Учреждение образования

«Гродненский государственный политехнический колледж»

Специальность: 2 – 40 01 01 «Программное обеспечение информационных

технологий»

Специализация: 2 – 40 01 01 35 «Программное обеспечение обработки

экономической и деловой информации»

Дисциплина: «Технология разработки программного обеспечения»

Группа: ПЗТ – 39

**ТЕХНО-РАБОЧИЙ ПРОЕКТ**

Техническое задание на разработку электронной книга «Тригонометрия»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Разработал И.С.Стручинский

Руководитель проекта О.И. Кизер

2023

**Содержание**

Изм.

Кол

Лист №док

Подпись

Дата

Лист

48

ТЗ 2-40 01 01.33.39.21.23 ТРПО

Разраб.

Стручинский

Пров.

Кизер

Н. контр.

Утв.

Техническое задание на разработку Электронной книги «Тригонометрия»

Стадия

Листов

УО ГГПК

1 Постановка задачи 3

1.1Организационно-экономическая сущность задачи 3

1.2.Функциональные требования 3

1.3 Эксплуатационные требования 6

1.4 Инструменты разработки 7

[2 Проектирование задачи](#_Достоинства_и_недостатки) 9

[2.1](#_Toc406878086) Разработка структуры сайта, системы меню, навигации 9

[2.2](#_Toc406878088) Организация данных 9

2.3 Разработка UML-диаграмм 9

2.3.1 Диаграмма последовательности 9

2.3.2 Модель данных 9

2.4 Диаграмма вариантов использования 9

2.5 Диаграмма классов 10

2.6 Диаграмма деятельности 10

2.7 Диаграмма коопирации (объектов) 10

2.8 Моделирование бизнес-процессов 10

## 2.9 Структурное проектирование 11

[3 Реализация](#_Достоинства_и_недостатки) 12

3.1 Руководство программиста 12

3.2 Спецификация программы 16

4 Тестирование 18

[4.1](#_Toc406878088) Тесты на использование 18

[4.2](#_Toc406878086) Отчет о результатах тестирования 20

[5](#_Toc406878086) Руководство пользователя 21

Заключение 22

Список использованных источников 23

[Приложение А Диаграмма последовательности](#_Toc406878088)  24

Приложение Б Модель данных 26

Приложение В Диаграмма вариантов использования 28

Приложение Г Диаграмма классов 30

Приложение Д Диаграмма деятельности 32

Приложение Е Диаграмма коопирации (объектов) 34

Приложение Ж Моделирование бизнес-процессов 36

Приложение 3 Структурное программирование ……38

1. **Постановка задачи**

**1.1Организационно-экономическая сущность задачи**

Наименование задачи: Электронная книга «Тригонометрия»;

Цель разработки: создание электронной книги для получения и поддержки уже имеющихся знаний ;

Назначение: данный программный продукт будет является информационным контентом для ознакомления с основами тригонометрии, для решения практических задач.

Периодичность использования: при желании.

**1.2.Функциональные требования**

Описание перечня функций и задач, которые должен выполнять будущий ПП:

Пользователь: все те же функции, что может выполнять гость, но так же появляется:

1. авторизация
2. редактировать профиль
3. поиск темы по названию
4. выполнение заданий
5. Выбор и поиск заданий на выполнение
6. чтение книги
7. просмотр содержания
8. просмотр тем
9. просмотр выполненных заданий
10. сделать закладку

Администратор:

1. авторизация

2. Редактирование разделов книги

3. Добавление нового материала/заданий в книгу

Описание процессов с входной, выходной и условно-постоянной информацией представлены в таблице 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| № | Категория пользователей | Наименование процесса | Краткое описание алгоритма  выполнения процесса | Входная информация | Выходная информация | Условно-постоянная  информация |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  | Пользователь | Авторизация | Ввод имени и пароля | Имя,пороль | Профиль | Отсутствует |
|  | Пользователь | редактирование профиля | Редактируется профиль пользователя | Имя ,пороль,класс | Обновленная информация | Отсутствует |
|  | Пользователь | Поиск темы по названию | Поиск темы | тема | тема | Отсутствует |
|  | Пользователь | Выбор и поиск заданий на выполнение | Задания расположены после тем | Номер задания | Текст заданмя | Отсутствует |
|  | Пользователь | Выполнение заданий | Изначально необходимо найти тему | Задание | Отметка | Отсутствует |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Таблица 1-продолжение | | | | | | |
|  | Пользователь | Чтение книги | Пользователь читает какую-либо тему | Отсутствует | Материал | Отсутствует |
|  | Пользователь | просмотр содержания | Пользователь просматривает содержание | Отсутствует | содержание | Отсутствует |
|  | Пользователь | просмотр тем | Пользователь просматривает  уже прочитанные темы в содержании | Отсутствует | Список тем | Отсутствует |
|  | Пользователь | просмотр выполненных заданий | Пользователь просматривает  уже выполненные задания в содержании | Отсутствует | содержание с выполненными заданиями | Отсутствует |
|  | Пользователь | сделать закладку | Пользователь может сделать закладку на той теме, где остановился | Место в теме | содержане  (пользователь видит свою закладку в содержании) | Отсутствует |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Таблица 1-продолжение | | | | | | |
|  | Администратор | Авторизация | Ввод имени и пароля | Имя,пороль | Профиль | Отсутствует |
|  | Администратор | Редактирование разделов книги | Администратор редактирует разделы в книге | Раздел | Отредактированный раздел | Отсутствует |
|  | Администратор | Добавление нового материала/заданий в книгу | Администратор добавляет информацию/заданий в разделы/темы | разделы/темы | Отредактированные  разделы/темы | Отсутствует |

**1.3.Эксплуатационные требования**

Требования к применению:

Требования к реализации: Для реализации статических страниц и шаблонов должны использоваться языки HTML и CSS. Для реализации интерактивных элементов клиентской части должны использоваться языки C#. Для реализации динамических страниц должен использоваться язык PHP.

Требования к надежности: Система может быть недоступна не более чем 24 часа в год. У администратора сайта должна быть возможность выгрузить и загрузить копию сайта.

Требования к интерфейсу**:** При разработке книги должны быть использованы преимущественного желтые/светло-коричневые оттенки. Основные разделы книги должны быть доступны с первой страницы. После регистрации перед пользователем появляется его личный кабинет. В личном кабинете будет кнопка «Читать книгу». Грамотный пользовательский интерфейс. Слева будет отображаться содержание книги. Закладки будут отображаться в содержании значком. Книга должна адаптироваться под компьютер.

Требования к хостингу:

– Поддерживает 1С-Битрикс, WordPress, Joomla!, Drupal и любые другие CMS

|  |  |
| --- | --- |
| – | Эффективная защита от спама и взлома |
| – | Базы данных хранятся на SSD-дисках. |
| – | Максимальный размер базы данных - 2 Гб |
| – | Все данные проходят через процедуру ежедневного резервного |

копирования. Копии хранятся в течение 20 дней.

– Поддержка MySQL,Python,PHP, Ruby,Node.js

**1.4 Инструменты разработки**

Для разработки данного web-ресурса будет выбран конструктор сайтов Microsoft Visual Studio Community 2022 (64-разрядная версия) – Current Версия 17.7.4, который является наиболее актуальным и рейтинговым конструктором сайтов для данного вида программного продукта.

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 PowerPoint – для создания презентации, которая будет использована на защите проекта;

- WEB-ресурс Figma– будет использоваться для создания UX,UI ;

Разработка проекта будет происходить на компьютере со следующими параметрами:

- Intel(R) Core(TM) i3-11800H @ 2.30GHz 2.30 GHz;

- объем ОЗУ 8Гб;

- объем места на жестком диске 1000 GB;

- графический адаптер nvidia gforce 3060;

[**2 Проектирование задачи**](#_Достоинства_и_недостатки)

[**2.1**](#_Toc406878086) **Разработка структуры приложения, системы меню, навигации**

Разработка программы на C#, подготовка исходников для данной программы. Чтобы увидеть UX,UI прототипы приложения необходимо перейти по ссылке “ https://www.figma.com/file/CHkNmP93quG2mB1hzGWr29/Untitled-(2)-(1)?type=design&node-id=0-1&mode=design&t=Zt0tg52pfjcaXQ0v-0 ”.

[**2.2**](#_Toc406878088) **Организация данных**

Данные находятся в приложении.

**2.3 Разработка UML-диаграмм**

**2.3.1 Диаграмма последовательности**

Диаграмма последовательности UML — такая диаграмма, на которой показаны взаимодействия объектов, упорядоченные по времени их проявления. Основные элементы диаграммы последовательности это: обозначения объектов (прямоугольники), вертикальные линии, отображающие течение времени при деятельности объекта, и стрелки, показывающие выполнение действий объектами.

При разработке программного продукта была поставлена задача разработать диаграмму последовательности, которая показывает порядок взаимодействия пользователя с системой при оформлении заказа. С данной диаграммой можно ознакомиться в Приложении Б.

**2.3.2 Модель данных**

Модель данных – это абстрактное, самодостаточное, логическое определение объектов, операторов и прочих элементов, в совокупности составляющих абстрактную машину доступа к данным, с которой взаимодействует пользователь. С данной моделью можно ознакомиться в Приложении Б.

**2.4 Диаграмма вариантов использования**

Диаграмма вариантов использования – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей. С прототипом главной страницы вы можете ознакомиться в Приложении В.

**2.5 Диаграмма классов**

**Диаграмма классов** определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами. Вид и интерпретация диаграммы классов существенно зависит от точки зрения (уровня абстракции): классы могут представлять сущности предметной области (в процессе анализа) или элементы программной системы (в процессах проектирования и реализации). С данной системой вы можете ознакомиться в Приложении Г.

**2.6 Диаграмма деятельности**

**Диаграмма деятельности**— это графическое представление процессов и задач, выполняемых в рамках определенного проекта или деятельности. Эта диаграмма используется для анализа и оптимизации бизнес-процессов, планирования работы, принятия решений и управления проектами. Диаграмма деятельности состоит из набора символов и линий, каждый из которых имеет свое значение и назначение. С данной схемой вы можете ознакомиться в Приложении Д.

**2.7Диаграмма кооперации(объектов)**

Диаграмма кооперации предназначена для описания поведения системы на уровне отдельных объектов, которые обмениваются между собой сообщениями, чтобы достичь нужной цели или реализовать некоторый вариант использования. С данной схемой вы можете ознакомиться в Приложении Е.

**2.8 Моделирование бизнес-процессов**

Моделирование бизнес-процессов (BPM) в [управлении бизнес-процессами](https://ru.wikipedia.org/wiki/BPM_(%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BA%D0%BE%D0%BD%D1%86%D0%B5%D0%BF%D1%86%D0%B8%D1%8F)) и [системной инженерии](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%B0%D1%8F_%D0%B8%D0%BD%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D0%B8%D1%8F) — это деятельность по представлению процессов предприятия, позволяющая анализировать, улучшать и автоматизировать текущие [бизнес-процессы](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81). С данным моделированием вы можете ознакомиться в Приложении Ж.

**2.9 Структкрное проектирование**

Структурное проектирование понимается как методология построения алгоритмов, программ и систем, в том числе информационных, в основе которой лежит выявление структуры задачи, определение составляющих компонент и выделение связей между ними. Процесс разделения сложных задач (объектов, систем) на относительно независимые друг от друга подзадачи (части, подсистемы) называется декомпозицией. С данным моделированием вы можете ознакомиться в Приложении З.

**3 Реализация**

**3.1 Руководство программиста**

Данный программный продукт был разработан с помощью конструктора сайтов Microsoft Visual Studio Community 2022 (64-разрядная версия) – Current Версия 17.7.4.

В разделе представлены компоненты и функции программы, которые работают совместно для решения поставленной задачи. Данный раздел содержит перечень основных процедур и функций, используемых в приложении для достижения поставленных целей. С основными процедурами и функциями можно ознакомиться в таблице 2.

Таблица 2 – Процедуры и функции(Microsoft Visual Studio Community 2022 (64-разрядная версия) – Current Версия 17.7.4)

|  |  |  |
| --- | --- | --- |
| Функции и методы | В каком модуле  находится | Назаначение |
| 1 private void button1\_Click(object sender, EventArgs e) | Form1.cs | Переход на другую форму |
| 2 private void button2\_Click(object sender, EventArgs e) | Form1.cs | Переход на другую форму |
| 3 private void button1\_Click(object sender, EventArgs e) | Form2.cs | Переход на другую форму |
| 4 private void button2\_Click(object sender, EventArgs e) | Form2.cs | Переход на другую форму |
| 5 private void button3\_Click(object sender, EventArgs e) | Form2.cs | Переход на другую форму |
| 6 private void button4\_Click(object sender, EventArgs e) | Form2.cs | Переход на другую форму |
| 7 private void button6\_Click | Form2.cs | Переход на другую форму |
|  |  |  |
| Таблица 2 - Продолжение |  |  |
| 8 private void button5\_Click(object sender, EventArgs e) | Form2.cs | Переход на другую форму |
| 9 private void button7\_Click(object sender, EventArgs e) | Form2.cs | Изменение цвета текста |
| 10 private void button8\_Click(object sender, EventArgs e) | Form2.cs | Изменение цвета текста |
| 11 private void button11\_Click(object sender, EventArgs e) | Form2.cs | Изменение цвета текста |
| 12 private void button10\_Click(object sender, EventArgs e) | Form2.cs | Изменение цвета текста |
| 13 private void button9\_Click(object sender, EventArgs e) | Form2.cs | Изменение цвета текста |
| 14 private void button1\_Click(object sender, EventArgs e) | Form3.cs | Переход на другую форму |
| 15 private void button3\_Click(object sender, EventArgs e) | Form3.cs | Переход на другую форму |
| 16 private void button4\_Click(object sender, EventArgs e) | Form3.cs | Переход на другую форму |
| 17 private void button6\_Click(object sender, EventArgs e) | Form3.cs | Переход на другую форму |
|  |  |  |
| Таблица 2 - Продолжение |  |  |
| 18 private void button5\_Click(object sender, EventArgs e) | Form3.cs | Переход на другую форму |
| 19 private void button2\_Click(object sender, EventArgs e) | Form4.cs | Происходит проверка логина и пароля, введенных пользователем |
| 20 private void button1\_Click(object sender, EventArgs e) | Form5.cs | Переход на другую форму |
| 21 private void button1\_Click(object sender, EventArgs e) | Form6.cs | Переход на другую форму |
| 22 private void button2\_Click(object sender, EventArgs e) | Form6.cs | Переход на другую форму |
| 23 private void button1\_Click(object sender, EventArgs e) | Form7.cs | Переход на другую форму |
| 24 private void button2\_Click(object sender, EventArgs e) | Form7.cs | Переход на другую форму |
| 25 private void button1\_Click(object sender, EventArgs e) | Form8.cs | Переход на другую форму |
| 26 private void button2\_Click(object sender, EventArgs e) | Form8.cs | Переход на другую форму |
| 27 private void button1\_Click(object sender, EventArgs e) | Form9.cs | Переход на другую форму |
| 28 private void button2\_Click(object sender, EventArgs e) | Form9.cs | Переход на другую форму |
| 29 private void button2\_Click(object sender, EventArgs e) | Form10.cs | Переход на другую форму |
|  |  |  |
| Таблица 2 - Продолжение | | |
| 30 private void button3\_Click(object sender, EventArgs e) | Form10.cs | Переход на другую форму |
| 31 private void button1\_Click(object sender, EventArgs e) | Form10.cs | Происходит проверка занчения,введенного пользователем |
| 32 private void button2\_Click(object sender, EventArgs e) | Form11.cs | Переход на другую форму |
| 33 private void button3\_Click(object sender, EventArgs e) | Form11.cs | Переход на другую форму |
| 34 private void button1\_Click(object sender, EventArgs e) | Form11.cs | Происходит проверка занчения,введенного пользователем |
| 35 private void button2\_Click(object sender, EventArgs e) | Form12.cs | Переход на другую форму |
| 36 private void button3\_Click(object sender, EventArgs e) | Form12.cs | Переход на другую форму |
| 37 private void button1\_Click(object sender, EventArgs e) | Form12.cs | Происходит проверка занчения,введенного пользователем |
| 38 private void button2\_Click(object sender, EventArgs e) | Form13.cs | Переход на другую форму |
| 39 private void button3\_Click\_1(object sender, EventArgs e) | Form13.cs | Переход на другую форму |
| 40 private void button1\_Click\_1(object sender, EventArgs e) | Form13.cs | Происходит проверка занчения,введенного пользователем |

## 3.2 Спецификация программы

Спецификация программного обеспечения приведена в таблице 3.

Таблица 3– Спецификация программы

|  |  |
| --- | --- |
| Form1.cs | Определяет логику и поведение “Form1” |
| Form1.Designer.cs | Содержит автоматически сгенерированный код, созданный средой разработки для визуального дизайна формы “Form1” |
| Form1.resx | Содержит ресурсы, такие как строки, изображения и другие данные, связанные с формой “Form1” |
| Form2.cs | Определяет логику и поведение “Form2” |
| Form2.Designer.cs | Содержит автоматически сгенерированный код, созданный средой разработки для визуального дизайна формы “Form2” |
| Form2.resx | Содержит ресурсы, такие как строки, изображения и другие данные, связанные с формой “Form2” |
| Form3.cs | Определяет логику и поведение “Form3” |
| Form3.Designer.cs | Содержит автоматически сгенерированный код, созданный средой разработки для визуального дизайна формы “Form3” |
| Form3.resx | Содержит ресурсы, такие как строки, изображения и другие данные, связанные с формой “Form3” |
| Form4.cs | Определяет логику и поведение “Form4” |
| Form4.Designer.cs | Содержит автоматически сгенерированный код, созданный средой разработки для визуального дизайна формы “Form4” |
| Form4.resx | Содержит ресурсы, такие как строки, изображения и другие данные, связанные с формой “Form4” |
| Form5.cs | Определяет логику и поведение “Form5” |
| Form5.Designer.cs | Содержит автоматически сгенерированный код, созданный средой разработки для визуального дизайна формы “Form5” |
| Form5.resx | Содержит ресурсы, такие как строки, изображения и другие данные, связанные с формой “Form5” |
| Form6.cs | Определяет логику и поведение “Form6” |
| Form6.Designer.cs | Содержит автоматически сгенерированный код, созданный средой разработки для визуального дизайна формы “Form6” |
|  |  |
| Таблица 3–Прододжение |  |
| Form6.resx | Содержит ресурсы, такие как строки, изображения и другие данные, связанные с формой “Form6” |
| Form7.cs | Определяет логику и поведение “Form7” |
| Form7.Designer.cs | Содержит автоматически сгенерированный код, созданный средой разработки для визуального дизайна формы “Form7” |
| Form7.resx | Содержит ресурсы, такие как строки, изображения и другие данные, связанные с формой “Form7” |
| Form8.cs | Определяет логику и поведение “Form8” |
| Form8.Designer.cs | Содержит автоматически сгенерированный код, созданный средой разработки для визуального дизайна формы “Form8” |
| Form8.resx | Содержит ресурсы, такие как строки, изображения и другие данные, связанные с формой “Form8” |
| Form9.cs | Определяет логику и поведение “Form9” |
| Form9.Designer.cs | Содержит автоматически сгенерированный код, созданный средой разработки для визуального дизайна формы “Form9” |
| Form9.resx | Содержит ресурсы, такие как строки, изображения и другие данные, связанные с формой “Form9” |
| Form10.cs | Определяет логику и поведение “Form10” |
| Form10.Designer.cs | Содержит автоматически сгенерированный код, созданный средой разработки для визуального дизайна формы “Form10” |
| Form10.resx | Содержит ресурсы, такие как строки, изображения и другие данные, связанные с формой “Form10” |
| Form11.cs | Определяет логику и поведение “Form11” |
| Form11.Designer.cs | Содержит автоматически сгенерированный код, созданный средой разработки для визуального дизайна формы “Form11” |
| Form11.resx | Содержит ресурсы, такие как строки, изображения и другие данные, связанные с формой “Form11” |
| 777.exe | Приложение |
| 777.sln | Программа |

[**4 Тестирование**](#_Достоинства_и_недостатки)

**4.1 Тесты на использование**

При разработке ресурса Электронная книга «Тригонометрия» многие возникающие ошибки и недоработки были исправлены на этапе реализации программного продукта. После завершения испытания реализации ресурса было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программного продукта в автономном режиме. Тесты на использование представлены в таблице 4.

Таблица4 –Тесты на использование

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Идентиф. теста | Требования | Модель приложения | Приоритет | Краткое заглавие теста | Шаги для выполнения | Ожидаемые результаты | Статус |
| 1 | 2 | Файл приложения | Средний | Проверка на начало работы приложения при запуске | Открыть приложение | Приложение открылось | Пройден |
| 2 | 3 | Файл приложения | Средний | Проверка функционала кнопки «Войти» | Нажать на кнопку «Войти» | Открытие экрана «Регистрация» | Пройден |
| 3 | 6 | Файл приложения | Средний | Проверка функционала прохождения авторизации | 1.В поле для пароля ввести пароль 2. В воле для логина ввести логин | Открытие главной формы | Пройден |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Таблица 4–Продолжение | | | | | | | |
| 4 | 8 | Файл приложения | Средний | Проверка функциональности кнопки «Задания» | Нажать на кнопку «Задания» | Открытие экрана «Задания» | Пройден |
| 5 | 10 | Файл приложения | Средний | Проверка функциональности кнопки «Содержание» | Нажать на кнопку «Содержание» | Открытие экрана «Содержание» | Пройден |
| 6 | 11 | Файл приложения | Средний | Проверка функциональности кнопок «Оставить закладку» | Нажать на кнопки «Оставить закладку» | 1.Происходит изменение цвета текста | Пройден |
| 7 | 13 | Файл приложения | Средний | Проверка функциональности кнопок «Задания» | 1.Перейти в раздел заданий  2.Нажать на кнопку «Задания» | Открытие экрана с заданиями | Пройден |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Таблица 4–Продолжение | | | | | | | |
| 8 | 14 | Файл приложения | Средний | Проверка функциональности кнопки «Выполнить» | 1.Ввести ответ на вопрос  2.Нажать на кнопку «Выполнить» | Открытие экрана с результатом | Пройден |
| 9 | 15 | Файл приложения | Средний | Проверка функциональности кнопки «Помощь» | 1.Перейти в раздел заданий  2.Нажать на кнопку «Задания»  3.Нажать на кнопку«Помощь» | Открытие экрана с темой данного вопроса | Пройден |

**4.2 Отчет о результатах тестирования**

В результате проведения тестирования выяснилось, что все ранее оговоренные функции и требования, были разработаны, а также протестированы. Тесты показали, что все функции работают правильно. Следовательно разработанный ресурс можно передать заказчику.

В ходе тестирования программного обеспечения продукта на устройстве не было выявлено каких-либо ошибок, так как адаптивность ресурса была проведена на всех стадиях разработки.

**5 Руководство пользователя**

Разрабатываемый проект «Тригонометрия» будет является информационным для ознакомления с основами тригонометрии, для решения практических задач , связанных с измерением углов, расстояний и высот в геометрии и геодезии, а также для использования в физике, инженерных расчетах и других областях науки и техники. Проект будет включать в себя материалы о тригонометрических функциях, формулах преобразования углов, методах решения треугольников, прикладных задачах и примерах использования тригонометрии в реальных ситуациях.

Действия пользователей по работе с программным обеспечением:

Администратор:

1. авторизация;
2. Редактирование разделов книги;
3. Добавление нового материала/заданий в книгу.

Пользователь:

1. Авторизация;
2. редактировать профиль;
3. поиск темы по названию;
4. выполнение заданий;
5. Выбор и поиск заданий на выполнение;
6. чтение книги;
7. просмотр содержани;я ;
8. просмотр тем;
9. просмотр выполненных ;
10. заданийсделать закладку.

[**Заключение**](#_Toc406878088)

Целью данного проекта заключалась в создании электронной книги для получения и поддержки уже имеющихся знаний

В целом при реализации программного продукта, были выполнены все условия, поставленные на начальном этапе разработки. Из чего мы можем сделать вывод, что ресурс можно передавать заказчику и вводить в эксплуатацию.

Так же в процессе создания программного продукта была подготовлена программная документация. Был разработан перечень диаграмм и проведены тесты для проверки правильности функционирования ПО.

**Список использованных источников**

1. Руководство по программированию на C# [Электронный ресурс]. – Электронные данные. – Режим доступа: https://learn.microsoft.com/ru-ru/dotnet/csharp/programming-guide/.
2. Википедия [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://ru.wikipedia.org/wiki/Заглавная_страница>.

***Приложение А*** (Диаграмма последовательности)

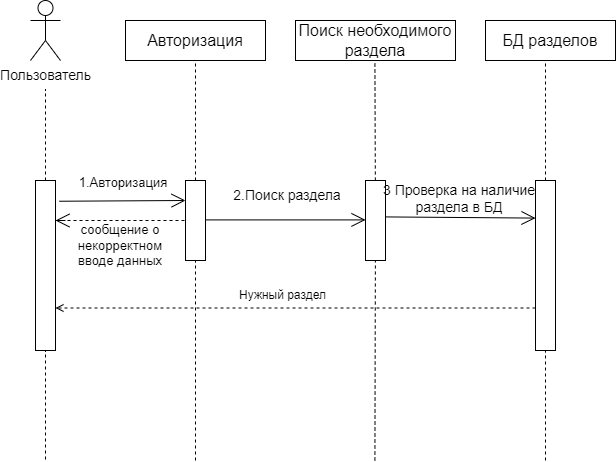


Рисунок 1- Диаграмма последовательности

***Приложение Б*** (Модель данных)

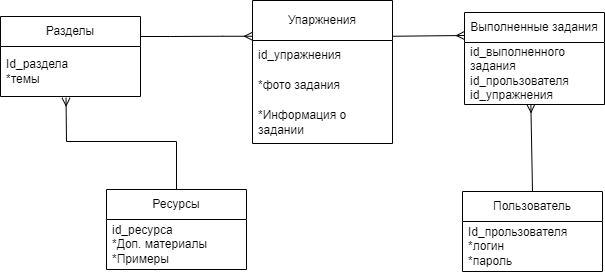


Рисунок 2- Модель данных

***Приложение В*** (Диаграмма вариантов использования)

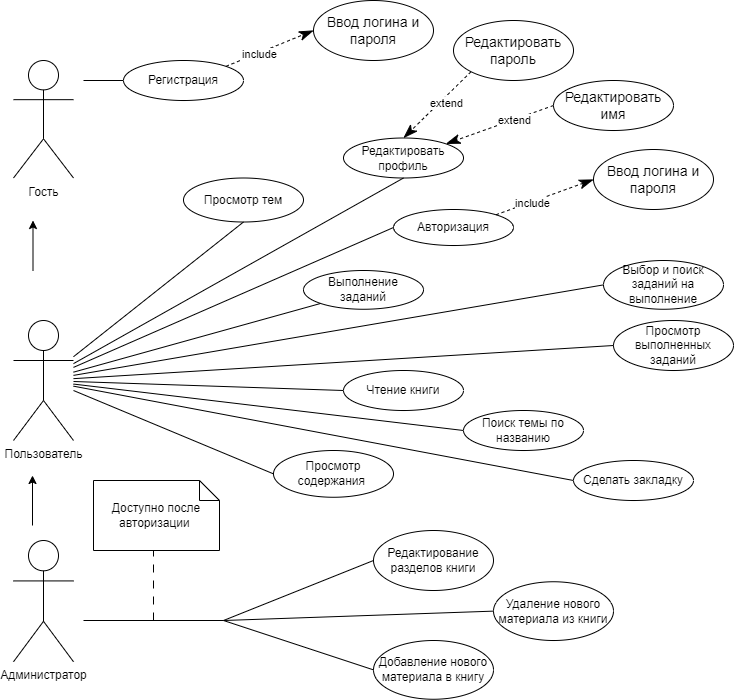


Рисунок 3- Диаграмма вариантов использования

***Приложение Г*** (Диаграмма классов)

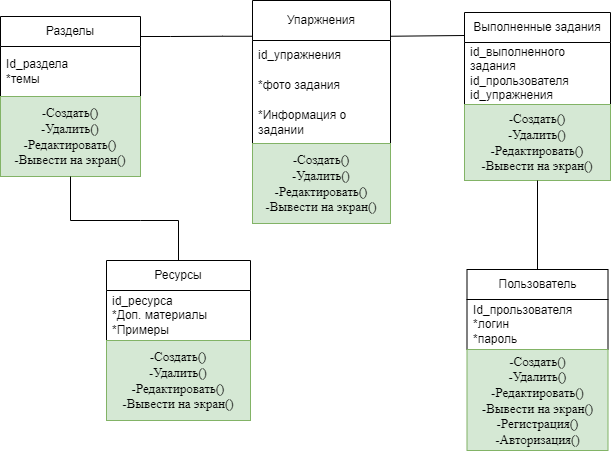


Рисунок 4- Диаграмма классов

***Приложение Д*** (Диаграмма деятельности)

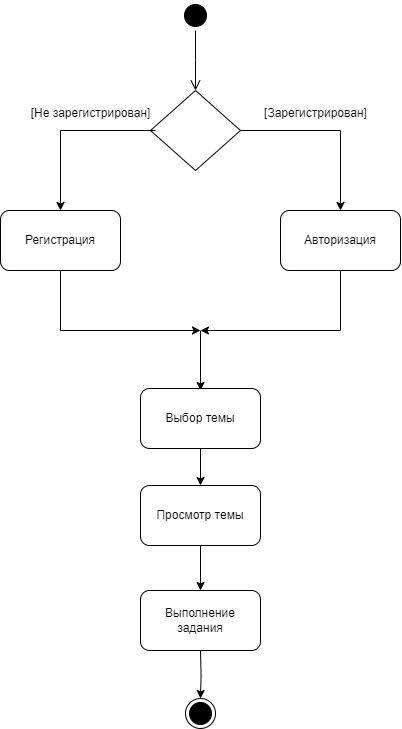


Рисунок 5- диаграмма деятельности

***Приложение Е*** (Диаграмма кооперации(объектов))

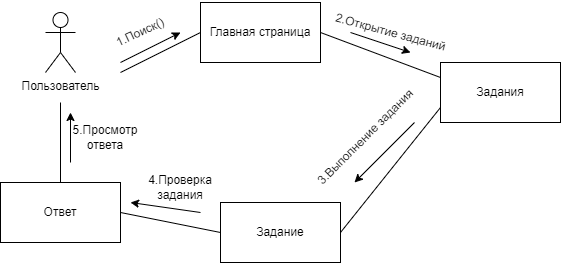
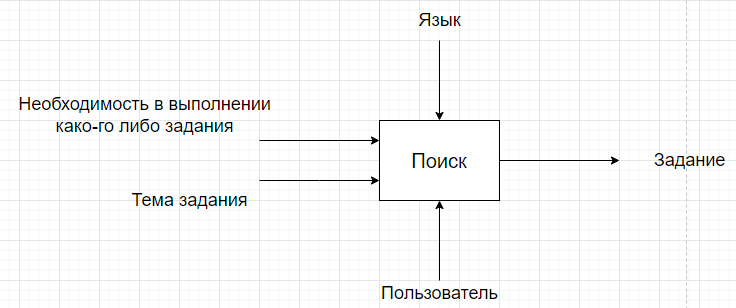
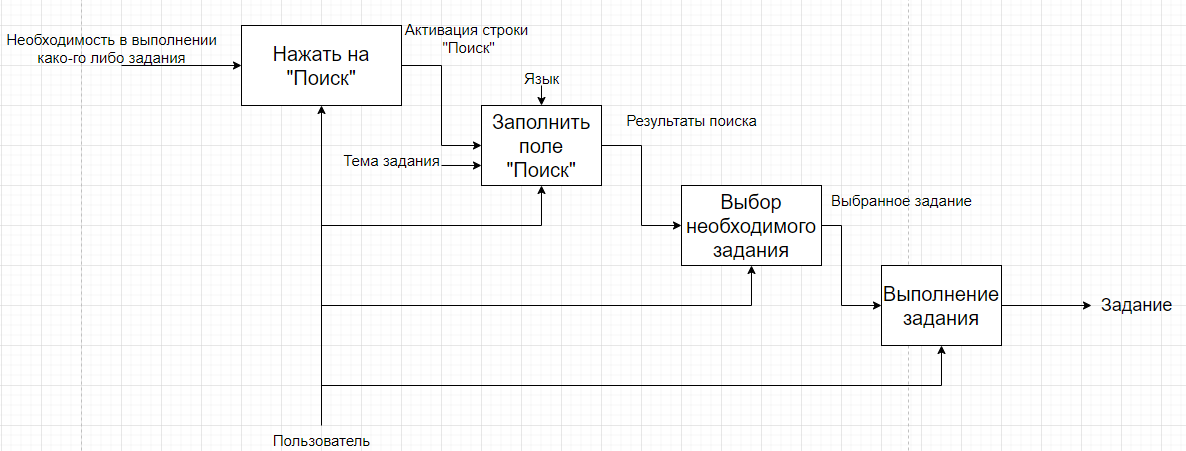


Рисунок 6- Диаграмма кооперации(объектов)

***Приложение Ж*** (Моделирование бизнес-процессов)



 Рисунок 7- Моделирование бизнес-процессов

***Приложение З*** (Структурное проектирование)

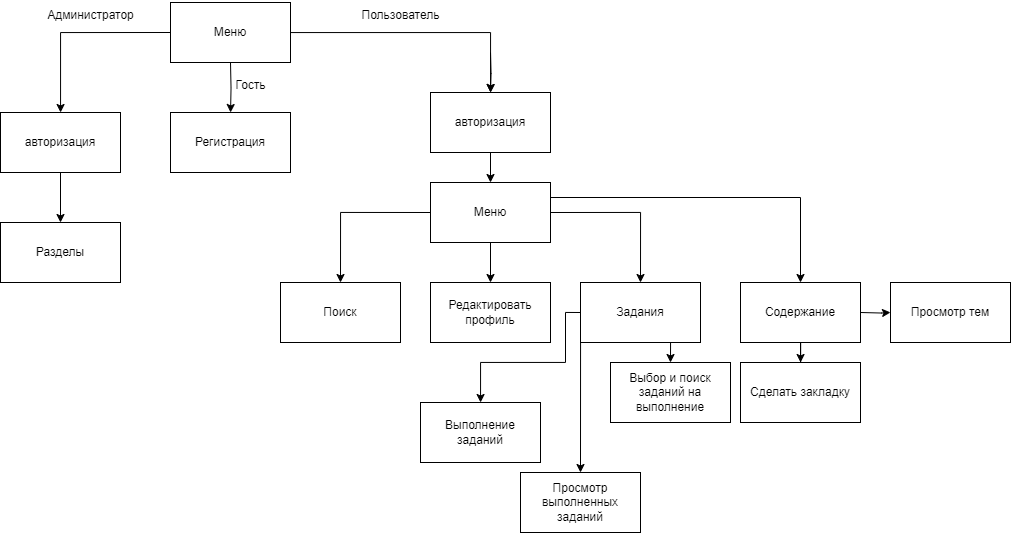


Рисунок 8- Структурное проектирование

***Приложение И*** (Листинг программы)

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_777

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Form1.ActiveForm.Hide();

Form2 MyForm2 = new Form2();

MyForm2.ShowDialog();

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

Form1.ActiveForm.Hide();

Form3 MyForm3 = new Form3();

MyForm3.ShowDialog();

Close();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

}

}

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Form2.ActiveForm.Hide();

Form1 MyForm1 = new Form1();

MyForm1.ShowDialog();

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

Form2.ActiveForm.Hide();

Form5 MyForm5 = new Form5();

MyForm5.ShowDialog();

Close();

}

private void button3\_Click(object sender, EventArgs e)

{

Form2.ActiveForm.Hide();

Form6 MyForm6 = new Form6();

MyForm6.ShowDialog();

Close();

}

private void button4\_Click(object sender, EventArgs e)

{

Form2.ActiveForm.Hide();

Form7 MyForm7 = new Form7();

MyForm7.ShowDialog();

Close();

}

private void button6\_Click(object sender, EventArgs e)

{

Form2.ActiveForm.Hide();

Form8 MyForm8 = new Form8();

MyForm8.ShowDialog();

Close();

}

private void button5\_Click(object sender, EventArgs e)

{

Form2.ActiveForm.Hide();

Form9 MyForm9 = new Form9();

MyForm9.ShowDialog();

Close();

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void button7\_Click(object sender, EventArgs e)

{

label3.ForeColor = Color.Red;

}

private void button8\_Click(object sender, EventArgs e)

{

label4.ForeColor = Color.Red;

}

private void button11\_Click(object sender, EventArgs e)

{

label5.ForeColor = Color.Red;

}

private void button10\_Click(object sender, EventArgs e)

{

label7.ForeColor = Color.Red;

}

private void button9\_Click(object sender, EventArgs e)

{

label6.ForeColor = Color.Red;

}

private void label6\_Click(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void label7\_Click(object sender, EventArgs e)

{

}

}

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Form3.ActiveForm.Hide();

Form1 MyForm1 = new Form1();

MyForm1.ShowDialog();

Close();

}

private void button3\_Click(object sender, EventArgs e)

{

Form3.ActiveForm.Hide();

Form10 MyForm10 = new Form10();

MyForm10.ShowDialog();

Close();

}

private void button4\_Click(object sender, EventArgs e)

{

Form3.ActiveForm.Hide();

Form11 MyForm11 = new Form11();

MyForm11.ShowDialog();

Close();

}

private void button6\_Click(object sender, EventArgs e)

{

Form3.ActiveForm.Hide();

Form12 MyForm12 = new Form12();

MyForm12.ShowDialog();

Close();

}

private void button5\_Click(object sender, EventArgs e)

{

Form3.ActiveForm.Hide();

Form13 MyForm13 = new Form13();

MyForm13.ShowDialog();

Close();

}

private void label1\_Click(object sender, EventArgs e)

{

}

public partial class Form4 : Form

{

public Form4()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

String UserName = "Igor";

String Password = "1234";

if (UserName ==textBox1.Text && Password== textBox2.Text)

{

MessageBox.Show("Успешно");

Form4.ActiveForm.Hide();

Form1 MyForm1 = new Form1();

MyForm1.ShowDialog();

Close();

}

else

{

MessageBox.Show("Некоректные логин или пароль");

}

}

}

public partial class Form5 : Form

{

public Form5()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Form5.ActiveForm.Hide();

Form2 MyForm2 = new Form2();

MyForm2.ShowDialog();

Close();

}

private void Form5\_Load(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

}

public partial class Form6 : Form

{

public Form6()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Form6.ActiveForm.Hide();

Form2 MyForm2 = new Form2();

MyForm2.ShowDialog();

Close();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

Form6.ActiveForm.Hide();

Form10 MyForm10 = new Form10();

MyForm10.ShowDialog();

Close();

}

private void Form6\_Load(object sender, EventArgs e)

{

}

}

public partial class Form7 : Form

{

public Form7()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Form7.ActiveForm.Hide();

Form2 MyForm2 = new Form2();

MyForm2.ShowDialog();

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

Form7.ActiveForm.Hide();

Form11 MyForm11 = new Form11();

MyForm11.ShowDialog();

Close();

}

}

public partial class Form8 : Form

{

public Form8()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Form8.ActiveForm.Hide();

Form2 MyForm2 = new Form2();

MyForm2.ShowDialog();

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

Form8.ActiveForm.Hide();

Form12 MyForm12 = new Form12();

MyForm12.ShowDialog();

Close();

}

}

public partial class Form9 : Form

{

public Form9()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Form9.ActiveForm.Hide();

Form2 MyForm2 = new Form2();

MyForm2.ShowDialog();

Close();

}

private void button2\_Click(object sender, EventArgs e)

{

Form9.ActiveForm.Hide();

Form13 MyForm13 = new Form13();

MyForm13.ShowDialog();

Close();

}

}

public partial class Form10 : Form

{

public Form10()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

Form10.ActiveForm.Hide();

Form3 MyForm3 = new Form3();

MyForm3.ShowDialog();

Close();

}

private void button3\_Click(object sender, EventArgs e)

{

Form10.ActiveForm.Hide();

Form6 MyForm6 = new Form6();

MyForm6.ShowDialog();

Close();

}

private void button1\_Click(object sender, EventArgs e)

{

String Password = "Нечетная";

if (Password == textBox1.Text)

{

MessageBox.Show("Верно");

}

else

{

MessageBox.Show("Неверно");

}

}

}

public partial class Form11 : Form

{

public Form11()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

Form11.ActiveForm.Hide();

Form3 MyForm3 = new Form3();

MyForm3.ShowDialog();

Close();

}

private void button3\_Click(object sender, EventArgs e)

{

Form11.ActiveForm.Hide();

Form7 MyForm7 = new Form7();

MyForm7.ShowDialog();

Close();

}

private void button1\_Click(object sender, EventArgs e)

{

String Password = "Четная" ;

if (Password == textBox1.Text)

{

MessageBox.Show("Верно");

}

else

{

MessageBox.Show("Неверно");

}

}

}