## СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 4](#_Toc145122594)

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ 5](#_Toc145122595)

[ВВЕДЕНИЕ 6](#_Toc145122596)

[ОСНОВНАЯ ЧАСТЬ 7](#_Toc145122597)

[1 *UML* диаграммы вариантов использования 7](#_Toc145122598)

[2 *UML* диаграммы классов 7](#_Toc145122599)

[3 Описание классов образующих связь типа «общее-частное» 9](#_Toc145122600)

[4 Дерево ветвлений *Git* 10](#_Toc145122601)

[5 Тестирование 11](#_Toc145122602)

[ЗАКЛЮЧЕНИЕ 13](#_Toc145122603)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 14](#_Toc145122604)

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

*Общие положения*

Полное наименование: Приложение ведения списка геометрических фигур с возможностью поиска и поддержкой *GUID* операций.

Краткое наименование: Приложение ведения списка геометрических фигур.

Вид разработки: *Windows Forms Desktop App*.

*Основание для разработки*

Разработка программного продукта ведется на основании учебного плана учебного заведения.

*Плановые сроки начала и окончания выполнения работы*

Плановый срок начала работ – 01.09.2023

Плановый срок окончания работ – 10.09.2023

*Назначение и цель разработки*

Ведения списка геометрических фигур с возможностью поиска, добавления, удаления и изменения элементов трёх типов: круг, прямоугольник, треугольник.

Требования к функциональным характеристикам

Программа должна быть исполнена в виде *Windows Forms* *C*# приложения.

Перечень функций и их спецификации:

1. Работа с файлами расширения .*cringe*:

открытие и работа с файлами с расширением .*cringe*;

сохранение данных в файл с расширением .*cringe* перед закрытием программы.

1. Поддержка нескольких типов геометрических фигур:

поддержка фигуры «круг»;

поддержка фигуры «прямоугольник»;

поддержка фигуры «треугольник».

1. Ведение списка фигур:

возможность удалять элементы списка;

возможность добавлять новые элементы списка;

возможность изменять существующие элементы списка.

1. Выполнение поиска по списку.

Требования к надежности

Возникновение каких-либо внутренних ошибок не должно приводить к утере данных. Программа должна в правильной форме реагировать на действия пользователей.

*Условия эксплуатации*

Системные требования к составу и параметрам технических средств:

процессор: *Intel* ® *Xeon* ® *E*-2234 @ 3.6 ГГц;

ОЗУ: 2 Гб;

периферия: монитор, клавиатура, мышь;

# ВВЕДЕНИЕ

Назначение программной документации заключается в демонстрации возможностей итогового программного продукта посредством *UML* диаграмм классов и *UML* диаграмм вариантов использования и описанием классов, образующих связь типа «общее-частное», включающее имеющиеся поля, свойства и методы класса, их типы и входные параметры в случае методов класса. Также приведено техническое задание к разработанной программе, дерево ветвлений *git* и результаты функционального и модульного тестирования программного продукта.

# ОСНОВНАЯ ЧАСТЬ

1 *UML* диаграммы вариантов использования

*UML* диаграмма вариантов использования для разработанной программы представлена на рисунке 1.

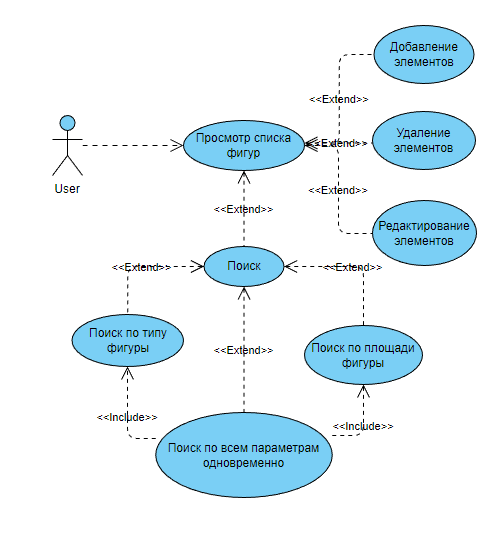
****

Рисунок 1 – *UML* диаграмма вариантов использования

2 *UML* диаграммы классов

*UML* диаграмма классов библиотеки *Model* для разработанной программы представлена на рисунке 2.

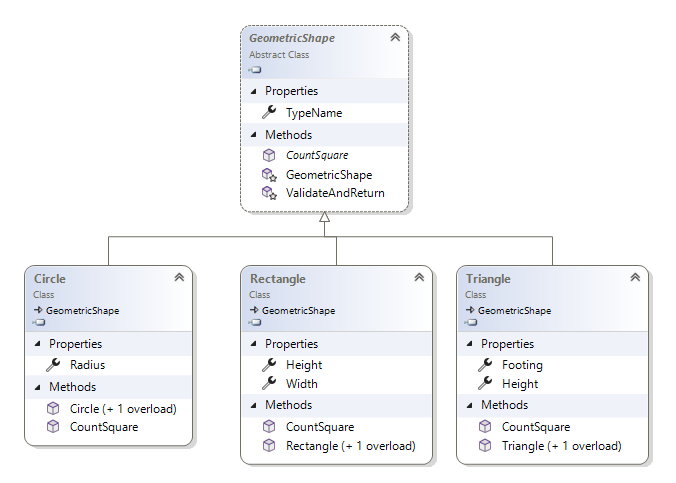


Рисунок 2 – *UML* диаграмма классов библиотеки *Model*

*UML* диаграмма классов форм и пользовательских элементов управления для разработанной программы представлена на рисунке 3.

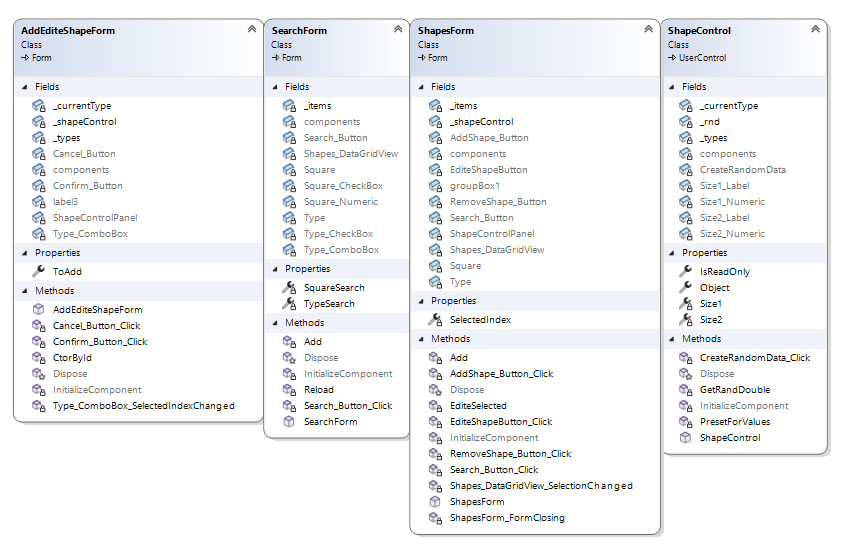


Рисунок 3 – *UML* диаграмма классов форм и пользовательских элементов управления

*UML* диаграмма вспомогательного класса для загрузки и сохранения данных представлена на рисунке 4.

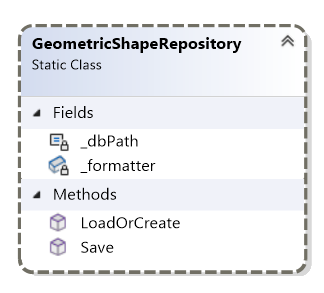


Рисунок 3 – *UML* диаграмма вспомогательного класса для загрузки и сохранения данных

*UML* диаграмма вспомогательного класса для загрузки и сохранения данных представлена на рисунке 4.

3 Описание классов образующих связь типа «общее-частное»

Таблица 1 описывает класс *GeometryShape*.

Таблица 1 – Класс *GeometryShape*

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс *GeometryShape* — сущность для описания абстрактной геометрической фигуры в программе | | |
| Свойства | | |
| *TypeName* | *string* | Наименование типа фигуры |
| Методы | | |
| *abstract CountSquare()* | *double* | Подсчёт площади фигуры |
| *ValidateAndReturn(double size)* | *double* | Проверяет размерный параметр на валидность. Если данные валидны – возвращает их, если нет – выбрасывает ошибку |

Таблица 2 описывает класс *Circle*.

Таблица 2 – Класс *Circle*

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс *Circle* — сущность для описания геометрической фигуры круга в программе | | |
| Свойства | | |
| *Radius* | *double* | Радиус круга |
| Методы | | |
| *Circle(double radius)* |  | Конструктор задающий радиус |
| *Circle()* |  | Конструктор по умолчанию |

Таблица 3 описывает класс *Rectangle*.

Таблица 3 – Класс *Rectangle*

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Описание | |
| Описание класса | | | |
| Класс *Rectangle* — сущность для описания геометрической фигуры прямоугольника в программе | | | |
| Свойства | | | |
| *Height* | *double* | Высота прямоугольника | |
| *Width* | *double* | Ширина прямоугольника | |
| Методы | | |
| *Rectangle* *(double heght, double width)* |  | Конструктор задающий высоту и ширину |
| *Rectangle* *()* |  | Конструктор по умолчанию |

Таблица 4 описывает класс *Triangle*.

Таблица 4 – Класс *Triangle*

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс *Triangle* — сущность для описания геометрической фигуры треугольника в программе | | |
| Свойства | | |
| *Height* | *double* | Высота треугольника |
| *Footing* | *double* | Основа треугольника |
| Методы | | |
| *Triangle* *(double heght, double width)* |  | Конструктор задающий высоту и основу |
| *Triangle()* |  | Конструктор по умолчанию |

4 Дерево ветвлений *Git*

Дерево ветвлений *Git* приведено на рисунке 4.

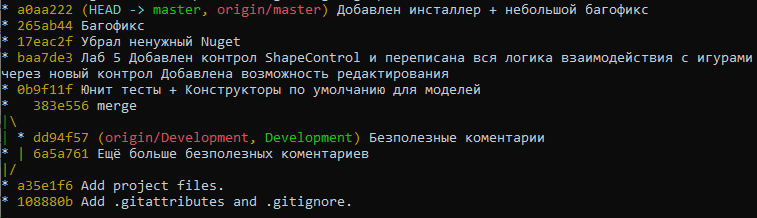


Рисунок 4 – Дерево ветвлений *Git*

5 Тестирование

Таблица 5 описывает функциональное тестирование программы.

Таблица 5 – Функциональное тестирование программы

|  |  |
| --- | --- |
| Тестовый случай | Результат |
| Пользователь открывает программу, нажимает кнопку «*Add Shape*» выбирает тип «*Rectangle*», заполняет поля «*Width*», «*Height*» и нажимает кнопку «*Confirm*» | В список фигур добавиться фигура типа «*Rectangle*» с заданными пользователем параметрами. |
| Пользователь открывает программу, нажимает кнопку «*Add Shape*» не выбирает тип, заполняет все доступные поля значениями равными или меньшими нуля. | По умолчанию выбирается фигура типа «*Circle*». Программа не дает заполнять поля формы значениями меньше нуля. При попытке нажатия кнопки «*Confirm*» с полями равными нулю программа выдаст сообщение о том, что параметр размера фигуры не может быть меньше или равен нулю. |
| Пользователь открывает программу, выбирает один из элементов и нажимает кнопку «*Remove Shape*» | Выбранная фигура удаляется из списка |
| Пользователь открывает программу, не выбирает ни одн из элементов и нажимает кнопку «*Remove Shape*» | По умолчанию выбрана первая фигура из списка. Она и будет удалена |
| Пользователь открывает программу, в которой ещё нет добавленных элементов, и нажимает кнопку «*Remove Shape*» | Ничего не произойдёт |

Рисунок 5 демонстрирует модульное тестирование программы.

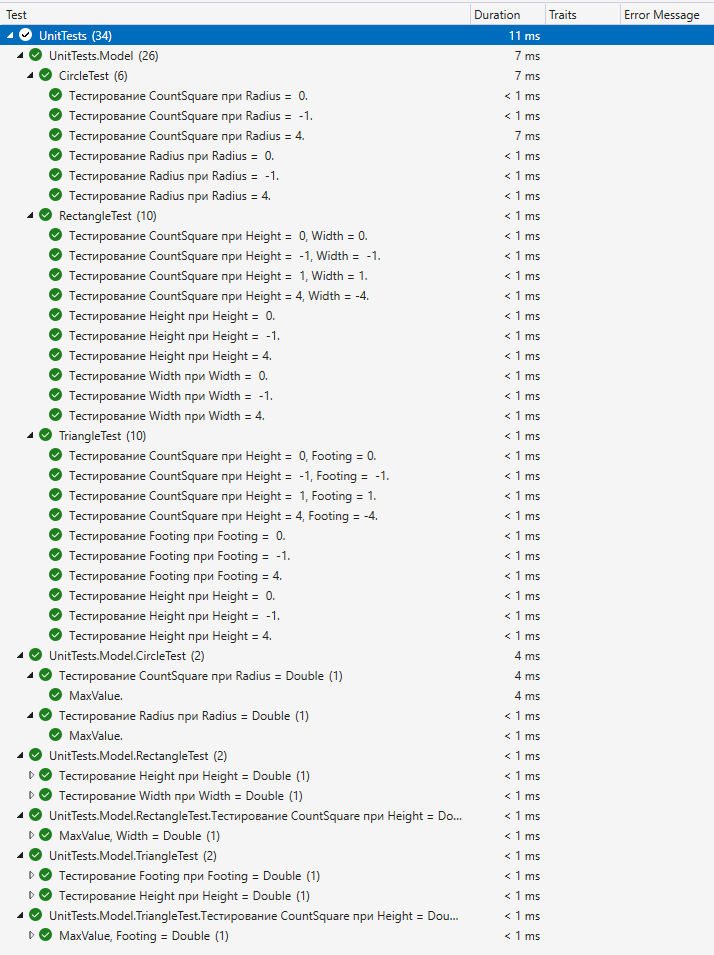


Рисунок 4 – Структура тестового проекта в Visual Studio

# ЗАКЛЮЧЕНИЕ

В данной программной документации была приведена демонстрации возможностей итогового программного продукта посредством *UML* диаграмм классов и *UML* диаграмм вариантов использования и описанием классов, образующих связь типа «общее-частное», включающее имеющиеся поля, свойства и методы класса, их типы и входные параметры в случае методов класса. Также было приведено техническое задание к разработанной программе, дерево ветвлений *git* и результаты функционального и модульного тестирования программного продукта.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Регулярные выражения. Википедия, свободная энциклопедия [Электронный ресурс]. — URL: https://ru.wikipedia.org/wiki/Регулярные\_выражения (дата обращения 01.09.2023).

2. Элементы языка регулярных выражений — краткий справочник. Microsoft Software Developer Network [Электронный ресурс]. — URL: http://msdn.microsoft.com/ru-ru/library/az24scfc(v=vs.110).aspx (дата обращения 01.09.2023).

3. Регулярные выражения в .NET Framework. Microsoft Software Developer Network [Электронный ресурс]. — URL: http://msdn.microsoft.com/ru-ru/library/hs600312(v=vs.110).aspx (дата обращения 01.09.2023).

4. Git — Book [Электронный ресурс]. — URL: http://git-scm.com/book/ru/v1 (дата обращения 01.09.2023).

5. GitHub [Электронный ресурс]. — URL: https://github.com (дата обращения 01.09.2023).

6. Git — Downloading Package [Электронный ресурс]. — URL: http://git-scm.com/download/win (дата обращения 01.09.2023).

7. gitignore/VisualStudio.gitignore [Электронный ресурс]. — URL: https://github.com/github/gitignore/blob/master/VisualStudio.gitignore (дата обращения 01.09.2023).

8. Free Mercurial and Git Client for Windows and Mac | Atlassian SourceTree [Электронный ресурс]. — URL: http://www.sourcetreeapp.com/ (дата обращения 01.09.2023).

9. Git — GUI Clients [Электронный ресурс]. — URL: http://git-scm.com/downloads/guis (дата обращения 01.09.2023).

10. A successful Git branching model// nvie.com [Электронный ресурс]. —URL: http://nvie.com/posts/a-successful-git-branching-model/ (дата обращения 01.09.2023).

11. Удачная модель ветвления для Git // Хабрахабр [Электронный ресурс]. — URL: http://habrahabr.ru/post/106912/ (дата обращения 01.09.2023).

12. Калентьев А. А. Новые технологии в программировании : учеб.пособие / А. А. Калентьев, Д. В. Гарайс, А. Е. Горяинов. — Томск : ЭльКонтент, 2014. — 176 с.

13. Работы студенческие по направлениям подготовки и специально-стям технического профиля. Общие требования и правила оформления. ОСТУСУР 01-2013 53 с.

14. Microsoft Software Developer Network [Электронный ресурс]. —URL: http://msdn.microsoft.com/ru-ru/default.aspx (дата обращения: 01.09.2023).