**3 ПРОЦЕДУРЫ И ФУНКЦИИ – МЕТОДЫ КЛАССА**

Задание №1. Разработать метод , который для заданного натурального числа n и вещественного x находит значение выражения . Вычислить с помощью данного метода значение выражения:

Листинг программы:

static void Main(string[] args)

{

static double Formula(int n, double x)

{

return Math.Pow(x, n) / n;

}

Console.WriteLine(Formula(2, 4) + Formula(4, 4) + Formula(6, 4));

}

Таблица 3.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2, 4, 4, 4, 6, 4 | 754,6666 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 3.1.



Рисунок 3.1 – Результат работы программы

Источник: собственная разработка

Задание №2. Построить таблицу значений функции для c шагом h. Для решения задачи использовать вспомогательный метод:

Листинг программы:

static void Main(string[] args)

{

Console.Write("a=");

double a = double.Parse(Console.ReadLine());

Console.Write("b=");

double b = double.Parse(Console.ReadLine());

Console.Write("h=");

double h = double.Parse(Console.ReadLine());

for (double i = a; i <= b; i += h)

Console.WriteLine("f({0:f2})={1:f4}", i, f(i));

Console.WriteLine("Перегрузка!--------------");

double y = 0;

for (double i = a; i <= b; i += h)

{

f(i, out y);

Console.WriteLine("f({0:f2})={1:f4}", i, y);

}

}

public static double f(double x)

{

double y;

if (x > 1) y = 1 / Math.Pow(1 + x, 2);

else if (x < 1) y = Math.Pow(Math.Pow(x, 2) - 1, 2);

else y = 0;

return y;

}

public static void f(double x, out double y)

{

if (x > 1) y = 1 / Math.Pow(1 + x, 2);

else if (x < 1) y = Math.Pow(Math.Pow(x, 2) - 1, 2);

else y = 0;

}

Таблица 3.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4, 7, 8 | 0,0400 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 3.2.

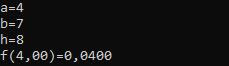


Рисунок 3.2 – Результат работы программы

Источник: собственная разработка

Задание №3.Перегрузите метод из предыдущего раздела так, чтобы его сигнатура (заголовок) соответствовала виду static void f (double x, double y). Продемонстрируйте работу перегруженных методов.

Листинг программы:

public static void f(double x, out double y)

{

if (x > 1) y = 1 / Math.Pow(1 + x, 2);

else if (x < 1) y = Math.Pow(Math.Pow(x, 2) - 1, 2);

else y = 0;

}

Таблица 3.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4, 7, 8 | 0,0400 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 3.3.

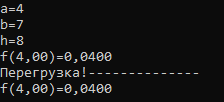


Рисунок 3.3 – Результат работы программы

Источник: собственная разработка

Задание №4. Создайте проект, в котором опишите класс для решения задачи Вашего варианта. Разрабатываемый класс должен содержать следующие элементы: скрытые и открытые поля, конструкторы, перегруженные операции. В программе должна выполняться проверка всех разработанных элементов класса.

Описать класс для работы с двумерным массивом вещественных чисел. Обеспечить добавление к первому столбцу столбца заданного номера (перегрузка операции +).

Листинг программы:

class Array

{

private double[,] \_data;

public int RowCount => \_data.GetLength(0);

public int ColumnCount => \_data.GetLength(1);

public double this[int x, int y]

{

get { if (x < 0 || x > RowCount || y < 0 || y > ColumnCount) { throw new Exception(); } return \_data[x, y]; }

set { \_data[x, y] = value; }

}

public Array(int dim1, int dim2)

{

\_data = new double[dim1, dim2];

}

public void Print()

{

for (int i = 0; i < RowCount; i++)

{

for (int j = 0; j < ColumnCount; j++)

{

Console.Write(\_data[i, j] + " ");

}

Console.WriteLine();

}

}

public int GetLength(int d)

{

return \_data.GetLength(d);

}

public double GetValue(int i1, int i2)

{

return \_data[i1, i2];

}

public void SetValue(int i1, int i2, int val)

{

\_data[i1, i2] = val;

}

public static Array operator +(Array array, int index)

{

if (index < 0 || index > array.ColumnCount)

{

throw new ArgumentException();

}

for (int i = 0; i < array.RowCount; i++)

{

array[i, 0] += array[i, index];

}

return array;

}

}

class Temp

{

static void Main(string[] args)

{

Console.Write("Создаём двумерный массив [{0},{1}]...", 5, 5);

Array array1 = new Array(5, 5);

Console.Write("\nСоздаём двумерный массив [{0},{1}]...", 5, 5);

Array array2 = new Array(5, 5);

Console.WriteLine("\n");

int min = -10, max = 10;

Random r = new Random();

Console.WriteLine("Заполняем массив 1 случайными числами из отрезка [{0},{1}]", min, max);

for (int i = 0; i < array1.GetLength(0); i++)

{

for (int j = 0; j < array1.GetLength(1); j++)

{

array1.SetValue(i, j, r.Next(min, max));

Console.Write("{0,3} ", array1.GetValue(i, j));

}

Console.WriteLine();

}

Console.WriteLine("Заполняем массив 2 случайными числами из отрезка [{0},{1}]", min, max);

for (int i = 0; i < array2.GetLength(0); i++)

{

for (int j = 0; j < array2.GetLength(1); j++)

{

array2.SetValue(i, j, r.Next(min, max));

Console.Write("{0,3} ", array2.GetValue(i, j));

}

Console.WriteLine();

}

Console.WriteLine("Добавление 4-ого столбца к первому для перваго массива: ");

array1 += 4;

array1.Print();

Console.WriteLine();

}

}

Таблица 3.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5, 5 | Array |

Источник: собственная разработка

Анализ результатов представлен на рисунке 3.4.

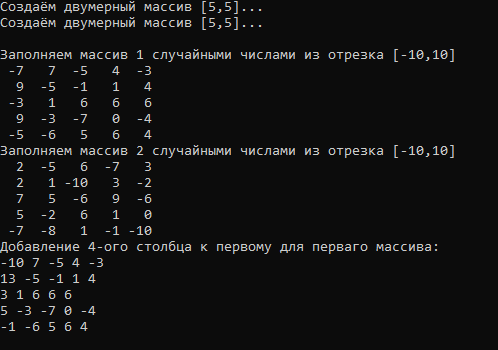


Рисунок 3.4 – Результат работы программы

Источник: собственная разработка

Задание №5. Каждый разрабатываемый класс должен, как правило, содержать следующие элементы:

скрытые поля;

конструкторы с параметрами и без;

параметров;

методы;

свойства;

перегруженные операции.

В программе должна выполняться проверка всех разработанных элементов класса.

Описать класс «предметный указатель». Каждый компонент указателя содержит слово и номера страниц, на которых это слово встречается.

Количество номеров страниц, относящихся к одному слову, от одного до десяти. Предусмотреть возможность формирования указателя с клавиатуры и из файла, вывода указателя, вывода номеров страниц для заданного слова, удаления элемента из указателя.

Написать программу, демонстрирующую все разработанные элементы класса.

Листинг программы:

class Book

{

private Dictionary<string, int[]> \_index = new Dictionary<string, int[]>();

public Dictionary<string, int[]> Index

{

get { return \_index; }

}

public string LoadFromFile(string filename)

{

var file = new System.IO.StreamReader(filename);

return file.ReadToEnd();

}

public List<int> GetPages(string word)

{

List<int> pages = new List<int>();

foreach (KeyValuePair<string, int[]> item in \_index)

{

if (item.Key == word)

{

pages.AddRange(item.Value);

}

}

return pages;

}

}

class Program

{

static void Main(string[] args)

{

var filePath = "Content.txt";

var text = LoadFromFile(filePath);

var content= PageParser.Parse(text);

var ukazatel = new Ukazatel(content);

var sections= ukazatel.GetSections();

PrintSections(sections);

Console.ReadLine();

}

private static void PrintSections(Section[] sections)

{

foreach (var section in sections)

{

Console.WriteLine("---------------");

Console.WriteLine(section.Letter);

foreach (var word in section.Words)

{

Console.WriteLine($"\t {word} | {string.Join(" ",word.Value)}");

}

}

}

public static string LoadFromFile(string filename)

{

var file = new System.IO.StreamReader(filename);

return file.ReadToEnd();

}

}

Таблица 3.5 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Content.txt | Dictionary<> |

Источник: собственная разработка

Анализ результата формирования указателя из файла представлен на рисунке 3.5.

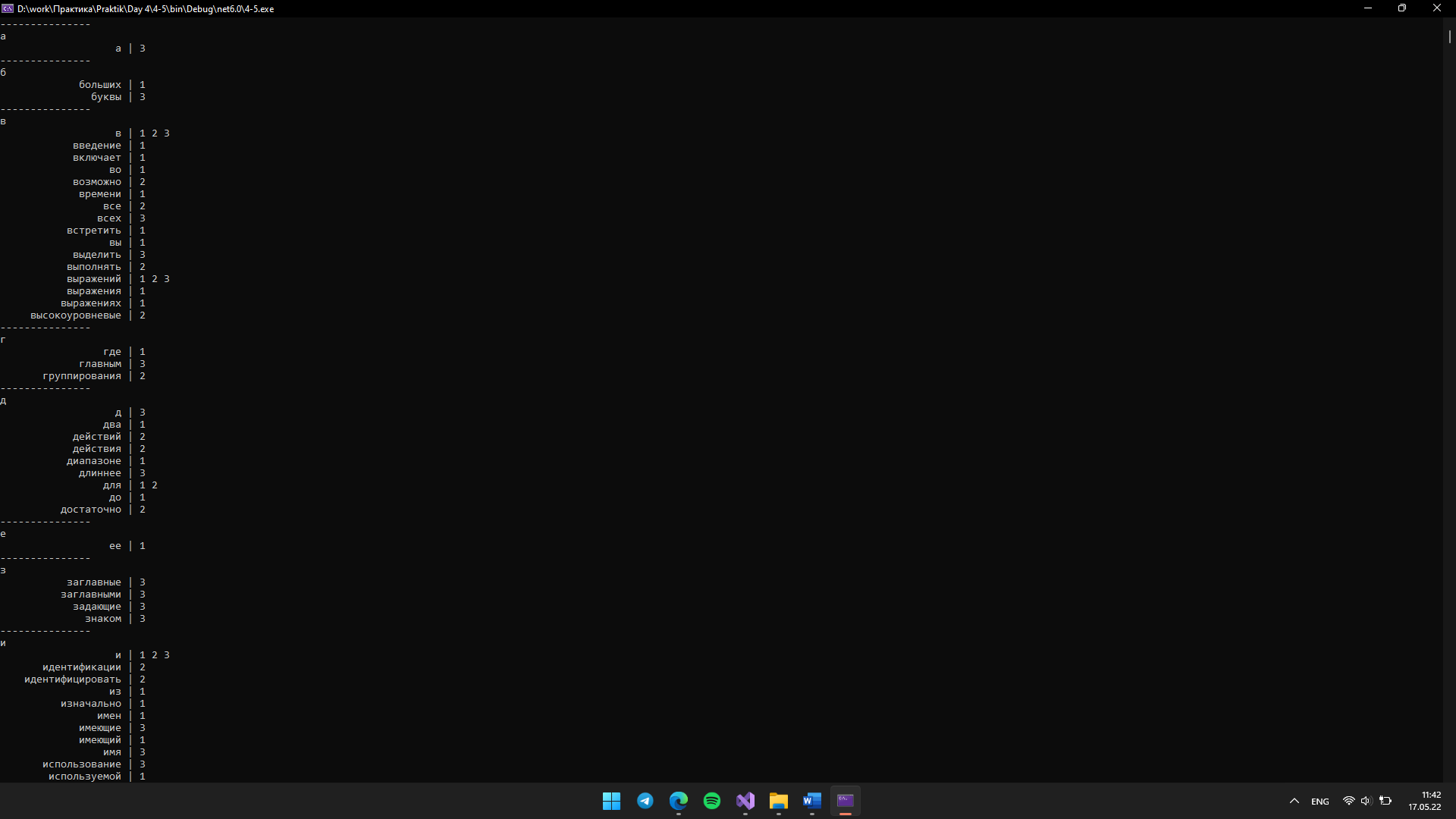


Рисунок 3.5 – Результат работы программы

Источник: собственная разработка