

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

Направление подготовки бакалавриата
01.03.02 Прикладная математика и информатика

Отчет о практике по научно-исследовательской работе

ПОМОЩНИК-ТРЕНАЖЕР ПО ДИСЦИПЛИНАМ КУРСА ПМИИ

Выполнил:

студент 3 курса группы 22304

А. Р. Артамонов _____
подпись

Место прохождения практики:

Кафедра информатики и математического обеспечения

Период прохождения практики:

02.09.24-26.05.25

Руководители:

Ю. А. Богоявленский, к.т.н., доцент

подпись

Д.Б. Чистяков, ст. преподаватель

подпись

Итоговая оценка

оценка

Петрозаводск — 2025

Содержание

Введение	3
1. Аналоги приложения-тренажера и инструменты для разработки	5
1.1. Аналоги приложения-тренажера	5
1.2. Язык программирования Java	7
1.3. Android Studio и Git	8
1.4. База данных Google Firebase	9
2. Используемые алгоритмы решения заданий, сбора и аналитики информации	12
2.1. Алгоритм нахождения определенного интеграла	13
2.2. Алгоритм нахождения предела функции	14
2.3. Алгоритмы для работы с матрицами	15
2.4. Алгоритмы для работы с векторами и координатами точек	17
2.5. Алгоритмы сбора статистики	19
3. Реализация приложения	21
3.1. Дизайн приложения	21
3.2. Логика приложения	26
Заключение	29
Список использованных источников	31

Введение

Студенты направления «Прикладная математика и информатика» (ПМиИ) часто сталкиваются с трудностями в освоении сложного и многогранного учебного материала. Высокая интенсивность обучения, необходимость одновременно изучать несколько дисциплин и ограниченные возможности для индивидуальной работы с преподавателем приводят к пробелам в знаниях и снижению успеваемости. Это создает потребность в инструменте, который позволит организовать обучение более эффективно и индивидуализировано.

Для решения этой проблемы было решено разработать помощник-тренажер, предназначенный для поддержки студентов в изучении дисциплин ПМиИ и упрощения работы преподавателей. Приложение имеет высокую актуальность, так как студентам необходимо успевать разбираться в нескольких новых дисциплинах одновременно и восполнять пробелы в пройденном материале.

Приложение будет предоставлять студентам доступ к большому количеству математических заданий, обеспечивать автоматическую проверку решений и предлагать персонализированные рекомендации для устранения пробелов в знаниях. С учетом доступности мобильных устройств приложение станет удобным инструментом, которым можно пользоваться в любое время и в любой обстановке.

В отличие от существующих аналогов, приложение будет ориентировано на специфику направления ПМиИ, что обеспечит его практическую ценность и эффективность как для студентов, так и для преподавателей.

Цель работы — разработка приложения-помощника для студентов и преподавателей направления ПМиИ, которое обеспечит поддержку в изучении и преподавании математических дисциплин.

Для достижения цели были поставлены следующие задачи:

1. Изучить язык программирования Java для разработки мобильных приложений.
2. Проанализировать содержание дисциплин и тем, изучаемых в рамках направления ПМиИ.
3. Ознакомиться с возможностями Google Firebase и изучить его использование в ка-

честве облачного решения для хранения и синхронизации данных в мобильных приложениях.

4. Изучить алгоритмы для автоматической проверки и анализа математических заданий.

5. Разработать концепцию и структуру мобильного помощника-тренажёра, ориентированного на поддержку учебного процесса студентов и преподавателей.

6. Реализовать сбор статистики решения заданий и анализ данных с целью получения информации об успеваемости и востребованности различных заданий.

7. Реализовать прототип приложения, включающий базовую функциональность, взаимодействие с базой данных и автоматическую проверку решений.

Ожидается, что разработанное приложение станет полезным инструментом для улучшения качества учебного процесса, предоставляя студентам возможности для самостоятельной работы и повышения уровня знаний, а преподавателям — средства для управления и анализа учебного процесса.

1. Аналоги приложения-тренажера и инструменты для разработки

В самом начале, было решено рассмотреть аналогичные приложения, которые реализуют тот же функционал, узнать о их достоинствах и недостатках. Прежде всего, такие приложения направлены на массового пользователя и часто не содержат специфических тем для того или иного направления. Но также существуют узконаправленные приложения, использующиеся для изучения иностранных языков или решения математических задач без образовательного контента.

Далее было необходимо определиться, какие инструменты будут использоваться для достижения поставленных задач. Поскольку мобильное устройство стало платформой, на которой будет создано приложение-тренажер, нужно было подобрать соответствующие инструменты для разработки. Выбор пал на язык программирования Java, среду разработки Android Studio и систему контроля версиями Git.

Для хранения пользовательских данных, заданий и статистики решённых примеров было решено использовать облачную платформу Google Firebase. Данный инструмент предоставляет ряд сервисов, необходимых для мобильной разработки, включая облачную базу данных NoSQL (Firebase Realtime Database) и аутентификацию пользователей. Благодаря удобному интерфейсу и хорошей интеграции с Android, Firebase стал оптимальным выбором для реализации серверной части приложения без необходимости настройки отдельного backend-сервера.

1.1. Аналоги приложения-тренажера

Рассмотренные аналогичные приложения популярны по всему миру, они обладают необходимым функционалом для изучения как математических дисциплин, так и программирования. Однако имеются некоторые недостатки, из-за чего они подойдут не каждому студенту. Далее будут описаны некоторые из них.

Coursera - платформа для онлайн-обучения, имеющая как платные так и бесплатные курсы. Она доступна в веб-версии, а также в мобильном приложении для Android и IOS. Coursera предлагает огромное количество курсов по самым различным темам: от личност-

ного здоровья до технических наук. Большинство из курсов выложены на английском языке, но также есть курсы на китайском, испанском, русском и других языках. Но на русский язык переведено только 37 курсов, что может привести к некоторым трудностям в изучении. Курсы и задания создаются профессионалами из самых известных компаний. Но главное достоинство и одновременно недостаток этой платформы - ее обширность. Она не затрагивает специфические темы для направления, а доступные курсы могут распространяться на платной основе, поэтому у студентов не всегда есть финансовая возможность, чтобы приобрести необходимый курс.

Khan Academy - некоммерческая образовательная организация, которая предоставляет различные курсы для изучения множества дисциплин. Платформа имеет веб-версию и также мобильное приложение. Она содержит большое количество видео с обучением тем или иным темам, поддерживает возможность решения практических заданий, изучения программирования и прохождения стандартных тестов. Главное достоинство - доступность и наглядное объяснение материала в формате видео, недостаток - ограниченность в темах высшей математики, поскольку платформа разработана по большей части для школ.

Wolfram Alpha - мощнейший инструмент для решения математических задач. Он подробно описывает решение данной ему задачи, имеет возможность решать интегралы, пределы, матрицы и многое другое. Но его главный недостаток - отсутствие образовательного контента. Инструмент способен решать сложные задачи и отображать ход решения, но не дает пользователю никакой справки по использованным методам и алгоритмам, что категорически не подходит для обучения.

Brilliant - коммерческая компания и связанное с ней сообщество. Включает в себя более 100 различных курсов. Большинство из них платные, только малая часть курсов распространяется бесплатно. В веб-версии выкладываются математические и естественнонаучные головоломки и предоставляются для решения пользователю. Компания делает акцент на решении головоломок и задач с упором на теорию, но без особой поддержки тем по направлениям. Главный недостаток - большое количество платного контента и отсутствие полноценных курсов по программированию и математическим дисциплинам.

Каждая рассмотренная платформа имеет свои достоинства, благодаря которым она стала популярна. Многие из них поддерживаются известными компаниями и личностями,

что является несомненно плюсом. Но выяснилось, что ни одно приложение не обладает полным набором желаемых функций. Поэтому приложение-тренажер будет актуально для студентов, так как не требует постоянного подключения к сети и подходит для изучения дисциплин их направления.

1.2. Язык программирования Java

Java - очень популярный язык программирования, зарекомендовавший себя как один из самых лучших. На нем написаны многие программы, а его универсальность позволяет использовать его для создания программного обеспечения как для персональных компьютеров, так и телефонов.

Выбор обусловлен универсальностью и доступностью данного языка. Найти документацию и различные материалы для изучения Java очень просто, так как это популярный язык программирования.

Язык Java был отмечен лозунгом "написать один раз, запускать где угодно". Это означает, что код написанный, к примеру, на операционной системе Windows может быть легко перенесен на любую другую платформу без существенного переписывания, в нашем случае Android. Это возможно благодаря особенности компиляции Java-программы.[1] Компилятор создает файл байт кода .class, который может работать в любой операционной системе, где установлена виртуальная машина Java, которая служит дополнительным уровнем абстракции между платформой Java и базовым аппаратным обеспечением машины. Данная технология объединяет в себе методы компиляции и интерпретации, ранее языки программирования использовали один из методов. Многоплатформенность языка Java является одной из его главных особенностей.[2]

Следующее не менее важное преимущество данного языка заключается в том, что Java - объектно-ориентированный язык программирования. Это означает, что взаимодействие в нем происходит через объекты. От программиста требуется создать отдельные блоки, которые будут взаимодействовать друг с другом. Это существенно упрощает расширение и масштабирование каждого блока.[3]

Язык Java относительно простой в изучении и чрезвычайно популярен среди компа-

ний, занимающихся программным обеспечением и технологиями.

Для выполнения курсовой работы на Java нам потребуется JDK (Java Development Kit) - специальный набор инструментов. Существуют разные реализации JDK, но все они используют один и тот же язык - Java. Наиболее популярные реализации - Oracle JDK и OpenJDK.

Oracle JDK разрабатывается компанией Oracle, а OpenJDK - открытый проект, развиваемый сообществом Java-разработчиков, включая Oracle, Red Hat и другие компании.

Основные различия между Oracle JDK и OpenJDK связаны с лицензированием и поддержкой. Oracle JDK можно использовать бесплатно для личных нужд, разработки, тестирования и демонстрации. Для получения коммерческой поддержки требуется платная лицензия. А OpenJDK полностью бесплатна.

В функциональном плане Oracle JDK и OpenJDK практически идентичны. Однако есть отзывы, что Oracle JDK может работать немного быстрее, а OpenJDK может быть немного менее стабильной. мы будем использовать Oracle JDK, но использование OpenJDK также допустимо и не вызовет никаких проблем.

Кроме Oracle и OpenJDK, существуют и другие реализации JDK от различных компаний. Например, компания Microsoft предоставляет свою собственную реализацию JDK, которая может устанавливаться по умолчанию на некоторые версии Windows.[4]

1.3. Android Studio и Git

Для эффективной и удобной работы над приложением, также необходимо воспользоваться некоторыми инструментами, такими как Android Studio и Git.

Android Studio - интегрированная среда разработки, созданная Google для разработки Android-приложений. Android Studio предоставляет возможность разработки приложений на языке программирования Java официально рекомендованном Google для создания Android-приложений.

Одним из главных преимуществ Android Studio является его способность автоматически находить и исправлять ошибки в коде, а также генерировать код. В Android Studio также предусмотрен встроенный ИИ ассистент - Studio Bot, с которым можно общаться на естественном языке.

Для тестирования приложений, Android Studio предлагает эмулятор операционной системы Android, который позволяет запускать готовые пресеты виртуального устройства или создавать собственные конфигурации с определенной версией операционной системы. Это дает возможность протестировать приложение на широком спектре различных устройств.

Наконец, Android Studio предоставляет возможность привести разработанное приложение к необходимому формату для публикации в различных магазинах приложений для Android, таких как Google Play, AppGallery, RuStore и других.[5]

Использование данной интегрированной среды разработки значительно упростит и ускорит процесс разработки приложения.

Git - это система контроля версий, которая позволяет отслеживать изменения в файлах и работать с ними совместно с другими разработчиками. С ее помощью можно создавать ветки разработки, коммитить изменения, сливать изменения из разных веток, откатывать изменения к предыдущим версиям и многое другое. Git позволяет эффективно работать над проектом, делать изменения без опасения потерять данные и контролировать историю изменений. Git является одним из наиболее популярных инструментов для контроля версий в сообществе разработчиков. Использование системы контроля версий поможет достаточно удобно переключаться между версиями разрабатываемого приложения и в случае непредвиденных проблем сохранять рабочую версию продукта.

1.4. База данных Google Firebase

В процессе разработки мобильного приложения важную роль играет выбор подходящей платформы для хранения и управления данными. В рамках настоящей работы в качестве основной облачной платформы была выбрана Google Firebase, представляющая собой комплексное решение для быстрого создания, запуска и масштабирования мобиль-

ных и веб-приложений.

Firebase представляет собой набор облачных сервисов, предоставляемых компанией Google, который включает в себя всё необходимое для современной разработки: от баз данных и аутентификации пользователей до хостинга и аналитики. Основное преимущество данной платформы заключается в высокой степени интеграции между сервисами и простоте их использования, что особенно важно при создании прототипов и MVP-приложений в ограниченные сроки.

Firebase предоставляет разработчику следующие ключевые компоненты:

- **Firebase Realtime Database** — облачная NoSQL база данных, обеспечивающая синхронизацию данных между клиентами в реальном времени. Это позволяет пользователям приложения моментально видеть изменения, внесённые другими участниками, без необходимости ручного обновления.
- **Firebase Authentication** — готовое решение для авторизации пользователей с поддержкой множества способов входа, включая электронную почту и пароли, а также OAuth-провайдеров, таких как Google или GitHub.
- **Firebase Cloud Storage** — надёжное облачное хранилище для файлов, которое позволяет загружать и извлекать изображения, документы и другие пользовательские ресурсы.
- **Firebase Hosting** — хостинг-решение для размещения веб-приложений и статических ресурсов с возможностью автоматического деплоя из интерфейса среды разработки.
- **Firebase Analytics и мониторинг** — средства анализа поведения пользователей и оценки стабильности работы приложения, включая отчёты о сбоях (Crashlytics), метриках производительности и событиях взаимодействия.

Особое внимание заслуживает недавно представленная разработка компании Google — Firebase Studio, анонсированная в апреле 2025 года. Это облачная среда, предоставляющая интегрированный цикл разработки с элементами искусственного интеллекта. Платформа была создана для того, чтобы ускорить разработку приложений и упростить интеграцию AI-функционала. Firebase Studio объединяет возможности облачной IDE, фреймворка

Genkit для генеративного ИИ и инструментов визуального прототипирования. Приложения можно создавать с помощью текстового описания, загруженных изображений или макетов, что особенно удобно на этапе проектирования интерфейса.

Кроме того, Firebase Studio предоставляет встроенную поддержку генерации фронтенда и бэкенда, автоматическую настройку зависимостей и возможность деплоя в один клик на сервисы Google Cloud. Благодаря интеграции с моделью Gemini, платформа позволяет описывать поведение приложения на естественном языке и получать соответствующую реализацию кода в интерфейсе, построенном на базе Visual Studio Code. Такие функции делают Firebase Studio особенно полезной при быстром прототипировании и тестировании гипотез, снижая порог входа для начинающих разработчиков.

Одним из ключевых факторов выбора Firebase в рамках данного проекта стала его тесная интеграция с Android-средой разработки и высокая надёжность. Firebase не требует настройки собственного сервера, обеспечивает масштабируемость и безопасное хранение данных, а также предлагает гибкую систему управления доступом. Все это делает платформу удобным и мощным инструментом для реализации учебных и исследовательских проектов.

Таким образом, использование Google Firebase позволяет значительно ускорить разработку, уменьшить количество рутинной работы и сосредоточиться на основной функциональности приложения, что особенно важно в условиях ограниченного времени и ресурсов, характерных для студенческих проектов.

2. Используемые алгоритмы решения заданий, сбора и аналитики информации

Для создания тренажёра было необходимо изучить возможные алгоритмы решения математических примеров, но перед этим нужно было определиться с дисциплинами, которые будут рассматриваться в приложении. Поскольку приложение разрабатывалось для студентов ПМиИ, выбор пал на три основные математические дисциплины и программирование. Линейная алгебра, математический анализ и аналитическая геометрия — три основные дисциплины, которые необходимо знать каждому студенту. Они являются базовыми, и знания, полученные в ходе их изучения, помогут быстрее осваивать следующий более сложный материал. Эти дисциплины тесно связаны с остальными разделами, например, с функциональным анализом.

Далее было необходимо проанализировать темы, которые рассматриваются в соответствующих дисциплинах. Было решено реализовать несколько алгоритмов для каждой из них. Для математического анализа были выбраны алгоритмы нахождения определённых интегралов и пределов, для линейной алгебры — вычисление определителя, ранга матрицы, транспонирование и выполнение элементарных операций с матрицами. Для аналитической геометрии реализованы задачи на нахождение скалярного произведения векторов, расстояния между двумя точками, а также определение угла между прямыми. В рамках дисциплины программирования был добавлен редактор кода, позволяющий вносить изменения в файлы с кодом.

Каждое задание сопровождается автоматической проверкой введённого ответа, что требует точной реализации алгоритмов и учёта допустимой погрешности при работе с вещественными числами. Все примеры были классифицированы по уровню сложности, а при разработке интерфейса было предусмотрено отображение подсказок, которые можно вызвать по запросу пользователя.

Особое внимание было уделено системе сбора и анализа данных о действиях пользователей. В процессе решения примеров система автоматически сохраняет информацию о количестве попыток, правильности ответов. Эти данные отправляются в базу данных Firebase и в дальнейшем используются для анализа успешности усвоения тем, выявления наиболее сложных задач, а также для адаптации уровня сложности под каждого пользователя. Такая система позволяет преподавателю отслеживать прогресс студентов,

а студенту — видеть собственные достижения и зоны для улучшения. Анализ накопленной статистики также поможет в будущем расширить и улучшить набор задач, ориентируясь на реальные трудности пользователей.

В следующих разделах будет подробно рассмотрена реализация каждого из выбранных алгоритмов, структура представления заданий и механизм автоматической проверки.

2.1. Алгоритм нахождения определенного интеграла

В качестве основного алгоритма был выбран метод прямоугольников. Метод прямоугольников — это численный метод, используемый для приближенного вычисления определенного интеграла. Принцип его работы заключается в разбиении области под кривой на несколько прямоугольников и суммировании их площадей. В этом методе функция оценивается в середине каждого интервала разбиения.

Алгоритм работает в четыре этапа:

1. Определение размера интервала:

Размер каждого интервала вычисляется путем деления разности между верхним и нижним пределами интегрирования на количество интервалов.

2. Инициализация суммы:

Некоторая переменная используется для накопления общей площади всех прямоугольников.

3. Итерация по интервалам:

3.1. Для каждого интервала вычисляется точка в середине интервала.

3.2. Определяется значение функции в этой точке.

3.3. Площадь прямоугольника, основание которого равно размеру интервала, а высота равна значению функции в середине интервала, добавляется к сумме.

4. Возврат результата:

После завершения цикла возвращается накопленная сумма, которая является приближенным значением определенного интеграла.

Метод прямоугольников, позволяет приближенно вычислять определенные интегралы с использованием численных методов. Этот метод прост в реализации и применим для различных функций. Такой подход к численному интегрированию может быть полезен в различных приложениях, требующих оценки интегралов, особенно когда аналитическое решение затруднительно или невозможно.

Алгоритм вызывается при каждом открытии соответствующего ему примера, но процесс вычисления не происходит, если в базе данных пример помечен как решенный, в противном случае алгоритм выполняется и сохраняет ответ в базу данных. Важной особенностью алгоритма также является автоматическое разбиение строки с примером в формате записи LaTeX и получение необходимых для решения выражений.

Важно сохранять возможность находить решение автоматически, что в будущем позволит легко расширять список задач.

2.2. Алгоритм нахождения предела функции

Вычисление предела функции является одной из фундаментальных задач математического анализа. Предел позволяет формализовать понятие стремления переменной к определённому значению и используется при изучении непрерывности, производных и интегралов. В рамках приложения был реализован численный метод приближения предела, основанный на последовательном уменьшении шага приближения к заданной точке a и оценке значений функции вблизи этой точки.

Алгоритм работы метода:

- **Установка начальных параметров:**

- `epsilon` — точность вычисления, определяющая допустимую разницу между последовательными значениями функции;
- `delta` — начальный шаг приближения к точке a .

- **Инициализация начальных значений:**

- `previousValue` — значение функции в точке $a - \delta$;

- `limitValue` — начальное предполагаемое значение предела, изначально равное `previousValue`.
- **Цикл уменьшения шага приближения:**
 - На каждом шаге вычисляется текущее значение функции в точке $a - \delta$;
 - Если разница между текущим и предыдущим значением функции меньше заданного `epsilon`, предполагаемое значение предела обновляется, и цикл прерывается;
 - В противном случае текущее значение функции становится предыдущим для следующей итерации, а `delta` уменьшается вдвое.
- **Округление результата:**
 - После завершения цикла вычисленное значение предела округляется до одного знака после запятой для представления в интерфейсе.

Данный численный метод позволяет приближённо оценить значение предела функции при стремлении переменной к заданной точке. Он особенно полезен в тех случаях, когда аналитическое вычисление предела затруднено или невозможно. Алгоритм выполняется аналогично алгоритму нахождения определённого интеграла и запускается по мере необходимости. Примеры для проверки формируются автоматически, включая извлечение данных из LaTeX-представления формул.

2.3. Алгоритмы для работы с матрицами

Работа с матрицами является неотъемлемой частью курса линейной алгебры и широко применяется в прикладных задачах, включая решение систем линейных уравнений, анализ линейных преобразований и статистическую обработку данных. В рамках приложения были реализованы ключевые алгоритмы для работы с матрицами: вычисление определителя, ранга, следа, транспонирование, а также операции сложения и умножения матриц.

Определитель матрицы — это числовая характеристика квадратной матрицы, которая используется во многих областях математики и её приложениях. В приложении реализован универсальный алгоритм вычисления определителя, основанный на рекурсивном разложении по строкам (метод миноров).

Описание алгоритма:

- **Базовые случаи:** для матрицы размером 1×1 определитель равен единственному элементу; для матрицы 2×2 определитель вычисляется как разность произведений диагональных элементов.
- **Рекурсивное разложение:** для матриц большего размера используется разложение по первой строке. Для каждого элемента этой строки формируется минор — подматрица, полученная удалением текущей строки и столбца. Определитель минора вычисляется рекурсивно, а итоговая сумма складывается с учётом знака по формуле $(-1)^{i+j}$, где i, j — индексы элемента.
- **Результат округляется** до одного знака после запятой для отображения в интерфейсе.

Ранг матрицы — это максимальное число линейно независимых строк (или столбцов). В приложении используется метод приведения матрицы к ступенчатому виду с помощью элементарных строчковых преобразований, аналогичный алгоритму Гаусса.

Описание алгоритма:

- Определяются размеры матрицы и инициализируется счётчик ранга.
- Для каждой строки проверяется наличие ведущего (ненулевого) элемента. Если он отсутствует, происходит попытка замены строки.
- При наличии ведущего элемента выполняются элементарные преобразования: зачистка всех элементов ниже ведущего путём вычитания из них текущей строки, умноженной на соответствующий коэффициент.
- После приведения к ступенчатому виду возвращается количество ненулевых строк — это и есть ранг матрицы.

Алгоритмы для вычисления определителя и ранга позволяют выполнять ключевые операции, необходимые как в теоретической математике, так и в прикладных задачах, включая анализ свойств матриц, решение СЛАУ и численные расчёты. Метод вычисления определителя обеспечивает универсальность за счёт поддержки матриц произвольного размера, а метод вычисления ранга даёт точную оценку линейной независимости строк с

использованием устойчивого и надёжного подхода.

След матрицы реализуется как сумма элементов главной диагонали. Эта операция особенно полезна при работе с квадратными матрицами и отображает сумму собственных значений в случае диагонализуемых операторов.

Операции транспонирования, сложения и умножения матриц также реализованы в виде отдельных методов. Транспонирование осуществляется заменой строк на столбцы. При сложении происходит поэлементное суммирование двух матриц одинаковой размерности с последующим округлением результата. Умножение реализовано согласно стандартному правилу произведения матриц: каждый элемент результата получается как сумма произведений соответствующих элементов строки первой матрицы и столбца второй.

Все алгоритмы сопровождаются вспомогательными функциями для округления и форматирования вывода. Результаты представляются в удобном для отображения строковом виде, что облегчает использование алгоритмов в интерфейсе приложения.

Таким образом, модуль работы с матрицами обеспечивает выполнение основных операций, необходимых для закрепления знаний студентов в области линейной алгебры, и может быть легко расширен для поддержки более сложных вычислений в будущем.

2.4. Алгоритмы для работы с векторами и координатами точек

В рамках дисциплины «Аналитическая геометрия» студенты изучают пространственные и плоские фигуры, операции над векторами и работу с координатами точек. Эти темы составляют основу для последующего изучения линейной алгебры, математического анализа и физики. Для автоматизации решения соответствующих задач в приложении были реализованы базовые геометрические алгоритмы, охватывающие основные операции над точками и векторами.

Разработанный модуль включает следующие функции:

- вычисление скалярного произведения двух векторов;
- нахождение расстояния между двумя точками;

- определение длины вектора;
- получение координат вектора по началу и концу;
- вычисление длины вектора по координатам двух точек;
- нахождение угла между двумя векторами в градусах.

Скалярное произведение двух векторов вычисляется как сумма произведений их соответствующих компонент. Этот результат используется, в частности, при нахождении угла между векторами и в проверке взаимной ортогональности. Перед выполнением операции осуществляется проверка на совпадение размерностей векторов. Результат округляется до одного знака после запятой.

Расстояние между двумя точками определяется по формуле Евклидовой метрики: сначала вычисляется сумма квадратов разностей координат, затем извлекается квадратный корень из результата. Алгоритм универсален и применим как для двумерного, так и для трёхмерного пространства.

Длина вектора рассчитывается аналогичным образом — как корень квадратный из суммы квадратов его компонент. Такая реализация позволяет использовать алгоритм при любом числе координат.

Координаты вектора, заданного по начальной и конечной точке, определяются как разности соответствующих координат этих точек. Функция возвращает новый вектор, который может быть передан в другие методы, например, для определения длины или угла.

Длина вектора по точкам реализована как последовательное применение функций: сначала вычисляются координаты вектора, затем вызывается метод определения длины. Это повышает модульность кода и снижает дублирование.

Угол между двумя векторами вычисляется по формуле:

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|}$$

где $\vec{a} \cdot \vec{b}$ — скалярное произведение векторов, а $|\vec{a}|$ и $|\vec{b}|$ — их длины. Затем значение $\cos \theta$ ограничивается в пределах от -1 до 1 для защиты от возможных ошибок округления, и

находится угол с помощью арккосинуса. Результат преобразуется в градусы и округляется до десятых долей.

Все функции сопровождаются защитой от некорректного ввода (например, попытки вычисления угла между нулевыми векторами или векторами разной размерности), а результаты форматируются для корректного сохранения в базу данных.

Таким образом, разработанные алгоритмы позволяют выполнять базовые операции с точками и векторами, необходимые при изучении аналитической геометрии. Универсальность и надёжность этих методов делают их пригодными для автоматической проверки решений и анализа ошибок студентов при выполнении соответствующих заданий.

2.5. Алгоритмы сбора статистики

Для эффективного использования приложения как учебного тренажёра важным компонентом является система сбора и анализа статистических данных. Она позволяет отслеживать прогресс студентов, выявлять наиболее проблемные темы, а также повышать мотивацию к обучению за счёт визуализации успехов. Система статистики реализована с учётом двух ролей пользователей: **преподавателя** и **студента**, каждая из которых имеет доступ к различному набору функций.

Преподаватель имеет доступ к следующей информации:

- **Общая статистика по всем студентам** — данная категория включает в себя отслеживание общего количества студентов, курсы их обучения, наиболее сложные примеры, представленные в списке.
- **Статистика по конкретным дисциплинам** (например, алгебра, геометрия, математический анализ) — позволяет оценить, какие темы вызывают наибольшие затруднения, посмотреть общее количество примеров, сводку по всем примерам, тепловую карту попыток решения заданий.
- **Статистика отдельного студента** — преподаватель может выбрать любого обучающегося и получить полную информацию о его активности: количество решённых заданий, темы с наибольшим количеством ошибок, количество попыток до верного ответа.

Студент имеет доступ к персональной статистике:

- **Успеваемость по дисциплинам** — отображается количество пройденных заданий, число попыток, статус задания.
- **История выполненных заданий** — позволяет вернуться к ранее решённым примерам и при необходимости повторить их.
- **Мотивационные индикаторы** — отображение общего прогресса.

При каждом взаимодействии пользователя с заданием (начало решения, отправка ответа) в базу данных автоматически записывается соответствующая информация. Структура базы включает отдельные ветки данных для каждого пользователя, сгруппированные по дисциплинам и примерам. Все записи включают идентификаторы заданий и флаги правильности решения.

Для сбора агрегированной статистики реализованы специальные алгоритмы, которые проходят по всем пользователям, фильтруя и обрабатывая только тех, кто относится к отслеживаемому направлению (ПМиИ). Это позволяет преподавателю видеть только релевантные данные и формировать отчёты на основе реальной активности студентов своей группы.

Система сбора статистики является важной составляющей приложения, объединяющей процессы обучения, контроля и аналитики. Она позволяет обеспечить индивидуальный подход к каждому студенту и предоставляет преподавателю инструмент для мониторинга учебного процесса на всех уровнях — от дисциплины до конкретного задания.

3. Реализация приложения

После реализации алгоритмов для решения математических примеров была начата работа над созданием полноценного прототипа мобильного приложения. Для реализации клиентской части было выбрано средство разработки Android Studio и язык программирования Java, обладающий широкой поддержкой для мобильной разработки. Разработка велась с использованием встроенного эмулятора Android, что позволяло оперативно тестировать интерфейс, поведение приложения и корректность работы алгоритмов.

Изначально была заложена прочная основа для построения расширяемого и поддерживаемого мобильного приложения, способного развиваться как инструмент для изучения и автоматизированной практики по основным дисциплинам прикладной математики и информатики. Особое внимание уделялось не только текущей функциональности, но и будущему расширению приложения. Предполагалось, что в дальнейшем в проект будут добавляться новые дисциплины, примеры и обучающие элементы. Поэтому архитектура приложения проектировалась с ориентацией на гибкость, масштабируемость и повторное использование кода.

3.1. Дизайн приложения

Был разработан простой и минималистичный дизайн мобильного приложения на платформе Android. Использовались как стандартные макеты, заготовленные в Android Studio, так и собственноручно реализованные элементы интерфейса. Особое внимание уделялось повторному использованию компонентов: были созданы универсальные макеты, которые значительно сократили объём конечного кода приложения и устранили необходимость разрабатывать отдельные шаблоны для каждой вкладки.

Одними из первых были реализованы экраны регистрации и авторизации. При запуске приложения пользователь может либо войти в систему через уже зарегистрированную учётную запись, либо создать новую, заполнив поля для электронной почты, пароля и типа аккаунта (студент или преподаватель). Данные пользователя сохраняются в облачной базе Firebase, а в дальнейшем загружаются в приложение для персонализации интерфейса и доступа к соответствующему функционалу.

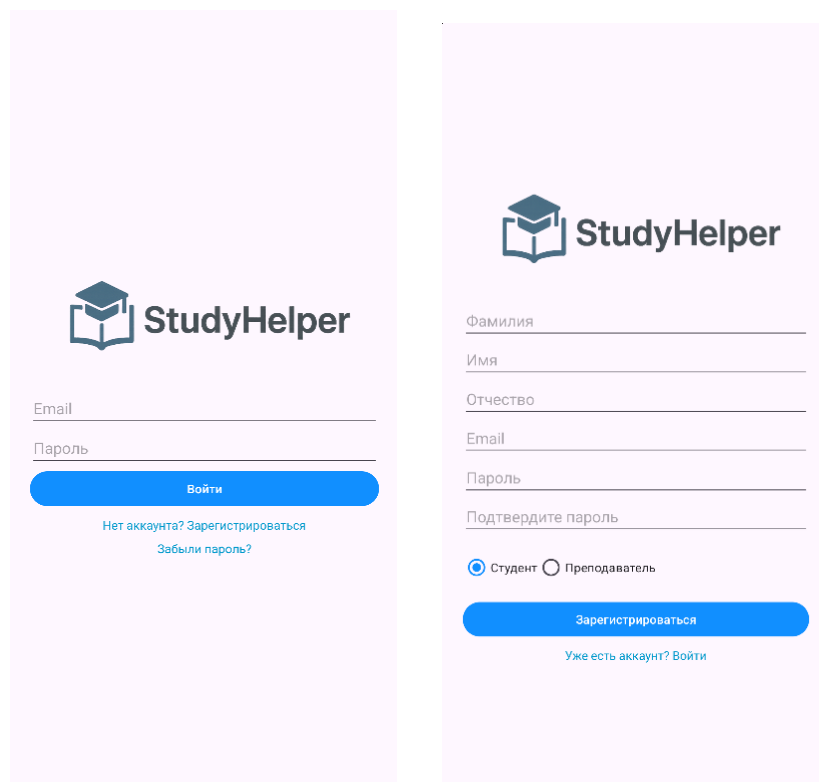


Рис. 1: Экраны входа и регистрации пользователя

После входа пользователь попадает в главное меню приложения. На главном экране отображаются кнопки для перехода к четырём основным дисциплинам: математический анализ, аналитическая геометрия, алгебра и программирование. Соответствующие элементы расположены в виде прокручиваемого списка, что обеспечивает масштабируемость и позволяет без изменений в дизайне добавлять новые дисциплины.

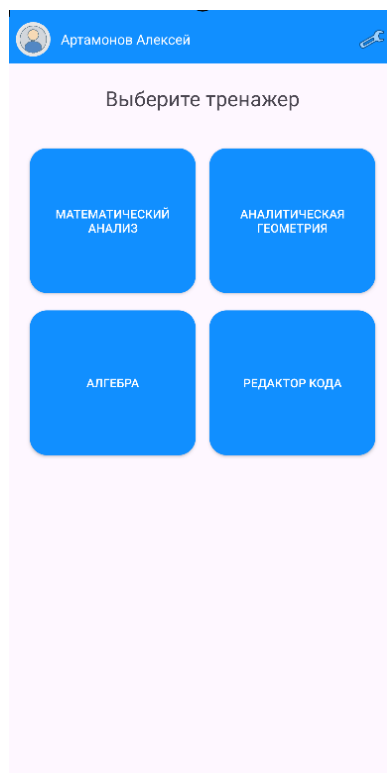


Рис. 2: Главное меню

Для отображения заданий по дисциплинам был реализован универсальный макет. Экран каждой дисциплины разделён на две основные части: «Заготовленные предметы» и «Пользовательские примеры». В каждой категории автоматически создаются кнопки перехода на примеры, загружая их из базы данных. Кнопки примеров подсвечиваются разным цветом в зависимости от статуса решения. Использование вертикальной прокрутки позволяет избежать перегрузки экрана при добавлении большого количества заданий.

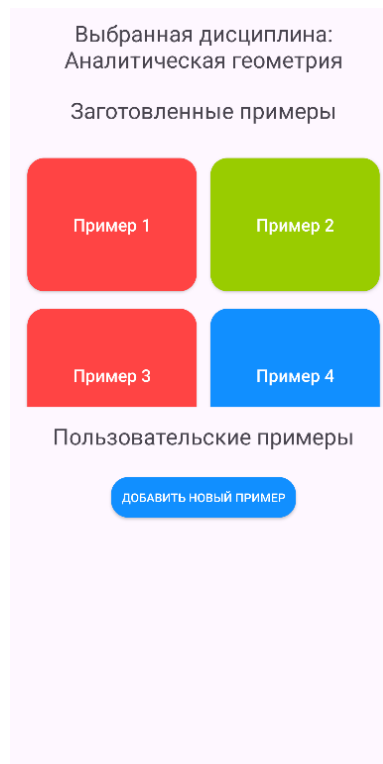


Рис. 3: Окно выбора примеров по дисциплине

Для задач по программированию предусмотрено отдельное окно — редактор кода. Этот экран реализован на отдельном макете и включает в себя текстовое поле с возможностью открытия и сохранения файлов. В будущем планируется добавить в этот раздел автоматическую проверку решений и поддержку основных языков программирования.

Также был создан универсальный макет отображения конкретного примера. Каждый пример включает текстовое описание задания, поле для ввода ответа, кнопки «Проверить», «Подсказка» и «Назад». При нажатии на кнопку «Подсказка» открывается окно с пояснением к решению. Вся логика проверки встроена в макет и обеспечивает мгновенную обратную связь по результату введённого ответа.

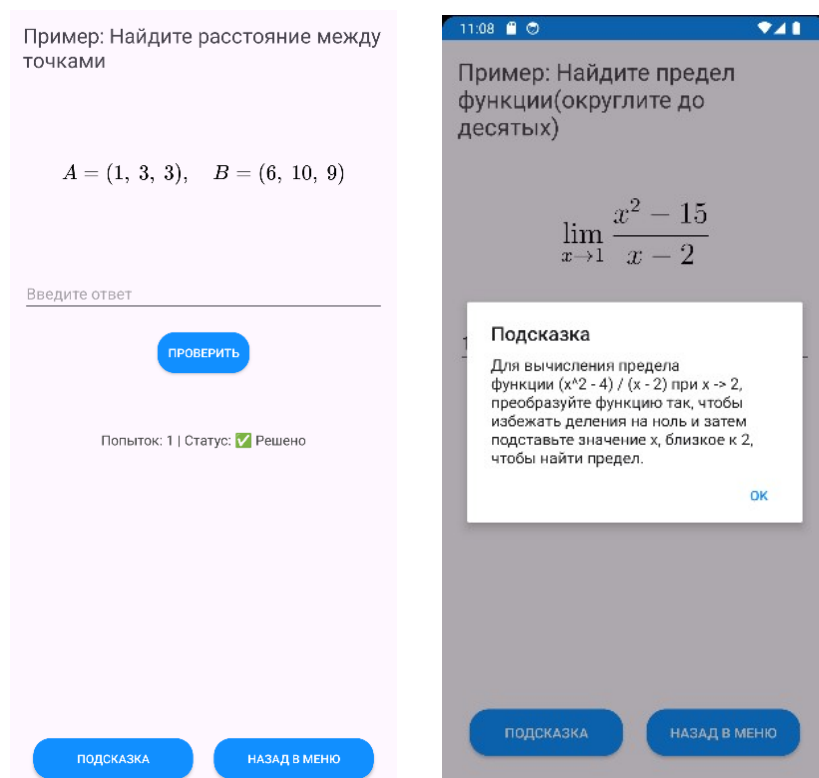


Рис. 4: Экраны входа и регистрации пользователя

Особое внимание уделено экрану профиля, который реализован в двух вариантах — для студента и преподавателя. В профиле студента отображаются персональные данные, прогресс по каждой дисциплине, количество решённых примеров и среднее число попыток. Профиль преподавателя предоставляет доступ к общей статистике по студентам направления, к успеваемости по каждой дисциплине, а также к аналитике по отдельным студентам. Информация обновляется в реальном времени и извлекается из базы данных Firebase.

Сменить аватар

Артамонов

Алексей

Романович

ПМИ

3

22304

ФИО преподавателя

Математика

Алгебра

Геометрия

Редактировать

Выйти

Удалить аккаунт

Фамилия

Имя

Преподавателя

Мат. анализ

242

Отслеживаемые направления:

ПМИ

Общая статистика

Общая информация по студентам

Студентов всего: 3

1 курс: 0

2 курс: 0

3 курс: 3

4 курс: 0

Примеры (по убыванию попыток)

Пример 3 из категории geometry

Всего попыток: 23, Студентов решило: 1

Редактировать

Выйти

Удалить аккаунт

Рис. 5: Экраны входа и регистрации пользователя

Дизайн приложения обеспечивает удобную навигацию, поддержку расширяемости и соответствие различным пользовательским ролям, что делает интерфейс как интуитивно понятным для студентов, так и функционально насыщенным для преподавателей.

3.2. Логика приложения

При написании логики приложения необходимо было сохранить возможность простого модифицирования программы и при этом сделать код более компактным и удобочитаемым. Поскольку язык Java является объектно-ориентированным, создание универсальных методов и классов для реализации логики приложения упростилось.

Чтобы код можно было в дальнейшем легко модифицировать и дополнять, были выделены несколько ключевых абстрактных классов, от которых наследуются остальные компоненты. Классы `BaseExampleActivity` и `BaseExampleListActivity` являются универсальными шаблонами, содержащими абстрактные методы, переопределяемые в специализированных активностях. Это позволило реализовать единый механизм отображения заданий и списков примеров для всех дисциплин.

На раннем этапе разработки было реализовано ограниченное количество заданий, в том числе для математического анализа — в классах `MathExampleActivity` и `MathExampleListActivity`. Эти классы используют алгоритмы из `IntegralOperations` и `LimitsOperations`, позволяющие вычислять определённые интегралы и пределы. Проверка ответов осуществляется локально: результат вычисляется с использованием математического метода и сравнивается с пользовательским вводом с учётом округления. В случае совпадения выводится сообщение о правильном решении, в противном случае — о наличии ошибки.

Аналогичная логика применена и к другим дисциплинам. Классы `AlgebraExampleActivity`, `GeometryExampleActivity` и соответствующие списковые активности реализуют примеры, основанные на вычислениях из `MatrixOperations` и `VectorOperations`. Таким образом, каждая дисциплина подключает свой набор алгоритмов, но использует общую архитектуру приложения.

Дополнительно был реализован экран `CodeEditorActivity`, содержащий простой текстовый редактор с возможностью загрузки и сохранения кода. Этот компонент предназначен для задач по программированию и может быть расширен в будущем для поддержки автопроверки решений.

Одним из важнейших компонентов логики приложения стала реализация системы регистрации и авторизации пользователей. При первом запуске пользователь может либо зарегистрироваться, выбрав свою роль (студент или преподаватель), либо войти в существующую учётную запись. Для этого созданы специальные модули `RegisterActivity`, `LoginActivity` и `ResetPasswordActivity`, взаимодействующие с сервисом `Firebase Authentication` и предоставляющие возможность авторизации, регистрации и восстановления пароля. Данные о пользователе (тип аккаунта, имя, учебная информация) сохраняются в облачной базе данных и загружаются при последующих входах в систему.

Роль пользователя напрямую влияет на доступный функционал приложения. В интерфейсе студента доступны задания и личная статистика по дисциплинам. Преподаватель получает доступ к расширенной аналитике, включая общую статистику по направлению, информацию по дисциплинам и возможность просмотра успеваемости каждого студента.

Для поддержки этих функций реализованы алгоритмы сбора и анализа метрик, которые автоматически фиксируют действия пользователя при взаимодействии с заданиями: количество попыток, успешность ответа и др. Эти данные структурируются в базе данных Firebase и используются для отображения в профиле, формирования индивидуальной и групповой статистики, а также для оценки прогресса студентов и определения наиболее сложных тем.

Приложение реализует не только интерфейс взаимодействия с учебными заданиями, но и полноценную логику учёта прогресса, дифференцированного функционала по ролям и модульную архитектуру, ориентированную на развитие и масштабирование.

Заключение

В ходе выполнения данной курсовой работы были рассмотрены и реализованы методы для работы с матрицами, интегралами, пределами и векторами на языке программирования Java. Были разработаны алгоритмы для вычисления определителя матрицы, вычисления ранга матрицы, нахождения определённого интеграла и предела, а также вычисления скалярного произведения векторов и расстояния между точками в пространстве.

Анализ и реализация данных алгоритмов позволяют эффективно выполнять ключевые операции в области линейной алгебры, математического анализа и аналитической геометрии. Написанный код обладает высокой универсальностью и применим в различных сферах науки и техники. Все поставленные задачи были достигнуты.

На основе разработанных алгоритмов был реализован прототип мобильного Android-приложения с интуитивно понятным интерфейсом. Приложение включает систему регистрации и авторизации, позволяющую пользователю выбрать роль — студент или преподаватель. В зависимости от роли доступен различный функционал: студент может проходить задания и отслеживать свою успеваемость, а преподаватель — просматривать агрегированную статистику по дисциплинам и конкретным студентам.

Особое внимание в процессе разработки было уделено архитектуре приложения: реализованы универсальные макеты интерфейса и классы-наследники, упрощающие расширение функционала. Логика приложения позволяет легко подключать новые алгоритмы и примеры за счёт модульной структуры и выноса вычислительных методов в отдельные классы.

Система сбора метрик реализована с использованием облачной базы данных Firebase. Она позволяет сохранять данные о решениях заданий, количестве попыток, правильности ответов и времени выполнения. Эти данные автоматически обрабатываются и отображаются в профиле пользователя, формируя основу для индивидуального и группового анализа успеваемости.

Дальнейшее развитие данного приложения может включать:

- расширение набора задач по математике и программированию;

- загрузку пользовательских примеров и реализацию их редактирования;
- улучшение механизма подсказок;
- внедрение полноценного редактора кода с поддержкой выбора языка и возможностью компиляции;
- расширение визуализации статистики и графического представления прогресса.

В целом, выполнение данной курсовой работы позволило не только глубже понять основы линейной алгебры, аналитической геометрии и математического анализа, но и приобрести практические навыки в области проектирования архитектуры мобильного приложения, работы с облачной базой данных и реализации интерфейса с учётом пользовательского опыта. В конечном итоге был создан функциональный и расширяемый прототип обучающего приложения, готовый к дальнейшему развитию.

Список использованных источников

- [1] Что такое Java? Руководство для начинающих по Java / [Электронный ресурс] // Microsoft Azure : [сайт]. — URL: <https://azure.microsoft.com/ru-ru/resources/cloud-computingdictionary/java-programming-language> (дата обращения: 7.12.2024).
- [2] Анастасия Р., Климент Л. Всё, что нужно знать новичку о Java / Анастасия Р., Климент Л. [Электронный ресурс] // Яндекс Практикум : [сайт]. — URL: <https://practicum.yandex.ru/java-dlya-nachinayuschih/> (дата обращения: 10.12.2024).
- [3] Что такое Java? - Описание языка программирования Java - AWS / [Электронный ресурс] // Amazon : [сайт]. — URL: <https://aws.amazon.com/ru/what-is/java/> (дата обращения: 11.12.2024).
- [4] Введение в Java / [Электронный ресурс] // METANIT.COM : [сайт]. — URL: <https://metanit.com/java/> (дата обращения: 11.12.2024).
- [5] Мария Т. Что такое Android Studio и как ей пользоваться / Мария Т. [Электронный ресурс] // SkillBox Media : [сайт]. — URL: <https://skillbox.ru/media/code/chto-takoeandroid-studio-i-kak-ey-polzovatsya/> (дата обращения: 25.12.2024).
- [6] Google Firebase Studio – облачная AI-платформа для разработки приложений / [Электронный ресурс] // Хабр : [сайт]. — URL: <https://habr.com/ru/companies/selectel/articles/908888/> (дата обращения: 22.04.2025).
- [7] Google for Developers. Документация / Google for Developers. [Электронный ресурс] // Firebase : [сайт]. — URL: <https://firebase.google.com/docs/build> (дата обращения: 28.04.2025)
- [8] Java Documentation / [Электронный ресурс] // Java Documentation - Get started : [сайт]. — URL: <https://docs.oracle.com/en/java/> (дата обращения: 28.04.2025).
- [9] Android Mobile App Developers Tools - Android Developers. URL: <https://developer.android.com/tools> (дата обращения: 30.04.2025).