|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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 практической работе № 7**

по дисциплине «Разработка мобильных приложений»

|  |  |
| --- | --- |
| **Выполнил:**  Студент группыИКБО-03-21 | Хречко С.В. |
| **Проверил:**  Ассистент | Новичков Д.Е. |

2023 г.

Цель работы

Целью данной работы является получение навыков работы c различными паттернами архитектуры мобильных приложений.

Задание

1. Обновить структуру разработанного на прошлых практических работах мобильного приложения по выбранной предметной области на основе общих рекомендаций к архитектуре мобильных приложений,

1.1. Реализовать или реструктурировать уже реализованные файлы с программным кодом пользовательского интерфейса и хранилищ состояния пользовательского интерфейса в подпакет UI слоя.

1.2. Реализовать или реструктурировать уже реализованные файлы с программным кодом моделей данных и источниками данных в подпакет слоя данных.

1.3. Реализовать при необходимости предметной области или реструктурировать уже реализованные файлы с программным кодом переиспользуемой бизнес-логики в подпакет доменного слоя.

2. Реализовать архитектуру разрабатываемого приложения по паттерну MVVM с использованием компонентов LiveData в элементах паттерна.

2.1. Реализовать по предметной области модели данных, используемых в приложении.

2.2. Реализовать по предметной области и/или реструктуризировать ранее разработанные классы пользовательского интерфейса в компоненты View паттерна MVVM. В приложении должно быть не менее 5 отдельных компонентов View, обусловленных предметной областью и имеющей логическую и смысловую нагрузку, связанных между собой данными и навигационными переходами. Обязательной парой таких компонентов должны быть выполненные в практической работе №4 экран со списком элементов и экран с демонстрацией информации элемента списка.

2.3. Реализовать по предметной области делегаты ViewModel, необходимые для работы компонентов View в разрабатываемом приложении.

2.4.Реализовать по предметной области Repository с несколькими простейшими DataSource классами. Данные в DataSource-классах могут храниться в виде массивов и ассоциативных массивов.

Решение

Были созданы дополнительные подпакеты dataLayerPackage и UIPackage. Новое расположение компонентов представлено на рисунке 1.

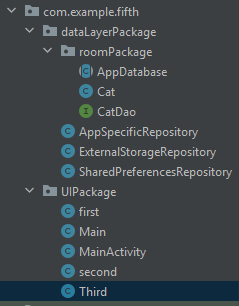


Рисунок 1 – Новое расположение компонентов

Были созданы классы, отвечающие за управление данными программы. Вся логика работы с данными была перенесена в них. Пример такого класса представлен на рисунке 2.

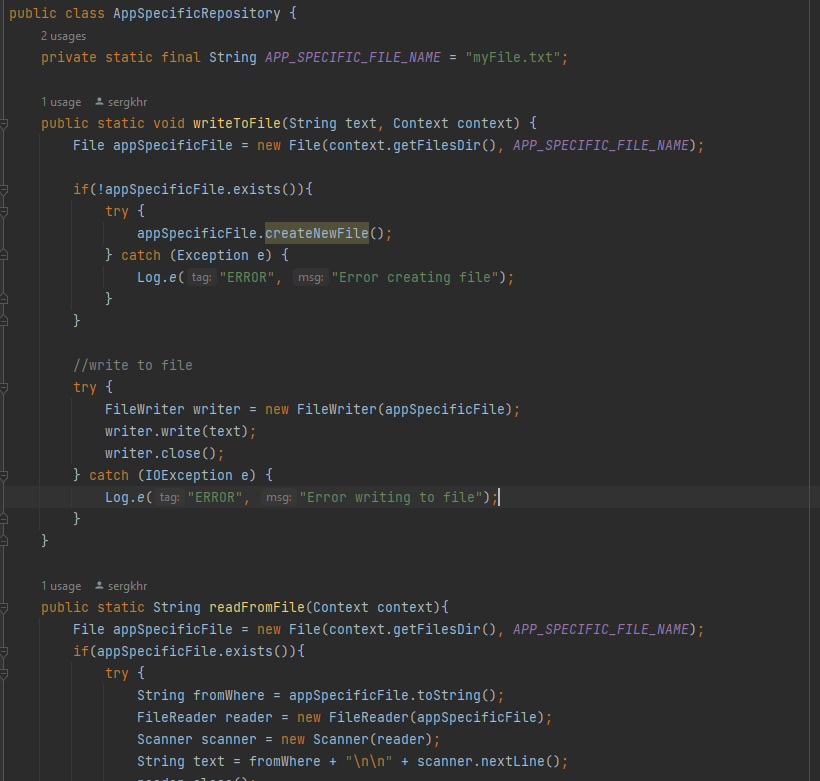


Рисунок 2 – Пример класса данных

Модель данных, используемая в работе, представлена на рисунке 3.

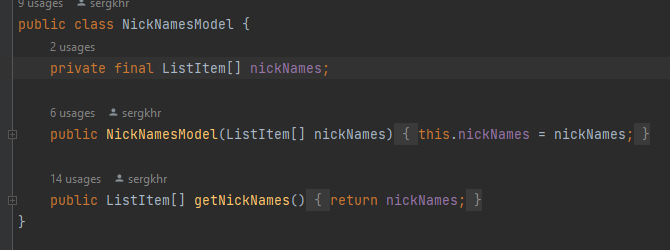


Рисунок 3 – Модель данных

На рисунке 4 представлен механизм отслеживания изменений данных в модели. Аналогично выстроен механизм во всех элементах view.

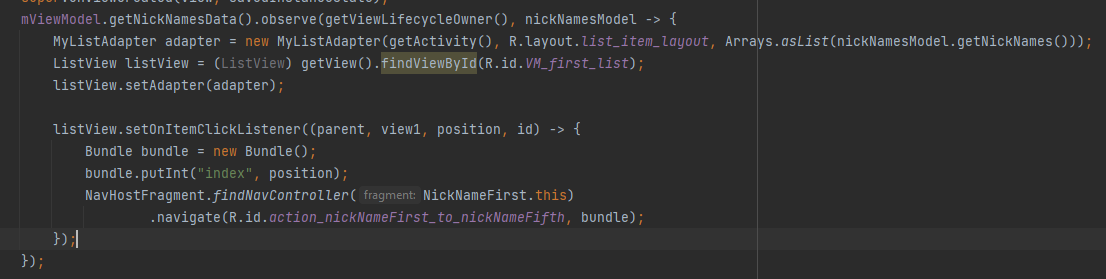


Рисунок 4 – Метод observe

На рисунке 5 представлена настройка LiveData для viewModel, класса в котором представлены методы для обработки запросов от view.

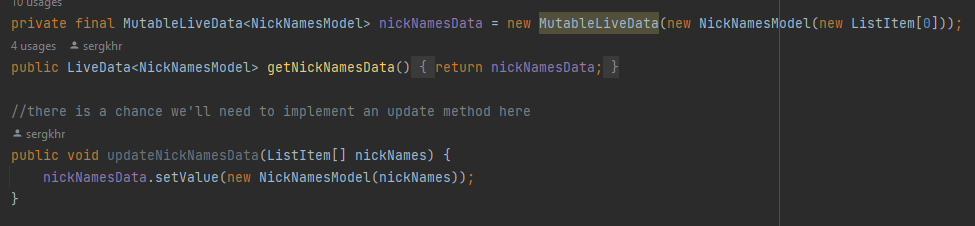


Рисунок 5 – Создание LiveData

На рисунках 6-10 представлены пять элементов view, созданные в процессе выполнения работы.

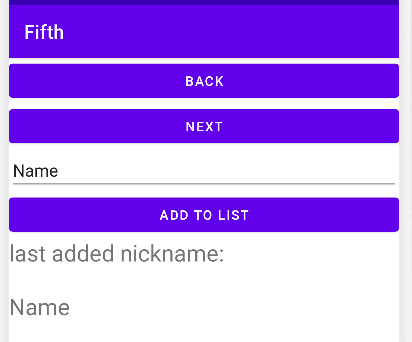


Рисунок 6 – View добавления имени

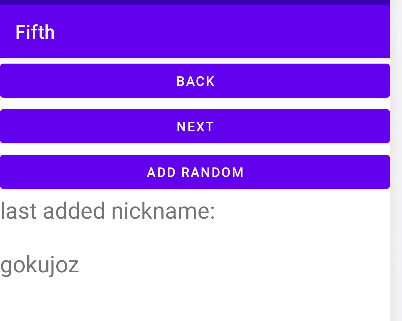


Рисунок 7 – View добавления случайного имени

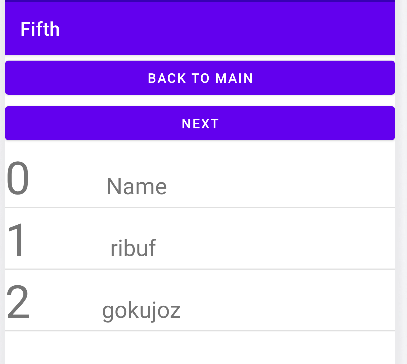


Рисунок 8 – View списка имен

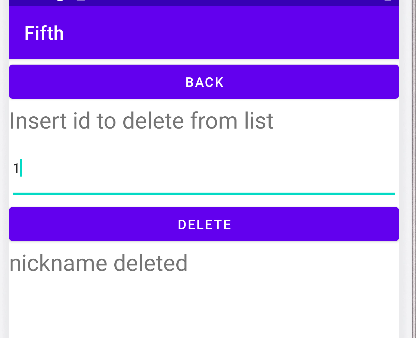


Рисунок 9 – View удаления имени

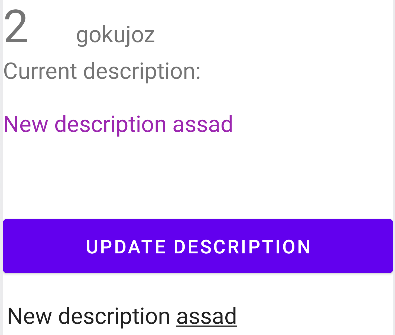


Рисунок 10 – View описания одного элемента списка

Вывод

В процессе выполнения работы были получены навыки работы с паттерном MVVM. Был переформатирован код, для более удобной работы с ним. Были разработаны несколько элементов view, а также настроена работа с моделей данных через viewModel.