# Лабораторная работа №19 Отображение списков и таблиц

1. Цель работы
   1. Изучить процесс настройки и использования элементов управления для отображения списков в приложении Android.
2. Литература
   1. Jetpack Compose | Руководство. metanit.com – Текст : электронный // metanit.com, 2023. – URL: https://metanit.com/kotlin/jetpack/ – гл.5.
3. Подготовка к работе
   1. Повторить теоретический материал (см.п.2).
   2. Изучить описание лабораторной работы.
4. Основное оборудование
   1. Персональный компьютер.
5. Задание
   1. Настройка карточки товара с использованием элемента Card
      1. Добавить в приложение класс Product для хранения данных о товаре:

- артикул,

- название,

- цена,

- идентификатор изображения из ресурсов (тип данных: Int)

* + 1. Добавить в приложение список из 20 товаров для последующего использования в других функциях.
    2. Добавить в приложение composable-функцию с параметром product для вывода информации о товаре при помощи компонента Card. Слева должно быть изображение товара. Справа от изображения – название товара, под названием – цена товара.
    3. Добавить в приложение composable-функцию с параметром product для вывода информации о товаре при помощи компонента Card. Сверху должно быть фото товара в виде круглой картинки, под картинкой – название товара и его цена.
    4. Настроить у карточек границы, цвет фона, цвет текста, тень, отступы.
  1. Отображение элементов списка в LazyColumn
     1. Добавить в приложение composable-функцию. В ней реализовать отображение карточек из п.5.1.3 в виде списка LazyColumn. При необходимости для наглядности добавить элементам отступы, чтобы список не помещался на экран полностью.
     2. Изменить вывод, чтобы в первой строке LazyColumn отображалась метка с названием выбранного товара.
  2. Отображение элементов списка в LazyRow
     1. Добавить в приложение composable-функцию. В ней реализовать отображение данных о товарах в виде списка карточек из п.5.1.4 в виде LazyRow.
  3. Реализация перехода к первому элементу
     1. Внести изменения в функцию из п.5.2, чтобы поверх LazyColumn справа снизу отображалась кнопка «Наверх».
     2. Реализовать видимость кнопки, если текущий элемент – не первый. Для этого:

- добавить в функцию переменную listState:

val listState = rememberLazyListState()

- добавить в LazyColumn свойство state:

LazyColumn(state = listState)

- добавить в функцию переменную

val showButton = remember { derivedStateOf { listState.firstVisibleItemIndex > 0 } }

- связать свойство доступности кнопки с переменной showButton

* + 1. Реализовать переход к первому элементу списка. Для этого:

- добавить в функцию переменную coroutineScope:

val coroutineScope = rememberCoroutineScope()

- при нажатии на кнопку «Наверх» выполнить следующий код:

coroutineScope.launch { listState.scrollToItem(index = 0) }

* 1. Отображение элементов списка в виде сетки LazyVerticalGrid
     1. Добавить в приложение composable-функцию ProductVerticalGrid. В ней использовать список из п.5.1.2.
     2. Реализовать отображение в сетке карточек из п.5.1.4 с данными о товарах, при необходимости измените макет так, чтобы в одной строке помещалось несколько карточек
  2. Отображение элементов списка в виде сетки LazyHorizontalGrid
     1. Добавить в приложение composable-функцию ProductHorizontalGrid. В ней использовать список из п.5.1.2.
     2. Реализовать отображение в сетке карточек из п.5.1.4 с данными о товарах, при необходимости измените макет так, чтобы в одном столбце помещалось несколько карточек

1. Порядок выполнения работы
   1. Выполнить все задания из п.5.
   2. Ответить на контрольные вопросы.
2. Содержание отчета
   1. Титульный лист
   2. Цель работы
   3. Ответы на контрольные вопросы
   4. Вывод
3. Контрольные вопросы
   1. Что такое «LazyColumn»?
   2. Что такое «LazyRow»?
   3. Что такое «LazyVerticalGrid»?
   4. Что такое «LazyHorizontalGrid»?