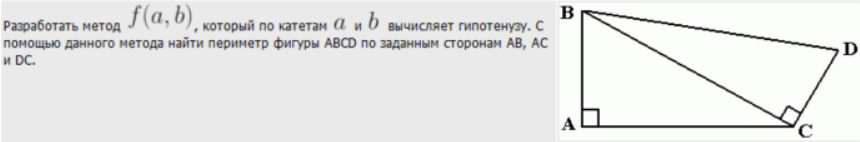
**ПРАКТИЧЕСКОЕ ЗАДАНИЕ 3**

**ПРОЦЕДУРЫ И ФУНКЦИИ – МЕТОДЫ КЛАССА**

Задание 1.



Листинг программы:

namespace ConsoleApp

{

internal class Program

{

static double func(double x, double y, double z)

{

return ((x + y + z) / 2);

}

static void Main(string[] args)

{

double a, b, c, d, e, f, g, s1, s2, s3, s, p1, p2, p3;

Console.Write("Сторона a=");

a = Convert.ToInt32(Console.ReadLine());

Console.Write("Сторона b=");

b = Convert.ToInt32(Console.ReadLine());

Console.Write("Сторона c=");

c = Convert.ToInt32(Console.ReadLine());

Console.Write("Сторона d=");

d = Convert.ToInt32(Console.ReadLine());

Console.Write("Сторона e=");

e = Convert.ToInt32(Console.ReadLine());

Console.Write("Сторона f=");

f = Convert.ToInt32(Console.ReadLine());

Console.Write("Сторона g=");

g = Convert.ToInt32(Console.ReadLine());

p1 = Program.func(a, b, f);

p2 = Program.func(f, g, c);

p3 = Program.func(e, g, d);

s1 = Math.Sqrt(p1 \* (p1 - a) \* (p1 - b) \* (p1 - f));

s2 = Math.Sqrt(p2 \* (p2 - f) \* (p2 - g) \* (p2 - c));

s3 = Math.Sqrt(p3 \* (p3 - e) \* (p3 - g) \* (p3 - d));

s = s1 + s2 + s3;

Console.WriteLine("Площадь пятиугольника равна = {0:F3}", s);

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Сторона a=7  Сторона b=6  Сторона c=4  Сторона d=5  Сторона e=4  Сторона f=3  Сторона g=6 | Площадь пятиугольника равна = 24,199 |

Анализ результатов:

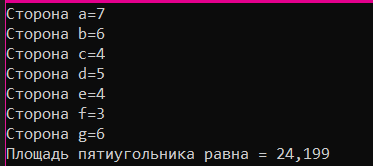
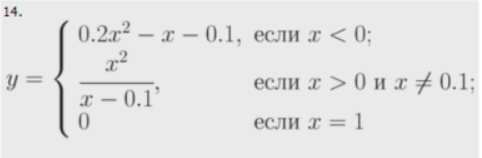
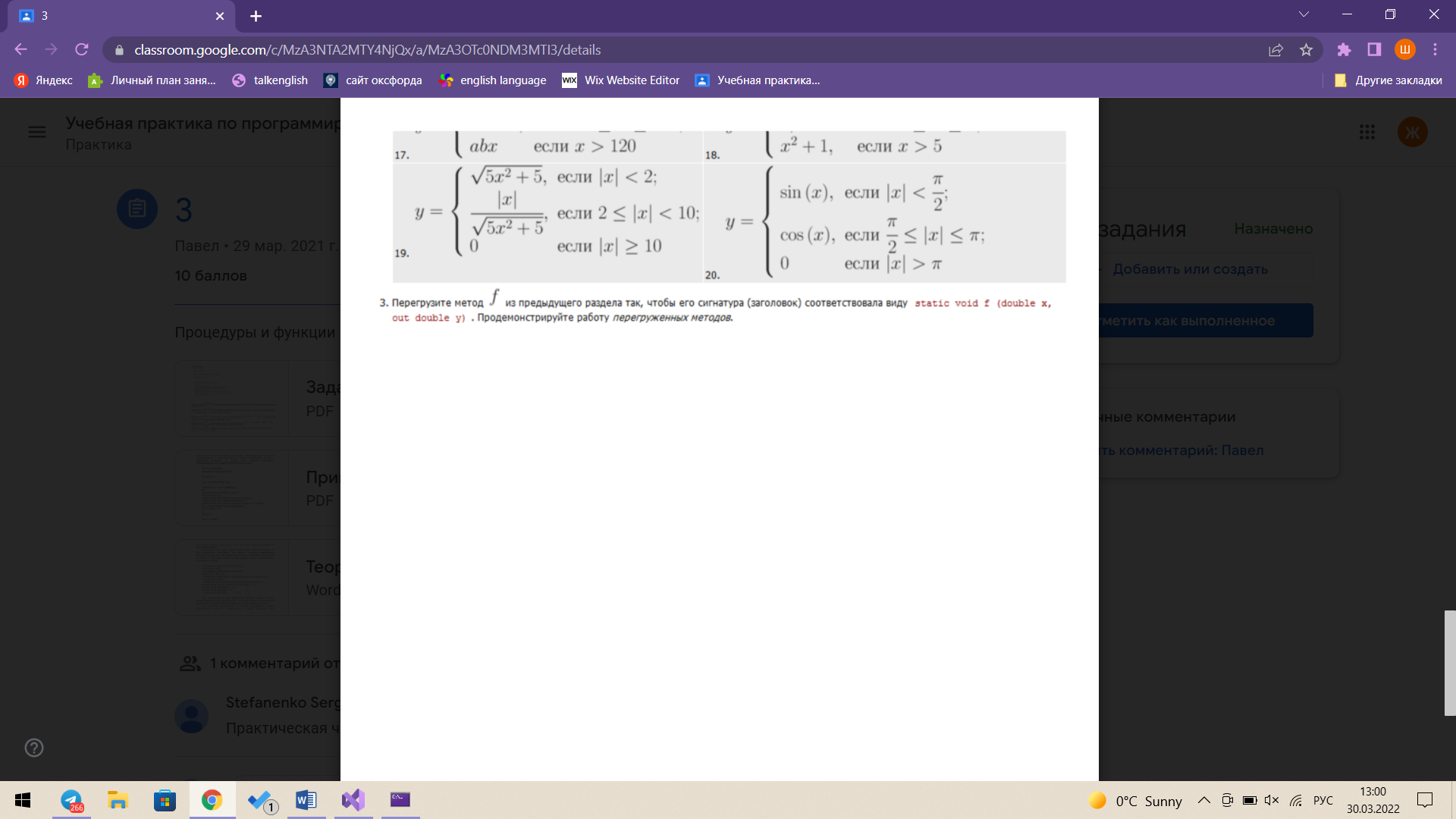


Рисунок 1.1 – Результат работы программы

Задание 2, задание 3.





Листинг программы:

class Program

{

static double f(double x)

{

double y = 0;

if (x < 0) y = (Math.Pow(0.2 \* x, 2) - x - 0.1);

else if (x >= 1) y = 0;

else if ((x > 0) & (x != 0.1)) y = Math.Pow(x, 2) / (x - 0.1);

return y;

}

static void f(double x, out double yk)

{

yk = 0;

if (x < 0) yk = (Math.Pow(0.2 \* x, 2) - x - 0.1);

else if (x >= 1) yk = 0;

else if ((x > 0) & (x != 0.1)) yk = Math.Pow(x, 2) / (x - 0.1);

}

static void Main(string[] args)

{

try

{

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());

double yk = 0;

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

{

f(i, out yk);

Console.WriteLine($"result = {yk}");

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| a = -1  b = 1  h = 1 | result = 0,9400000000000001  result = 0  result = 0 |

Анализ результатов:

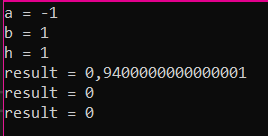
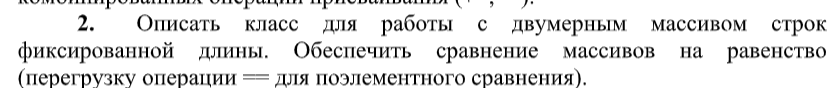


Рисунок 1.2 – Результат работы программы

Задание 4



Листинг программы:

public class Programm

{

static void Main(string[] args)

{ var arrToFill = new string[] { "one", "two", "three", "four", "five", "six", "seven", "eight", "nine", "ten" };

var arr1 = new Massiv<string>(5, 5);

var arr2 = new Massiv<string>(5, 5);

arr1.Fill(arrToFill);

arr2.Fill(arrToFill);

arr1.Print();

Console.WriteLine();

arr2.Print();

if (arr1 == arr2) Console.WriteLine("Arrays are equal!");

else Console.WriteLine("Arrays are not equal!");

Console.ReadLine();

}

}

public class Massiv<T>

{

private readonly T[,] \_arr;

public int RowCount { get; set; }

public int ColumnCount { get; set; }

public Massiv(int rows, int columns)

{

RowCount = rows;

ColumnCount = columns;

\_arr = new T[RowCount, ColumnCount];

}

public void Print(int coloredRow = -1, int coloredColumn = -1)

{

var maxLength = MaxLength();

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

{

Console.Write("|");

if (coloredRow == i) Console.ForegroundColor = ConsoleColor.Red;

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

{ if (coloredColumn == j) Console.ForegroundColor = ConsoleColor.Red;

var anz = maxLength - \_arr[i, j].ToString().Length;

var l = Spaces(anz);

if (coloredRow != i && coloredColumn != j) Console.ResetColor();

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

}

Console.ResetColor();

Console.Write("|\n");

}

}

private static string Spaces(int count)

{

var result = "";

for (int i = 0; i < count; i++) result += " ";

return result;

}

private int MaxLength()

{

int max = int.MinValue;

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

{

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

{

if (\_arr[i, j].ToString().Length > max) max = \_arr[i, j].ToString().Length;

}

}

return max;

}

public void Fill(T[] toFill)

{

var rnd = new Random();

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

{

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

{

T value = toFill[rnd.Next(toFill.Length)];

\_arr[i, j] = value;

}

}

}

public static bool operator ==(Massiv<T> arr1, Massiv<T> arr2)

{

for (int i = 0; i < arr1.RowCount; i++)

{

for (int j = 0; j < arr2.ColumnCount; j++)

{

if (!arr1.\_arr[i, j].Equals(arr2.\_arr[i, j])) return false;

}

}

return true;

}

public static bool operator !=(Massiv<T> arr1, Massiv<T> arr2)

{

return !(arr1 == arr2);

}

public T this[int row, int column]

{

get { return \_arr[row, column]; }

set { \_arr[row, column] = value; }

}

}

Анализ результатов:

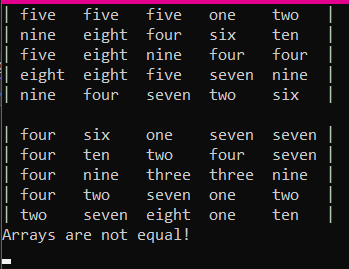


Рисунок 1.3 – Результат работы программы

Задание 5.

Описать класс «склад», содержащий закрытый массив товаров.

Обеспечить следующие возможности:

вывод информации о товаре по номеру с помощью индекса;

вывод на экран информации о товаре, название которого введено с

клавиатуры; если таких товаров нет, выдать соответствующее сообщение;

сортировку товаров по названию магазина, по наименованию и по

цене;

перегруженную операцию сложения товаров, выполняющую

сложение их цен.

Написать программу, демонстрирующую все разработанные элементы

классов.

Листинг программы:

try

{

Sklad sklad = new Sklad();

Tovar tovar = new Tovar("Молоко", "Santa", 150);

Tovar tovar4 = new Tovar("Молоко", "Almi", 100);

Tovar tovar1 = new Tovar("Мороженное", "Almi", 270);

Tovar tovar2 = new Tovar("Яйца", "OldCity", 250);

Tovar tovar3 = new Tovar("Хлеб", "Korona", 400);

sklad.AddTovarRange(tovar, tovar1, tovar2, tovar3, tovar4);

sklad.PrintTovar();

sklad.GetInforation(0);

sklad.GetInformationTovar("Молоко");

sklad.InfoSortPrice();

Console.ReadLine();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

class Tovar

{

private string \_name;

private string \_nameOfShop;

private decimal \_price;

public Tovar(string Name, string NameOfShop, decimal Price)

{

\_name = Name;

\_nameOfShop = NameOfShop;

\_price = Price;

}

public string Name

{

get { return \_name; }

set { \_name = value; }

}

public string NameOfShop

{

get { return \_nameOfShop; }

set { \_nameOfShop = value; }

}

public decimal Price

{

get { return \_price; }

set { \_price = value; }

}

public static decimal operator +(Tovar plane1, Tovar plane2) //перегрузка операторов

{

return plane1.Price + plane2.Price;

}

public override string ToString()

{

return $"{\_name},{\_nameOfShop},{\_price}";

}

}

class Sklad

{

List<Tovar> tovars;

public Sklad()

{

tovars = new List<Tovar>();

}

public void AddTovar(Tovar plane)

{

tovars.Add(plane);

}

public void AddTovarRange(params Tovar[] Planes)

{

tovars.AddRange(Planes);

}

public void GetInforation(int index) // получение информации по индексу

{

if (index > tovars.Count - 1)

{

throw new Exception("Товара не существует");

}

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

Console.WriteLine("Наименование товара: " + tovars[index].Name);

Console.WriteLine("Наименование магазина: " + tovars[index].NameOfShop);

Console.WriteLine("Цена: " + tovars[index].Price);

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

}

public void GetInformationTovar(string userString) // получение информации о товаре по наименованию

{

for (int i = 0; i < tovars.Count; i++)

{

if (userString == tovars[i].Name)

{

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

Console.WriteLine($"Наименование товара: {tovars[i].Name}");

Console.WriteLine($"Наименование магазина: {tovars[i].NameOfShop}");

Console.WriteLine($"Цена: {tovars[i].Price}");

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

}

else

{

Console.WriteLine("Товара не существует");

}

}

}

public void PrintTovar() // получение информации о товаре по наименованию

{

for (int i = 0; i < tovars.Count; i++)

{

Console.WriteLine($"Наименование товара: {tovars[i].Name}");

Console.WriteLine($"Наименование магазина: {tovars[i].NameOfShop}");

Console.WriteLine($"Цена: {tovars[i].Price}");

}

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

}

public void InfoSortPrice() //сортировка по цене

{

var temp = tovars[0];

for (int i = 0; i < tovars.Count - 1; i++)

{

for (int j = i + 1; j < tovars.Count; j++)

{

if (tovars[i].Price > tovars[j].Price)

{

temp = tovars[i];

tovars[i] = tovars[j];

tovars[j] = temp;

}

}

}

Console.WriteLine("Сортировка по цене:");

for (int i = 0; i < tovars.Count; i++)

{

Console.WriteLine(tovars[i]);

}

}

}

Анализ результатов:

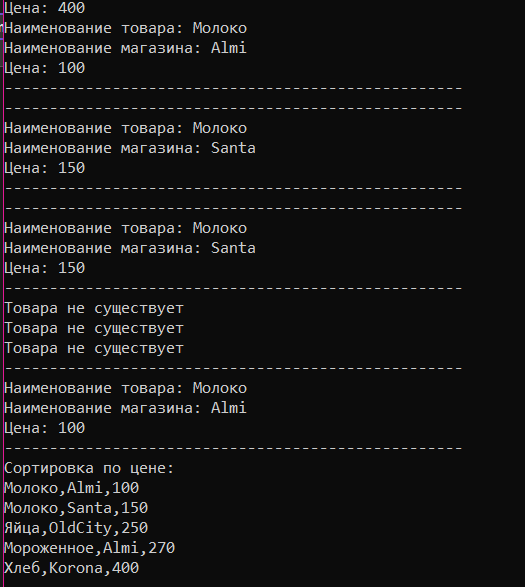


Рисунок 1.4 – Результат работы программы