АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

имени Гумарбека Даукеева

Институт автоматизации и информационных технологий

Кафедра IT-инженерии

**Лабораторная работа №1**

# **Создание простого меню**

По дисциплине: Разработка программных приложений на базе Android

Образовательная программа: 6B06103 – Вычислительная техника и программное обеспечение

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

Проверил: ст.преподаватель: Ахсутова А. А.

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

Алматы 2024

**Содержание**

[Цель работы. 3](#_Toc177904290)

[Листинг программы 4](#_Toc177904291)

[Скриншоты приложение 7](#_Toc177904292)

[Заключение 9](#_Toc177904293)

[Список литературы 10](#_Toc177904294)

Цель работы.

1. **Изучение процесса работы с Android Studio**: В рамках данной задачи необходимо детально ознакомиться с инструментом разработки Android Studio. Это включает установку, настройку, использование встроенных функций для создания и отладки приложений. Android Studio предоставляет широкий спектр возможностей для разработки, включая поддержку Kotlin, Java, инструментов для работы с макетами, симуляторами устройств и анализа производительности приложений. Важно освоить основные элементы интерфейса и понять, как эффективно организовывать рабочий процесс.
2. **Работа с компонентом DropdownMenu**: DropdownMenu — это элемент интерфейса, позволяющий пользователю выбирать из списка опций. В ходе работы необходимо разобраться с созданием и настройкой данного компонента, его параметрами и поведением в приложении. Изучение позволит на практике понять, как настраивать выпадающие меню, обеспечивая пользователю интуитивно понятный интерфейс для выбора.
3. **Работа с компонентом ContextMenu**: Контекстное меню используется для предоставления пользователю дополнительных действий, доступных в контексте определённого элемента интерфейса. В ходе работы необходимо разобраться с тем, как создать и настроить контекстное меню, реагирующее на длительные нажатия или другие жесты пользователя. Также важно понять, как организовать взаимодействие с данным меню для выполнения специфических действий в зависимости от выбора пользователя.

Листинг программы

class MainActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *enableEdgeToEdge*()  
 *setContent* {  
 Lab2Theme {  
 Scaffold(  
 topBar = { TopBar("lab 2") },  
 modifier = Modifier.*fillMaxSize*()  
 ) { innerPadding ->  
 App(modifier = Modifier.*padding*(innerPadding))  
 }  
 }  
 }  
 }  
}  
@Composable  
private fun TopBar(title: String = "") {  
 TopAppBar(  
 title = { Text(text = title, color = MaterialTheme.colorScheme.onPrimary) },  
 colors = TopAppBarDefaults.topAppBarColors(  
 containerColor = MaterialTheme.colorScheme.primary,  
 )  
 )  
}  
@Composable  
private fun App(  
 modifier: Modifier = Modifier  
) {  
 Column(  
 modifier = modifier.*fillMaxSize*()  
 ) {  
 Row(  
 modifier = Modifier.*fillMaxWidth*(),  
 horizontalArrangement = Arrangement.Start  
 ) {  
 Button(onClick = { Log.d("Running", "Start button clicked") }) {  
 Text(text = "start")  
 }  
 }  
 Row(  
 modifier = Modifier.*fillMaxWidth*(),  
 horizontalArrangement = Arrangement.Center  
 ) {  
 Button(  
 modifier = Modifier.*padding*(end = 100.*dp*),  
 onClick = { Log.d("Running", "Center button clicked") }  
 ) {  
 Text(text = "center")  
 }  
 }  
 Row(  
 modifier = Modifier.*fillMaxWidth*(),  
 horizontalArrangement = Arrangement.End  
 ) {  
 Button(onClick = { Log.d("Running", "End button clicked") }) {  
 Text(text = "end")  
 }  
 }  
 }  
}

Описание кода

1. Функция HomeScreen:

* Основная функция, которая отображает главный экран.
* Используются вспомогательные переменные (menuExpanded, extensionMenuExpanded, contextMenuExpanded, homeScreenText, homeTextColor, homeTextSize) для управления состоянием экрана и меню.
  + menuExpanded — определяет, раскрыто ли главное меню.
  + contextMenuExpanded — определяет, открыто ли контекстное меню (всплывающее при долгом нажатии).
  + homeTextColor и homeTextSize — контролируют цвет и размер текста на экране.
* Контент экрана состоит из:
  + TopAppBar (верхняя панель) с названием и кнопкой для вызова меню.
  + DropdownMenu — выпадающее меню, которое управляет размером и цветом текста.
  + Текст посередине экрана. На текст можно долго нажать, чтобы открыть контекстное меню.

1. Меню (HomeScreenDropdownMenu):

* Определяет два набора меню:
  + Главное меню (например, "Reset", "Increase text size", "Decrease text size").
  + Расширенное меню (например, "Set red color", "Set yellow color", "Set green color").
* Выпадающее меню показывается при нажатии на кнопку с иконкой (три точки).
* Для каждой опции меню вызывается обработчик нажатий, и текст обновляется в зависимости от выбранного действия (изменение цвета, увеличение/уменьшение размера текста).

1. Контекстное меню (HomeContextMenu):

* Открывается при долгом нажатии на текст, расположенный в центре экрана.
* Это меню также состоит из пунктов основного меню и расширенного меню для изменения текста.
* Принцип работы аналогичен выпадающему меню.

1. Компонент HomeScreenTopAppBar:

* Верхняя панель с названием и кнопкой для вызова главного меню.
* IconButton реагирует на нажатие и вызывает раскрытие меню через передачу функции onMenuClicked.

1. Обработка состояний:

* Переменные menuExpanded, contextMenuExpanded, homeTextColor и homeTextSize хранят состояния, которые изменяются при взаимодействии с элементами меню.
* Например, при выборе "Increase text size" текст увеличивается, при выборе цвета текст изменяет свой цвет.

1. Использование Toast:

* В каждой функции меню, при нажатии на пункт меню, вызывается Toast, который показывает сообщение с названием выбранного пункта меню.

Скриншоты приложение

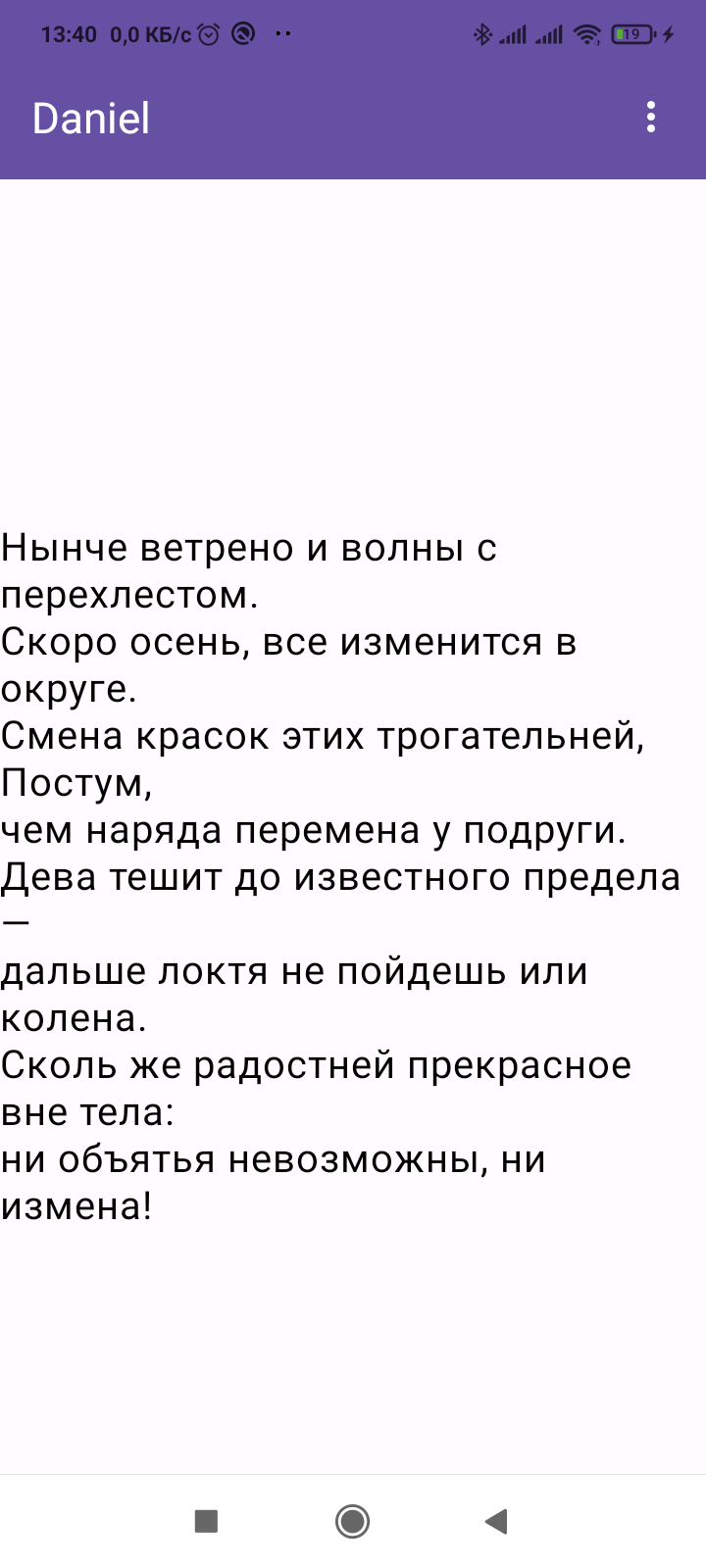


Рисунок 1 – Скриншоты главного экрана

Заключение

В ходе выполнения данной работы были изучены основы создания интерфейсов в Android с использованием Jetpack Compose. Основное внимание было уделено созданию компонентов через программный код, без применения XML-верстки. Также были рассмотрены базовые компоненты, такие как Column и Row, позволяющие гибко управлять расположением элементов на экране.

Кроме того, была реализована работа с кнопками и их позиционированием в разных частях экрана, что позволяет понять, как работает система компоновки в Compose. Использование Modifier для настройки компонентов продемонстрировало его мощь в управлении размерами, отступами и положением элементов.

Полученные знания и навыки позволяют уверенно разрабатывать интерфейсы в Compose, эффективно использовать его декларативный подход и грамотно управлять пользовательским взаимодействием через логику внутри компонентов.

Список литературы

1. Филлипс Б., Стюарт К. Программирование под Android. Большая книга рецептов. — М.: Вильямс, 2021.
2. Ломбардо М. В. Android. Программирование для профессионалов. — СПб.: Питер, 2020.
3. Google. Официальная документация по Android. <https://developer.android.com/docs>.
4. Мёрфи М. Программирование под Android. Полное руководство для профессионалов. — М.: Диалектика, 2019.