МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«Новосибирский национальный исследовательский государственный университет»

(Новосибирский государственный университет, НГУ)

Структурное подразделение Новосибирского государственного университета – Высший колледж информатики Университета (ВКИ НГУ)

КАФЕДРА ИНФОРМАТИКИ

ОТЧЕТ КУРСОВОГО ПРОЕКТА

Разработка мобильных приложений

ПРИЛОЖЕНИЕ «MatrixEditor»

|  |  |
| --- | --- |
| Руководитель  доцент ВКИ НГУ | Никифоров А.А  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2024 г. |
| Студентка 3 курса  гр. 107сб | Косинова А.И  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2024 г. |

Новосибирск

2024

**СОДЕРЖАНИЕ**

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ 4](#_Toc151776047)

[ВВЕДЕНИЕ 5](#_Toc151776048)

[1 ПОСТАНОВКА ЗАДАЧИ 6](#_Toc151776049)

[1.1 Бизнес-требования 6](#_Toc151776050)

[1.2 Пользовательские требования 6](#_Toc151776051)

[1.3 Системные требования 6](#_Toc151776052)

[1.4 Требования к графическому пользовательскому интерфейсу 7](#_Toc151776053)

[2 АНАЛИЗ ТРЕБОВАНИЙ И ОПРЕДЕЛЕНИЕ СПЕЦИФИКАЦИЙ 8](#_Toc151776054)

[2.1 Описание предметной области задачи 8](#_Toc151776055)

[2.1.1 Информационные объекты предметной области и взаимосвязи между ними 8](#_Toc151776056)

[2.1.2 Информационные и функциональные потребности пользователей разрабатываемой ПС (ПМ) 8](#_Toc151776057)

[2.1.3 Методы работы с информационными объектами предметной области 8](#_Toc151776058)

[2.1.5 Концептуальное обоснование разработки 9](#_Toc151776059)

[2.2 Классы и характеристики пользователей 9](#_Toc151776060)

[2.3 Функциональные требования 9](#_Toc151776061)

[2.3.1 Определение функциональных возможностей ПС (ПМ) 9](#_Toc151776062)

[2.3.2 Описание прецедентов 9](#_Toc151776063)

[2.4 Нефункциональные требования 10](#_Toc151776064)

[3 ВЫБОР ПРОГРАММНЫХ СРЕД И СРЕДСТВ РАЗРАБОТКИ 11](#_Toc151776065)

[3.1 Сравнительный анализ имеющихся возможностей по выбору средств разработки 11](#_Toc151776066)

[3.2 Характеристика выбранных программных сред и средств 12](#_Toc151776067)

[4 АЛГОРИТМ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ 13](#_Toc151776068)

[4.1 Этапы реализации ПС (ПМ) 13](#_Toc151776069)

[4.2 Пользовательский интерфейс ПС (ПМ) 14](#_Toc151776070)

[4.2.1 Взаимодействие пользователей с ПС (ПМ) 14](#_Toc151776071)

[4.2.2 Проектирование пользовательских сценариев 14](#_Toc151776072)

[4.2.3 Определение операций пользователей 14](#_Toc151776073)

[4.2.4 Составление функциональных блоков 15](#_Toc151776074)

[4.2.5 Проектирование структуры экранов ПС (ПМ) и схемы навигации 15](#_Toc151776075)

[4.3 Входные, выходные и промежуточные данные 16](#_Toc151776076)

[4.4 Алгоритмы использования применяемых программных технологий 16](#_Toc151776077)

[4.5 Архитектура и схема функционирования ПС (ПМ) 17](#_Toc151776078)

[6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 18](#_Toc151776079)

[ЗАКЛЮЧЕНИЕ 24](#_Toc151776080)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 26](#_Toc151776081)

# ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ

SQLite – встроенная база данных для хранения необходимой информации.

ИП (Интерфейс Пользователя) – визуальные компоненты, обеспечивающие взаимодействие с пользователем.

CRUD (Create, Read, Update, Delete) – основные операции управления данными матриц.

SQL (Structured Query Language) – язык для управления и запросов к базам данных.

XML (eXtensible Markup Language) – язык разметки для хранения и передачи данных.

SDK (Software Development Kit) – набор программных инструментов для разработки приложений.

# ****ВВЕДЕНИЕ****

В настоящее время, с растущей зависимостью от информационных технологий и мобильных устройств, мобильные приложения играют ключевую роль в удовлетворении потребностей пользователей[1]. Однако, в контексте разработки и использования мобильных приложений, существует значительная потребность в инструментах, которые могли бы эффективно работать с данными, особенно в случаях, когда эти данные представлены в виде матриц. В данном контексте представляется актуальной задача создания и использования мобильного приложения "MatrixEditor" для операционной системы Android.

"MatrixEditor" — это мобильное приложение, разработанное для удобного отображения, редактирования и анализа матриц, хранящихся в базе данных. Способность просматривать и взаимодействовать с матрицами имеет огромное значение в различных областях.

Приложение "MatrixEditor" предоставляет пользователю возможность не только просматривать матрицы, но и редактировать их, строить новые, выделять цветом важные элементы, а также выбирать тип матрицы, что позволяет более гибко и эффективно работать с данными. Это особенно полезно в контексте различных задач, где матрицы играют ключевую роль, таких как анализ данных, моделирование и принятие решений.

С учетом актуальности вопросов обработки данных и возрастающей потребности в удобных инструментах для работы с ними, мобильное приложение "MatrixEditor" представляет собой ценный ресурс для пользователей Android-платформы, открывая новые перспективы для более удобной и эффективной работы с матрицами и данными в целом.

# ****1 ПОСТАНОВКА ЗАДАЧИ****

## 1.1 Бизнес-требования

MatrixEditor – это приложение для операционной системы Android, разработанное на языке программирования Java. Основная цель приложения заключается в предоставлении пользователям инструмента для работы с матрицами: создания, редактирования и сохранения. Приложение рассчитано на широкую аудиторию, включая студентов. Главная задача MatrixEditor – упростить процесс работы с матрицами, обеспечив удобный и интуитивно понятный интерфейс.

## 1.2 Пользовательские требования

1) Создание матриц: возможность указания размеров матрицы. Заполнение матрицы значениями.

2) Редактирование матриц: изменение значений элементов матрицы. Изменение размера матрицы.

3) Сохранение матриц: возможность сохранения созданных или отредактированных матриц.

Система управления пользователями для хранения матриц по профилю.

4) Управление матрицами: просмотр списка сохраненных матриц.

Возможность выбора матрицы для последующего редактирования.

5) Интерфейс: интуитивно понятный интерфейс для облегчения взаимодействия пользователя с приложением.

## 1.3 Системные требования

1. Модуль создания и редактирования матриц:

1.1 Реализация функционала для создания и редактирования матриц.

1.2 Интеграция с базой данных для хранения матриц.

2. Модуль хранения матриц в базе данных:

2.1.Реализация базы данных для эффективного хранения матриц различных размеров.

2.2. Обеспечение безопасного взаимодействия с базой данных.

3. Модуль управления пользователями и матрицами:

3.1.Создание системы управления пользователями.

3.2. Интеграция с функционалом сохранения и выбора матриц.

## 1.4 Требования к графическому пользовательскому интерфейсу

1. Главный экран: кнопки для создания новой матрицы или выбора существующей.
2. Экран создания/редактирования матрицы: ввод размеров матрицы. Заполнение значений матрицы.
3. Интерфейс взаимодействия: интуитивно понятные элементы управления. Адаптация интерфейса к разным размерам экранов.
4. Единый стиль оформления.

# ****2 АНАЛИЗ ТРЕБОВАНИЙ И ОПРЕДЕЛЕНИЕ СПЕЦИФИКАЦИЙ****

## ****2.1 Описание предметной области задачи****

### 2.1.1 Информационные объекты предметной области и взаимосвязи между ними

1) Матрица:

1.1. Определение размеров и значений элементов.

1.2. Уникальный идентификатор.

2) База данных:

2.1 Хранение матриц с учетом их параметров.

2.2. Управление данными пользователей.

3) Пользователь:

3.1. Регистрация и аутентификация.

3.2. Связь с сохраненными матрицами.

### 2.1.2 Информационные и функциональные потребности пользователей разрабатываемой ПС (ПМ)

1. Создание новых матриц с заданными параметрами.
2. Редактирование существующих матриц.
3. Сохранение матриц для последующего использования.
4. Удобное управление базой данных матриц.

### 2.1.3 Методы работы с информационными объектами предметной области

1. Создание матриц: ввод размеров и значений. Генерация уникального идентификатора.

2. Редактирование матриц: изменение значений элементов. Изменение размера матрицы.

3. Сохранение матриц: запись матрицы в базу данных. Связь с профилем пользователя.

### 2.1.5 Концептуальное обоснование разработки

Разработка MatrixEditor обоснована необходимостью создания универсального инструмента для работы с матрицами на платформе Android, обладающего функциональностью редактирования, создания и управления матрицами, а также интеграцией с базой данных для обеспечения сохранности данных пользователей.

## 2.2 Классы и характеристики пользователей

Студенты:

Используют для выполнения учебных задач.

Основной функционал: создание и редактирование матриц.

## 2.3 Функциональные требования

### 2.3.1 Определение функциональных возможностей ПС (ПМ)

Пользователь может:

1. Создавать новые матрицы с указанием размеров.
2. Редактировать значения элементов матрицы.
3. Сохранять созданные или отредактированные матрицы.
4. Выбирать матрицу для редактирования.

### 2.3.2 Описание прецедентов

1. Создание матрицы: ввод пользователем размеров матрицы. Генерация уникального идентификатора. Возможность ввода значений элементов.

2. Редактирование матрицы: выбор матрицы из списка. Изменение значений элементов. Изменение размера матрицы.

3. Сохранение матрицы: сохранение матрицы в базе данных. Автоматическое привязывание к профилю пользователя.

## 2.4 ****Нефункциональные требования****

Группы нефункциональных требований:

1. Производительность и масштабируемость: время отклика системы не более 1 секунды.
2. Переносимость и совместимость: приложение должно работать на устройствах с Android 5.0 и выше. Совместимость с различными разрешениями экранов.
3. Надёжность, доступность, ремонтопригодность: вероятность сбоев не более 1%. Восстановление после сбоя не более 5 секунд.
4. Удобство использования (юзабилити): интуитивно понятный интерфейс. Адаптация под разные размеры экранов Android-устройств.

# ****3**** ****ВЫБОР ПРОГРАММНЫХ СРЕД И СРЕДСТВ РАЗРАБОТКИ****

## 3.1 Сравнительный анализ имеющихся возможностей по выбору средств разработки

1. Android Studio:

Достоинства:

Официальная IDE для Android-разработки.

Интеграция с Android SDK.

Поддержка языка программирования Java.

Недостатки:

Высокие системные требования.

Некоторая сложность для новичков.

1. IntelliJ IDEA:

Достоинства:

Полная поддержка Java.

Удобный интерфейс и множество плагинов.

Широкие возможности по настройке.

Недостатки:

Платная версия содержит больше функций.

Не так сильно ориентирована на мобильную разработку, как Android Studio.

1. Eclipse:

Достоинства:

Бесплатная и открытая среда разработки.

Широкий выбор плагинов и расширений.

Недостатки:

Уступает Android Studio в интеграции с Android SDK.

Менее современный интерфейс.

## 3.2 Характеристика выбранных программных сред и средств

1. Android Studio – официальная IDE для разработки приложений под Android.

Интеграция с Android SDK обеспечивает доступ к необходимым инструментам.

Достоинства:

Официальная поддержка Google.

Широкие возможности для работы с Android-проектами.

Недостатки:

Высокие требования к ресурсам компьютера.

1. SQLite – встроенная система управления базами данных, идеально подходящая для мобильных приложений.

Достоинства:

Легкая интеграция в Android-приложения.

Надежная и компактная база данных.

Недостатки:

Менее подходит для крупных и сложных баз данных.

Android Studio является наилучшим выбором для разработки приложения MatrixEditor. Он предоставляет официальную и полноценную поддержку Android, интегрируется с необходимыми инструментами, и обладает обширным сообществом разработчиков. Хоть IntelliJ IDEA также является мощным инструментом, Android Studio предоставляет специфические возможности, необходимые для разработки под Android. Использование SQLite для базы данных обеспечит компактность и легкость интеграции.

# ****4**** АЛГОРИТМ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ

## Этапы реализации ПС (ПМ)

1. Подготовительный этап:

Определение требований к приложению.

Анализ существующих аналогов.

Выбор программных средств разработки.

1. Проектирование:

Разработка структуры базы данных для хранения матриц.

Проектирование пользовательского интерфейса.

Определение операций, доступных пользователям.

1. Реализация:

Написание кода приложения в Android Studio.

Создание и настройка базы данных SQLite.

Разработка логики взаимодействия с базой данных.

Реализация функций создания, редактирования и сохранения матриц.

1. Тестирование:

Проведение модульного тестирования отдельных компонентов.

Проверка взаимодействия приложения с базой данных.

Проведение интеграционного тестирования.

1. Отладка и оптимизация:

Выявление и исправление ошибок.

Оптимизация производительности приложения.

1. Документирование:

Написание технической документации.

Подготовка руководства пользователя.

## 4.2 Пользовательский интерфейс ПС (ПМ)

### 4.2.1 Взаимодействие пользователей с ПС (ПМ)

1. Создание матрицы:

Пользователь выбирает опцию "Создать матрицу" из меню.

Вводит необходимые параметры (размерность, элементы).

Нажимает кнопку "Создать".

1. Редактирование матрицы:

Пользователь выбирает матрицу из списка.

Вносит изменения в элементы матрицы.

Сохраняет изменения.

1. Хранение матрицы в базе данных:

При создании или редактировании матрицы, данные сохраняются в базе данных SQLite.

### Проектирование пользовательских сценариев

Сценарий 1. Создание матрицы:

1. Ввод параметров (размерность, элементы).
2. Визуализация созданной матрицы.
3. Редактирование матрицы:
4. Выбор матрицы из списка.
5. Редактирование элементов.
6. Подтверждение изменений.

### Определение операций пользователей

1. Создание матрицы:

Ввод размерности и элементов.

1. Редактирование матрицы:

Выбор матрицы.

Изменение значений.

1. Удаление матрицы:

Выбор матрицы.

Подтверждение удаления.

### Составление функциональных блоков

Проектируемые функциональные блоки для приложения MatrixEditor могут включать следующие:

1. Блок Редактирования матрицы:
2. Возможность ввода значений элементов матрицы.
3. Операции редактирования, такие как изменение значений, добавление и удаление строк и столбцов.
4. Блок Создания матрицы:
5. Возможность создания новых матриц с заданными размерами.
6. Инициализация матрицы нулями или случайными значениями.
7. Блок Хранения матриц в базе данных:
8. Операции сохранения матрицы в базе данных.
9. Загрузка матрицы из базы данных для последующего редактирования.
10. Блок Навигации и Управления:
11. Меню и элементы управления для перехода между различными экранами и выполнения операций.

### Проектирование структуры экранов ПС (ПМ) и схемы навигации

Проектируемая структура экранов включает в себя:

1. Главный экран:  
   Где пользователь может переходить к страницам редактирования, создания и настройкам.
2. Экран Редактирования Матрицы:

Где пользователь может вводить и изменять значения элементов матрицы.

1. Экран Создания Матрицы:

Для создания новых матриц с выбранными параметрами.

1. Экран Настроек:

Настройки приложения, такие как предпочтительные параметры отображения.

## Входные, выходные и промежуточные данные

Входные данные:

1. Для редактирования матрицы: пользователь вводит значения элементов матрицы через интерфейс.
2. Для создания матрицы: пользователь указывает размеры и параметры новой матрицы.
3. Для хранения в базе данных: данные о матрицах сохраняются в виде записей в базе данных.

Промежуточные данные:

1. Матрицы, созданные или редактированные пользователем, хранятся в оперативной памяти до сохранения или загрузки в базу данных.

Выходные данные:

1. Отредактированная матрица выводится на экран пользователя.
2. При сохранении в базе данных, информация о матрице добавляется в соответствующую таблицу.

## Алгоритмы использования применяемых программных технологий

Применяемые технологии:

1. Android Studio (Java): Для разработки мобильного приложения под Android.
2. SQLite: Для работы с встроенной реляционной базой данных на устройстве.

Алгоритмы использования:

Работа с Android Studio:

1. Разработка пользовательского интерфейса.
2. Реализация логики взаимодействия с пользователем.

Использование SQLite:

1. Создание базы данных и таблиц в приложении.
2. Реализация операций сохранения и загрузки матриц в/из базы данных.

## 4.5 Архитектура и схема функционирования ПС (ПМ)

Функциональные блоки:

1. Редактирование матрицы.
2. Создание матрицы.
3. Хранение и управление матрицами.
4. Навигация и управление приложением.

Описание функциональных блоков:

Редактирование матрицы: позволяет пользователю вводить и изменять значения элементов матрицы.

Создание матрицы: обеспечивает возможность создания новых матриц с заданными параметрами.

Хранение и управление матрицами: включает операции сохранения, загрузки и удаления матриц из базы данных.

Навигация и управление приложением: обеспечивает переход между экранами и выполнение основных операций.

# ****6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ****

Установка приложения MatrixEditor:

1. Скачайте приложение
2. Установите приложение на вашем устройстве.

Главный экран приложения отображает набор кнопок, через которые можно взаимодействовать со всеми интерфейсами приложения. Данный экран изображен на рисунке 6.1

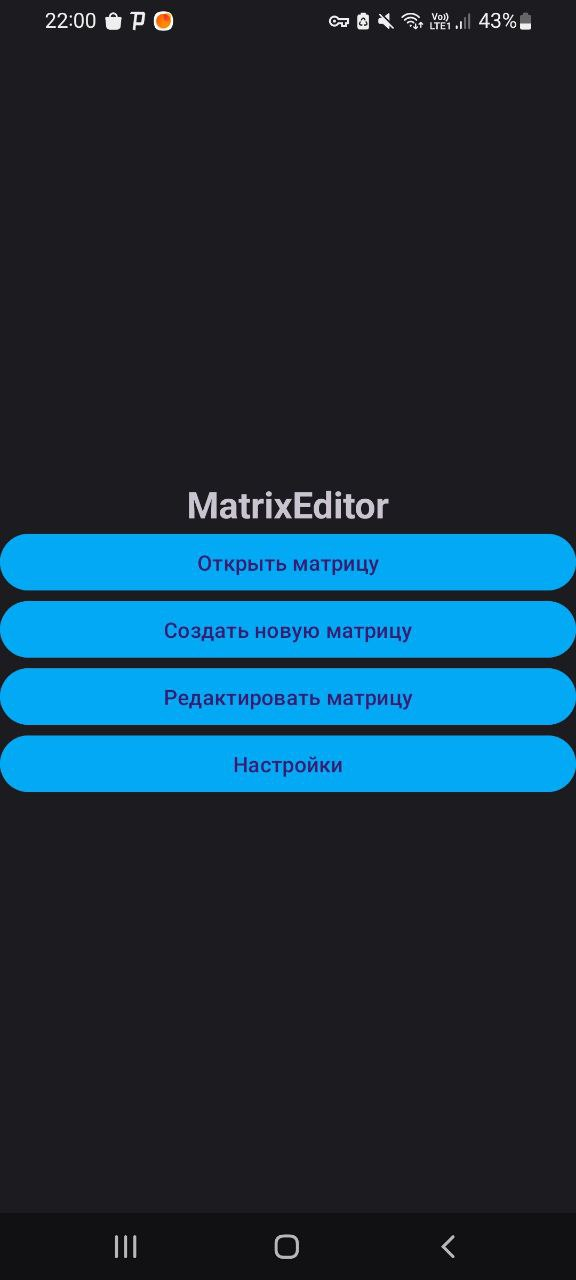


Рисунок 6.1 – Главный экран

Настройка приложения:

1. После установки запустите приложение.
2. Зайдите в «Настройки» из главного окна (рис. 6.6)
3. Измените необходимые параметры.

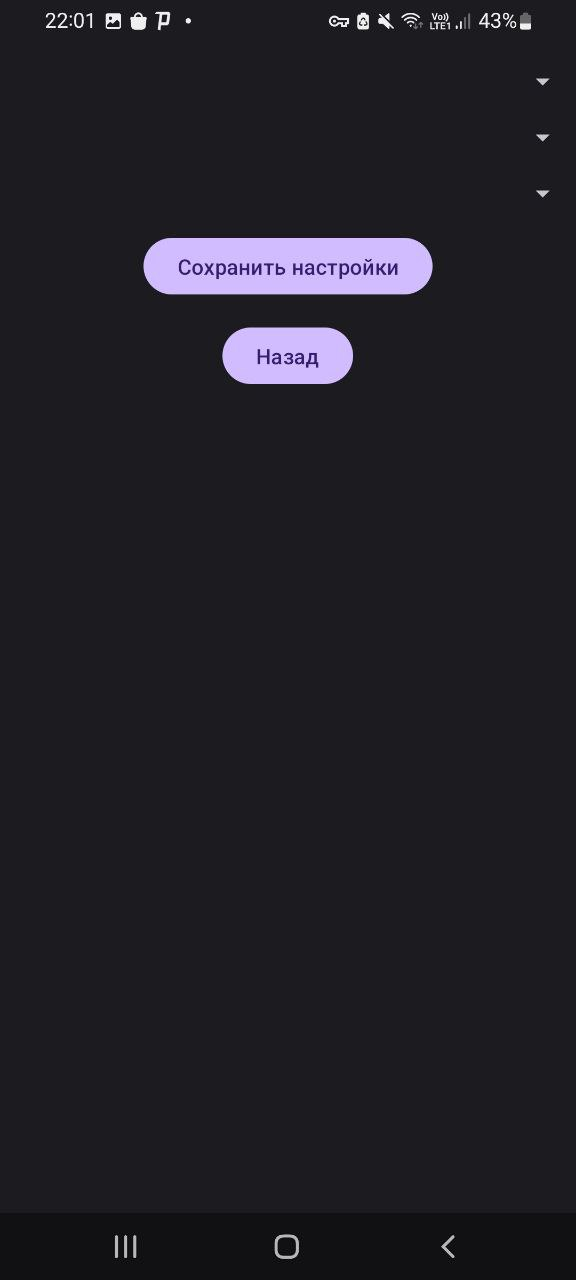


Рисунок 6.6 – Окно настроек для приложения

Просмотр матрицы:

Чтобы просмотреть матрицу, нужно нажать на кнопку «Открыть матрицу», где можно увидеть матрицу. Данный экран изображен на рисунке 6.2



Рисунок 6.2 – Экран с отображением матрицы

Чтобы вывести матрицу из базы данных, которая установлена по умолчанию, можно нажать на кнопку «Матрица по умолчанию». Данное действие отображено на рисунке 6.3

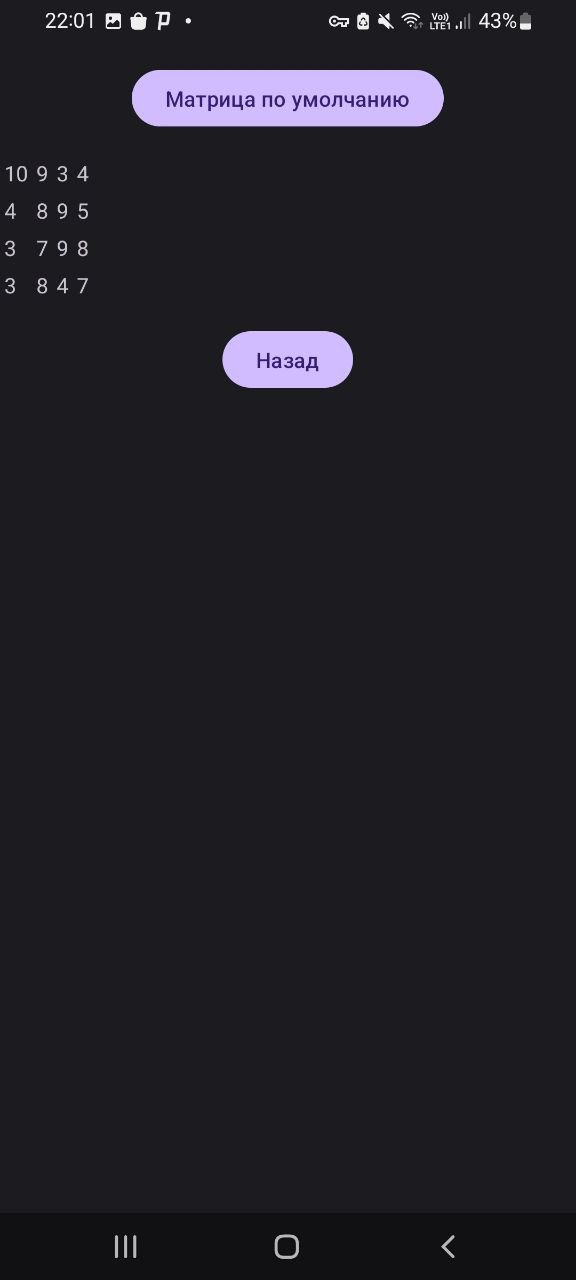


Рисунок 6.4 – Вывод матрицы по умолчанию

Если нажать кнопку «Назад», приложение вернется на главный экран. Оттуда можно перейти в окно «Создать новую матрицу». Данный экран изображен на рисуноке 6.5.

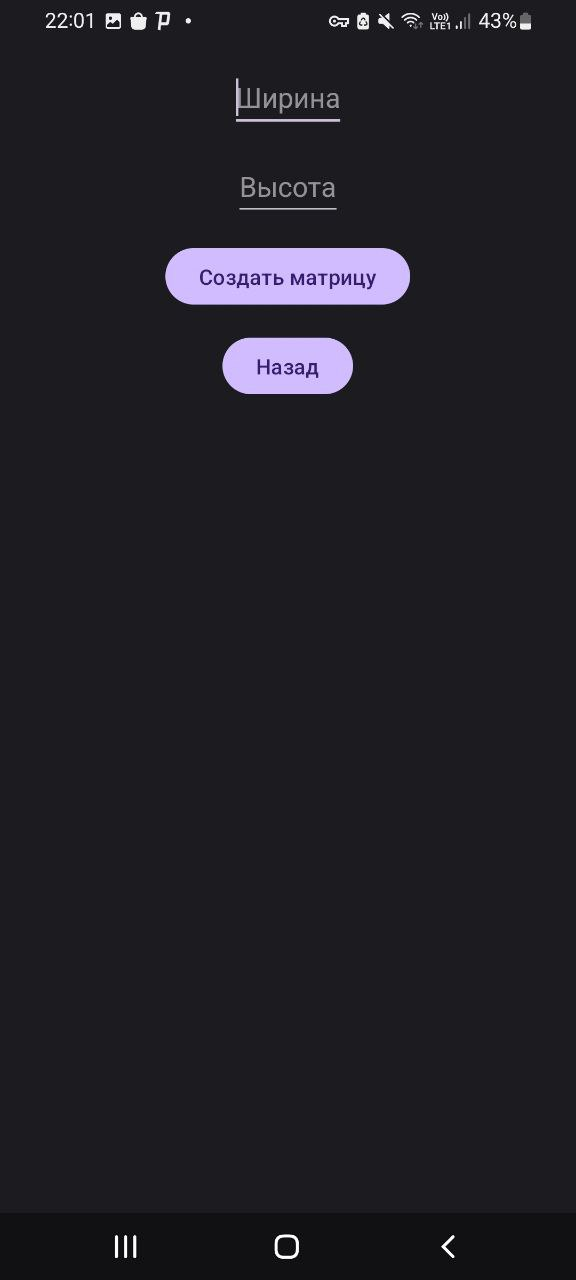


Рисунок 6.5 – Окно создания новой матрицы

В данной форме можно ввести свои данные в базу данных, задав ей длину и ширину. Пример изображен на странице 6.6.

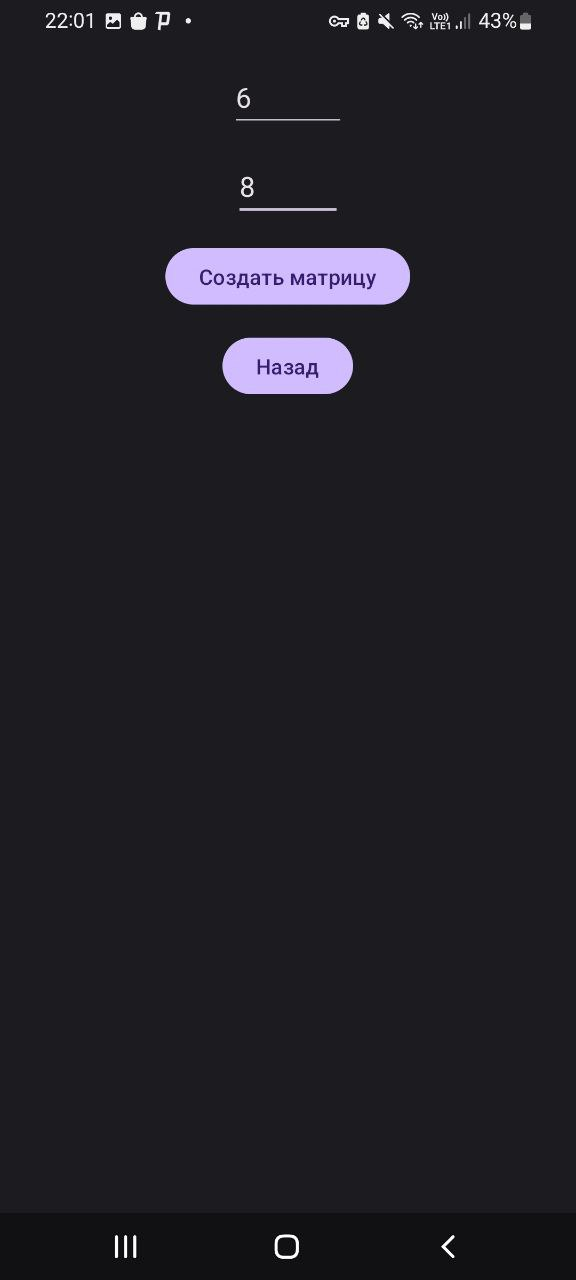


Рисунок 6.6 – Ввод параметров для матрицы

Также есть возможность задать конкретные значения для матрицы в окне «Редактировать матрицу».

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы было рассмотрено создание приложения для редактирования, создания и хранения матриц на платформе Android. Разработано приложение под названием MatrixEditor, написанное на языке программирования Java в среде разработки Android Studio.

На этапе анализа предметной области были выделены основные информационные объекты, взаимосвязи между ними и функциональные потребности пользователей. Это позволило четко определить цели и задачи приложения, а также выбрать подходящие технологии для реализации.

Основной целью приложения стало обеспечение пользователей возможностью удобного создания, редактирования и хранения матриц. Проанализированы существующие программные решения, что позволило выделить уникальные черты и особенности разрабатываемого приложения.

На этапе проектирования была разработана структура базы данных для хранения матриц, а также определены основные элементы пользовательского интерфейса. Это включает в себя возможность создания, редактирования и сохранения матриц в базе данных.

В процессе реализации был написан код приложения, включающий в себя логику работы с базой данных SQLite, создание пользовательского интерфейса и обработку пользовательских действий. Важным аспектом было обеспечение стабильной работы и удобства использования.

Проведено тестирование, отладка и оптимизация приложения с целью выявления и устранения ошибок, а также повышения производительности. Результатом стало стабильное приложение, готовое к использованию конечными пользователями.

Созданная документация, включая техническую документацию и руководство пользователя, предоставляет подробное описание функционала приложения, его возможностей и способов использования.

Таким образом, приложение MatrixEditor успешно разработано и готово к внедрению в повседневную практику пользователей, предоставляя им удобный инструмент для работы с матрицами на мобильных устройствах.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. ОТЧЕТ РЫНКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ ЗА 2022 ГОД URL : <https://asomobile.net/blog/otchet-rynka-mobilnyh-prilozhenij-za-2022-god/>.