**3 ПРОЦЕДУРЫ И ФУНКЦИИ. ПЕРЕГРУЗКА ФУНКЦИЙ (МЕТОДОВ) И ОПЕРАТОРОВ**

Задание 1. Написать программу, которая вычисляет значение функции.



Рисунок 1.1 – Уравнение функции

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyNameSpace

{

class Program

{

static double min(double a, double b)

{

return (a < b) ? a : b;

}

static void Main(string[] args)

{

Console.Write("x=");

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

Console.Write("y=");

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

Console.Write("z=");

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

Console.Write("v=");

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

Console.WriteLine(min(min(x, y), min(z, v)));

}

}

}

Таблица 1.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5, 4, 3, 2 | 2 |

Анализ результатов:

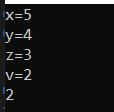
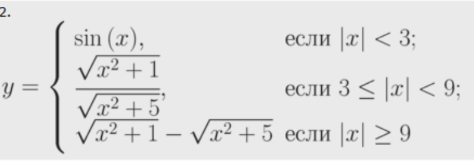


Рисунок 1.2 – Результат работы программы

Источник: собственная разработка

Задание 2. Написать программу. Построение таблицу значений функции.



Листинг программы:

using static System.Console;

namespace MyNamespace

{

class Program

{

public static void Main()

{

WriteLine("Введите x");

double x = Convert.ToDouble(Console.ReadLine());

Console.WriteLine(CalculateX(x));

}

static double CalculateX(double x)

{

double y = 0;

if (Math.Abs(x) < 3)

{

y = Math.Sin(x);

}

else if (3 <= Math.Abs(x) && Math.Abs(x) < 9)

{

y = (Math.Sqrt(x\*x + 1)) / (Math.Sqrt(x\*x+5));

}

else if (Math.Abs(x) >= 9)

{

y = (Math.Sqrt(x\*x+1)) - (Math.Sqrt(x\*x+5));

};

return y;

}

}

}

Таблица 1.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5 | 0,9309493362512626 |

Анализ результатов:

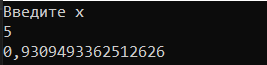


Рисунок 1.3 – Результат работы программы

Источник: собственная разработка

Задание 3. Описать класс для работы с двумерным массивом строк фиксированной длины. Обеспечить сравнение массивов на равенство (перегрузку операции == для поэлементного сравнения).

Листинг программы:

using System.Runtime.CompilerServices;

using static System.Console;

namespace MyNamespace

{

class Program

{

public static void Main()

{

Array first = new Array();

first.Length = 10;

int[,] First = new int[first.Length, first.Length];

Console.WriteLine("Массив");

first.fillArr(First, first.Length);

Console.WriteLine("Остатки:");

first.arrDiv(First, 2);

Console.ReadLine();

}

class Array

{

public int Length;

int[] Arr;

public void fillArr(int[,] Arr, int Length)

{

Random r = new Random();

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

{

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

{

Arr[i, j] = r.Next(0, Length + 1);

Console.Write("{0}" + ", ", Arr[i, j]);

}

}

Console.ReadLine();

}

public void arrDiv(int[,] a, int temp)

{

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

{

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

{

a[i, j] %= temp;

Console.Write("{0}" + ", ", a[i, j]);

}

}

}

}

}

}

Таблица 1.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Массив |

Анализ результатов:

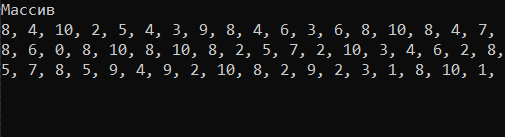


Рисунок 1.4 – Результат работы программы

Источник: собственная разработка

Задание 4. Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (по автору, по году издания или категории), добавления книг в библиотеку, удаления книг из нее, доступа к книге по номеру. Написать программу, демонстрирующую все разработанные элементы

класса.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

public class Library

{

public int Number

{

get; // исправить, получение значения свойства

set; // установка значения свойства (методы доступа)

}

public string Author

{

get;

set;

}

public int Year

{

get;

set;

}

public string Category

{

get;

set;

}

List<Library> LibraryList = new List<Library>();

public void Add(Library Library)

{

LibraryList.Add(Library);

}

public void Remove(Library Library)

{

LibraryList.Remove(Library);

}

public override bool Equals(object obj)

{

if (obj == null) return false;

Library objLibrary = obj as Library;

if (objLibrary == null)

return false;

return true;

}

public override string ToString()

{

return Number + " " + Author + " " + Year + " " + Category;

}

public void ShowScreen()

{

foreach (var VARIABLE in LibraryList)

{

Console.WriteLine(VARIABLE.ToString());

}

Console.WriteLine("\n");

}

public void Search(string str, int number)

{

foreach (var VARIABLE in LibraryList)

{

if (number == 1 && VARIABLE.Author == str)

{

Console.WriteLine(VARIABLE.ToString());

}

else if (number == 2 && VARIABLE.Year == Convert.ToInt32(str))

{

Console.WriteLine(VARIABLE.ToString());

}

else if (number == 3 && VARIABLE.Category == str)

{

Console.WriteLine(VARIABLE.ToString());

}

else if (number == 4 && VARIABLE.Number == Convert.ToInt32(str))

{

Console.WriteLine(VARIABLE.ToString());

}

}

}

static void Main(string[] args)

{

Library Library = new Library();

Library.Add(new Library { Number = 1, Author = "Ессенин", Year = 1900, Category = "Стихи" });

Library.Add(new Library { Number = 2, Author = "Пушкин", Year = 1910, Category = "Рассказы" });

Library.Add(new Library { Number = 3, Author = "Лермонтов", Year = 1920, Category = "Басни" });

Library.Add(new Library { Number = 4, Author = "Достоевский", Year = 1910, Category = "Стихи" });

Library.ShowScreen();

Library.Remove(new Library { Number = 3, Author = "Лермонтов", Year = 1920, Category = "Басни" });

Library.ShowScreen();

Console.WriteLine("По какому критерию будем производить поиск? (По автору-1, год издания-2, по категории-3," +

"по номеру -4)");

int number = int.Parse(Console.ReadLine());

switch (number)

{

case 1:

Console.WriteLine("Введите писателя");

Library.Search(Console.ReadLine(), number);

break;

case 2:

Console.WriteLine("Введите год издания");

Library.Search(Console.ReadLine(), number);

break;

case 3:

Console.WriteLine("Введите категорию");

Library.Search(Console.ReadLine(), number);

break;

case 4:

Console.WriteLine("Введите номер книги");

Library.Search(Console.ReadLine(), number);

break;

default:

Console.WriteLine("Не правильный критерий");

break;

}

Console.ReadLine();

}

}

}

Таблица 1.4 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1, Пушкин | 2 Пушкин 1910 Рассказы |

Анализ результатов:

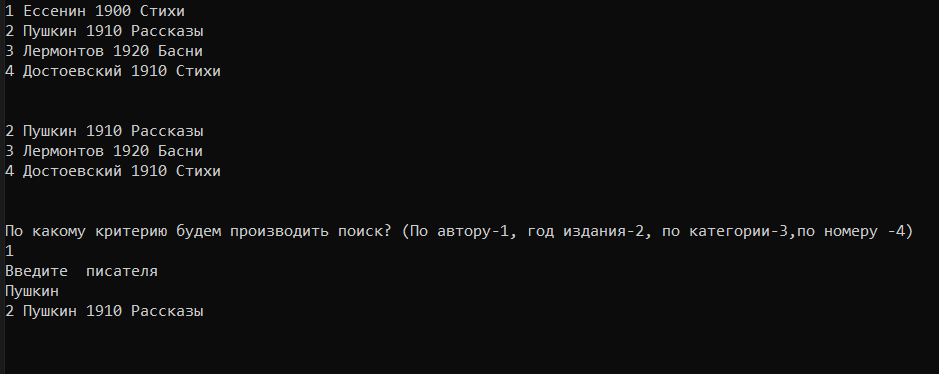


Рисунок 1.5 – Результат работы программы

Источник: собственная разработка