**3 ПРОЦЕДУРЫ И ФУНКЦИИ – МЕТОДЫ КЛАССА**

Задание №1. Разработать метод , который по длинам сторон треугольника вычисляет его площадь. С помощью данного метода по заданным вещественным числам найти площадь пятиугольника, изображённого на рисунке 3.1.

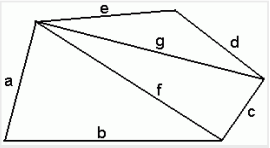


Рисунок 3.1 – Заданный пятиугольник

Источник: собственная разработка

Листинг программы:

static double Square(double[] parametrs)

{

double p = 0;

foreach (var item in parametrs)

{

p += item;

}

p /= 2;

return Math.Sqrt(p \* (p - parametrs[0]) \* (p - parametrs[1]) \* (p - parametrs[2]));

}

double PentagonSquare = Square(new double[] { 5, 6, 7 }) + Square(new double[] { 5, 6, 7 }) + Square(new double[] { 5, 6, 7 });

Console.WriteLine($"Traiangle square - {Math.Round(Square(new double[] { 5, 6, 7 }), 4)}");

Console.WriteLine($"Pentagon square - {Math.Round(PentagonSquare, 4)}");

Таблица 3.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5, 6, 7, 5, 6, 7, 5, 6, 7 | 14.6969, 44.0908 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 3.2.



Рисунок 3.2 – Результат работы программы

Источник: собственная разработка

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

Листинг программы:

static void Main(string[] args)

{

Console.Write("Enter a value: ");

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

Console.Write("Enter b value: ");

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

Console.Write("Enter h value: ");

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("Overload.");

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) > 1) y = -1;

else if ((x - 1) < 1) y = 1;

else y = 0;

return y;

}

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

{

if ((x - 1) > 1) y = -1;

else if ((x - 1) < 1) y = 1;

else y = 0;

}

Таблица 3.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1, 4, 1 | 1.0000, 0.0000, -1.0000, -1.0000 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 3.3.

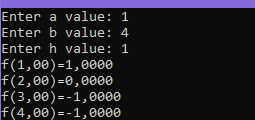


Рисунок 3.3 – Результат работы программы

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) > 1) y = -1;

else if ((x - 1) < 1) y = 1;

else y = 0;

}

Таблица 3.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5, 10, 1 | -1.0000, -1.0000, -1.0000, -1.0000, -1.0000, -1.0000, |

Источник: собственная разработка

Анализ результатов представлен на рисунке 3.4.

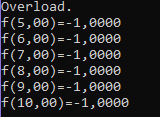


Рисунок 3.4 – Результат работы программы

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 целых чисел. Реализовать возможность выполнения операции нахождения остатков от деления всех элементов массива на заданное число.

Листинг программы:

class Array

{

private int[,] \_data;

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

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

public int 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 int[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 double RemainderSumDevision(int number)

{

double remainderSum = 0;

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

{

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

{

remainderSum += Math.Abs(this.GetValue(i, j)) % number;

}

}

return remainderSum;

}

}

class Temp

{

static void Main(string[] args)

{

Console.Write("Forming two-dimensional array[{0},{1}]...", 10, 10);

Array array1 = new Array(10, 10);

Console.Write("\nCreating two-dimensional array[{0},{1}]...", 10, 10);

Array array2 = new Array(10, 10);

Console.WriteLine("\n");

int min = -10, max = 10;

Random r = new Random();

Console.WriteLine("Filling first array [{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("Filling second array [{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($"Remainder of division on 7: {array1.RemainderSumDevision(7)}");

}

}

Таблица 3.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 10, 10, 7 | Array, 229 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 3.5.

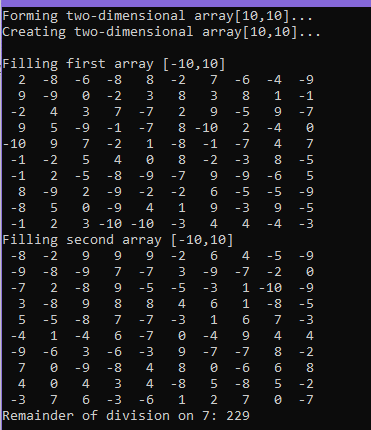


Рисунок 3.5 – Результат работы программы

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 «самолет», содержащий следующие закрытые поля:

название пункта назначения;

шестизначный номер рейса;

время отправления.

Предусмотреть свойства для получения состояния объекта.

Листинг программы:

internal class Plane

{

private string \_Destination;

private int \_Number;

private DateTime \_Time;

public Plane(string destitnation, int number, DateTime time)

{

\_Destination = destitnation;

\_Number = number;

\_Time = time;

}

public string destination { get { return \_Destination; } set { \_Destination = value; } }

public int number { get { return \_Number; } set { \_Number = value; } }

public DateTime time { get { return \_Time; } set { \_Time = value; } }

}

using System;

using z5.Modules;

namespace z5

{

class Program

{

static void Main(string[] args)

{

Plane plane = new Plane("Tokyo", 771199, new DateTime(2022, 07, 20, 18, 35, 5));

Console.WriteLine($"Destination town: {plane.destination}");

Console.WriteLine($"Plane ID: {plane.number}");

Console.WriteLine($"Departure time {plane.time}");

}

}

}

Таблица 3.5 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| \_Destination, \_Number; \_Time; | Tokyo, 771199, 20.07.2022, 18:35:05 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 3.6.



Рисунок 3.6 – Результат работы программы

Источник: собственная разработка