Приложение В30

Приложение Г32

Приложение Д34

Приложение Е36

Приложение Ж38

Приложение И40

Приложение К42

Приложение Л47

**Введение**

На учебной практике была поставлена задача разработать программный сервис для решения задач по линейному программированию, включающий в себя графический метод, симплекс-метод и транспортную задачу.

Данный проект направлен на создание приложения, которое позволит пользователям эффективно решать задачи линейного программирования, что значительно облегчит выполнение сложных вычислений и анализ данных. Основная цель проекта заключается в разработке удобного и интуитивно понятного инструмента, способного выполнять точные и надёжные расчёты.

Разрабатываемое приложение ориентировано на широкий круг пользователей, включая студентов, преподавателей в области математического моделирования и оптимизации. Программный сервис будет полезен как для учебных целей, так и для практического применения в различных областях науки и техники, предоставляя возможность углубленного анализа и решения задач.

Приложение будет интересно всем, кто занимается изучением методов линейного программирования и их практическим применением, а также тем, кто стремится повысить свои навыки в данной области.

Далее приведено краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ задачи». В нем можно ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, инструменты разработки (будет рассмотрена среда, в которой создается данный проект), диаграмму Ганта и выбор модели жизненного цикла программного обеспечения. Также в этом разделе можно узнать о том, как данная задача решается в настоящее время. Все входные и выходные данные тоже будут описаны в первом разделе.

В разделе «Проектирование задачи» будут рассмотрены основные аспекты разработки десктопного приложения. Здесь можно узнать об организации данных в контексте среды разработки. В данном разделе будет описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации.

«Реализация» – это третий раздел отчёта, в котором описываются все элементы и объекты, которые будут использованы при реализации данного приложения.

Четвёртый раздел – «Тестирование». В нём будет описано полное и функциональное тестирование данной программы. Будут смоделированы все возможные действия пользователя при работе с приложением, начиная от открытия приложения и заканчивая его закрытием.

В разделе «Руководство пользователя» будет описано назначение, область применения, среда функционирования данного программного продукта.

«Заключение» будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств.

В разделе «Список использованных источников» будет приведён список используемых при разработке источников.

В приложении А представлена диаграмма вариантов использования.

В приложении Б представлена диаграмма Ганта.

В приложении В представлена диаграмма последовательности графического метода.

В приложении Д представлена диаграмма последовательности симплекс метода.

В приложении Е представлена диаграмма последовательности графического метода.

В приложении Ж представлен UX прототип приложения.

В приложении И представлен UI прототип приложения.

В приложении К представлено функциональное тестирование.

В приложении Л представлен листинг программы.

1. **Анализ задачи**
   1. **Постановка задачи**
      1. **Организационно-экономическая сущность задачи**

Наименование задачи: Разработка программного сервиса для решения задач по линейному программированию.

Цель разработки: Создание приложения для выполнения вычислений, связанных с математическим моделированием.

Назначение: Данный программный продукт предназначен для преподавателей, занимающихся математическим моделированием, и позволяет им эффективно проводить вычисления.

Программный продукт будет использоваться по мере необходимости.

Источники и способы получения данных: ввод пользователем.

* + 1. **Функциональные требования**

К поставленной задаче были заявлены следующие функциональные требования, которые может выполнять гость:

* Построение графиков
* Построение таблиц
* Построение матриц
* Просмотр файлов
  + 1. **Описание входной, выходной и условно-постоянной информации**

Входной информацией являются вводимые пользователем данные.

Выходной информацией является результаты вычислений, представленные в текстовом или графическом виде.

Условно-постоянной информацией являются значения настроек приложения.

* + 1. **Нефункциональные (эксплуатационные) требования**

Требования к применению: удобный, интуитивный пользовательский интерфейс, функциональным и простым в использовании. Он не должен перегружать пользователя ненужной информацией.

Требования к производительности: приложение должно обеспечивать высокую точность и надёжность вычислений. Для повышения производительности необходима стабильная работа без сбоев и перебоев.

Требования к реализации: соответствие современным стандартам разработки приложения.

Требования к надежности: Программное обеспечение должно обеспечивать точность и надёжность вычислений, а также стабильную работу без сбоев и перебоев.

Требования к интерфейсу:

* главная страница: меню навигации с основными разделами и настройками приложения;

1. секция калькуляторов: калькуляторы для решения задач по математическому моделированию;
   1. **Диаграмма вариантов использования**

Диаграмма вариантов использования – диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

В разрабатываемом проекте можно выделить одного актёра: пользователь. Для пользователя доступны следующие варианты использования: просмотр теоретического материала, использование калькуляторов, ввод данных, выбор операций, выбор темы приложения.

Разработанная диаграмма вариантов использования представлена в приложении А.

* 1. **Разработка плана работы над проектом**

Диаграмма Ганта предназначена для визуализации графика выполнения задач в проекте. Она помогает планировать проект, определяя последовательность и продолжительность задач, а также позволяет визуализировать сроки начала и завершения каждой из них. Этот инструмент способствует оптимизации распределения ресурсов, выявлению зависимостей между задачами и мониторингу прогресса выполнения. Кроме того, диаграмма Ганта улучшает коммуникацию, позволяя представлять информацию о проекте заинтересованным сторонам. Созданная диаграмма представлена в приложении Б.

* 1. **Выбор стратегии разработки и модели жизненного цикла**

Таблица 1 Выбор модели жизненного цикла на основе характеристик требований.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Да | Нет | Да | Нет |
| 2. | Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Нет | Нет | Нет |
| 3. | Часто ли будут изменяться требования на протяжении ЖЦ? | Нет | Нет | Нет | Нет | Да | Да |
| 4. | Нужно ли демонстрировать требования с целью их определения? | Нет | Нет | Нет | Да | Да | Да |
| 5. | Требуется ли проверка концепции программного средства или системы? | Нет | Да | Да | Да | Нет | Да |
| 6. | Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? | Нет | Нет | Да | Да | Да | Да |
| 7. | Нужно ли реализовать основные требования на ранних этапах разработки? | Да | Да | Нет | Нет | Нет | Нет |

Вычисления: 4 за каскадную, 5 за V- образную, 7 за RAD, 4 за инкрементную, 3 за быстрого прототипирования и 3 за эволюционную.

Итог: На основе результатов заполнения табл. 3, подходящей является RAD модель.

Таблица 2 **–** Выбор модели жизненного цикла на основе характеристик команды разработчиков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории команды разработчиков  проекта | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | Да | Да |
| 2. | Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? | Нет | Нет | Да | Да | Да | Да |
| 3. | Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 4. | Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да | Нет | Нет | Нет | Нет |
| 5. | Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да | Нет | Нет |
| 6. | Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да | Да | Да | Да | Да |

Вычисления: 5 за каскадную, 5 за V-образную, 5 за RAD, 4 за инкрементную, 2 за быстрого прототипирования и 2 за эволюционную.

Итог: На основе результатов заполнения табл. 4 подходящими являются каскадная и v-образная и RAD модели.

Таблица 3 **–** Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории коллектива пользователей | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да | Да | Нет | Нет | Нет |
| 2. | Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? | Нет | Нет | Нет | Да | Да | Да |
| 3. | Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет | Нет | Да | Да | Да |
| 4. | Будет ли заказчик отслеживать ход выполнения проекта? | Да | Да | Да | Да | Да | Да |

Вычисления: 4 за каскадную, 4 за V-образную, 4 за RAD, 1 за инкрементную, 1 за быстрого прототипирования и 1 за эволюционную.

Итог: На основе результатов заполнения табл. 5 подходящей является каскадная, V-образная и RAD модели.

Таблица 4 **–** Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории типов проекта и рисков | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Разрабатывается ли в проекте продукт нового для организации направления? | Нет | Нет | Нет | Да | Да | Да |
| 2. | Будет ли проект являться расширением существующей системы? | Да | Да | Да | Да | Нет | Нет |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Продолжение таблицы 4 | | | | | | | |
| 3. | Будет ли проект крупно- или среднемасштабным? | Нет | Нет | Нет | Да | Да | Да |
| 4. | Ожидается ли длительная эксплуатация продукта? | Да | Да | Нет | Да | Нет | Да |
| 5. | Необходим ли высокий уровень надежности продукта проекта? | Нет | Да | Нет | Да | Нет | Да |
| 6. | Предполагается ли эволюция продукта проекта в течение ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7. | Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Нет | Да | Да | Да |
| 8. | Является ли график сжатым? | Нет | Нет | Да | Да | Да | Да |
| 9. | Предполагается ли повторное использование компонентов? | Нет | Нет | Да | Да | Да | Да |
| 10. | Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? | Нет | Нет | Нет | Нет | Да | Да |

Вычисления: 4 за каскадную, 5 за V-образную, 5 за RAD, 5 за инкрементную, 5 за быстрого прототипирования и 7 за эволюционную.

Итог: На основе результатов заполнения табл. 6 подходящей является эволюционная модель.

Общий итог: Результаты заполнения табл. 3-6: каскадная – 17, V-образная – 19, RAD – 21, Инкрементная -14, Быстрого прототипирования – 11, Эволюционная – 13, в итоге заполнения табл. 3 – 6 наиболее подходящей является RAD модель.

* 1. **Инструменты разработки**

Для разработки данного проекта будет выбрана среда разработки Visual Studio, которая является наиболее актуальной средой для создания приложений данного типа.

Разработка будет производится на языке программирования, как: С# - язык программирования был выбран потому что является ООП языком и обладает такими необходимыми качествами как стабильность, адаптивность, производительность, совместимость, а так же понятность и простота. Данный язык обладает необходимым инструментарием, в приложении будет использоваться WinForms.

Иные инструменты, используемые при разработке и написании сопутствующей документации:

* WEB-ресурс DRAW.IO – будет использоваться для создания графической части и разработки UML-диаграмм;
* Microsoft Office Word – для написания документации к программному продукту;
* Microsoft Office Project – для создания диаграммы Ганта;
* Figma – для постройки UX-UI прототипов;
* Xmind – для создания структуры сайта;
* OperaGX – для поиска информации.

Разработка проекта будет происходить на компьютере со следующими параметрами:

− процессор Intel® Core ™ i3-7100U CPU @ 2.40GHz;

− объем оперативной памяти 16.00 GB;

− объем места на жестком диске 256 GB;

− видеокарта Intel(R) HD Graphics 620;

− ОС Windows 10 Pro.

**2 Проектирование задачи**

1. **Разработка главного меню**

Эффективная система меню и продуманная система навигации являются основой удобства использования приложения и влияют на впечатление пользователей о нем. Грамотная организация меню и навигационных элементов помогает пользователям быстро находить нужную информацию, легко перемещаться между разделами и страницами, а также улучшает восприятие контента. На этапе разработки структуры важно учитывать потребности целевой аудитории, принципы юзабилити, а также технические особенности проекта, чтобы создать логичную и интуитивно понятную навигацию.

Главное меню представлена на рисунке 1.

Рисунок 1 – Главное меню

1. **Разработка UML-диаграмм**
2. **Диаграмма деятельности**

В первом разделе «1.1 Постановка задачи» были описаны функциональные требования к проекту. На их основе построена диаграмма последовательности. Диаграмма последовательности – это диаграмма, предназначенная для моделирования взаимодействия объектов системы во времени, а также обмена сообщениями между ними. Графические диаграммы последовательности имеют два представления. Одно – слева направо – в виде вертикальных линий, которые соответствуют линии жизни отдельного участка взаимодействия. Второе –вертикальная временная ось, направленная сверху вниз. На диаграмме отображен процесс регистрации. Диаграмма последовательности представлена в Приложении В.

1. **Диаграмма компонентов**

Диаграмма объектов – это диаграмма, предназначенная для моделирования статических связей между объектами системы в конкретный момент времени. В отличие от диаграммы классов, которая отображает классы и их связи в общем виде, диаграмма объектов фокусируется на конкретных экземплярах классов и их связях, показывая состояние системы в определенной ситуации. Связи между объектами показывают, как взаимодействуют конкретные экземпляры классов в ходе выполнения сценария. На диаграмме изображён процесс оформления заказа. Диаграмма объектов представлена в Приложении Г.

1. **Разработка пользовательского интерфейса**

Важным элементом проектирования данного программного продукта является описание внешнего интерфейса разрабатываемого продукта.

Для разработки визуального дизайна использовались сдержанные, мягкие цвета для удобства использования программного продукта. В ходе разработки были спроектированы дизайны всех страниц форм, как видимых пользователю, так и невидимых. Ранее разработанная главное меню расположена на рисунке 1.

Для организации эффективной работы пользователя нужно создать целостный программный продукт данной предметной области, в котором все компоненты будут сгруппированы по функциональному назначению. Все исходные данные будут разделены на несколько групп.

Прототип – это наглядная модель пользовательского интерфейса. В сущности, это «черновик», созданный на основе представления разработчика о потребностях пользователя. Итоговое отображение программы может отличаться от прототипа.

Прототипы UX представлены в приложении Ж, ссылка на UX: https://www.figma.com/design/L7zDbQoEBmB41Ngp29qjwk/Untitled?node-id=0-1&t=gQ8NPJjfGd8Hsna4-1. Прототипы UI представлены в приложении И, ссылка на UI: https://www.figma.com/design/SQmozLG6KwT8T7ZSBwbP99/Untitled?node-id=0-1&t=bPMWtpDuAGLiPWR2-1.

1. **Реализация**
   1. **Руководство программиста**

Программный продукт разработан с использованием среды разработки Visual Studio, которая является наиболее актуальной средой для создания приложений данного типа, обладает необходимым инструментарием, в приложении будет использоваться WinForms. Для разработки проекта будет выбран язык программирования С#, который является ООП языком и обладает такими необходимыми качествами как стабильность, адаптивность, производительность, совместимость, а так же понятность и простота.

1. **Создание приложения**

Данный курсовой проект содержит 7 модулей. Далее рассмотрим назначе-ние каждого модуля:

− Модуль MainMenu – Главное меню;

− Модуль Viewer – компонент просмотра PDF файлов;

− Модуль Settings – меню настроек;

− Модуль RoundedComponents – вспомогательные компоненты;

− Модуль Chart – график;

− Модуль SolvingGraphProblem – меню калькулятора;

− Модуль MainScreen – меню калькулятора;

− Модуль TransportProblemSolving – меню калькулятора;

− Модуль StepOfSolving – вспомогательны функции;

Таблица 5 – Классы

|  |  |  |
| --- | --- | --- |
| Имя класса | В каком модуле находится | Назначение |
| 1 | 2 | 3 |
| RoundedPanel | RoundedComponents.cs | Использование вместо стандартных компонентов |
| CustomButton | RoundedComponents.cs | Использование вместо стандартных компонентов |
| ImageButton | RoundedComponents.cs | Использование вместо стандартных компонентов |
| RoundedLabelWithShadows | RoundedComponents.cs | Использование вместо стандартных компонентов |
| Chart | Chart.cs | Рисовка графика |

|  |  |  |
| --- | --- | --- |
| Продолжение таблицы 5 | | |
| Constraint | Constraint.cs | Ограничения |
| Function | Function.cs | Представление ЦФ |
| Problem | Problem.cs | Данные задачи |
| SimplexSnap | SimplexSnap.cs | Преобразование данных |
| ProblemsService | ProblemsService.cs | Дефолтные данные |
| Simplex | Simplex.cs | Решение задачи |

Описание использованных компонентов приводится в таблице 6.

Таблица 6 – Использованные компоненты

|  |  |  |
| --- | --- | --- |
| Компонент | На какой форме расположен | Назначение |
| 1 | 2 | 3 |
| RoundedPanel | MainForm, Viewer, Settings | Используется как фон, тени |
| CustomButton | MainForm, Viewer, Settings | Используется для взаимодействия с пользователем |
| ImageButton | MainForm, Viewer, Settings | Используется для взаимодействия с пользователем |
| RoundedLabelWithShadows | MainForm, Viewer, Settings | Используется для отображения информации |
| TabControl | TransportProblemSolving | Используется для разделения решения на шаги |
| Button | TransportProblemSolving, MainScreen, GraphMethod | Используется для взаимодействия с пользователем |
| TextBox | TransportProblemSolving, MainScreen, GraphMethod | Используется для взаимодействия с пользователем |
| Label | TransportProblemSolving, MainScreen, GraphMethod | Используется для отображения информации |
| ComboBox | TransportProblemSolving, MainScreen, GraphMethod | Используется для взаимодействия с пользователем |

1. **Тестирование**
   1. **Тесты на использование**

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения испытания реализации программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Функциональное тестирование представлено в приложение К.

Расписание работ над проектом представлен в таблице 5.

Статистика по дефектам представлена в таблице 6.

* 1. **Отчет о результатах тестирования**

Таблица 5 Расписание тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Дата | Деятельность | Продолжительность ч. |
| Петько Алексей | 01.12.2024 | Разработка тест-кейсов | 6 |
| Петько Алексей | 01.12.2024 | Проведение тестирования | 6 |
| Петько Алексей | 01.12.2024 | Написание отчётов о дефектах | 6 |
| Петько Алексей | 01.12.2024 | Проведение регрессионного тестирования | 1 |
| Петько Алексей | 01.12.2024 | Написание отчёта о результатах тестирования | 1 |

Таблица 6 Статистика по дефектам

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Важность | | | |
| Статус | Количество | Низкая | Средняя | Высокая | Критическая |
| Найдено | 4 | 1 | 1 | 2 | 0 |
| Исправлено | 3 | 1 | 1 | 1 | 0 |
| Проверено | 4 | 1 | 1 | 1 | 0 |
| Открыто заново | 0 | 0 | 0 | 0 | 0 |
| Отклонено | 0 | 0 | 0 | 0 | 0 |

1. **Руководство пользователя**

Цель проекта заключается в создании десктопного приложения, которое предлагает калькуляторы для графического симплекс-метода и решения транспортной задачи. Приложение должно быть удобным и интуитивно понятным для пользователей, обеспечивая быстрый доступ к инструментам и результатам вычислений.

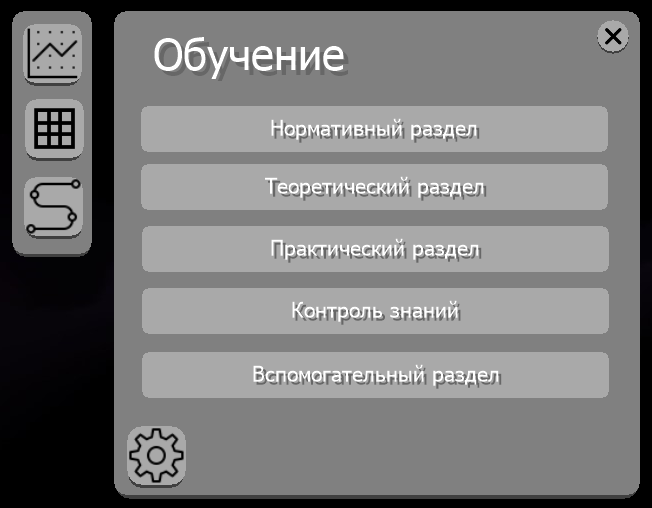
После открытия приложения открывается главное меню, которое представлено на рисунке 2.

Рисунок 2 – Главное меню

Меню приложения состоит из следующих пунктов:

* Кнопок навигации
* Калькулятор по графическому методу
* Калькулятор по симплекс методу
* Калькулятор по транспортной задаче
* Настройки

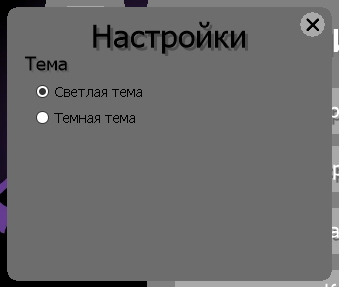
Настройки приложения представлены на рисунке 3.

Рисунок 3 – Настройки

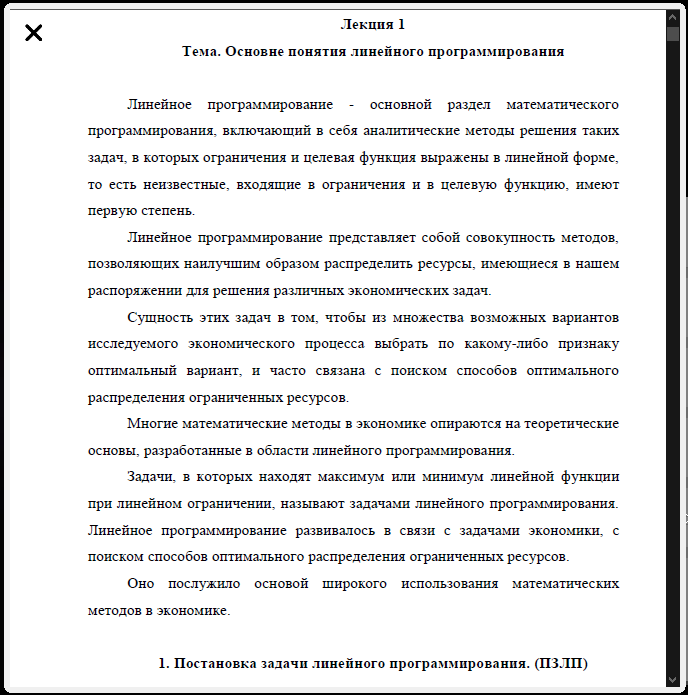
Кнопки навигации приводят пользователя к теоретическому материалу по математическому моделирования. Окно с теоретическим материалом представлено на рисунке 4

Рисунок 4 – Страница «Магазин»

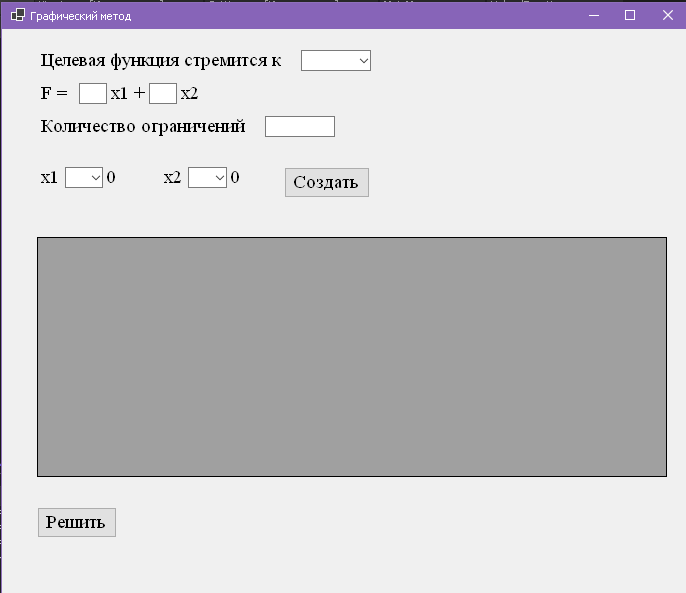
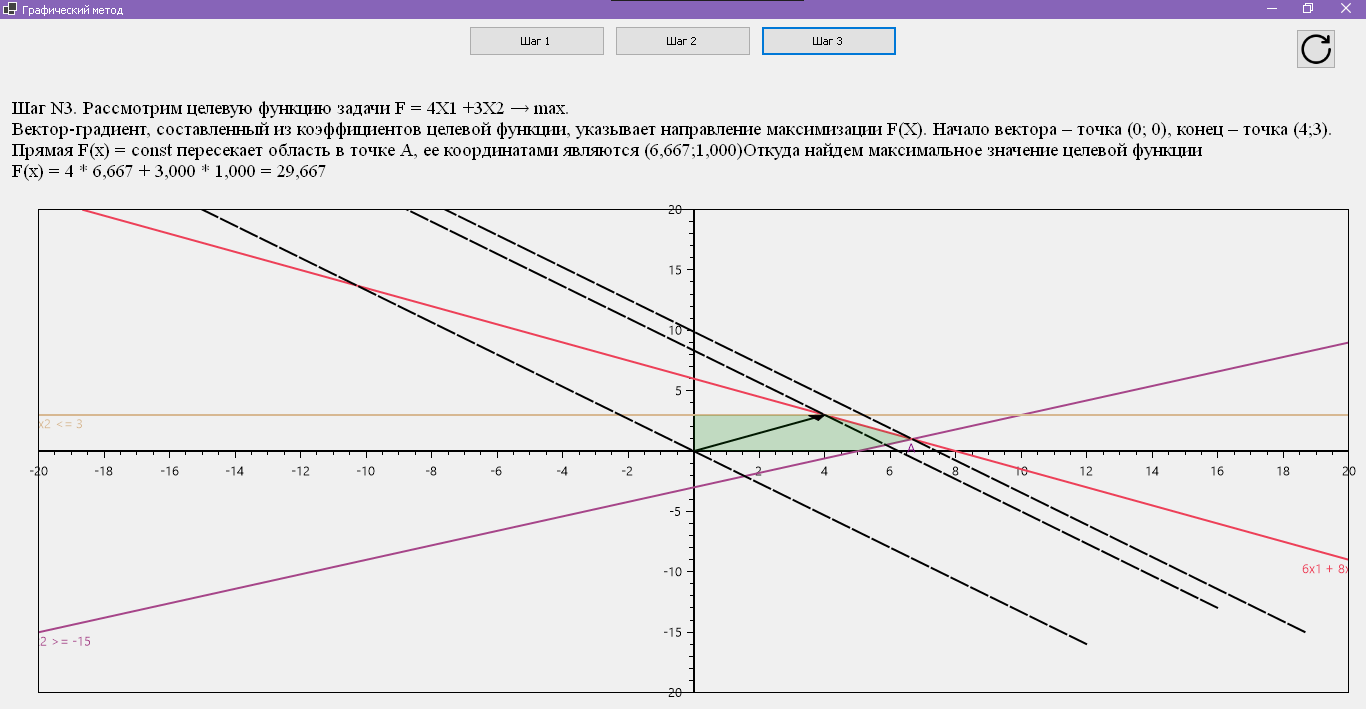
Далее нажав в меню на кнопку «Графический метод» пользователь попадёт на форму для ввода данных задачи с последующим её решением. Форма «Графический метод» представлена на рисунке 5.

Рисунок 5 – Форма «Графический метод»

Далее введя данные и нажав на кнопку «Решить», на форму появится решение задачи графическим методом. Форма с решением ЗЛП представлена на рисунке 6.

Рисунок 6 – Форма «Графический метод»

Далее перейдя назад в меню, предварительно закрыв калькулятор, пользователь, нажав на кнопку «Симплекс метод», попадёт на форму с вводом данных и последующим решением задачи симплекс методом. Форма «Симплекс метод» представлена на рисунке 7.

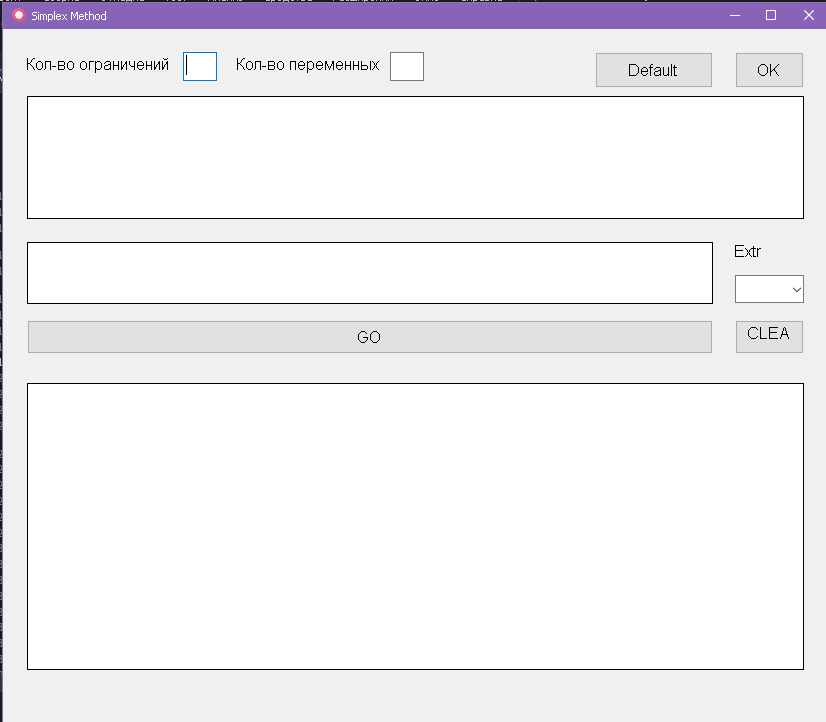


Рисунок 7 – Форма «Симплекс метод»

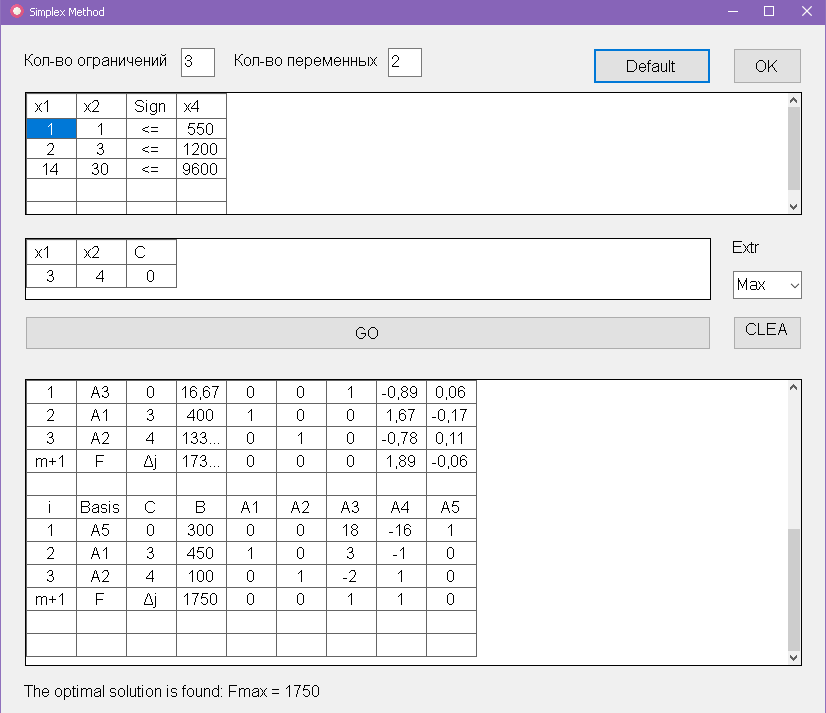
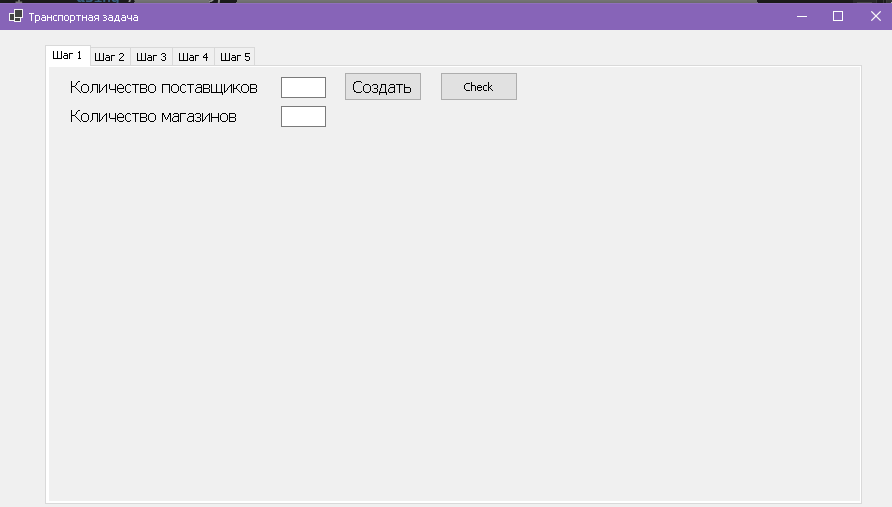
Далее введя данные и нажав на кнопку «Go», на форме появится решение задачи симплекс методом. Форма с решением ЗЛП представлена на рисунке 8.

Рисунок 8 – Форма «Симплекс метод»

Далее нажав в главном меню «Транспортная задача», пользователь попадёт на форму, где он может посмотреть решение транспортной задачи. Страница «Транспортная задача» представлена на рисунке 9.

Рисунок 9 – Форма «Транспортная задача»

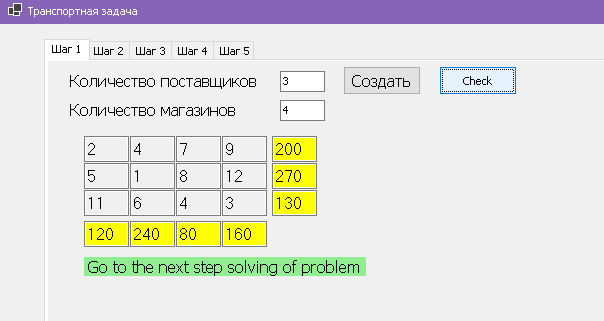
Далее введя данные и нажав на кнопку «Создать», на форме появится матрица распределения, которую надо заполнить, с последующим решением транспортной задачи. Форма с первым шагом решения представлена на рисунку 10.

Рисунок 10 -Форма «Транспортная задача»

**Заключение**

Целью нашего проекта было создание учебного десктопного приложения, которое предоставляет калькуляторы для графического, симплекс-метода и решения транспортной задачи. Приложение разработано с целью облегчить изучение и применение математического моделирования, предоставляя удобный и интуитивно понятный интерфейс для пользователей.

В процессе разработки были проанализированы требования к проекту, разработан структурный и визуальный дизайн, а также реализованы функциональные возможности для решения задач графическим, симплекс-методом и транспортной задачи. Приложение позволяет пользователям легко вводить данные, просматривать результаты расчетов, а также сохранять их для дальнейшего анализа. Эти функции способствуют упрощению учебного процесса и повышают эффективность изучения математического моделирования.

Приложение представляет собой многофункциональную и интуитивно понятную платформу, которая способствует расширению знаний в области математического моделирования и укреплению навыков пользователей. Подход к разработке, основанный на современных технологиях, позволяет обеспечить стабильность работы приложения, защиту данных и адаптивность для различных устройств, что гарантирует положительный опыт пользователей.

Реализация проекта способствует не только автоматизации и упрощению процесса обучения, но и укреплению имиджа учебного заведения или компании как надежного и инновационного партнера, готового к внедрению современных решений для удовлетворения образовательных потребностей.

.

**Список используемых источников**

1. Леонова, Задачи линейного программирования и методы их решения
2. Matecos [Электронный ресурс] – Электронные данные. ­– Режим доступа: <https://matecos.ru/mat/matematika> – Дата доступа 25.11.2024;
3. Math [Электронный ресурс] – Электронные данные. ­– Режим доступа: <https://math.semestr.ru/lp> – Дата доступа 06.12.2024;
4. Programforyou [Электронный ресурс] – Электронные данные. ­– Режим доступа: https://programforyou.ru/calculators/simplex-method – Дата доступа 01.12.2024;