**3.Процедуры и функции – методы класса**

Задание №1. Разработать метод f(x), который вычисляет значение по следующей формуле: f(x) = cos(2x) + sin(x-3). Определить, в какой из точек a или b, функция принимает наименьшее значение.

Листинг программы:

using static System.Math;

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

if (Func(numA) > Func(numB))

{

Console.WriteLine($"В точке b:{numB} со значением y:{Func(numB):.###} " +

$"функция принимает наименьшее значение");

}

else

{

Console.WriteLine($"В точке a:{numA} со значением y:{Func(numA):.###} " +

$"функция принимает наименьшее значение");

}

double Func(double x) => Cos(2 \* x) + Sin(x - 3);

Таблица 3.1 – Входные и выходные данные программы задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| a:4  b:5 | В точке b:5 со значением y:,07 функция принимает наименьшее значение |

Анализ результатов:

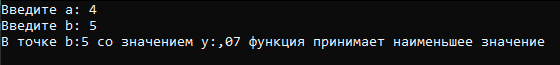


Рисунок 3.1 – Результат выполнения программы задание №1

Задание №2. Постройте таблицу значений функции y = f(x) для x принадлежащего [a,b] с шагом h.

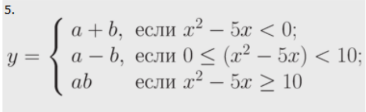


Рисунок 3.2 –Функция для задания №2

Листинг программы:

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

Console.Write("Введите шаг h: ");

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

for (double i = numA; i <= numB; i += h)

{

if (F.Fc(i) < 0)

{

//Задание 3

F.Fc(i, out double y);

Console.WriteLine($"x:{i} y:{y} a + b = {numA + numB}");

}

else if (0 <= F.Fc(i) && F.Fc(i) < 10)

{

Console.WriteLine($"x:{i} y:{F.Fc(i)} a - b = {numA - numB}");

}

else

{

Console.WriteLine($"x:{i} y:{F.Fc(i)} a \* b = {numA \* numB}");

}

}

class F

{

public static double Fc(double x) => x \* x - 5 \* x;

//Задание 3

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

{

y = 0;

y = x \* x - 5 \* x;

}

}

Таблица 3.2 – Входные и выходные данные программы задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| a:2  b:7  h:1 | x:2 y:-6 a + b = 9  x:3 y:-6 a + b = 9  x:4 y:-4 a + b = 9  x:5 y:0 a - b = -5  x:6 y:6 a - b = -5  x:7 y:14 a \* b = 14 |

Анализ результатов:

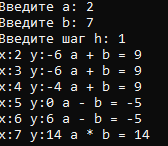


Рисунок 3.3 – Результат выполнения программы задание №2

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

Листинг программы:

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

Console.Write("Введите шаг h: ");

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

for (double i = numA; i <= numB; i += h)

{

if (F.Fc(i) < 0)

{

//Задание 3

F.Fc(i, out double y);

Console.WriteLine($"x:{i} y:{y} a + b = {numA + numB}");

}

else if (0 <= F.Fc(i) && F.Fc(i) < 10)

{

Console.WriteLine($"x:{i} y:{F.Fc(i)} a - b = {numA - numB}");

}

else

{

Console.WriteLine($"x:{i} y:{F.Fc(i)} a \* b = {numA \* numB}");

}

}

class F

{

public static double Fc(double x) => x \* x - 5 \* x;

//Задание 3

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

{

y = 0;

y = x \* x - 5 \* x;

}

}

Таблица 3.3 – Входные и выходные данные программы задание №3

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| a:2  b:7  h:1 | x:2 y:-6 a + b = 9  x:3 y:-6 a + b = 9  x:4 y:-4 a + b = 9  x:5 y:0 a - b = -5  x:6 y:6 a - b = -5  x:7 y:14 a \* b = 14 |

Анализ результатов:

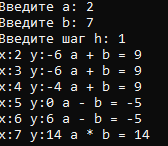


Рисунок 3.4 – Результат выполнения программы задание №3

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 строк. Обеспечить перегрузку операции + для построчного соединения элементов.

Листинг программы:

Файл Program:

using z3.Model;

Console.Write("Введите i: ");

int i = Convert.ToInt32(Console.ReadLine());

Console.Write("Введите j: ");

int j = Convert.ToInt32(Console.ReadLine());

TwodimensionalArray twArray = new TwodimensionalArray(i, j);

twArray.Write();

Console.WriteLine();

TwodimensionalArray twArray2 = new TwodimensionalArray(i, j);

twArray2.Write();

Console.WriteLine();

var twArray3 = twArray + twArray2;

twArray3.Write();

Console.WriteLine();

Console.WriteLine($"Кол-во элементов в twArray3 {twArray3.CountElem}");

Файл TwodinebsionalArray:

namespace z3.Model

{

public class TwodimensionalArray

{

private string[,] \_arr;

public int CountElem

{

get { return \_arr.Length; }

}

public TwodimensionalArray(int sizeI, int sizeJ)

{

\_arr = RandomArr(sizeI, sizeJ);

}

public static TwodimensionalArray operator +(TwodimensionalArray a1,

TwodimensionalArray a2)

{

if (a1.\_arr.GetUpperBound(0) == a2.\_arr.GetUpperBound(0) &&

a1.\_arr.GetUpperBound(1) == a2.\_arr.GetUpperBound(1))

{

TwodimensionalArray newArr =

new TwodimensionalArray(a1.\_arr.GetUpperBound(0) + 1, a1.\_arr.GetUpperBound(1) + 1);

for (int i = 0; i < a1.\_arr.GetUpperBound(0) + 1; i++)

{

for (int j = 0; j < a1.\_arr.GetUpperBound(1) + 1; j++)

{

newArr.\_arr[i, j] = a1.\_arr[i, j] + a2.\_arr[i, j];

}

}

return newArr;

}

return null;

}

public void Write()

{

for (int i = 0; i <= \_arr.GetUpperBound(0); i++)

{

for (int j = 0; j < \_arr.GetUpperBound(1) + 1; j++)

{

Console.Write($" {\_arr[i,j]}\t");

}

Console.WriteLine("");

}

}

private string[,] RandomArr(int sizeI, int sizeJ)

{

string[,] newArr = new string[sizeI, sizeJ];

Random random= new Random();

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

{

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

{

newArr[i,j] = random.Next(10).ToString();

}

}

return newArr;

}

}

}

Таблица 3.4 – Входные и выходные данные программы задание №4

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| i: 3  j:2 | 59 04  76 89  54 02  Кол-во элементов в twArray3 6 |

Анализ результатов:

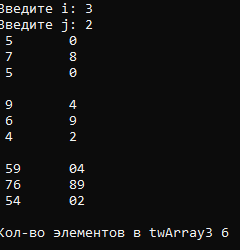


Рисунок 3.5 – Результат выполнения программы задание №4

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, реализующий тип данных «вещественная матрица» и работу с ними. Класс должен реализовывать следующие операции над матрицами:

-сложение, вычитание (как с другой матрицей, так и с числом);

-комбинированные операции присваивания (+=, -=);

-операции сравнения на равенство/неравенство;

-доступ к элементу по индексам.

Написать программу, демонстрирующую все разработанные элементы класса.

Листинг программы:

Файл Program:

using z4.Model;

Console.Write("Введите i: ");

int i = Convert.ToInt32(Console.ReadLine());

Console.Write("Введите j: ");

int j = Convert.ToInt32(Console.ReadLine());

RealMatrix rm = new RealMatrix(i, j);

rm.Write();

Console.WriteLine();

RealMatrix rm2 = new RealMatrix(i, j);

rm2.Write();

Console.WriteLine();

var rm3 = rm + rm2;

rm3.Write();

Console.WriteLine();

rm3 += 3;

rm3.Write();

Console.WriteLine();

rm3 -= 5;

rm3.Write();

Console.WriteLine();

Console.WriteLine($"Кол-во элементов в rm3 {rm3.CountElem}");

Console.WriteLine($"элемент 1 1 в rm3 = {rm3[1,1]}");

Файл RealMatrix:

namespace z4.Model

{

public class RealMatrix

{

private double[,] \_arr;

public int CountElem

{

get { return \_arr.Length; }

}

public double this[int i, int j]

{

get { return \_arr[i,j]; }

set { \_arr[i, j] = value; }

}

public RealMatrix(int sizeI, int sizeJ)

{

\_arr = RandomArr(sizeI, sizeJ);

}

public static RealMatrix operator +(RealMatrix a1,

RealMatrix a2)

{

if (a1.\_arr.GetUpperBound(0) == a2.\_arr.GetUpperBound(0) &&

a1.\_arr.GetUpperBound(1) == a2.\_arr.GetUpperBound(1))

{

RealMatrix newArr =

new RealMatrix(a1.\_arr.GetUpperBound(0) + 1, a1.\_arr.GetUpperBound(1) + 1);

for (int i = 0; i < a1.\_arr.GetUpperBound(0) + 1; i++)

{

for (int j = 0; j < a1.\_arr.GetUpperBound(1) + 1; j++)

{

newArr.\_arr[i, j] = a1.\_arr[i, j] + a2.\_arr[i, j];

}

}

return newArr;

}

throw new Exception("Разные размеры матриц");

}

public static RealMatrix operator +(RealMatrix a1,

double num)

{

RealMatrix newArr =

new RealMatrix(a1.\_arr.GetUpperBound(0) + 1, a1.\_arr.GetUpperBound(1) + 1);

for (int i = 0; i < a1.\_arr.GetUpperBound(0) + 1; i++)

{

for (int j = 0; j < a1.\_arr.GetUpperBound(1) + 1; j++)

{

newArr.\_arr[i, j] = a1.\_arr[i, j] + num;

}

}

return newArr;

}

public static RealMatrix operator -(RealMatrix a1,

RealMatrix a2)

{

if (a1.\_arr.GetUpperBound(0) == a2.\_arr.GetUpperBound(0) &&

a1.\_arr.GetUpperBound(1) == a2.\_arr.GetUpperBound(1))

{

RealMatrix newArr =

new RealMatrix(a1.\_arr.GetUpperBound(0) + 1, a1.\_arr.GetUpperBound(1) + 1);

for (int i = 0; i < a1.\_arr.GetUpperBound(0) + 1; i++)

{

for (int j = 0; j < a1.\_arr.GetUpperBound(1) + 1; j++)

{

newArr.\_arr[i, j] = a1.\_arr[i, j] - a2.\_arr[i, j];

}

}

return newArr;

}

throw new Exception("Разные размеры матриц");

}

public static RealMatrix operator -(RealMatrix a1,

double num)

{

RealMatrix newArr =

new RealMatrix(a1.\_arr.GetUpperBound(0) + 1, a1.\_arr.GetUpperBound(1) + 1);

for (int i = 0; i < a1.\_arr.GetUpperBound(0) + 1; i++)

{

for (int j = 0; j < a1.\_arr.GetUpperBound(1) + 1; j++)

{

newArr.\_arr[i, j] = a1.\_arr[i, j] - num;

}

}

return newArr;

}

public static bool operator >(RealMatrix a1,

RealMatrix a2)

{

if (a1.\_arr.GetUpperBound(0) == a2.\_arr.GetUpperBound(0) &&

a1.\_arr.GetUpperBound(1) == a2.\_arr.GetUpperBound(1))

{

for (int i = 0; i < a1.\_arr.GetUpperBound(0) + 1; i++)

{

for (int j = 0; j < a1.\_arr.GetUpperBound(1) + 1; j++)

{

if (a1.\_arr[i, j] < a2.\_arr[i, j])

{

return false;

}

}

}

return true;

}

throw new Exception("Разные размеры матриц");

}

public static bool operator <(RealMatrix a1,

RealMatrix a2)

{

if (a1.\_arr.GetUpperBound(0) == a2.\_arr.GetUpperBound(0) &&

a1.\_arr.GetUpperBound(1) == a2.\_arr.GetUpperBound(1))

{

for (int i = 0; i < a1.\_arr.GetUpperBound(0) + 1; i++)

{

for (int j = 0; j < a1.\_arr.GetUpperBound(1) + 1; j++)

{

if (a1.\_arr[i, j] > a2.\_arr[i, j])

{

return false;

}

}

}

return true;

}

throw new Exception("Разные размеры матриц");

}

public void Write()

{

for (int i = 0; i <= \_arr.GetUpperBound(0); i++)

{

for (int j = 0; j < \_arr.GetUpperBound(1) + 1; j++)

{

Console.Write($" {\_arr[i, j]}\t");

}

Console.WriteLine("");

}

}

private double[,] RandomArr(int sizeI, int sizeJ)

{

double[,] newArr = new double[sizeI, sizeJ];

Random random = new Random();

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

{

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

{

newArr[i, j] = random.Next(10);

}

}

return newArr;

}

}

}

Таблица 3.5 – Входные и выходные данные программы задание №5

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| i: 3  j: 4 | Кол-во элементов в rm3 12  элемент 1 1 в rm3 = 9 |

Анализ результатов:

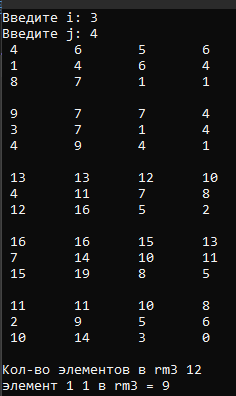


Рисунок 3.6 – Результат выполнения программы задание №5