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](#_Toc151320731)

[ВВЕДЕНИЕ 5](#_Toc151320732)

[1 ПОСТАНОВКА ЗАДАЧИ 6](#_Toc151320733)

[1.1 Бизнес-требования 6](#_Toc151320734)

[1.2 Пользовательские требования 6](#_Toc151320735)

[1.3 Системные требования 6](#_Toc151320736)

[1.4 Требования к графическому пользовательскому интерфейсу 7](#_Toc151320737)

[2 АНАЛИЗ ТРЕБОВАНИЙ И ОПРЕДЕЛЕНИЕ СПЕЦИФИКАЦИЙ 8](#_Toc151320738)

[2.1 Описание предметной области задачи 8](#_Toc151320739)

[2.1.1 Информационные объекты предметной области и взаимосвязи между ними 8](#_Toc151320740)

[2.1.2 Информационные и функциональные потребности пользователей разрабатываемой ПС (ПМ) 8](#_Toc151320741)

[2.1.3 Методы работы с информационными объектами предметной области 8](#_Toc151320742)

[2.1.5 Концептуальное обоснование разработки 9](#_Toc151320743)

[2.2 Классы и характеристики пользователей 9](#_Toc151320744)

[2.3 Функциональные требования 9](#_Toc151320745)

[2.3.1 Определение функциональных возможностей ПС (ПМ) 9](#_Toc151320746)

[2.3.2 Описание прецедентов 9](#_Toc151320747)

[2.4 Нефункциональные требования 10](#_Toc151320748)

[3 ВЫБОР ПРОГРАММНЫХ СРЕД И СРЕДСТВ РАЗРАБОТКИ 11](#_Toc151320749)

[3.1 Сравнительный анализ имеющихся возможностей по выбору средств разработки 11](#_Toc151320750)

[3.2 Характеристика выбранных программных сред и средств 12](#_Toc151320751)

[4 АЛГОРИТМ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ 13](#_Toc151320752)

[4.1 Этапы реализации ПС (ПМ) 13](#_Toc151320753)

[4.2 Пользовательский интерфейс ПС (ПМ) 14](#_Toc151320754)

[4.2.1 Взаимодействие пользователей с ПС (ПМ) 14](#_Toc151320755)

[4.2.2 Проектирование пользовательских сценариев 14](#_Toc151320756)

[4.2.3 Определение операций пользователей 14](#_Toc151320757)

[4.2.4 Составление функциональных блоков 15](#_Toc151320758)

[4.2.5 Проектирование структуры экранов ПС (ПМ) и схемы навигации 15](#_Toc151320759)

[4.2.6 Низкоуровневое проектирование 16](#_Toc151320760)

[4.3 Входные, выходные и промежуточные данные 16](#_Toc151320761)

[4.4 Разработка базы данных, реализуемой в рамках ПС (ПМ) 17](#_Toc151320762)

[4.6 Алгоритмы использования применяемых программных технологий 18](#_Toc151320763)

[4.7 Архитектура и схема функционирования ПС (ПМ) 18](#_Toc151320764)

[6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 20](#_Toc151320765)

[ЗАКЛЮЧЕНИЕ 21](#_Toc151320766)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 23](#_Toc151320767)

[ПРИЛОЖЕНИЯ 24](#_Toc151320768)

[Приложение А 24](#_Toc151320769)

[Приложение Б 24](#_Toc151320770)

[Приложение В 24](#_Toc151320771)

[Приложение Г 24](#_Toc151320772)

[Приложение Д 24](#_Toc151320773)

[Приложение Е 25](#_Toc151320774)

# 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] <https://asomobile.net/blog/otchet-rynka-mobilnyh-prilozhenij-za-2022-god/>. 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.

### 4.2.4 Составление функциональных блоков

{На этом этапе выделяются отдельные функциональные блоки. Под функциональным блоком подразумевают группу функций (операций), связанных по назначению или области применения или группу функций информационного наполнения.}

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст

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

{На этом этапе, основываясь на сценариях работы и ролях пользователей, формируется структура экранов системы, т.е. определяется количество экранов, функциональность каждого из них, навигационные связи между ними, формируется структура меню и других навигационных элементов.

Навигационная схема показывает механизм распределения функций и задач между экранами.

Навигационная схема определяет, каким образом пользователи смогут перемещаться между различными экранами.}

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст

### 4.2.6 Низкоуровневое проектирование

{ На данном этапе разрабатываются интерфейсы конкретных экранов системы (состав, взаимное расположение интерфейсных элементов).

Привести планы отдельных экранных форм, прототипы окон, страниц и т.п.}

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст

## 4.3 Входные, выходные и промежуточные данные

{В данном разделе описывается структура входных, выходных и про-межуточных данных. Например, входные данные поступают на вход программы в виде файла, значит, структура файла должна быть полностью описана. Аналогичное требование выполняется для промежуточных и выходных данных.

Например, для сайтов входной информацией могут быть текстовые файлы (указать в каком формате), графическая информация (указать формат) и т. п. Выходная информация – это HTML, PHP и т. п. страницы, видимые в окне браузеров (указать, каких).}

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст

## 4.4 Разработка базы данных, реализуемой в рамках ПС (ПМ)

{В этом разделе необходимо указать тип и описать структуру базы данных, если таковая имеется в проекте. Если приложение не использует базу данных, этот пункт опускается.

При описании структуры базы данных описываются все таблицы в виде:

- имя таблицы и её назначение, т.е. для хранения какой информации предназначена данная таблица;

- описание всех полей таблицы с указанием типа, назначения, первичных и внешних ключей.

Приводится ER-диаграмма, иллюстрирующая схему связей между таблицами.

Для объектной базы данных приводится описание структуры данных всех классов, аналогично тому, как это сделано для таблиц, добавляется только описание методов.

Если в выпускной квалификационной работе разработаны и созданы инструменты для работы с БД, их можно описать в данном разделе.

ЕСЛИ В ПРОЕКТЕ НЕ ИСПОЛЬЗУЕТСЯ БД, НО ПРЕДУСМОТРЕНО НЕКОЕ ДРУГОЕ ХРАНИЛИЩЕ ДАННЫХ (НАПРИМЕР СИСТЕМА КАТАЛОГОВ), СЛЕДУЕТ ЕЕ ОПИСАТЬ. НАЗВАНИЕ РАЗДЕЛА В ЭТОМ СЛУЧАЕ НАДО ИЗМЕНИТЬ.}

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст

## 4.6 Алгоритмы использования применяемых программных технологий

{При разработке сложных программных систем, использующих существующие программные технологии, основанные на мат.моделях и предназначенные для решения конкретных задач, следует привести алгоритмы применения этих технологий для решения задач ВКР.

*Например, разработка ПС с применением нейронных сетей включает следующий сценарий:*

*- выбор архитектуры нейронной сети,*

*- формирование датасета для обучения сети,*

*- обучение сети.*

*В данном случае, сама нейронная сеть не разрабатывается, применяется готовая программная технология, которая адаптируется под задачу. Алгоритм этой адаптации и следует описать.*}

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст

## 4.7 Архитектура и схема функционирования ПС (ПМ)

{В этом разделе следует показать организацию программной системы, структурных элементов и объединение этих элементов в более крупные подсистемы. Можно использовать статические UML диаграммы: диаграммы классов, диаграммы компонентов, диаграммы развертывания, диаграммы объектов.

Показать поведение системы во время выполнения основных функций, используя диаграммы активности, состояний UML.

В этом разделе должна быть описана структура ПС или ПТС, т. е.:

- из каких функциональных блоков (файлов, модулей, процедур, функций, классов) состоит ПС или ПТС;

- приведено описание каждого блока с его названием и назначением;

- приведена графическая схема взаимосвязи этих блоков.

Для описания схемы функционирования программного средства можно использовать диаграммы UML.

С использованием диаграммы компонентов UML (она иллюстрирует архитектуры компонентов программного обеспечения и зависимости между ними) показать, как компоненты соединяются вместе для формирования программной системы.

С помощью диаграммы развертывания UML показать, как программная система будет физически развернута на аппаратном обеспечении.}

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст

# ****6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ****

{В этот раздел могут быть включены следующие пункты:

- порядок установки и настройки программного средства на компьютер пользователя или первичная настройка технического средства;

- действия пользователя в случае сбоя ПС или ПТС;

- последовательность действий пользователя для решения своих основных задач при работе с программным или техническим средством.

Следует привести образы экранов (ScreenShots) с пояснениями для лучшего восприятия раздела. Приветствуется создание помощи по установке и эксплуатации ПС или ПТС.}

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст

# 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

{Список опубликованных печатных, либо интернет- источников информации, используемых при написании текста ПЗ.

Сведения об источниках можно располагать в порядке появления ссылок на источники в тексте документа.

Источники необходимо нумеровать арабскими цифрами без точки и печатать с абзацного отступа.

Ссылки на использованные источники следует приводить в квадратных скобках в тексте, в конце предложения.

Примеры библиографического описания источников}

1. Численное исследование процессов переноса и трансформации газовых и аэрозольных примесей в шлейфе выбросов Норильского промышленного района / В.Ф. Рапута, В. Симоненков, Б.Д. Белан, Т.В. Ярославцева / «Оптика атмосферы и океана», 31, № 6, 2018 – C. 438-439.
2. Вострикова Л.Г. Финансовое право [Электронный ресурс] : учебник / Л.Г. Вострикова. – Электрон. текстовые дан. – М.: Равновесие: Юстинформ, 2005. – 1 электрон. опт. диск (СD). – Загл. с контейнера.
3. Исследовано в России: многопредмет. науч. журн. / Моск. физ.-техн. ин-т. – Электрон. журн. – Долгопрудный : МФТИ, 1998.– URL : http://zhurnal.mipt.rssi.ru, свободный. –Яз. рус. – (Дата обращ. ……..)
4. Мозолин В. П. Модернизация права собственности в экономическом измере-нии [Электронный ресурс] / В. П. Мозолин. – Электрон. ст. – М., 2011. – URL : http://www.norma-verlag.com/journal/2011/1, свободный. – Яз. рус. – Аналог печат. изд. (Журнал российского права. – 2011. – № 1). – (Дата обращ. 23.06.2011).

# ПРИЛОЖЕНИЯ

## Приложение А

{Содержит **техническое задание на реализацию программной системы**, разработанное при непосредственном участии представителей заинтересованных организаций.}

## Приложение Б

{Содержит примеры входных данных в виде изображений, таблиц, фрагментов текстовых документов и.т.д.}

## Приложение В

{Содержит различные схемы, диаграммы, иллюстрирующие алгоритмы работы разработанной программной системы.}

## Приложение Г

{Содержит результаты тестовых испытаний на различных тестовых данных.

*Например, при наличии в составе ПС расчетно-графического модуля, полученные в результате работы, иллюстрации и таблицы, по которым можно судить о корректной работе программы*.}

## Приложение Д

{Содержит фрагменты листинга программного кода.

**Обязательно наличие поясняющего текста (для чего предназначен фрагмент кода, функция, класс)!**

Привести не менее 10 страниц кода. Текстовые пояснения оформляются в соответствии с ГОСТ.

Параметры оформления листинга кода: размер шрифта 10, допускается расположение в 2 столбца, одинарный междустрочный интервал..}

## Приложение Е

{При наличии, можно привести Акт о внедрении в эксплуатацию разработанной программной системы.}