МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

Институт среднего профессионального образования

Утверждаю

Зам. директора по УМР

Е.Г. Конакина

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**ОТЧЕТ**

**по учебной практике УП.01.01**

по профессиональному модулю

ПМ.01. Разработка модулей программного обеспечения для компьютерных систем

(код и наименование)

Специальность 09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студента III курса 32919/3 группы 

Дроздов Дмитрий Артёмович

(Фамилия, имя, отчество)

Место прохождения практики ФГАОУ ВО «СПбПУ» Институт среднего профессионального образования, пр. Энгельса, 23, Спб, 194156 

(наименование и адрес организации)

Период прохождения практики

с «09» марта 2021 г. по «03» апреля 2021 г.

Руководители практики:

От колледжа \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Янчук Т.А.

(подпись) (Ф.И.О.)

Итоговая оценка по практике (от колледжа): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2021

Утверждаю

Зам. директора по УМР

Е.Г. Конакина

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

**Задание на учебную практику УП.01.01**

по профессиональному модулю ПМ.01. Разработка модулей программного обеспечения для компьютерных систем

(код и наименование) 

Специальность 09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студенту III курса 32919/3 группы



(фамилия, имя, отчество)

Место прохождения практики ФГАОУ ВО «СПбПУ» Институт среднего профессионального образования, пр. Энгельса, 23, СПб, 194156

(наименование и адрес организации)

Период прохождения практики

с «09» марта 2021 г. по «03» апреля 2021 г.

**Виды работ, обязательные для выполнения**

1. Разработка алгоритма поставленной задачи;
2. Разработка кода программного продукта на основе готовых спецификаций на уровне модуля;
3. Использование инструментальных средств на этапе отладки программного продукта.

**Индивидуальное задание**

1. Вычисление функции с помощью разложения в ряд

2. Использование подпрограмм для обработки

3. Работа с графикой

4. Обработка одномерных массивов

5. Обработка двумерных массивов

6. Работа с файлами последовательного доступа

7. Динамические структуры данных

8. Работа с файловой системой

Задание выдал «11» января 2021 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Янчук Т.А.

(подпись) (Ф.И.О.)

С заданием ознакомлен:

«09» марта 2021 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_ Дроздов Д.А.

(подпись) (Ф.И.О. студента)

**АТТЕСТАЦИОННЫЙ ЛИСТ**

**ПО УЧЕБНОЙ ПРАКТИКЕУП.01.01**

по профессиональному модулю

ПМ.01. Разработка модулей программного обеспечения для компьютерных систем

(код и наименование)

Специальность 09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студента III курса 32919/3 группы

(Фамилия, имя, отчество)

Место прохождения практики ФГАОУ ВО «СПбПУ» Институт среднего профессионального образования, пр. Энгельса, 23, СПб, 194156

(наименование и адрес организации) 

Период прохождения практики

с «09» марта 2021 г. по «03» апреля 2021 г.

**Виды и качество выполнения работ**

|  |  |  |
| --- | --- | --- |
| Виды выполненных работ обучающимся  во время практики | Объем работ, час. | Качество выполнения работ (оценка по пятибальной системе) |
| Разработка алгоритма поставленной задачи | 24 |  |
| Разработка кода программного продукта на основе готовых спецификаций на уровне модуля; | 32 |  |
| Использование инструментальных средств на этапе отладки программного продукта. | 30 |  |
| Проектирование программных систем | 24 |  |
| Документирование программных продуктов | 20 |  |

**Характеристика учебной/профессиональной деятельности обучающегося во время производственной практики (по профилю специальности):**

Общие и профессиональные компетенции, предусмотренные программой практики, освоены/не освоены.

(нужное подчеркнуть)

Итоговая оценка по практике (от колледжа): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики Янчук Т.А. 

(Ф.И.О.) (подпись)

Дата «02» апреля 2021 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

Институт среднего профессионального образования

**ДНЕВНИК**

**прохождения учебной практики УП.01.01**

по профессиональному модулю

ПМ.01. Разработка модулей программного обеспечения для компьютерных систем

(код и наименование)

Специальность 09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студента III курса 32919/3 группы

(Фамилия, имя, отчество)

Место прохождения практики ФГАОУ ВО «СПбПУ» Институт среднего профессионального образования, пр. Энгельса, 23, СПб, 194156  
 (наименование и адрес организации)

Период прохождения практики

с «09» марта 2021 г. по «03» апреля 2021 г.

Руководители практики:

От колледжа \_\_\_\_\_\_\_\_\_\_\_\_ Янчук Т.А.

(подпись) (Ф.И.О.)

Санкт-Петербург

2021

**Содержание дневника**

|  |  |  |
| --- | --- | --- |
| Дата | Виды выполненных работ и заданий по программе практики | Подпись руководителя практики |
| 12.01 | Получение задания на учебную практику. |  |
| 13.01 | Выполнение задания по теме «Вычисление функции с помощью разложения в ряд». |  |
| 14.01 | Выполнение задания по теме «Вычисление функции с помощью разложения в ряд». |  |
| 15.01 | Выполнение задания по теме «Использование подпрограмм для обработки». |  |
| 16.01 | Выполнение задания по теме «Использование подпрограмм для обработки». |  |
| 18.01 | Выполнение задания по теме «Работа с графикой». |  |
| 19.01 | Выполнение задания по теме «Работа с графикой». |  |
| 20.01 | Выполнение задания по теме «Обработка одномерных массивов». |  |
| 21.01 | Выполнение задания по теме «Обработка одномерных массивов». |  |
| 22.01 | Выполнение задания по теме «Обработка двумерных массивов». |  |
| 23.01 | Выполнение задания по теме «Обработка двумерных массивов». |  |
| 25.01 | Выполнение задания по теме «Работа с файлами последовательного доступа». |  |
| 26.01 | Выполнение задания по теме «Работа с файлами последовательного доступа». |  |
| 27.01 | Выполнение задания по теме «Динамические структуры данных». |  |
| 28.01 | Выполнение задания по теме «Динамические структуры данных». |  |
| 29.01 | Выполнение задания по теме «Работа с файловой системой». |  |
| 30.01 | Выполнение задания по теме «Работа с файловой системой». |  |
| 01.02 | Выполнение задания по теме «Работа с файловой системой». |  |
| 02.02 | Выполнение задания по теме «Работа с файловой системой». |  |
| 03.02 | Выполнение задания по теме «Работа с файловой системой». |  |
| 04.02 | Выполнение задания по теме «Работа с файловой системой». |  |
| 05.02 | Оформление отчета. |  |
| 06.02 | Сдача отчета и защита практики. |  |

**ВВЕДЕНИЕ**

В ответ на усложнение и многочисленность задач, которые сегодня предъявляются к программному обеспечению, усложняется и сама система ПО, но при этом порождается и другая необходимость – упростить процесс создания, развития и модернизации программного обеспечения. Такой подход не только упрощает разработчикам решение своих задач, но и позволяет упростить и удешевить эксплуатацию ПО для пользователей.

Практически все сложное программное обеспечение имеет модульную структуру, то есть состоит из программных модулей, при этом разработка программных модулей программного обеспечения может быть выполнена:

1. Как составляющая часть процесса разработки ПО. В данном случае разработка модулей, в зависимости от проекта, может вестись последовательно и (или) параллельно, но всегда завершается их объединением в единую систему.
2. В процессе доработки, усовершенствования или модернизации ранее созданного ПО. Модульная конструкция программного обеспечения позволяет вносить изменения в отдельные модули, удалять некоторые из них или дополнять систему новыми модулями. При этом разработка программных модулей программного обеспечения и их внедрение, как и изменение или удаление, не заставляет переделывать всю систему ПО в целом.

В программном комплексе созданные модули могут представлять собой:

* законченные компоненты;
* функциональные группы;
* пакеты программ;
* единый программный комплекс – законченный программный продукт конкретного целевого назначения.

Практика будет выполняться на языке C#. Этот язык программирования был выбран для как один из самых популярных языков программирования и обладающий множеством возможностей для создания интерфейсов для Windows

Структура отчета состоит из ведения, 8 пунктов, в которых описано выполнение задачи по каждой из выданных тем, заключения и списка литературы.

1. **ВЫЧИСЛЕНИЕ ФУНКЦИИ С ПОМОЩЬЮ РАЗЛОЖЕНИЯ В РЯД**

Была разработана программа, вычисляющая значение функции с помощью разложения в ряд. Сумма ряда вычисляется при помощи цикла с неизвестным числом повторений, так как требуется найти значение с заданной точностью.

Вычисляемая функция – натуральный логарифм (рис. 1). Демонстрация выполнения приведена на рисунке 2. Фрагмент кода реализации показан на рисунке 3 и 4. Данный фрагмент описывает реализация задания по паттерну MVVM. Конструктор MainWindow записывает в DataContext ViewModel для этого экрана. В ViewModel содержаться все поля приложения, включая то, что вводит пользователь и что выводит сама программа. Интерфейс IDataErrorInfo позволяет wpf автоматически производить валидацию полей, используя индексатор. OnCalculateCommand() – производит вычисление по формуле и запись ответов в соответствующие поля.

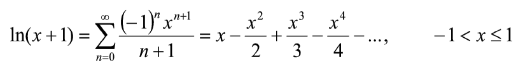


Рисунок 1 – Заданная функция.

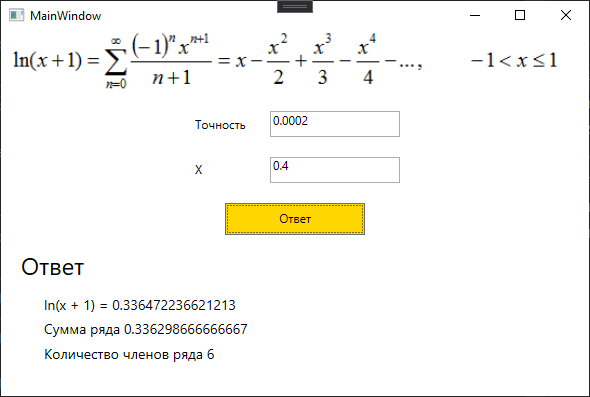


Рисунок 2 – Демонстрация выполнения.

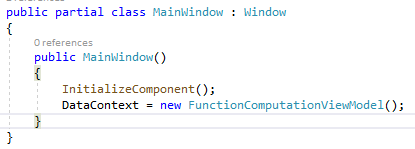


Рисунок 3 – Фрагмент кода реализации View.



Рисунок 4 – Фрагмент кода реализации ViewModel.

1. **ИСПОЛЬЗОВАНИЕ ПОДПРОГРАММ ДЛЯ ОБРАБОТКИ ПРОСТЫХ ТИПОВ ДАННЫХ**

Была разработана программа, которая демонстрирует использование функции программиста. Подпрограмма получает исходные данные через параметры, вычисляет по заданию результат и передает его в место вызова. Задание по полученному варианту приведено на рисунке 5, на рисунках 6 и 7 демонстрируется выполнение и показ фрагмента кода реализации соответственно. Данный фрагмент кода представляет методы, в котором находится условие для вычисления сторон треугольника и треугольников по формуле Герона, а так же запись ответа в поля и метод для вычисления по формуле Герона по 4 сторонам.

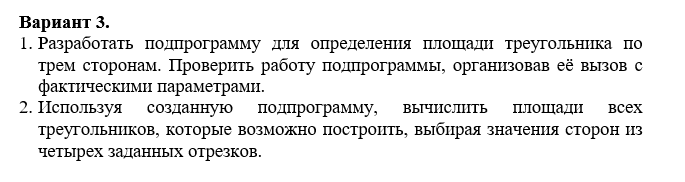


Рисунок 5 – Задание.

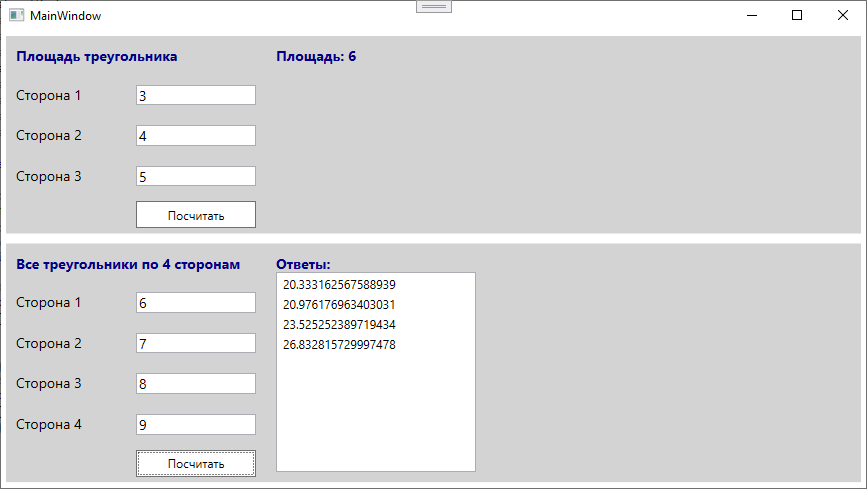


Рисунок 6 – Демонстрация выполнения.



Рисунок 7 – Фрагмент кода реализации.

1. **РАБОТА С ГРАФИКОЙ**

Была разработана программа движения фигуры по диагонали при помощи таймера, а также программа, работающая с таблицей используя задание 2 и программа, удаляющая данные из таблицы используя условия из задания 3 (рис. 7). В проект добавлена настройку параметров: выбор цвета фигуры с помощью диалогового окна выбора цвета и настройку размера фигуры. Настройка параметров происходит с помощью второй формы. Демонстрация работы задания 1 – рисунок 8, задания 2 – рисунок 9, задание 3 – рисунок 10, фрагмент кода реализации – рисунок 11 – 14.

На рисунке 13 продемонстрирован код, управляющий движением круга, метод OnTick вызывается каждый тик таймера, так же предоставлены методы изменения направления круга и методы определяющие текущее направление круга для полного контроля за движением фигуры. На рисунке 14 продемонстрировано создания круга и запуск таймера, в котором используется метод RenderTick, в этом методе .идет обращение к объекту управляющим фигурой, вызов OnTick. Так же вычисляется deltaTime для более плавного движения фигуры. На рисунке 15 показан обработчик нажатия на кнопку, вызывающий ShowColorDialog для выбора нового цвета фигуры и обработчик изменения слайдера, для изменения размера фигуры. На рисунке 11 продемонстрирован метод, изменяющий цвет клетки в промежутки, между первым четным и последним чётным числом. На рисунке 12 продемонстрированы реализации алгоритма удаления из задания 3, где удаляется элементы до тех пор, пока не останется только один.

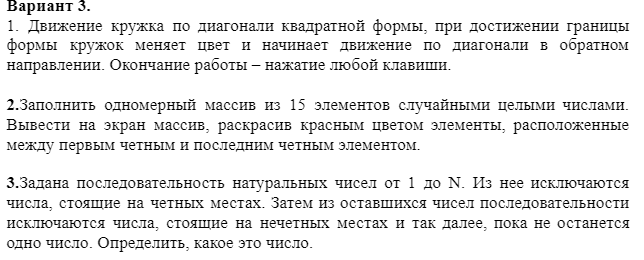


Рисунок 7 – Задание.

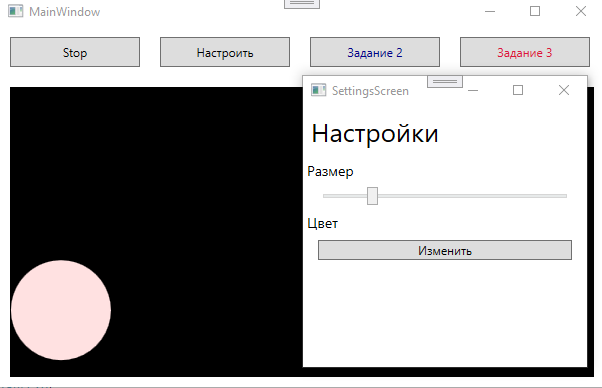


Рисунок 8 – Демонстрация работа 1 задания.

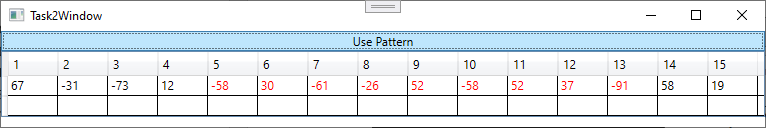


Рисунок 9 – Демонстрация работа задания 2.

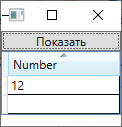
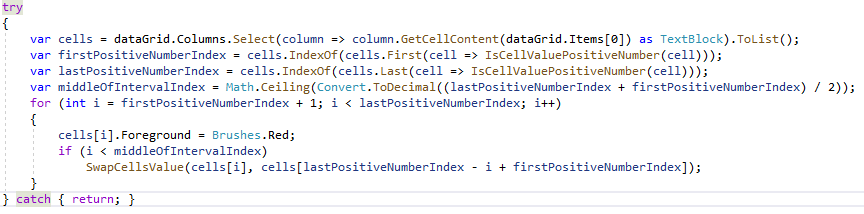


Рисунок 10 – Демонстрация работа задания 3.

 Рисунок 11 – Фрагмент кода реализации задания 2

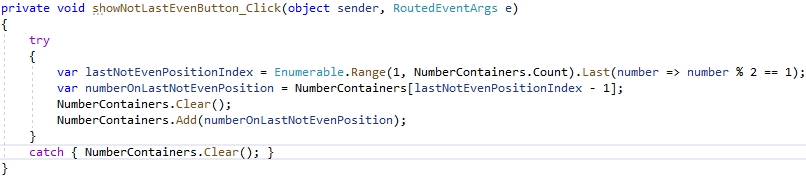


Рисунок 12 – Фрагмент кода реализации 3 задания.

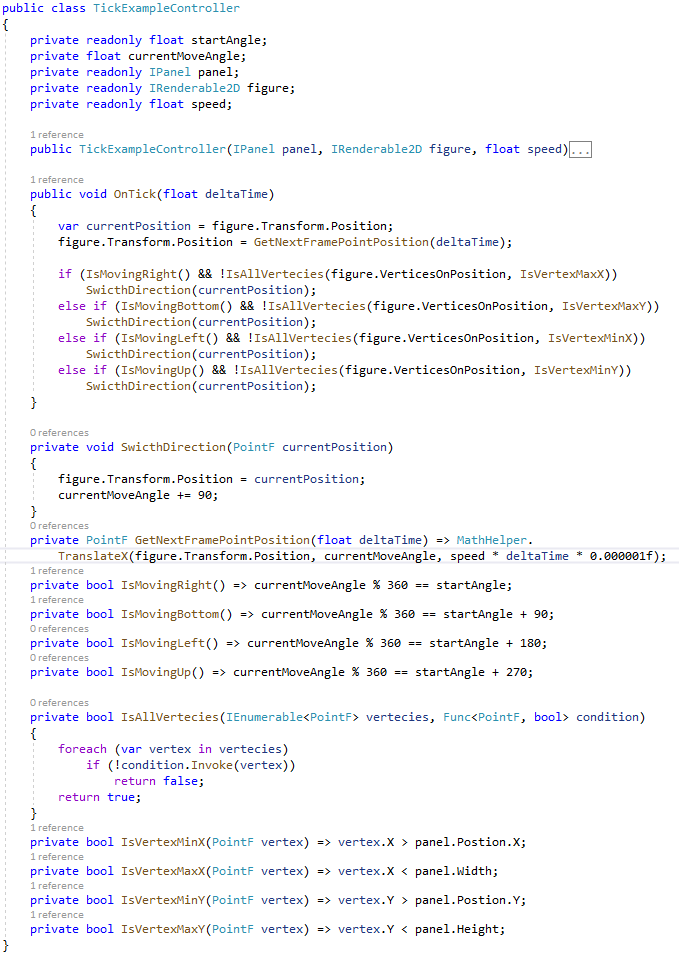


Рисунок 13 – Фрагмент кода реализации движения круга.

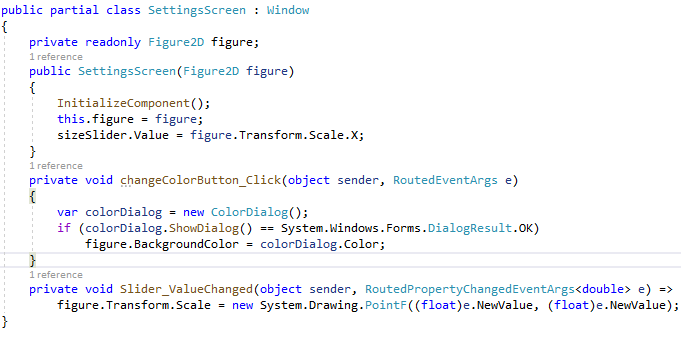


Рисунок 14 – Фрагмент кода экрана настройки.

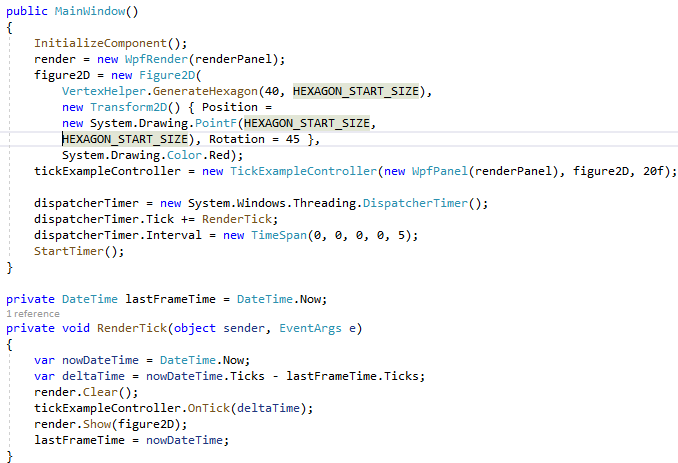


Рисунок 15 – Фрагмент кода генерации круга и работа таймера.

1. **ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ**

Была разработана программа для заполнения одномерного массива целыми числами и создана библиотека по работе с строками. (рис. 16). Демонстрация работы программы – рисунок 17. Демонстрация работы с библиотекой – рисунок 18. На рисунке 19 фрагменте представлен делегат, где используется алгоритм высчитывания произведения правых и левых четных значений и вывод ответа в экран. На рисунке 19 представлен код разработанной библиотеки для работы с строками, где есть метод RemoveSymbols, он удаляет из строки заданные символы, а .метод RemoveReapeatSymbols удаляет только те символы из списка, которые повторяются больше одного раза в строке.

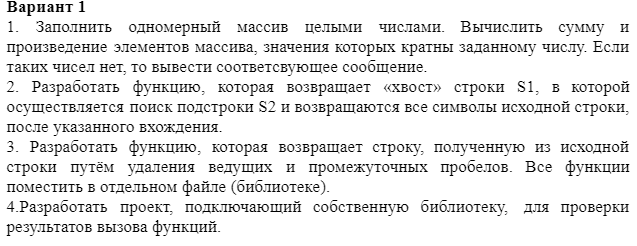
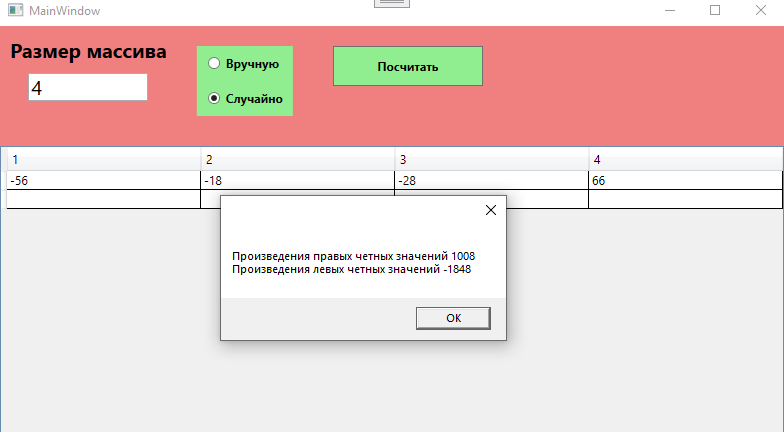
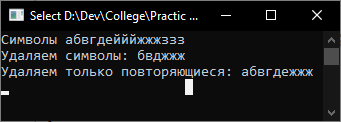


Рисунок 16 – Задание по варианту.



Задание 17 – Демонстрация работы задания 1.



Задание 18 – Демонстрация работы с библиотекой строк.

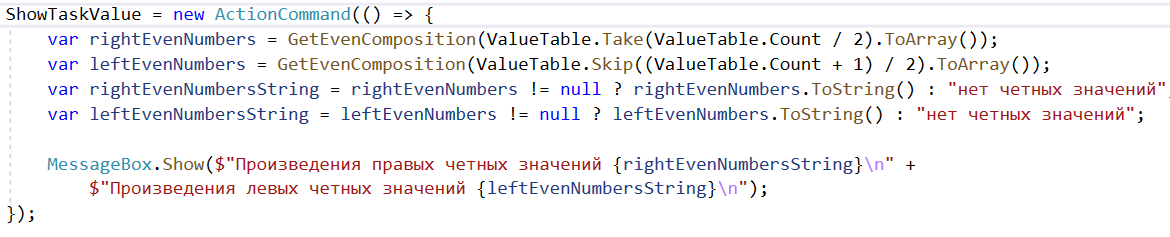


Рисунок 19 – Фрагмент кода реализации алгоритма для программы из задания 1.

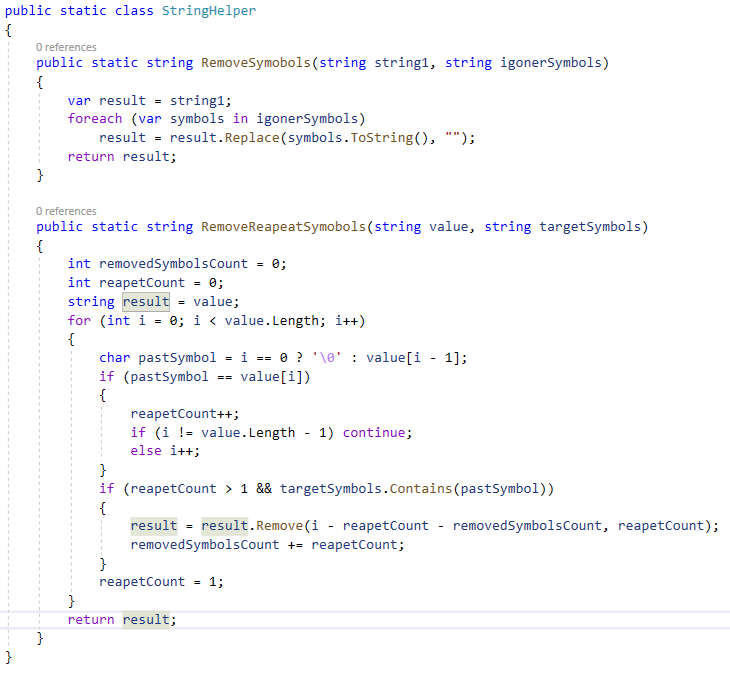


Рисунок 20 – Реализация библиотеки для работы с строками.

1. **ОБРАБОТКА ДВУМЕРНЫХ МАССИВОВ**

Выполнено конструирование пользовательского интерфейса для организации ввода двумерного массива двумя способами: случайным образом или вручную. Создано две функции: для вычисления количества положительных элементов двумерного массива и для вычисления количества элементов, больших заданного числа. Выполнено задание по выданному варианту (рис. 21). Демонстрация работы – рисунок 22, фрагмент кода реализации – рисунок 23. На фрагменте представлен алгоритм перебора двумерного массива для выбора одиночного элемента и использования его при подсчете произведения относительно всего массива с помощью LINQ и каждого его столбца отдельно и вывод результата пользователю. Далее идет обработчик нажатия на следующую кнопку, для определения кол-во элементов, больше заданного пользователем, определения так же идет через LINQ. multiplayNonZeroElments\_Click перебирает все элементы каждого столбца массива не равные 0, и умножает их и выводит ответ пользователю. showCountOneTwo\_Click выводит кол-во строки подходящих по описанию алгоритма связанного с последовательностью 1, 2, 3…N.

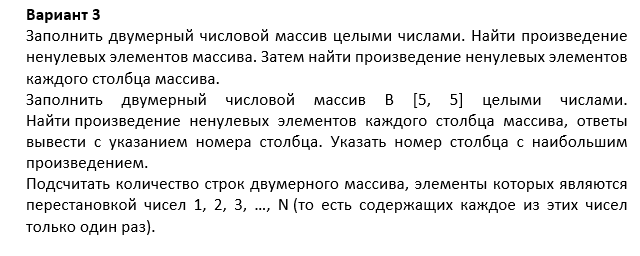


Рисунок 21 – Задание по варианту.

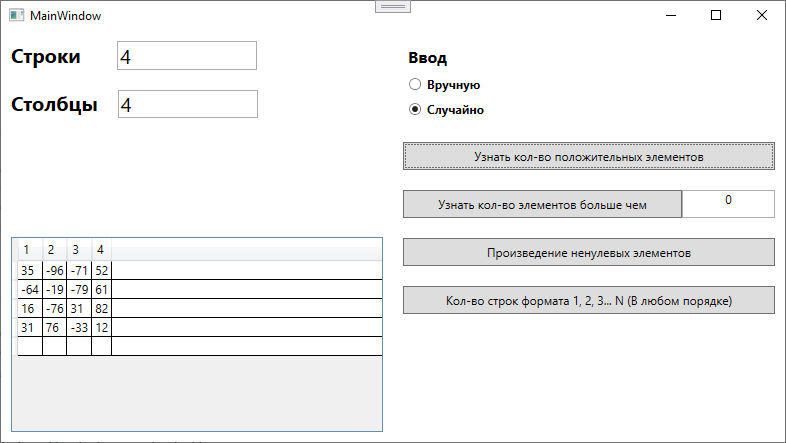


Рисунок 22 – Демонстрация работы приложения.



Рисунок 23 – Фрагмент кода реализации.

1. **РАБОТА С ФАЙЛАМИ ПОСЛЕДОВАТЕЛЬНОГО ДОСТУПА**

Была разработана программа, которая при нажатии кнопки «Открыть файл» обеспечивает открытие текстового файла по выбору пользователя с помощью диалогового окна и загрузку его содержимого в поле многострочного редактора. Обеспечен поиск заданного слова (или сочетания букв) в текстовом редакторе и вывод результата поиска в отдельном окне. Выполнено задание в соответствии с вариантом (рис. 24).

Так же была разработана программа для работы с двумя файлами и записью в конец одного из результата и программа сохраняющая файл в двоичные данных. Демонстрация работы – рисунок 25 задания 1, рисунок 26 задание 2 и 3. На рисунке 27 представлен код, открывающий файл и записывающий его содержимое в поля текста 1 и 2, так же код для поиска совпадений в двух текстовых полях. На рисунке 28 предоставлен фрагмент кода реализации сравнения двух файлов и записи в один из них результата в соответствие задание 2. На рисунке 29 предоставлено сохранение файла в байтовом формате и чтение этого файла в соответствие с заданием 3.

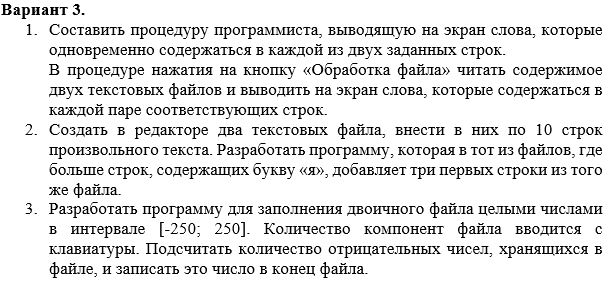


Рисунок 24 – Задание по варианту.

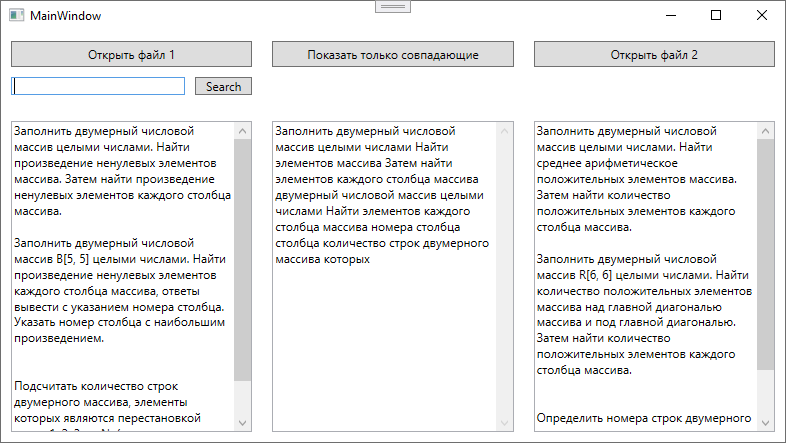


Рисунок 25 – Демонстрация работы задания 1.

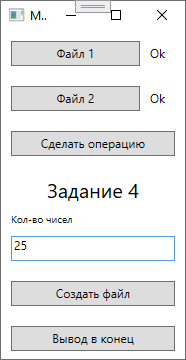


Рисунок 26 – Демонстрация работы задания 2 и 3.

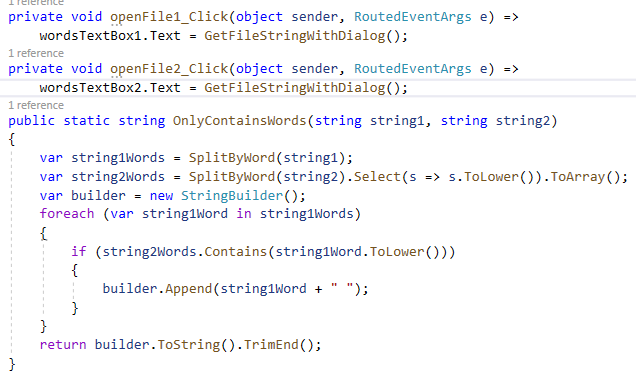


Рисунок 27 – Фрагмент кода задания 1.

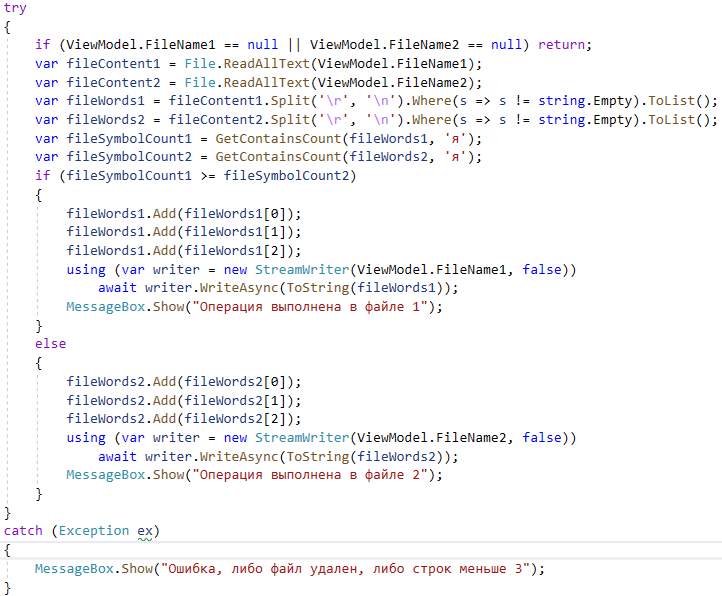


Рисунок 28 – Фрагмент кода реализации сравнения двух файлов и записи в один из них результата.

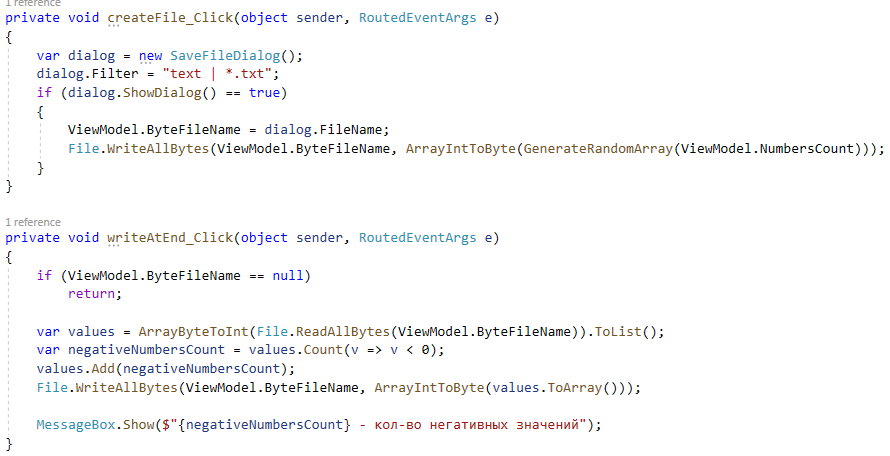


Рисунок 29 – Фрагмент кода реализации задания 3.

1. **ДИНАМИЧЕСКИЕ СТРУКТУРЫ ДАННЫХ**

Была разработана программа, которая получает текст, состоящий из непустой последовательности слов из латинских букв, разделённых запятыми, за последним словом – точка; каждое слово состоит не более, чем из 10 символов. Требуется найти k – количество слов, которые удовлетворяют условию, заданному вариантом задания (рис. 28). Демонстрация работы – рисунок 29. Так же требуется разработать программу, которая пользуется файлом как таблицей для записи продуктов, для последующего добавления, изменения, просмотра, удаления, фильтрации по дате и поиска по названию. Демонстрация работы – рисунок 30 и 31. По мимо этого нужно посчитать кол-во символов в тексте удовлетворяющего условию из варианта, пример рисунок 32. Фрагмент кода реализации первого задания – рисунок 33. Код для взаимодействия с файлом как с таблице предоставлен на рисунке 34. На рисунке 35 предоставлен пример работы программы над текстом, где программа считает кол-во слов повторяющийся буквой «н».

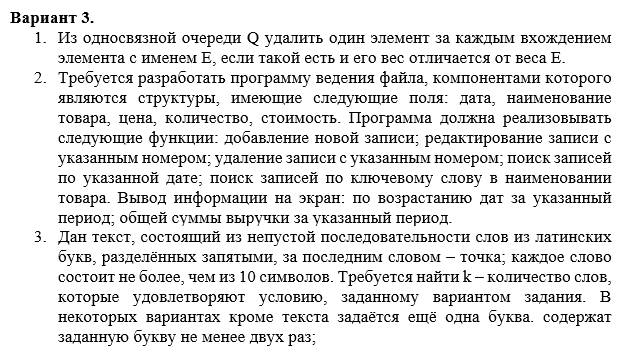


Рисунок 28 – Задание по варианту.

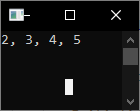


Рисунок 29 – Демонстрация работы программы из задания 1.

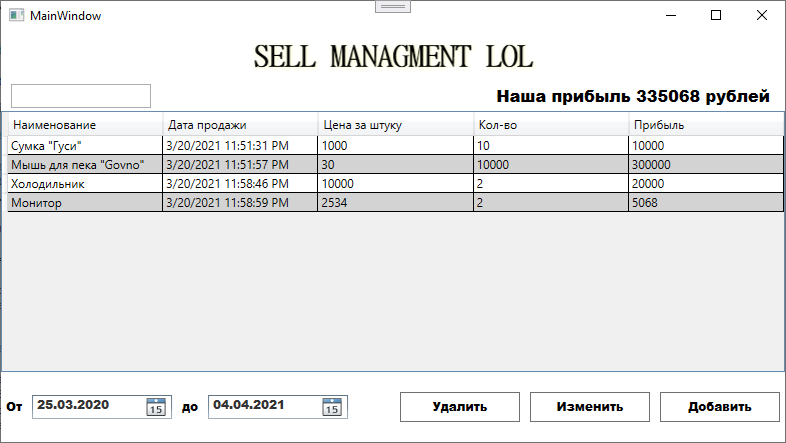


Рисунок 30 – Демонстрация работы программы из задания 2, список товаров.

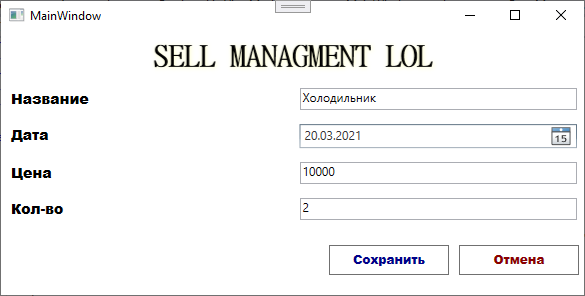


Рисунок 31 – Демонстрация работы программы из задания 2, добавления/изменения товара.

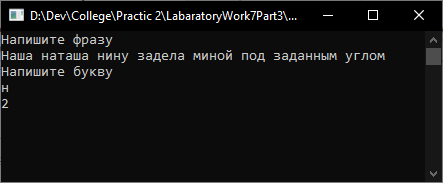


Рисунок 32 – Демонстрация работы программы из задания 3.

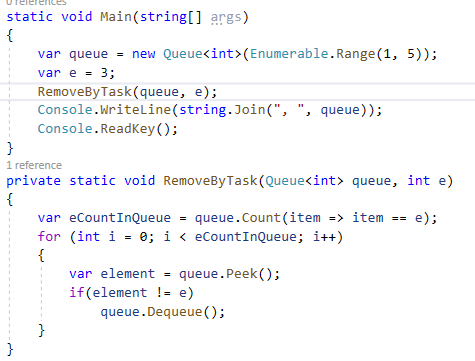


Рисунок 33 – Реализация программы из задания 1

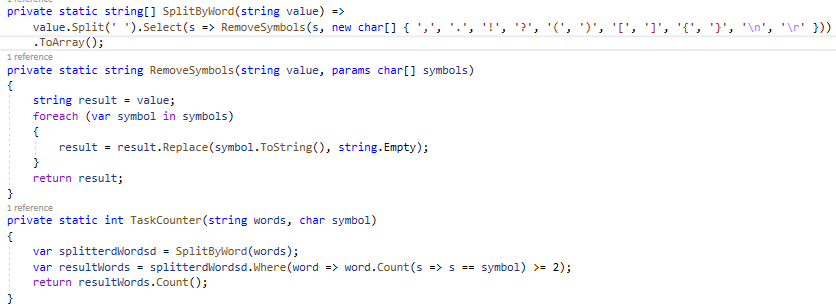


Рисунок 34 – Реализация программы из задания 3

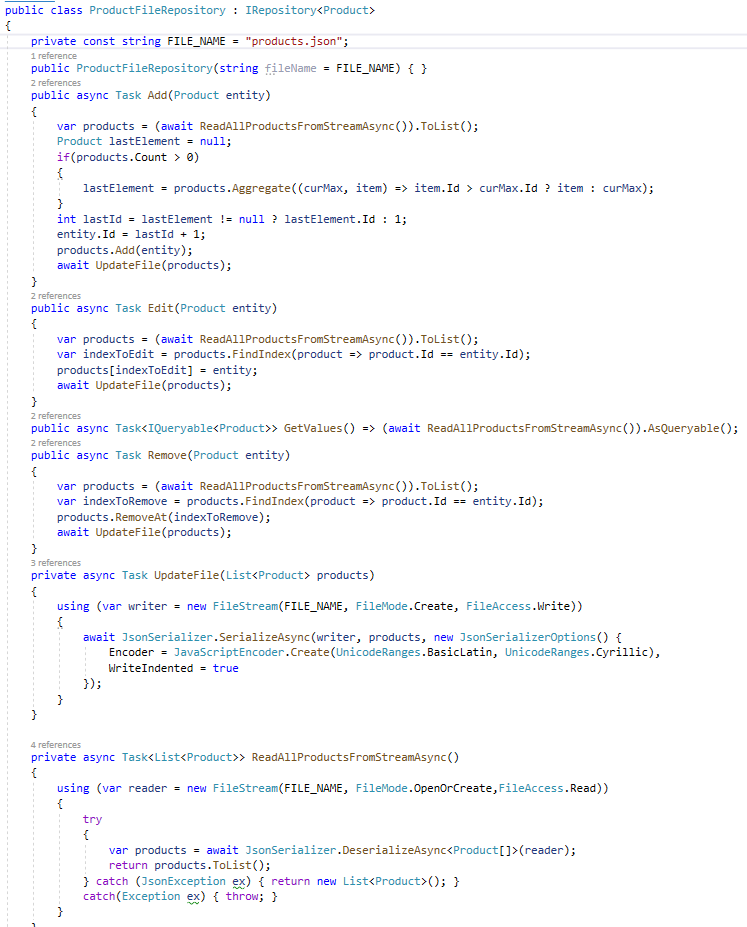


Рисунок 35 – Реализация работы с таблицей в файле из задания 2 .

1. **РАБОТА С ФАЙЛОВОЙ СИСТЕМОЙ**

Была разработана программа, которая получает символьный файл f. Считает число вхождений в файл каждой из букв a, b, c, d, e, f. Результат выводит в файл g в виде таблицы с комментариями (рис. 36). Демонстрация работы – рисунок 37, фрагмент кода реализации – рисунок 41. Так же создается файл table.txt, показывающий сколько раз встречается каждый конкретный символ – демонстрация файла 38, код для реализации – 42. Так была разработана программа для работы с коллекций книг заключенной в файл. Рисунок 39 – Демонстрация работы программы задания. Рисунок 43 – Код для получения и записи коллекции книг в файл.

По заданию 4 было разработана программа способная искать фразу в группе файлов расположенный в директории и ее поддиректориях. На рисунке 40 продемонстрирована работа приложения, а на рисунке 44 код реализующий поиск.

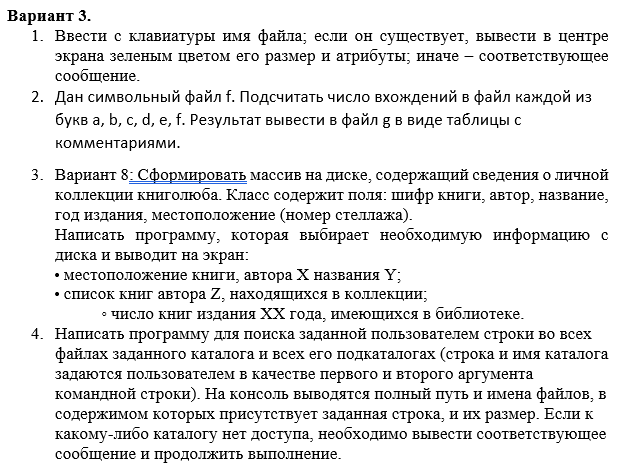


Рисунок 36 – Задание по варианту.

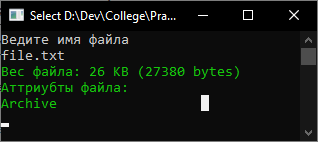


Рисунок 37 – Демонстрация работы программы задания 1.

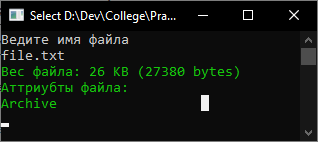


Рисунок 38 – Демонстрация работы программы задания 1.

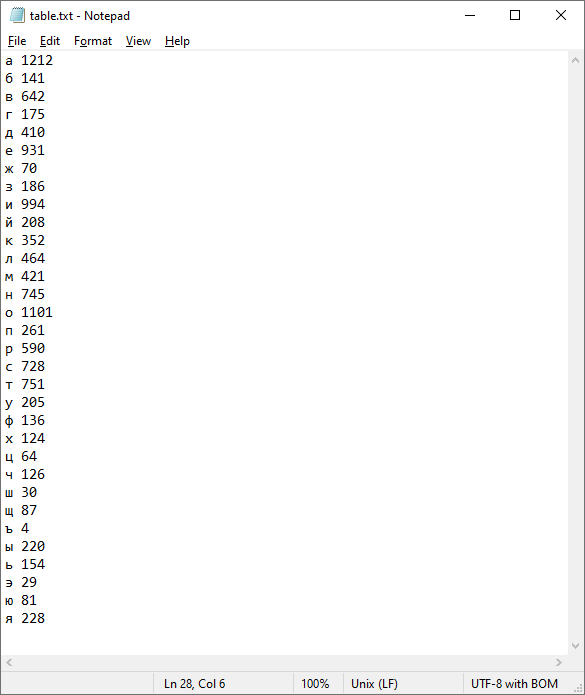


Рисунок 39 – Демонстрация работы программы задания 2.

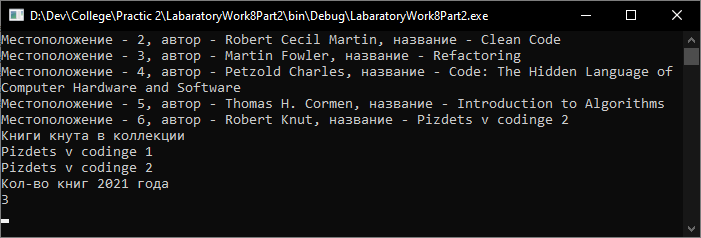


Рисунок 40 – Демонстрация работы программы задания 3

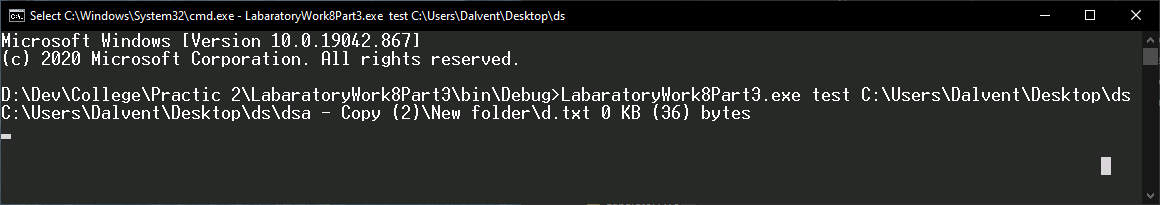


Рисунок 41 – Демонстрация работы программы задания 4



Рисунок 42 – Код, выводящий информацию о файле.

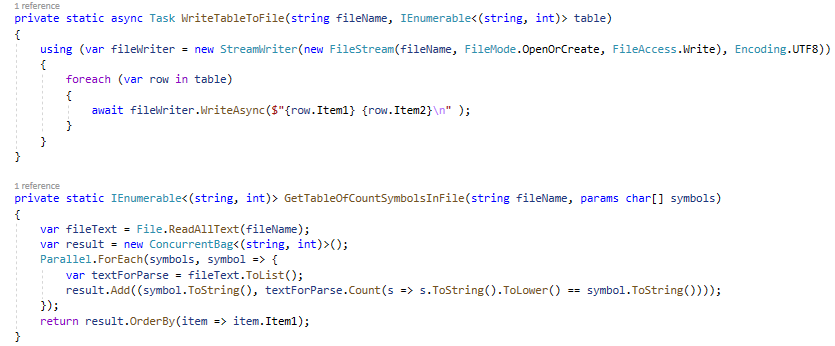


Рисунок 43 – Код для создания таблицы с информацией о кол-во символов в выбранном файле.



Рисунок 44 – Код для получения и записи коллекции книг в файл.

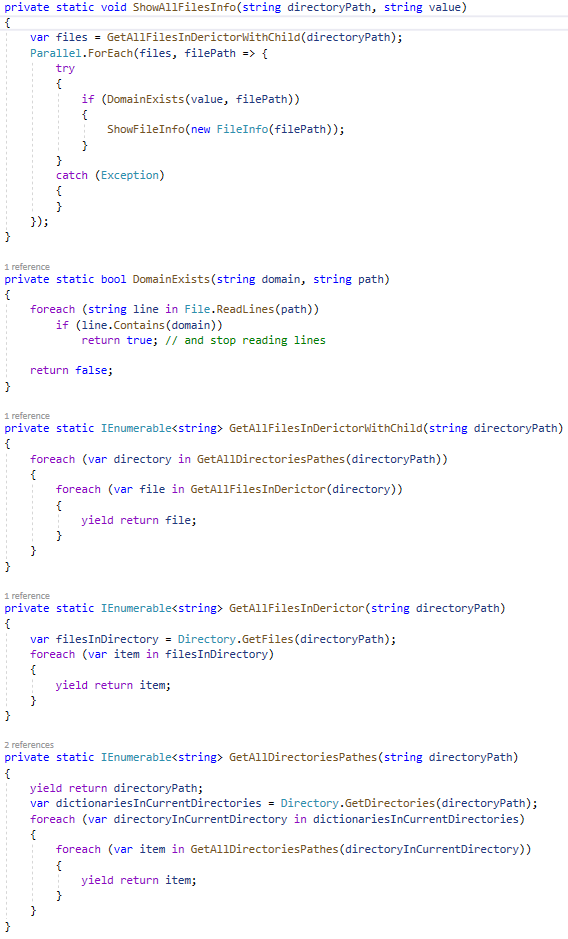


Рисунок 45 – Код реализующий поиск по файлам в директории.

**ЗАКЛЮЧЕНИЕ**

В результате практики получены следующие результаты:

1. Созданы приложения с использованием функций программиста для обработки данных.
2. Проведена работа с графикой в приложениях на базе .NET Framework и WPF.
3. Отработано конструирование интерфейса для пользователя.
4. Освоена работа с файлами и файловой системой операционной системы Windows.

Список литературы:

* <https://docs.microsoft.com/ru-ru/dotnet/api/?view=netframework-4.8> – официальная документация по API .NET Framework
* <https://metanit.com/sharp/> - сайт по языку программирования C# и платформе .NET.