**Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет технологий и управления имени К.Г. Разумовского**

**(Первый казачий университет)»**

|  |
| --- |
| Университетский колледж информационных технологий |

**Отчет**

**О прохождении практики**

**УП.02.01 Учебная практика**

**по профессиональному модулю**

**ПМ.02 Осуществление интеграции программных модулей**

*(вид, тип практики)*

|  |
| --- |
| **Обучающимся Мусатовым Даниилом Романовичем** |
| Направления подготовки/специальности  **09.02.07 Информационные системы и программирование** |
| Профиль подготовки (специализация/квалификация) программист |
| Форма обучения очная |
| Курс 3 |
| Группа 090207–9о–пр–21/1 |

практикант, ФГБОУ ВО «МГУТУ им. К.Г. Разумовского (ПКУ)» УКИТ

*(должность в которой проходил практику, наименование организации/предприятия)*

с «09» июня 2025 г. по «28» июня 2025 г.

Подпись обучающегося

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_Д.Р. Мусатов\_\_\_\_\_

*(подпись) (Ф.И.О.)*

«⠀⠀⠀⠀» ⠀⠀⠀⠀⠀⠀⠀⠀⠀2025 г.

Подпись руководителя

практики от Университета

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ⠀⠀ И.А. Озеркова⠀\_\_\_\_\_

*(подпись) (Ф.И.О.)*

«⠀⠀⠀⠀» ⠀⠀⠀⠀⠀⠀⠀⠀⠀2025 г.

Отчет принял

Заместитель директора по учебно–методической работе

Университетского колледжа информационных технологий

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ⠀⠀⠀⠀⠀ Е.В. Вернер⠀⠀⠀⠀\_⠀⠀

*(подпись) (Ф.И.О.)*

«⠀⠀⠀⠀» ⠀⠀⠀⠀⠀⠀⠀⠀⠀2025 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc201919113)

[ОСНОВНАЯ ЧАСТЬ 6](#_Toc201919114)

[1. Анализ предметной области 6](#_Toc201919115)

[1.1 Общая постановка задач 6](#_Toc201919116)

[1.2 Уточнение постановки задачи 6](#_Toc201919117)

[1.3 Математическая модель 9](#_Toc201919118)

[2. Обоснование выбора средств методов и средств 9](#_Toc201919119)

[2.1 Описание метода решения 9](#_Toc201919120)

[2.2 Анализ средств разработки 10](#_Toc201919121)

[3. Проектирование приложения 11](#_Toc201919122)

[3.1 UML - диаграммы 12](#_Toc201919123)

[3.2 Wireframe-эскизы приложения 15](#_Toc201919124)

[3.3 Описание выбранных форматов данных 21](#_Toc201919125)

[4. Разработка алгоритмов 23](#_Toc201919126)

[5. Разработка приложения 24](#_Toc201919127)

[6. Тестирование приложения 25](#_Toc201919128)

[7. Анализ качества моделирования 26](#_Toc201919129)

[ЗАКЛЮЧЕНИЕ 29](#_Toc201919130)

[СПИСОК ЛИТЕРАТУРЫ 30](#_Toc201919131)

[ПРИЛОЖЕНИЕ А 32](#_Toc201919132)

[Приложение Б 41](#_Toc201919133)

[ПРИЛОЖЕНИЕ В 42](#_Toc201919134)

# ВВЕДЕНИЕ

Учебная практика УП02.01 проходила в лаборатории колледжа «Университетский колледж информационных технологий им. К.Г. Разумовского».

Цель практики: разработка автоматизированной системы (АС), которая включает модули для моделирования и аналитической обработки данных. Система предназначена для наглядного представления методов анализа и визуализации при вычислении площади крупного сегмента, ограниченного окружностью радиуса R (с центром в точке (x₀, y₀)) и прямой, параллельной оси ординат и расположенной на расстоянии C от неё. Расчёты выполнялись с применением как аналитических формул, так и метода Монте-Карло.

В рамках практики решались следующие задачи:

* определение требований, необходимых для выполнения поставленной преподавателем задачи;
* разработка технического задания на основе исходной формулировки;
* изучение теоретических основ расчёта площади сегмента указанной фигуры;
* проектирование и реализация пользовательского интерфейса, обеспечивающего удобное и интуитивно понятное взаимодействие с программой;
* написание программного кода, реализующего алгоритм Монте-Карло для вычисления площади заданной области;
* создание интерфейсных элементов приложения, включая стартовый экран, основной модуль для ввода количества точек, интерфейсы численного и графического моделирования, формы вывода результатов, модуль анализа данных и окно с информацией о программе;
* интеграция встроенной справочной системы для помощи пользователям в работе с приложением.

В ходе практики предусматривалось развитие следующих профессиональных компетенций:

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

Также были продемонстрированы следующие общие компетенции:

ОК 01 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам

ОК 02 Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности

ОК 03 Планировать и реализовывать собственное профессиональное и личностное развитие

ОК 04 Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством,

ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста

ОК 06 Проявлять гражданскопатриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения

ОК 07 Содействовать сохранению окружающей среды, умения: соблюдать нормы экологической безопасности; определять направления ресурсосбережения в рамках профессиональной деятельности по специальности, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях

ОК 08 Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности

ОК 09 Использовать информационные технологии в профессиональной деятельности

ОК 10 Пользоваться профессиональной документацией на государственном и иностранном языках

ОК 11 Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере

# ОСНОВНАЯ ЧАСТЬ

## **Анализ предметной области**

## **Общая постановка задач**

Разработка направлена на создание программы, которая сможет вычислять площадь большей части круга, отсекаемой горизонтальной прямой, используя два подхода:

* аналитический - точный математический расчёт;
* стохастический - приближённый расчёт методом Монте-Карло.

Приложение должно содержать в себе следующие формы:

* заставка с указанием автора и варианта работы;
* основная форма с вводом данных, численным и графическим моделированием и сохранением данных моделирования;
* форма анализа моделирования, включающая в себя графический анализ всех выполненных попыток моделирования;
* форма «О программе»;
* встроенная справка по работе с программой.

Дальше речь пойдет об уточнении поставленной задачи.

## **Уточнение постановки задачи**

На основании технического задания (ТЗ) стандарт IDEF0 применяется для наглядного отображения логических взаимосвязей между процессами. Контекстная диаграмма и диаграмма декомпозиции, разработанные в соответствии с ТЗ, позволяют системно проанализировать структуру и взаимодействие всех компонентов системы.

В соответствии с техническим заданием, общая контекстная диаграмма (A0), представленная на рисунке 1, отображает основную функцию системы - "Вычислить площадь методом Монте-Карло" (A0).

Входные данные:

* количество точек

Выходные данные:

* результаты анализа вычислений
* графическое представление математического моделирования
* сохранённые результаты вычислений

Механизмы выполнения:

* программа Monte-Karlo
* база данных SQLite (для хранения результатов)

Управляющие элементы:

* методы статистического анализа
* техническое задание (ТЗ)
* вычислительные методы



Рисунок 1 – диаграмма IDEF0

Диаграмма декомпозиции (A1-A5) на рисунке 2 раскрывает внутреннюю структуру процесса A0, детализируя его на основные подпроцессы:

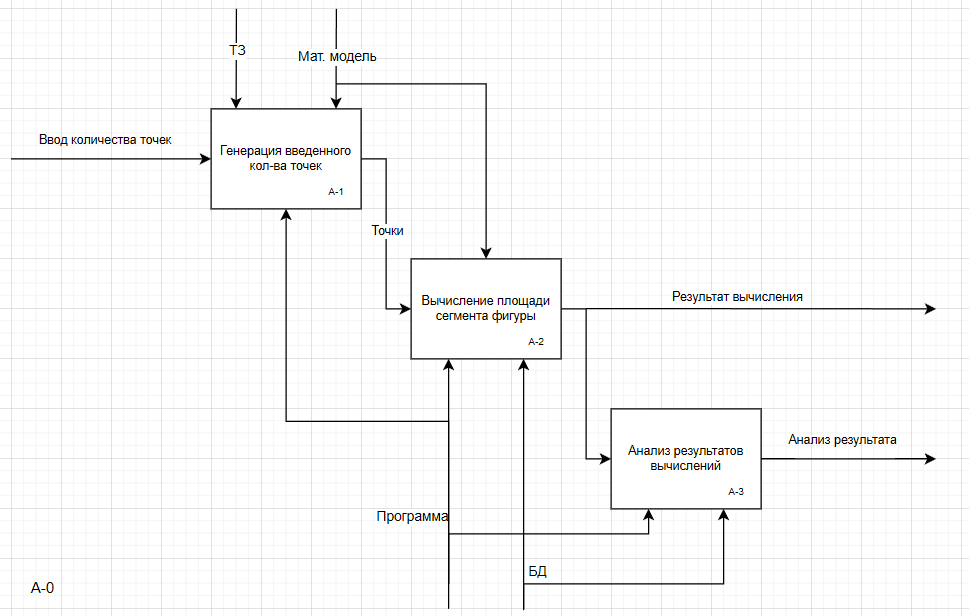


Рисунок 2 – диаграмма IDEF0 с декомпозицией

* A1 «Генерация введенного количества точек». Принимает на вход «количество точек». Результатом работы являются «Точки». Механизм – «Программа Monte-Karlo»;
* A2 «Вычисление площади сегмента фигуры». Использует «Точки» (выход A1). Для выполнения используется механизм «Программа Monte-Karlo», применяющий «Мат. модель». Возвращает «Результат вычислений»;
* A3 «Анализ результатов вычислений». Принимает результаты вычислений. Возвращает анализ результата. Механизм – «Программа Monte-Karlo»;

Ниже приводится описание математической модели.

## **Математическая модель**

Математическое моделирование проводится с помощью графического калькулятора Desmos и получается следующий результат, показанный на рисунке 3 при: x0 = -3, y0 = 1, R = 2, C = -2 и вертикально направленной линии.

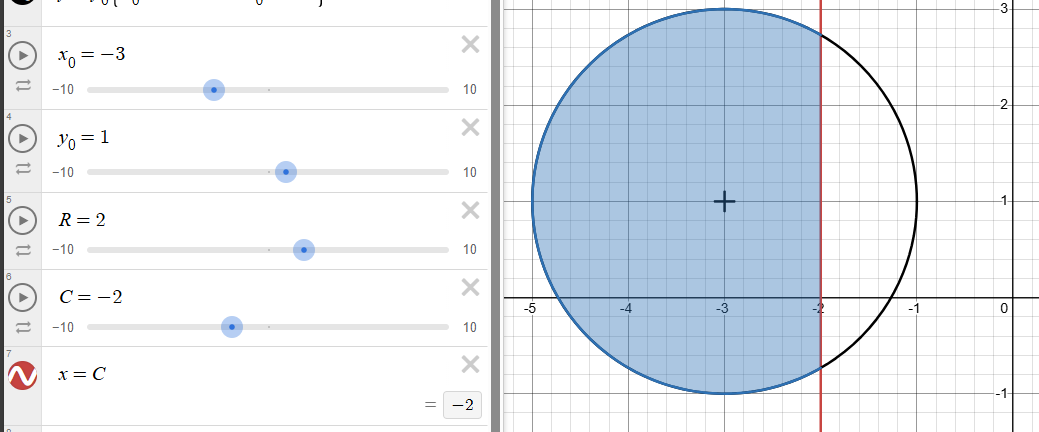


Рисунок 3 – математическое моделирование с помощью Desmos

Математическая модель наглядно иллюстрирует:

* Вывод расчётной формулы площади сегмента
* Фундаментальное свойство - суммарная площадь сегментов в точности соответствует площади всей окружности

## **Обоснование выбора средств методов и средств**

## **Описание метода решения**

Для определения площади большего сегмента будут использоваться 2 подхода:

* аналитический. Вычисляет точный результат площади сегмента. Формула площади сегмента для данного метода: S = R²arccos(d/R) - d√(R²-d²), где d = |C - x0| (или y0);
* стохастический. Будет использоваться метод Монте-Карло, основанный на генерации случайных точек в квадрате, и подсчёте точек, попавших в сегмент окружности. Формула для метода Монте-Карло: S ≈ (Nточек\_внутри/Nточек\_всего) \* Sквадрата.

Важно отметить, что так как метод Монте-Карло стохастический, он даёт приближённый результат, точность которого повышается с увеличением количества точек и проведённых экспериментов.

## **Анализ средств разработки**

Разработка программного обеспечения выполняется на языке C# в среде Microsoft Visual Studio, что обеспечивает тесную интеграцию с платформой .NET и доступ к современным инструментам разработки. C# был выбран благодаря своей производительности, безопасности и поддержке различных парадигм программирования, включая объектно-ориентированный подход и асинхронные операции. Богатая стандартная библиотека и активное сообщество разработчиков делают этот язык оптимальным выбором для реализации сложных вычислительных алгоритмов.

Для хранения данных используется легковесная СУБД SQLite в сочетании с Entity Framework Core. Такой подход позволяет эффективно работать с базой данных, используя преимущества объектно-реляционного отображения. Entity Framework Core значительно упрощает выполнение операций с данными и управление миграциями, что особенно важно при работе с результатами вычислительных экспериментов.

Средой разработки выбран Microsoft Visual Studio, так как он предоставляет разнообразные инструменты для написания, отладки и тестирования кода. Среди ключевых преимуществ:

* встроенный визуальный конструктор форм для Windows Forms, ускоряющий разработку интерфейсов;
* поддержка Entity Framework Core и других ORM, упрощающих работу с базами данных;
* интегрированные средства профилирования и диагностики производительности;
* поддержка Git и систем управления проектами.

Недостатки включают высокие требования к ресурсам (особенно в версии с полным функционалом) и ограниченную поддержку других ОС (хотя есть Visual Studio Code как альтернатива).

Система хранения данных: SQLite + Entity Framework Core. Комбинация SQLite и Entity Framework Core представляет собой мощное решение для работы с данными, обладающее рядом ключевых преимуществ:

* SQLite обеспечивает легковесное хранение информации, не требуя развертывания отдельного серверного приложения - все данные содержатся в едином файле, что значительно упрощает развертывание и распространение приложения. Несмотря на свою компактность, эта СУБД демонстрирует высокую производительность при работе с различными объемами данных.
* Entity Framework Core расширяет возможности работы с SQLite, предоставляя удобный механизм объектно-реляционного отображения. Благодаря поддержке LINQ разработчики получают гибкий инструмент для написания выразительных запросов непосредственно на языке C#, что повышает читаемость кода и ускоряет процесс разработки.

Из недостатков можно отметить ограничения SQLite при работе в многопользовательских сценариях, но для локального приложения это не критично.

Как итог, выбранная комбинация технологий представляет собой оптимальное решение для создания десктопного приложения с требованиями к хранению структурированных данных.

## **Проектирование приложения**

## **3.1 UML - диаграммы**

UML - диаграммы служат мощным инструментом визуализации архитектуры программного комплекса. Они позволяют комплексно отобразить:

* Структурные аспекты
* Поведенческие модели
* Логику взаимодействия

**3.1.1 Диаграмма прецедентов**

UML - диаграмма прецедентов показывает, как пользователь взаимодействует с системой. Диаграмма на рисунке 4 показывает, какие действия может выполнить пользователь и в какой последовательности может их выполнять.

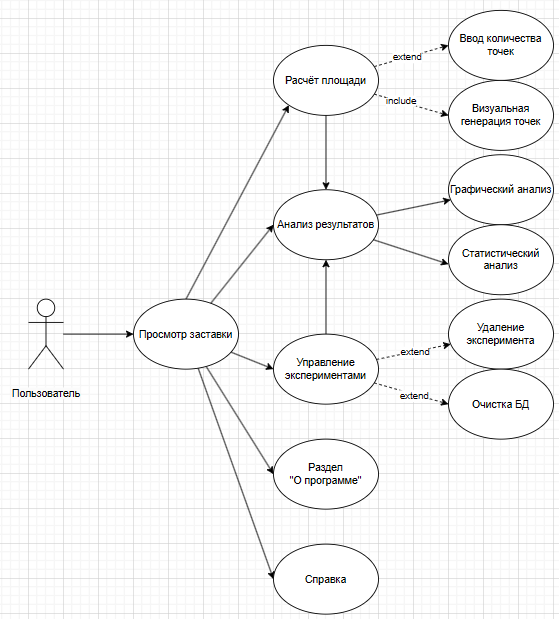


Рисунок 4 – диаграмма прецедентов

С помощью диаграммы прецедентов также можно запланировать какие функции к какой форме будут относиться.

**3.1.2 Диаграмма деятельности**

Диаграмма деятельности UML визуализирует логику работы системы, отображая последовательность выполняемых операций и их взаимосвязи. Этот тип диаграммы детально показывает все шаги процесса, включая ветвления и параллельные потоки выполнения. На рисунке 5 представлена обобщенная диаграмма деятельности для разрабатываемой автоматизированной системы.

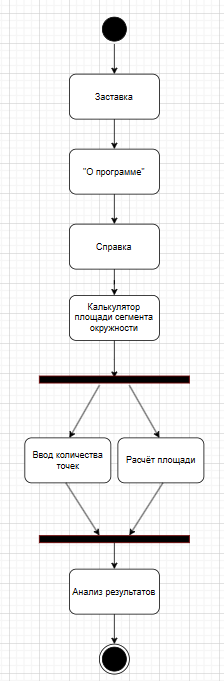


Рисунок 5 – диаграмма деятельности

С помощью этой диаграммы можно запланировать какие интерактивные элементы будут в программе и как будут вызываться функции в программе.

**3.1.3 Диаграмма последовательности**

Диаграмма последовательностей UML визуализирует динамику взаимодействия объектов в системе, отражая временной порядок выполнения операций. Этот тип диаграммы играет ключевую роль в проектировании, поскольку четко демонстрирует цепочку вызовов и обмена сообщениями между компонентами. В рамках автоматизированной системы, где задействовано множество взаимосвязанных объектов, на рисунке 6 (созданном в PlantUML) представлен наиболее сложный сценарий взаимодействия, а именно инициализация объектов и возможная генерация точек с отображением результата.

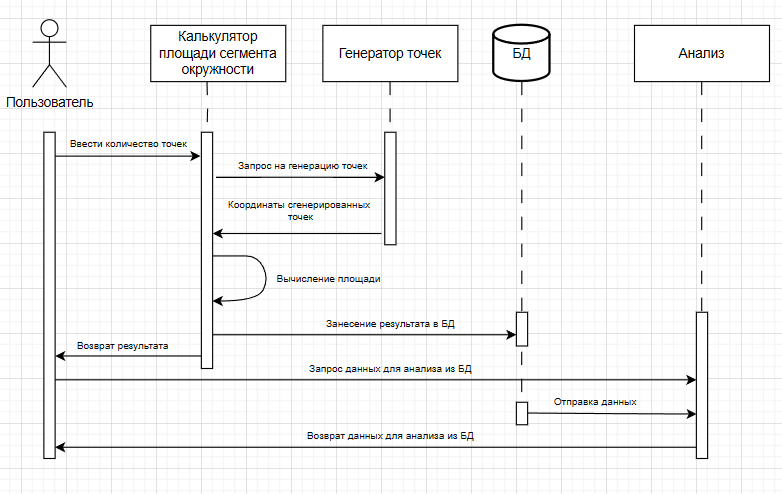


Рисунок 6 – диаграмма последовательности для инициализации объектов и возможной генерации точек

С помощью диаграммы можно понять сколько будут существовать объекты в программе и как часто они используются.

## **3.2 Wireframe-эскизы приложения**

Программа проектируется как многооконное приложение с продуманной структурой форм. Разработанные заранее Wireframe-эскизы служат наглядным прототипом интерфейса, позволяя до начала разработки определить внешний вид и логику работы каждой формы. Это существенно упрощает последующий процесс реализации пользовательского интерфейса.

Стартовая заставка, представленная на рисунке 7, формирует первое впечатление пользователя о приложении. Как видно из эскиза, она включает три ключевых элемента: предустановленные параметры для расчетов, информацию об авторе программы, а также пример графической визуализации. На рисунке 8 показан образец будущего графика с точками, который поможет пользователю сразу понять принцип отображения результатов моделирования.

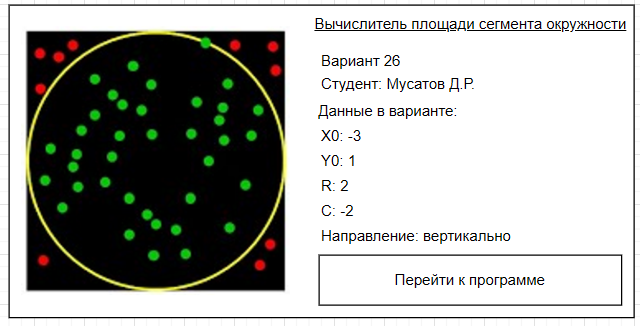


Рисунок 7 – эскиз заставки приложения

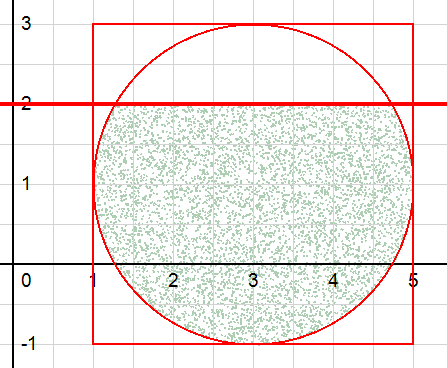


Рисунок 8 – картинка на заставке приложения

После стартовой заставки пользователь переходит в основное окно программы, представленное на рисунке 9. Этот центральный интерфейсный элемент включает несколько ключевых компонентов, организованных для удобства работы.

В левой части расположена панель управления, предназначенная для настройки параметров математического моделирования. Сверху находится навигационное меню, обеспечивающее доступ ко всем функциональным формам приложения. Основную площадь окна занимает область визуализации, расположенная справа от панели управления.

В графической зоне будет отображаться координатная плоскость с соответствующими осями и подписями. На этом поле программа рисует ограничивающий квадрат, внутри которого располагаются элементы моделирования: окружность заданного радиуса, секущая линия и сгенерированные точки. Особенностью визуализации является отображение только тех точек, которые попадают в больший сегмент окружности, что позволяет наглядно демонстрировать принцип работы метода Монте-Карло.

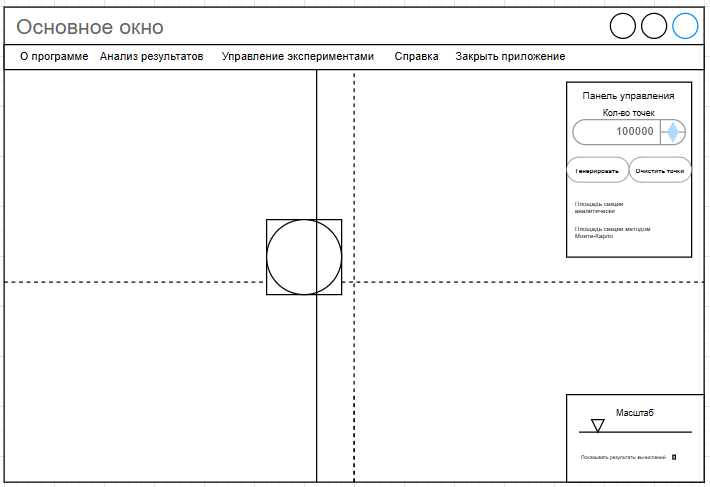


Рисунок 9 – эскиз основного окна

Справочная документация будет отображаться в браузере пользователя, используя стандартный дизайн, сгенерированный программой Dr.Explain. Это решение позволяет создать удобную и функциональную справку без необходимости разработки собственного интерфейса для документации.

Окно «О программе» разработано в минималистичном стиле и содержит всю необходимую информацию о приложении. В верхней части окна расположено название программы и крупноформатная иконка приложения, которая сразу привлекает внимание пользователя. Ниже указаны данные об авторе, текущая версия программного обеспечения и активная ссылка на GitHub-репозиторий, где можно ознакомиться с исходным кодом проекта.

Визуальное оформление окна «О программе» и дизайн иконки приложения можно увидеть на рисунках 10 и 11 соответственно. Эти элементы интерфейса созданы в едином стиле с основной программой, что обеспечивает целостность пользовательского опыта.

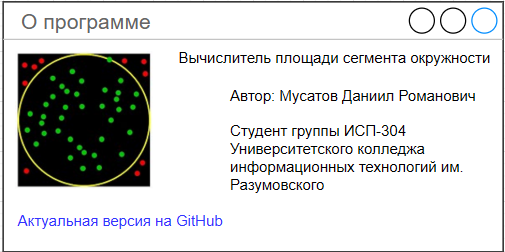


Рисунок 10 – эскиз окно «О программе»

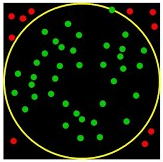
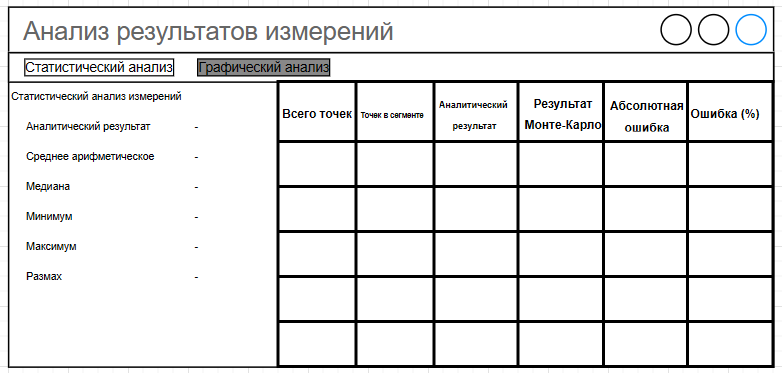


Рисунок 11 – иконка приложения

Окно «Анализ результатов» будет реализовано с интерфейсом на основе двух вкладок, позволяющих переключаться между различными представлениями данных. Первая вкладка посвящена числовой обработке результатов - здесь будет представлена детальная статистическая информация по проведенным измерениям, а также таблица с итоговыми значениями эксперимента.

Вторая вкладка, «Графический анализ», содержит визуальное представление данных. Основное пространство занимает точечная диаграмма, где каждая точка соответствует отдельному вычислению площади методом Монте-Карло. На этом же графике будут отображены линии, демонстрирующие результаты статистической обработки данных. Для удобства интерпретации в правом верхнем углу размещена цветовая легенда, поясняющая значение каждого элемента графика.



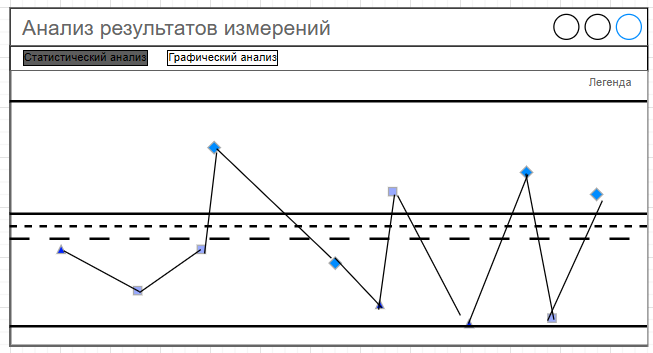


Рисунок 12 – эскиз вкладок в окне с анализом результатов

Окно управления экспериментами спроектировано как центральный узел для работы с данными исследований. Основное пространство интерфейса занимает таблица со списком всех проведенных экспериментов, где пользователь может выбирать нужные записи для дальнейших действий.

Нижняя панель содержит группу кнопок для операций над экспериментами:

* полная очистка всех экспериментальных результатов.
* удаление конкретного результата
* анализ конкретного результата

Визуальное оформление и компоновка элементов управления представлены на рисунке 13.



Рисунок 13 – эскиз окна для управления экспериментами

В целом дизайн приложения планируется простым и понятным для пользователя.

## **3.3 Описание выбранных форматов данных**

Для надежного сохранения результатов вычислений методом Монте-Карло была спроектирована реляционная модель данных, основанная на двух взаимосвязанных таблицах. Эта структура обеспечивает логичное разделение информации и поддерживает связи между сущностями.

Логическая структура базы данных с указанием типов полей, первичных ключей и связей между таблицами представлена на рисунке 14. Эта схема демонстрирует нормализованную структуру, оптимизированную для выполнения типичных запросов приложения.

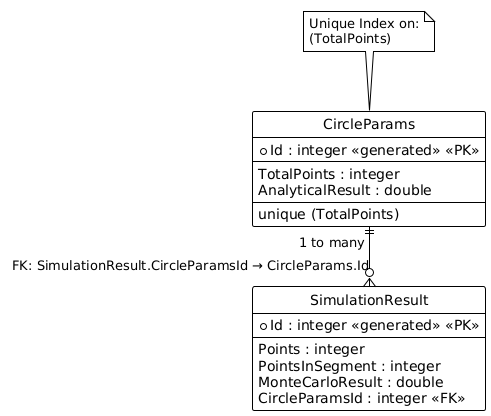


Рисунок 14 – логическая модель базы данных

Таблица CircleParams (Параметры окружности) содержит геометрические параметры фигур для экспериментов:

* Id (int, PK) – уникальный идентификатор записи;
* TotalPoints (int) – общее количество точек в эксперименте;
* AnalyticalResult (double) – аналитически рассчитанная площадь.

Таблица SimulationResult (Результаты симуляций) фиксирует статистику экспериментов:

* Id (int, PK) – уникальный идентификатор;
* Points (int) – точки внутри окружности;
* PointsInSegment (int) – точки в целевом сегменте;
* MonteCarloResult (double) – вычисленная площадь методом Монте-Карло;
* CircleParamsId (int, FK) – связь с параметрами эксперимента.

Между таблицами установлена связь один-ко-многим: один набор параметров окружности может соответствовать нескольким результатам экспериментов.

Использование реляционной модели в SQLite обеспечивает критически важные для научных расчетов преимущества:

* целостность данных – внешние ключи гарантируют отсутствие "сиротствующих" результатов экспериментов без соответствующих параметров;
* эффективность запросов – уникальный индекс ускоряет поиск дубликатов параметров и снижает избыточность хранения;
* точность вычислений – тип double гарантирует сохранение дробных значений без потери точности;
* масштабируемость – нормализованная структура позволяет добавлять новые параметры экспериментов без изменения схемы;
* атомарность операций – транзакционная модель SQLite защищает данные от частичных обновлений при сбоях.

Такой подход особенно эффективен для научных задач, где требуется сопоставление параметров экспериментов с их результатами, статистический анализ серий вычислений и верификация данных. Реляционная модель обеспечивает однозначную интерпретацию связей между геометрическими параметрами и результатами симуляций, что принципиально важно для воспроизводимости экспериментов.

## **Разработка алгоритмов**

Блок-схема, представленная на рисунке 15, отражает ключевые этапы работы основного алгоритма программы. Схематично показана логика выполнения операций на главной форме приложения, начиная от инициализации компонентов до вывода конечных результатов.

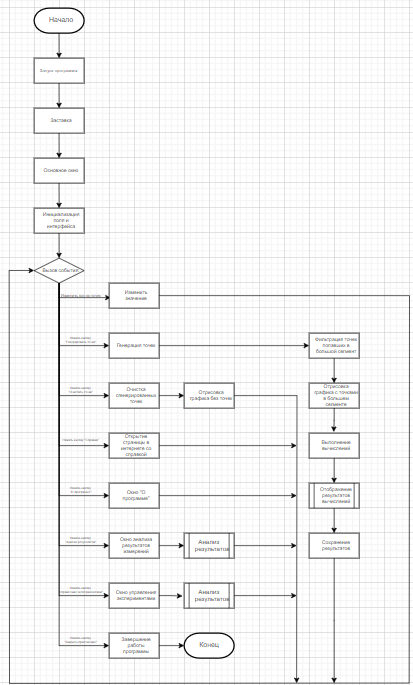


Рисунок 15 – общая блок схем приложения

Эта диаграмма поможет на этапе разработки приложения.

## **Разработка приложения**

Разработка приложения происходит в соответствии с её проектированием

При запуске исполняемого файла открывается заставка, показанная на рисунке В1 в приложении. Если программа уже была запущена, повторный дубликат программы не запустится, и активирует окно предыдущей запущенной версии.

Заставка показывается 3 секунды или пока не будет нажата кнопка «Перейти к программе». В любом из этих случаев произойдёт переход к основному окну.

В основном окне, показанном на рисунке В2 в приложении, происходит математическое моделирование методом Монте-Карло. В числовое поле количества точек, доступное для ввода с клавиатуры не получится ввести некорректные значения, такие как буквы или символы, а количество не получится задать меньше 1 и больше 108.

От основного окна с помощью верхнего меню происходит переход между другими окнами. Другие окна открываются как модальные, то есть пока открыто модальное окно, взаимодействовать с другими окнами приложения не получится. Кнопка «Закрыть приложение» просто закрывает приложение.

Нажав на справку, произойдёт запуск страницы с руководством пользователя в браузере по умолчанию. Пример этого показан на рисунке В3 в приложении.

В окне «О программе», показанном на рисунке В4 в приложении, находится информация об авторе и программе. Если нажать на ссылку на GitHub откроется репозиторий автора.

Окно с анализом, чьи вкладки показаны на рисунках В5 и В6 в приложении, отображают анализ экспериментов в разных форматах.

В окне с управлением экспериментами, показанном на рисунке В7 в приложении, происходит глобальное управление экспериментами. В нём можно удалить все результаты. Выбрав эксперимент нажатием на строчку с экспериментом, можно удалить его или посмотреть анализ.

Все операции с базой данных, кроме получения данных, записываются в файлы, в папке Logs.

## **Тестирование приложения**

Тестирование приложения проводилось комплексно с использованием юнит-тестов и интеграционных тестов, обеспечивающих проверку корректности работы всех компонентов системы. Всего было разработано 44 юнит-тестов и 1 интеграционный тест, что подтверждает высокую степень покрытия кода тестами. Результаты успешного прохождения тестов представлены на рисунке 16.

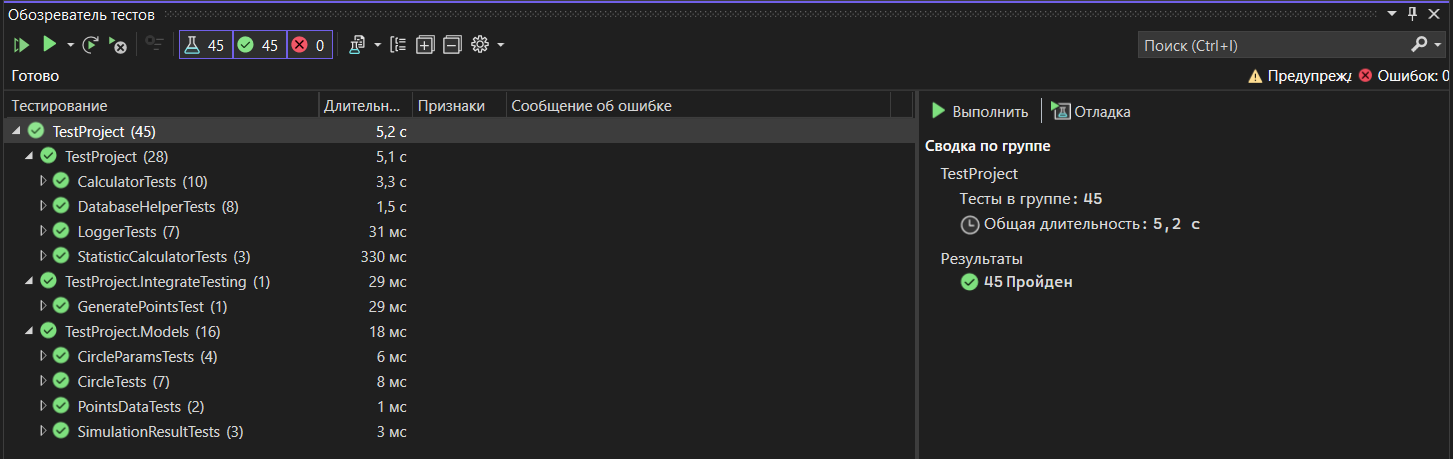


Рисунок 16 – результат прохождения авто тестов

Основные направления тестирования включают:

* тестирование математических вычислений. Были проверены все математические модули приложения и такие аспекты как: проверка корректности вычисления площади сегмента на основе соотношения точек, аналитические расчёты, тестирование граничных условий (нулевой радиус, секущая через центр) и обработка ошибок;
* тестирование моделей данных. Протестированы все сущности предметной области. Все сущности корректно инициализируются и работают, им нельзя присвоить некорректные параметры, и они правильно переводятся в строковую форму;
* тестирование работы с базой данных, а именно модуля DatabaseHelper. В нём проверялось создание и пересоздание базы, CRUD-операции, согласованность данных при добавлении связанных сущностей;
* тестирование системы логирования. Автоматическое создание директорий и файлов, корректность добавления временных меток к логам, тестирование записи стек трейсов ошибок и самих ошибок и проверка работы в условиях отсутствия прав на запись;
* тестирование статистических вычислений в модуле StatisticCalculator. В него входило проверка правильности вычисления моды, медианы, размаха, дисперсии, стандартного отклонения;
* интеграционное тестирование. Проведено тестирование комплексного сценария: создание 100 000 случайных точек, проверка попадания точек в сегмент и расхождение между аналитическим и вероятностным методом ≤ 5%.

Также проходило тестирование интерфейса, проверяющее, что все запланированные элементы корректно отображаются и что переход между вкладками происходит корректно.

## **Анализ качества моделирования**

Основным результатом моделирования в рамках данной работы является оценка площади большого сегмента, образованного окружностью и горизонтальной прямой, с использованием метода Монте-Карло (МК). Качество полученных оценок анализировалось по следующим ключевым аспектам:

* зависимость точности от объема выборки (N). Как и ожидалось теоретически, метод Монте-Карло демонстрирует приблизительный характер результатов. Главным фактором, определяющим точность оценки площади, является количество случайных точек (N), сгенерированных внутри ограничивающего прямоугольника. Наблюдается четкая закономерность: погрешность оценки уменьшается по мере увеличения N. Результаты, полученные при малом N (например, 100-1000 точек), характеризуются значительным разбросом и высокой относительной погрешностью относительно аналитического значения. При увеличении N до 10⁴, 10⁵ и более точек, результаты становятся существенно стабильнее и ближе к истинному значению площади. Эта зависимость напрямую связана со статистической природой метода Монте-Карло: ошибка оценки убывает пропорционально 1/√N (согласно закону больших чисел и центральной предельной теореме);
* стабилизация оценки при повторных экспериментах. Важным подтверждением надежности метода и качества моделирования стало наблюдение за поведением оценки при многократном повторении относительно малого N. Было установлено, что хотя результаты отдельных экспериментов с малым N могут сильно отклоняться от аналитического значения и друг от друга, их среднее арифметическое и медиана по серии таких экспериментов (например, 100 или 1000 запусков) устойчиво приближаются к точному аналитическому значению площади. Этот эффект является прямым следствием закона больших чисел: среднее результатов независимых испытаний сходится к математическому ожиданию (которое в данном случае равно искомой площади) при увеличении числа испытаний. Сходимость медианы также подтверждает устойчивость оценки;
* источники погрешности. Основным источником погрешности в данной реализации является статистическая погрешность, присущая самому методу Монте-Карло. Она обусловлена случайным характером генерации точек и конечностью выборки N. Величина этой погрешности может быть оценена (например, через стандартную ошибку среднего или с использованием доверительных интервалов), и она управляема через параметр N. Потенциальные систематические погрешности (например, связанные с некорректной генерацией точек или проверкой попадания) в корректно реализованной модели должны быть пренебрежимо малы по сравнению со статистической погрешностью при достаточно больших N.

Как итог, можно сказать, что разработанная модель на основе метода Монте-Карло эффективна для решения поставленной задачи оценки площади сегмента. Модель демонстрирует предсказуемое поведение, соответствующее теоретическим основам метода: точность результата прямо и управляемо зависит от количества сгенерированных случайных точек (N). Наблюдаемая сходимость среднего и медианы результатов множества экспериментов с малым N к аналитическому значению является сильным эмпирическим подтверждением корректности реализации модели и статистической обоснованности получаемых с ее помощью оценок. Для получения оценки с заданной точностью необходимо выбирать N, соответствующее требуемому уровню статистической погрешности. Альтернативно, при ограниченных вычислительных ресурсах (малое N), повышения надежности оценки можно достичь путем усреднения результатов большого количества независимых прогонов модели.

# ЗАКЛЮЧЕНИЕ

В ходе прохождения учебной практики по профессиональному модулю ПМ.02 Осуществление интеграции программных модулей в период с 09.06.2025 по 28.06.2025 в Университетском колледже информационных технологий ФГБОУ ВО «МГУТУ им. К.Г. Разумовского (ПКУ)» в Лаборатории программного обеспечения и сопровождения компьютерных систем, освоены необходимые практические умения и закреплены навыки работы по данному модулю.

Основной целью практики стало приобретение профессиональных навыков и компетенций через выполнение задач, соответствующих будущей специальности. В рамках учебной практики был получен ценный опыт интеграции программных модулей, а также освоены ключевые профессиональные компетенции.

В процессе работы проведен анализ требований к решению задачи вычисления площади сегмента окружности. На основании условий, предоставленных преподавателем, разработано техническое задание. Изучены математические основы расчетов с применением аналитического подхода и метода Монте-Карло.

Реализован удобный графический интерфейс, содержащий стартовый экран, основной рабочий модуль, компоненты численного и графического анализа, систему сохранения результатов и справочный раздел. Программно воплощен метод Монте-Карло с функцией визуализации процесса вычислений. Дополнительно интегрирована встроенная справочная система для помощи пользователям.

Завершающим этапом стало тестирование приложения и проверка точности расчетов. В результате успешно создано программное обеспечение, сочетающее аналитические и статистические методы вычислений с продуманным интерфейсом. Приобретенные навыки проектирования и разработки программных решений способствуют профессиональному росту в сфере создания программного обеспечения. Полученный опыт имеет значительную практическую ценность для дальнейшего обучения и профессиональной деятельности.

# СПИСОК ЛИТЕРАТУРЫ

1. Система вопросов и ответов о программировании: [Электронный ресурс]. URL: <https://stackoverflow.com/questions/> (дата обращения: 21.06.2024)
2. Система вопросов и ответов о программировании на русском языке: [Электронный ресурс]. URL: <https://ru.stackoverflow.com/questions/> (дата обращения: 22.06.2024)
3. Как оформить список литературы по госту: [Электронный ресурс]. URL: https://journal.tinkoff.ru/how-to-make-bibliography/ (дата обращения: 22.06.2024)
4. Документация по MySQL Workbench: [Электронный ресурс]. URL: <https://dev.mysql.com/doc/workbench/en/> (дата обращения: 15.06.2024)
5. MySQL 8.4 Reference Manual: [Электронный ресурс]. URL: <https://dev.mysql.com/doc/refman/8.4/en/> (дата обращения: 15.06.2024)
6. Исчерпывающее руководство по Юникоду и кодировке символов в Python: [Электронный ресурс]. URL: <https://tproger.ru/translations/unicode-and-encoding-python-guide> (дата обращения: 18.06.2024)
7. How to support full Unicode in MySQL databases: [Электронный ресурс]. URL: <https://mathiasbynens.be/notes/mysql-utf8mb4#character-sets> (дата обращения: 18.06.2024)
8. Справочник по Transact-SQL (ядро СУБД): [Электронный ресурс]. URL: <https://learn.microsoft.com/ru-ru/sql/t-sql/language-reference?view=sql-server-ver16> (дата обращения: 24.06.2024)
9. SQL Tutorial: [Электронный ресурс]. URL: <https://www.w3schools.com/sql/default.asp> (дата обращения: 25.06.2024)
10. Транзакции в MySQL: [Электронный ресурс]. URL: <https://www.garb.ru/blog/transaction.html> (дата обращения: 22.06.2024)
11. Сброс AUTO\_INCREMENT в MySQL: как принудительно начать с 1: [Электронный ресурс]. URL: <https://sky.pro/wiki/sql/sbros-auto-increment-v-my-sql-kak-prinuditelno-nachat-s-1/> (дата обращения: 21.06.2024)
12. Типы данных MySQL: [Электронный ресурс]. URL: <https://metanit.com/sql/mysql/2.3.php> (дата обращения: 18.06.2024)
13. Сообщение об ошибке «входная строка имела неверный формат» при отсутствии параметра decimal во время функции службы Web в Microsoft Dynamics NAV 2009: [Электронный ресурс]. URL: [https://support.microsoft.com/ru-ru/topic/сообщение-об-ошибке-входная-строка-имела-неверный-формат-при-отсутствии-параметра-decimal-во-время-функции-службы-web-в-microsoft-dynamics-nav-2009-8286ef77-cd0a-ae9a-d912-136c2fa2e5d1](https://support.microsoft.com/ru-ru/topic/%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D0%B5-%D0%BE%D0%B1-%D0%BE%D1%88%D0%B8%D0%B1%D0%BA%D0%B5-%D0%B2%D1%85%D0%BE%D0%B4%D0%BD%D0%B0%D1%8F-%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B0-%D0%B8%D0%BC%D0%B5%D0%BB%D0%B0-%D0%BD%D0%B5%D0%B2%D0%B5%D1%80%D0%BD%D1%8B%D0%B9-%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82-%D0%BF%D1%80%D0%B8-%D0%BE%D1%82%D1%81%D1%83%D1%82%D1%81%D1%82%D0%B2%D0%B8%D0%B8-%D0%BF%D0%B0%D1%80%D0%B0%D0%BC%D0%B5%D1%82%D1%80%D0%B0-decimal-%D0%B2%D0%BE-%D0%B2%D1%80%D0%B5%D0%BC%D1%8F-%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D1%8B-web-%D0%B2-microsoft-dynamics-nav-2009-8286ef77-cd0a-ae9a-d912-136c2fa2e5d1) (дата обращения: 24.06.2024)
14. Grant a user permission to only view a MySQL view and nothing else: [Электронный ресурс]. URL: https://stackoverflow.com/questions/3108656/grant-a-user-permission-to-only-view-a-mysql-view-and-nothing-else (дата обращения: 21.06.2024)
15. Использование триггеров базы данных MySQL: [Электронный ресурс]. URL: <https://www.digitalocean.com/community/tutorials/how-to-manage-and-use-mysql-database-triggers-on-ubuntu-18-04-ru> (дата обращения: 21.06.2024)

**ПРИЛОЖЕНИЕ А**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**Общие сведения**

Наименование: Вычислитель площади сегмента окружности.

Условное обозначение: Monte-Karlo v1.0.

Наименование организации: Университетский колледж информационных технологий им. Разумовского.

Разработчик: Мусатов Даниил Романович

Перечь документов, на основании которых создается АС утверждено руководителем практики 09.06.2025.

Плановые сроки начала и окончания работ по созданию АС

Начало работ: 09.06.2025

Окончание работ: 28.06.2025

Работа выполняется в рамках учебной практики. Финансирование не предусмотрено.

**Цели создание АС:**

* разработать программный продукт для численного моделирования методом Монте-Карло;
* автоматизировать вычисления площади сегмента окружности;
* визуализировать процесс моделирования;
* визуализировать процесс анализа;
* освоить практический навык программирования.

Критерии оценки достижения целей создания АС:

* корректность численных расчетов (погрешность < 5% при количестве точек > 10 000);
* время расчета: ≤1 сек для N ≤ 10000;
* отображать график, соответствующий результатам моделирования;
* удобство пользовательского интерфейса.

**Назначение АС**

Программа предназначена для поддержки учебного процесса, демонстрации метода Монте-Карло и автоматизации расчетов в учебных и исследовательских целях.

**Характеристика объектов автоматизации**

Объект автоматизации – процесс численного моделирования площади сегментов окружности.

**Характеристика окружающей среды**

Эксплуатация в стандартных офисных и учебных помещениях.

**4. Требования к автоматизированной системе**

4.1 Требования к структуре АС в целом

Структура программы:

* заставка;
* основная форма;
* форма анализа моделирования;
* форма «О программе»;
* форма управления экспериментами;
* встроенная справка.

4.1.1 Компоненты взаимодействуют через общий модуль данных.

4.1.2 Требования к характеристикам взаимосвязей создаваемой АС со смежными АС не предусмотрено.

4.1.3 Требование к режимам функционирования АС:

* интерактивный режим с визуализацией.

4.1.4. Требования по диагностированию АС

Обработка исключительных ситуаций:

* все исключения должны быть перехвачены. Отображение стеков трассировки пользователю недопустимо, следует показывать дружелюбные сообщения;
* каждое критическое событие должно фиксироваться в отдельном журнале («error.log») с указанием причины и времени возникновения.

4.1.5 Перспективы развития, модернизации АС:

* добавление поддержки других геометрических фигур;
* возможность задания координат центра окружности
* возможность задания радиуса окружности
* возможность задания произвольной прямой;
* многоязычная поддержка интерфейса;
* добавление динамического округления в зависимости от количества точек;
* добавление других способов анализ данных.

**4.2 Требования к функциям (задачам), выполняемым АС**

4.2.1 Перечень функций:

* прием и проверка исходных данных;
* аналитический расчет площади;
* численное моделирование методом Монте-Карло;
* графическое отображение процесса моделирования;
* сохранение результатов;
* графическое отображение анализа;
* проведение серии моделирований для анализа погрешности;
* управление результатами моделирований;
* логирование операций удаления и записи в базу данных и генерации данных;
* отображение справки.

4.2.2 Результаты выполнения функций:

* вывод значения площади сегмента;
* визуализация результатов;
* сохраненные файлы результатов;
* файлы логирования.

**4.3 Требования к видам обеспечения АС**

4.3.1 Требования к математическому обеспечению АС

Формулы расчета площади сегмента:

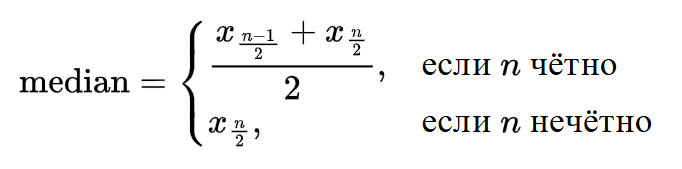
* формула площади круга: S = πR2
* формула площади квадрата, описывающего круг Sквадрата = 4R2
* формула площади сегмента: S = R²arccos(d/R) - d√(R²-d²), где d = |C - x0| (или y0)
* формула для метода Монте-Карло: S ≈ (Nвнутри/Nвсего) \* Sквадрата
* алгоритм генерации случайных точек: координата точки по оси X или Y это случайное число с плавающей точкой от 0 до 1, умноженное на удвоенный радиус минус радиус. Получается координата случайной точки в области [-R, R).

Методы оценки точности:

* абсолютная погрешность ∆X = Xд – X, где Xд – действительное значение, а X – измеренное значение;
* относительная погрешность δ = (|∆X| / Xд) \* 100%, где Xд – действительное значение, а ∆X – абсолютная погрешность измерений.

Методы статистического анализа:

* медиана, значение, разделяющее упорядоченную выборку на две равные части.



Где x0≤x1≤…≤ xn-1 - упорядоченная выборка, n - размер выборки.

* мода наиболее часто встречающееся значение в выборке.
* дисперсия среднее арифметическое квадратов отклонений от среднего значения.



Где x0≤x1≤…≤ xn-1 - упорядоченная выборка, n - размер выборки.

* стандартное отклонение корень квадратный из дисперсии.
* размах (Range) разность между максимальным и минимальным значениями выборки. Range = max(values) − min(values), где values – выборка.

4.3.2 Требования к информационному обеспечению:

* ввод исходных данных пользователем;
* хранение и сохранение результатов в базе данных.

4.3.3 Требования к лингвистическому обеспечению:

* интерфейс – русский язык.

4.3.4 Требования к программному обеспечению:

* язык программирования – C# 13;
* среда выполнения - .NET 9.0;
* операционная система – Windows 7/10/11;
* Библиотеки:
  + Math;
  + Microsoft.EntityFrameworkCore.SQLite.

4.3.5 Требования к техническому обеспечению:

* ПК с процессором не ниже Intel Core i3;
* ОЗУ не менее 512 МБ;
* клавиатура;
* мышь;
* цветной монитор 13 дюймов;
* место на диске 1 ГБ.

4.3.6 Требования к метрологическому обеспечению

* оценка погрешности < 5%, при количестве точек ≥ 10 000;
* округление абсолютной и относительной погрешности до двух значащих цифр.

4.3.7 Требования к организационному обеспечению не требуются.

4.3.8 Использовать методические материалы по практике дисциплины:

* ГОСТ 34.602-2020;
* ГОСТ 19.701-90;
* ГОСТ Р 7.0.80-2023;
* ГОСТ 7.32-2017;
* ГОСТ Р 7.05.-2008.

**4.4 Общие технические требования АС**

4.4.1 Требования к численности и квалификации персонала и пользователя АС

Пользователь – студент или преподаватель. Навыки: базовые математической теории вероятности.

4.4.2 Требования к показателям назначения:

* автоматизация расчета
* демонстрация метода Монте-Карло.

4.4.3 Требования к надежности:

* устойчивость к ошибкам;
* обработка недопустимых значений.

4.4.4 Требования по безопасности не требуют специальных мер.

4.4.5 Требования к эргономике и технической эстетике

* простой и понятный интерфейс;
* предупреждение для эпилептиков если есть моргающие элементы;
* графическая визуализация результатов;
* подписи всех элементов управления;
* цветовая дифференциация элементов на графике.

4.4.6 Требования к транспортабельности для подвижных АСне применимо.

4.4.7 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов АС не требуются.

4.4.9 Требования по сохранности информации при авариях:

* создание резервной копии пользователем.

4.4.10 Требования к защите от влияния внешних воздействий не требуются.

4.4.11 Требования к патентной чистоте и патентоспособности не требуются.

4.4.12 Требования по стандартизации и унификации

* использование стандартных и сторонних библиотек C#.

4.4.13 Дополнительные требования не предусмотрены.

5. **Состав и содержание работ по созданию автоматизированной системы**

* анализ требований к разработке ТЗ (09.06.2025);
* разработка ТЗ к разработке (10.06.2025);
* разработка вычислений методом Монте-Карло (11.06.2025 – 12.06.2025);
* разработка пользовательского интерфейса (13.06.2025 – 14.06.2025);
* разработка формы анализа (15.06.2025);
* подготовка автоматических тестов (16.06.2025);
* подготовка ручных тестов (17.06.2025);
* проведение тестирования (18.06.2025 – 20.06.2025);
* подготовка пояснительной записки (21.06.2025 – 26.06.2025);
* подготовка к приему разработанной АС (27.06.2025);
* прием разработанной АС (28.06.2025).

6. **Порядок разработки автоматизированной системы**

Разработка АС осуществляется по следующим этапам с учётом выбранной программно-технической платформы:

6.1 Формулировка цели и задач разработки

* определяются цели автоматизации, состав и назначение задач, решаемых системой;
* анализируется предметная область;
* оценивается эффективность внедрения.

Документ: Техническое задание (разделы 1 и 2 по ГОСТ 34.602-89).

6.2. Выбор среды и средств разработки

В качестве основной платформы выбрана:

* ОС: Windows 11;
* язык программирования: C# 13;
* среда выполнения: .NET 9.0 – windows;
* фреймворк: Windows Forms;
* математическая библиотека: Math.NET;
* работа с базой данных через Microsoft.EntityFrameworkCore
* среда разработки: Visual Studio 2022 или новее.

6.3. Разработка математического и алгоритмического обеспечения

* построение математических моделей процессов;
* использование стандартной библиотеки Math для реализации алгоритмов, включая:
* статистические расчёты;
* метод Монте-Карло;

6.4. Разработка программного обеспечения и пользовательского интерфейса

* реализация графического интерфейса с использованием Windows Forms;
* реализация функций сохранения данных в базу данных SQLite.

6.5. Проведение испытаний программного обеспечения

* модульное тестирование классов и алгоритмов с использованием MS Test;
* интеграционные тесты взаимодействия между модулями с использованием MS Test;
* сценарии ручного тестирования интерфейса;
* проведение нагрузочных и граничных тестов алгоритмов Монте-Карло;
* испытания соответствуют этапам: предварительные, приёмочные, опытная эксплуатация.

6.6. Оформление документации и подготовка к вводу в эксплуатацию:

* подготовка комплекта документации;
* создание установщика;
* подготовка системы к сдаче: финальное тестирование, резервное копирование, выдача документации заказчику.

**7. Порядок контроля и приемки автоматизированной системы**

* проверка ручным тестированием пользовательский интерфейс;
* автоматизированное тестирование математических расчетов;
* проверка корректности реагирование программы на ошибки;
* проверка графического отображения результатов;
* демонстрация работы программы на нескольких наборах исходных данных.

**8. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие**

* OC Windows 7/10/11;
* установка .NET v9;
* установка приложения;
* проверка корректной работы программы на целевом оборудовании.

Установка приложения происходит следующим образом:

1. Подготовка системы. Установите .NET 9 Runtime (требует прав администратора) и скачайте архив Monte-Karlo.rar с официального источника;
2. Распаковка. Скопируйте архив в целевую директорию (например, C:\Program Files\Monte-Karlo\) и распакуйте архив с сохранением структуры папок (используйте WinRAR/7-Zip);
3. Запуск программы. Откройте распакованную папку и запустите Monte-Karlo.exe. При первом запуске автоматически создадутся DataBase.db, хранилище экспериментов, и папка Logs\ с журналами ошибок в формате YYYY-MM-DD.log.

**9. Требования к документированию**

* инструкция пользователя;
* встроенная справка по работе с программой;
* форма «О программе» с указанием автора и версии;
* код программы.

**10. Источники разработки**

* задание на учебную практику;
* ГОСТ 34.602-2020;

# Приложение Б

**КОД ПРИЛОЖЕНИЯ**

# ПРИЛОЖЕНИЕ В

**ОКНА ПРОГРАММЫ**

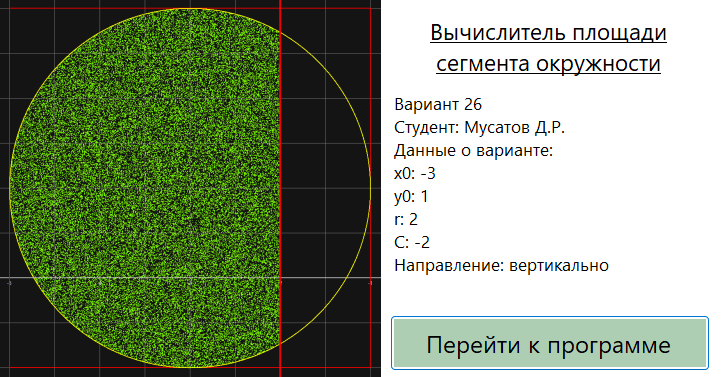


Рисунок В1 – Заставка программы

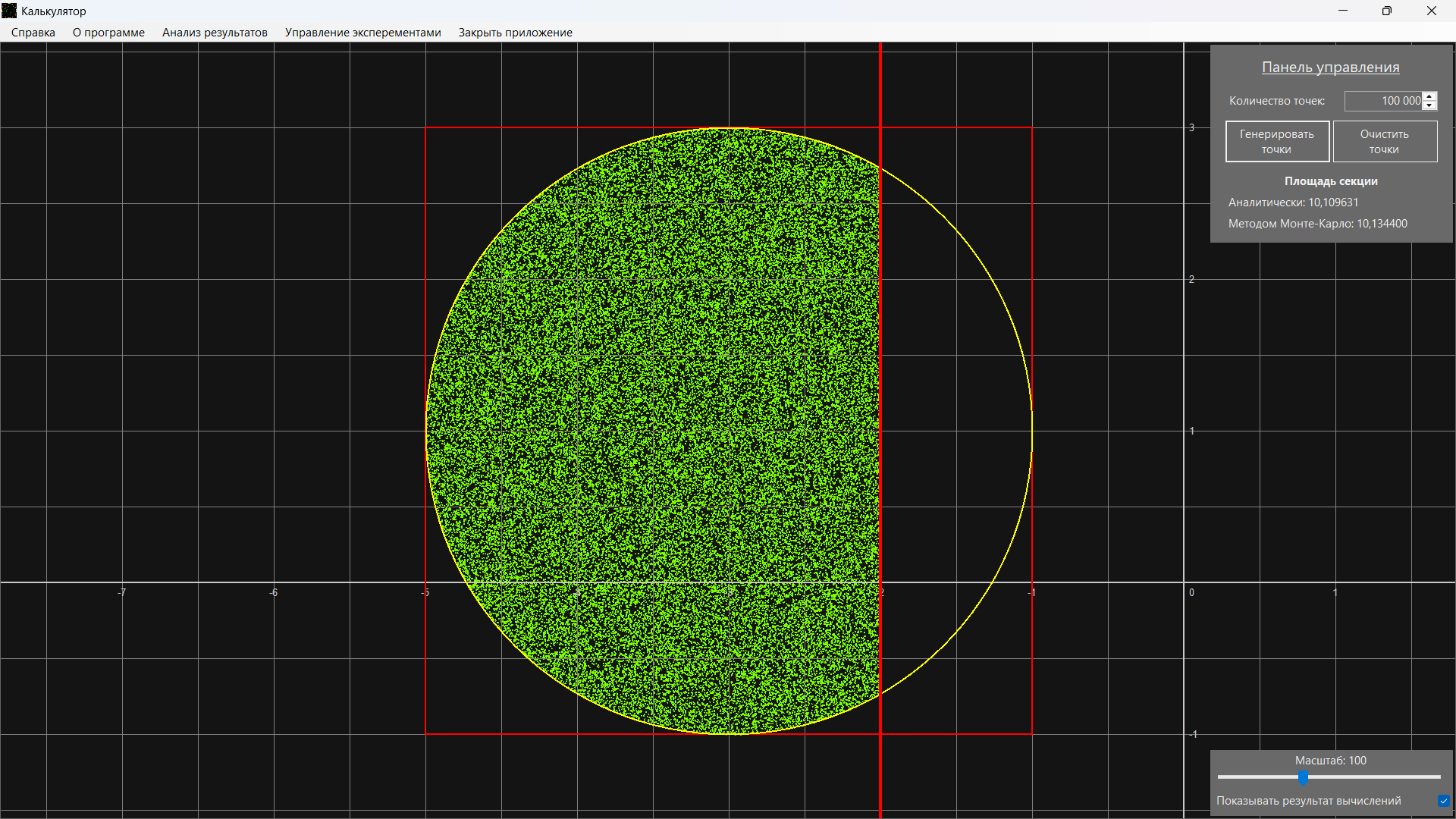


Рисунок В2 – Основное окно программы

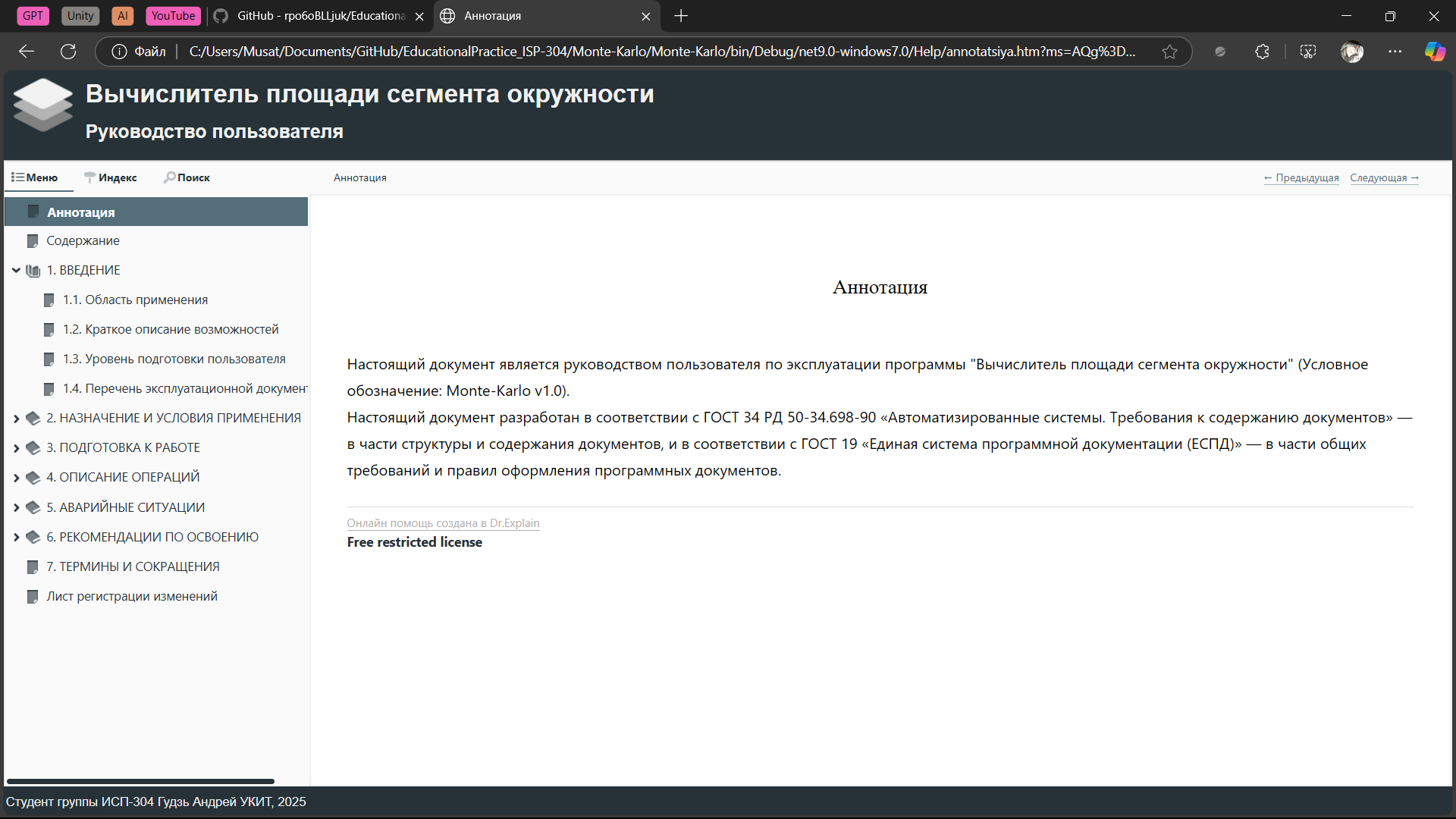


Рисунок В3 – справка, открываемая в браузере



Рисунок В4 – окно «О программе»

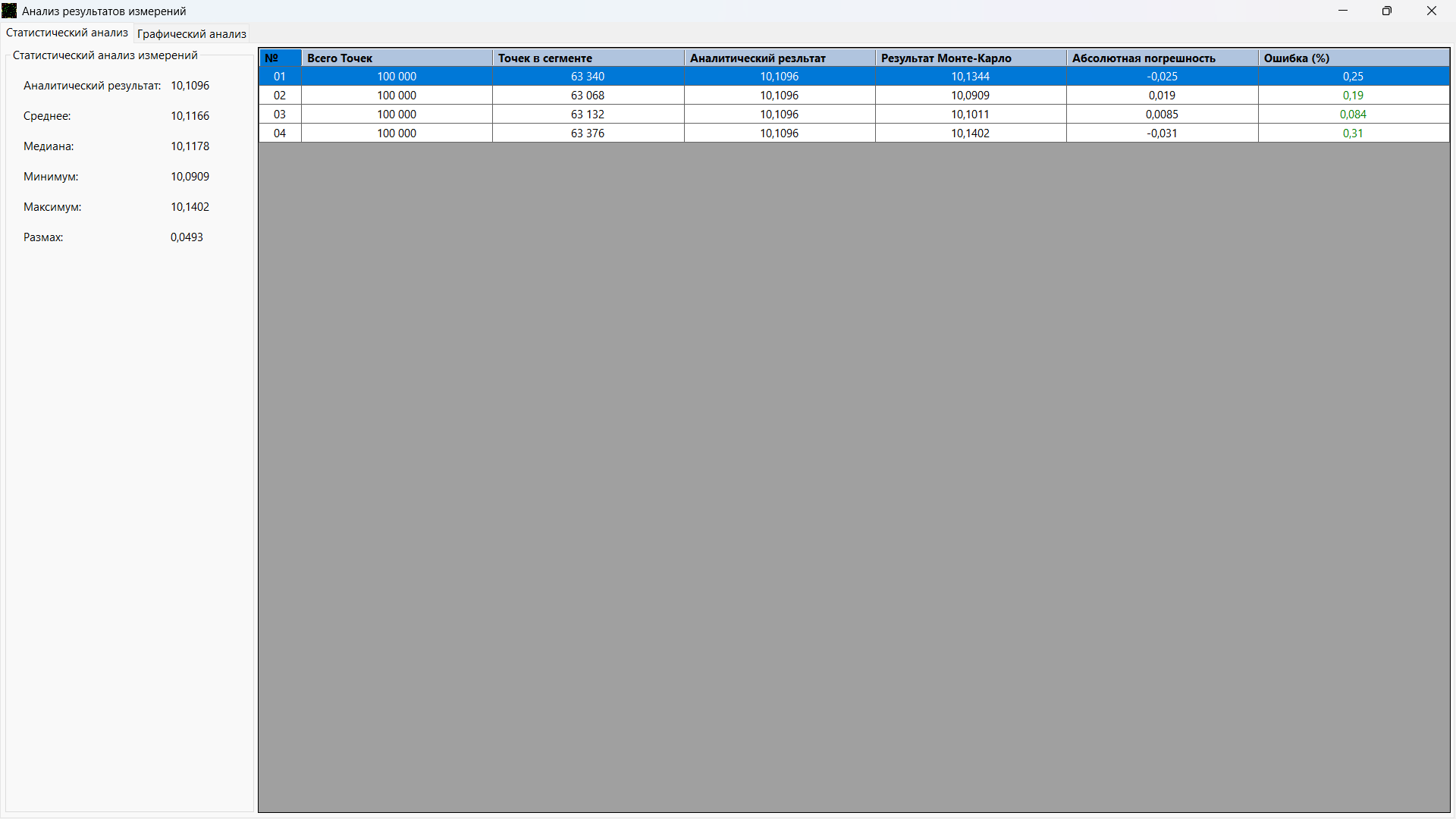


Рисунок В5 – вкладка статистического анализа в окне анализа результатов измерений

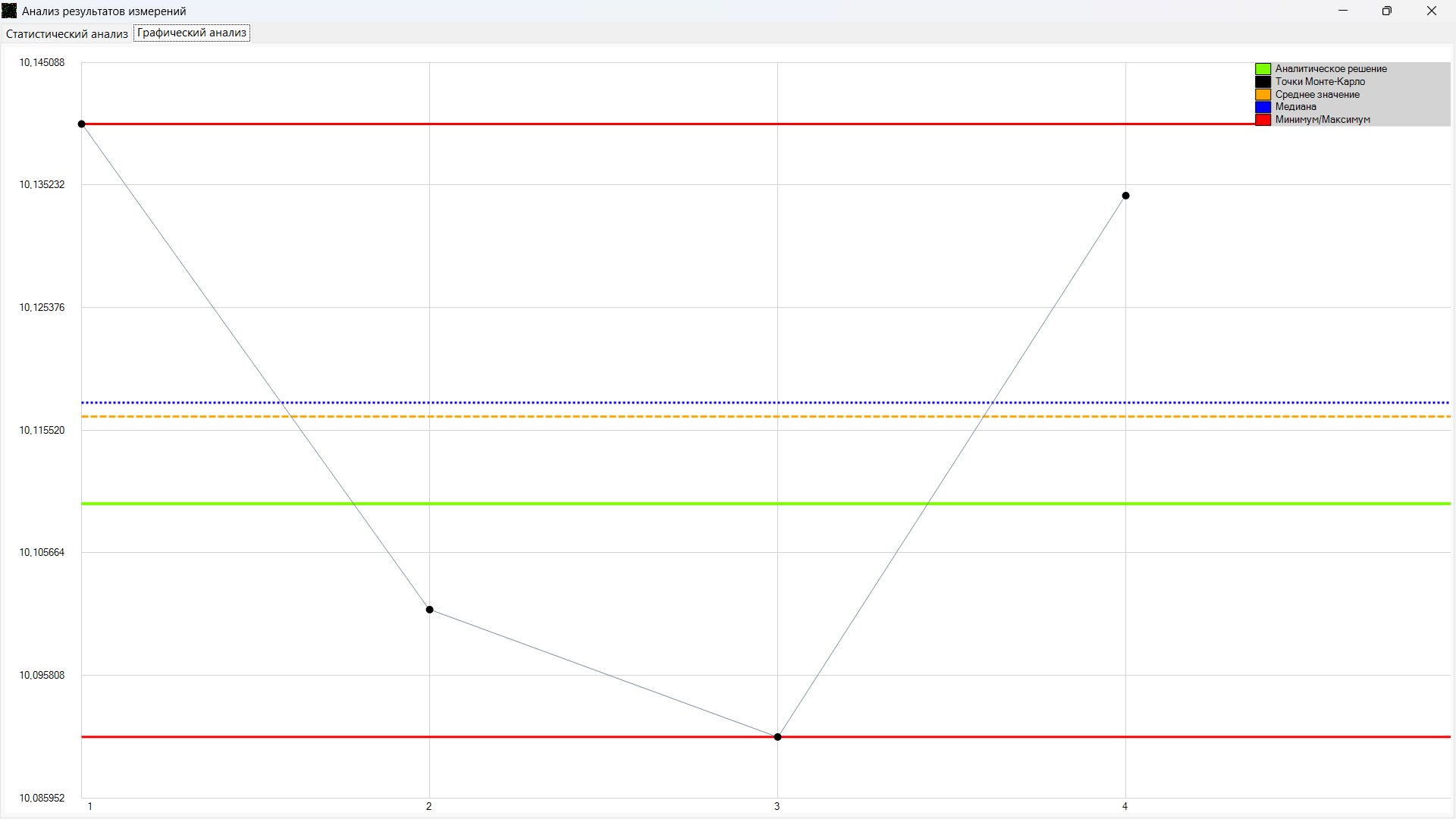


Рисунок В6 – вкладка графического анализа в окне анализа результатов измерений

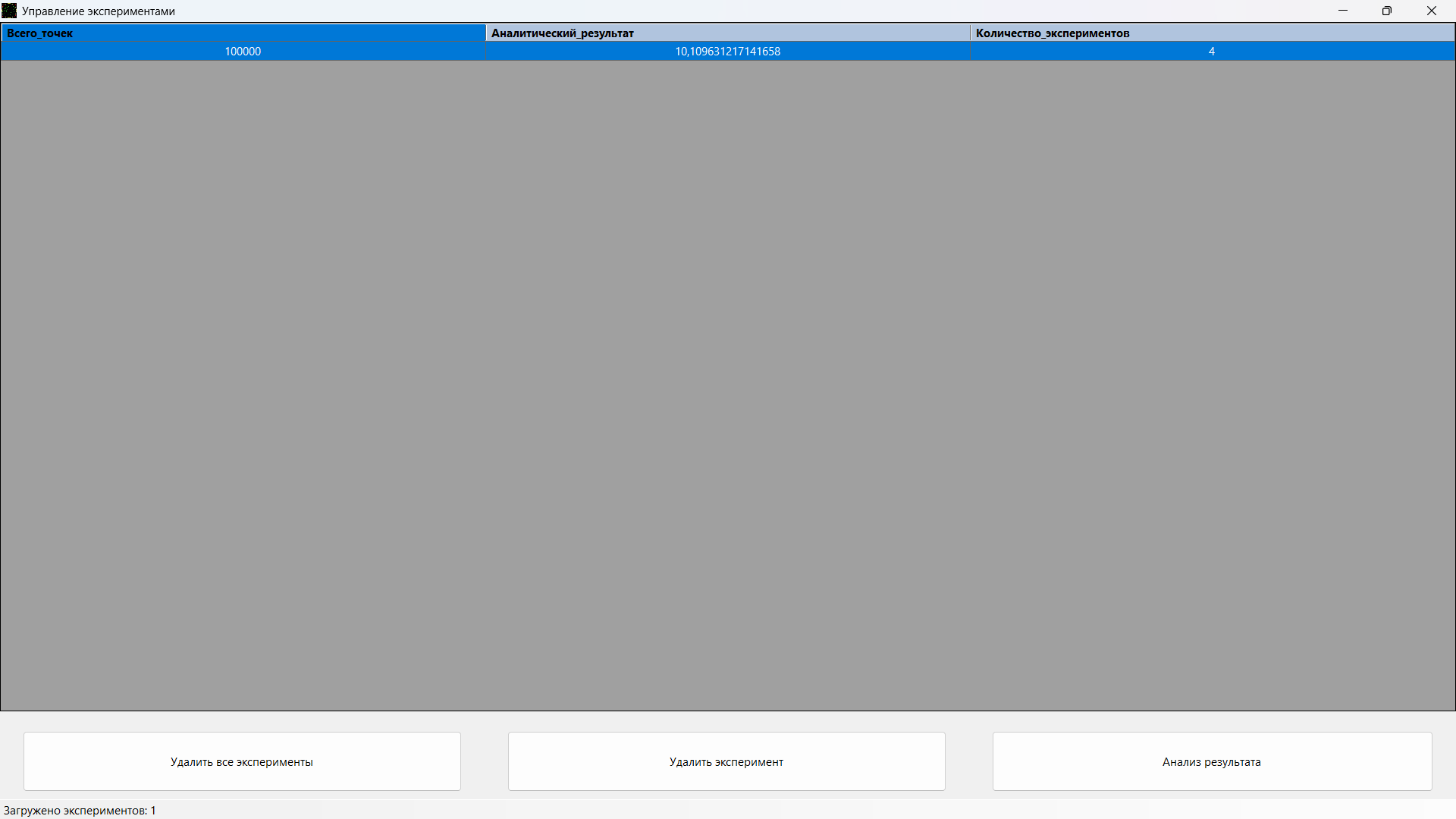


Рисунок В7 – окно управления экспериментами