавать структурированные списковые интерфейсы с чередованием фона и добавлять иконки в конец строки через комбинацию компоновщиков.

По заданию 8:
Освоил работу с состоянием (state) в Compose через mutableStateOf и remember. Научился создавать интерактивные интерфейсы, где состояние компонентов динамически изменяется в ответ на пользовательские действия, и реализовал бизнес-логику с условной обработкой данных (сброс счетчика, накопление серии дней).

По заданию 9:
Понял важность декларативного подхода в Compose и научился исправлять типичные ошибки, связанные с побочными эффектами. Освоил правильное вычисление производных значений (сумма, количество) на основе состояния и использование LaunchedEffect для обработки событий, зависимых от изменений состояния.

По заданию 10:
Освоил создание многоэкранных приложений с навигацией на основе состояния в Compose. Научился реализовывать полноценные циклы аутентификации (регистрация и логин) с валидацией полей, управлением видимостью пароля и сохранением пользовательских данных в состоянии приложения.

Ссылка на гит с кодами :
https://github.com/oggiorgi/MIREA/tree/main/5sem/Kotlin/PR2/codes