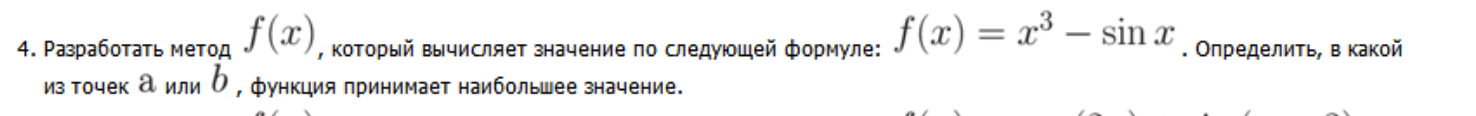
**3 Процедуры и функции - методы класса**

Задание №1.



Листинг программы:

static void Func(double x,out double z)

{

z = Math.Pow(x, 3) - Math.Sin(x);

Console.WriteLine(z);

}

Console.Write("Введите число а: ");

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

Console.Write("Введите число b: ");

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

double rez1, rez2;

Func(a, out rez1);

Func(b, out rez2);

if (rez1 > rez2) Console.WriteLine($"В точке {a} фунция принимает наибольшее значение. ");

else Console.WriteLine($"В точке {b} фунция принимает наибольшее значение. ");

Console.ReadLine();

Таблица А.1 – Входные и выходные данных задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3  15 | 26.85887999194013  3374.349712159843 |

Анализ результатов:

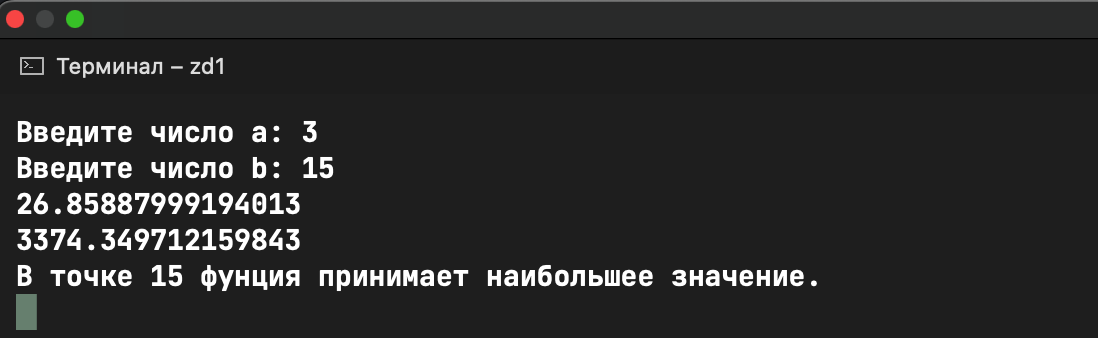
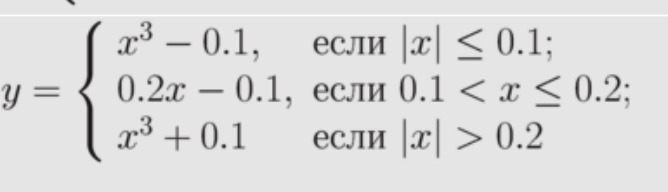


Рисунок А.1 – Результат работы программы из задания №1

Задание №2. Реализовать таблицу вычислений с шагом. 

Листинг программы:

Console.Write("Задайте значение числу а: ");

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

Console.Write("Задайте значение числу b: ");

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

Console.Write("Задайте значение шагу h: ");

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

double y = 0;

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

{

Console.WriteLine($"{i:.##} {Func(i):.##}");

}

Console.ReadLine();

static double Func(double x)

{

double y;

if (Math.Abs(x) <= 0.1) y = Math.Pow(x, 3) - 0.1;

else if (x > 0.1 && x <= 0.2) y = 0.2 \* x - 0.1;

else y = Math.Pow(x, 3) + 0.1;

return y;

}

Таблица А.2 – Входные и выходные данных задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4  5  3 | 64.1 |

Анализ результатов:

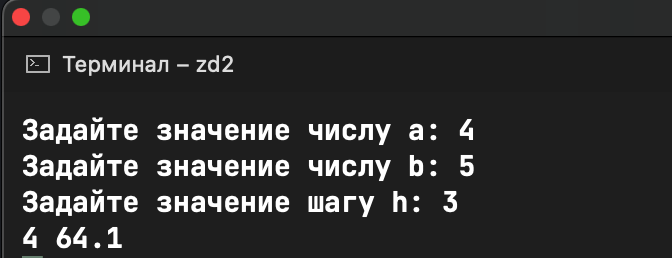


Рисунок А.2 – Результат работы программы из задания №2

Задание №3. Сделать перегрузку метода из задания 2.

Листинг программы:

Console.Write("Задайте значение числу а: ");

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

Console.Write("Задайте значение числу b: ");

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

Console.Write("Задайте значение шагу h: ");

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

double y = 0;

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

{

Console.WriteLine($"{i:.##} {F.Func(i):.##}");

}

Console.ReadLine();

class F

{

public static double Func(double x)

{

double y;

if (Math.Abs(x) <= 0.1) y = Math.Pow(x, 3) - 0.1;

else if (x > 0.1 && x <= 0.2) y = 0.2 \* x - 0.1;

else y = Math.Pow(x, 3) + 0.1;

return y;

}

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

{

y = 0;

if (Math.Abs(x) <= 0.1) y = Math.Pow(x, 3) - 0.1;

else if (x > 0.1 && x <= 0.2) y = 0.2 \* x - 0.1;

else y = Math.Pow(x, 3) + 0.1;

}

}

Таблица А.3 – Входные и выходные данных задание №3

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4  5  3 | 64.1 |

Анализ результатов:

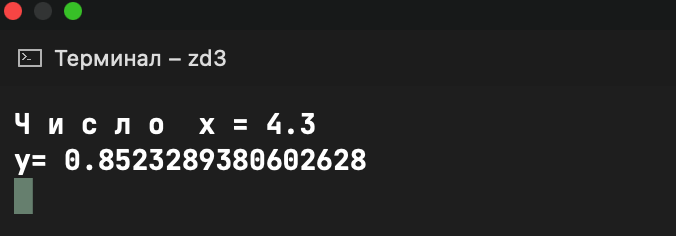


Рисунок А.3 – Результат работы программы из задания №3

Задание №4. Описать класс, реализующий тип данных «вещественная матрица». Реализовать вычитание строки заданного номера из всех остальных строк, кроме данной строки. 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.

Листинг программы:

using System;

Console.WriteLine("Введите размер матрицы");

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

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

Real\_Matrix matrix = new Real\_Matrix(n,m);

double[,] a = new double[n, m];

matrix.\_m = m;

matrix.\_n = n;

int i = 0;

int j = 0;

matrix.Write();

Console.WriteLine("Введите строку");

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

int bs = b - 1;

matrix.F(bs);

matrix.Write();

Console.ReadLine();

class Real\_Matrix

{

public int \_n;

public int \_m;

private double[,] \_matrix;

public Real\_Matrix(int n, int m)

{

\_n = n;

\_m = m;

\_matrix = GetRandomMatrix(\_n, \_m);

}

private double[,] GetRandomMatrix(int n, int m)

{

double[,] matrix = new double[n, m];

Random random = new Random();

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

{

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

{

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

}

}

return matrix;

}

public void Write()

{

for (int i = 0; i < \_n; i++)

{

for (int j = 0; j < \_m; j++)

{

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

}

Console.WriteLine("");

}

}

public void F(int numS)

{

for (int i = 0; i < \_n; i++)

{

if (i != numS)

{

for(int j = 0; j < \_m; j++)

{

\_matrix[i, j] -= \_matrix[numS, j];

}

}

}

}

}

Таблица А.4 – Входные и выходные данных задание №4

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3  3 |  |

Анализ результатов:

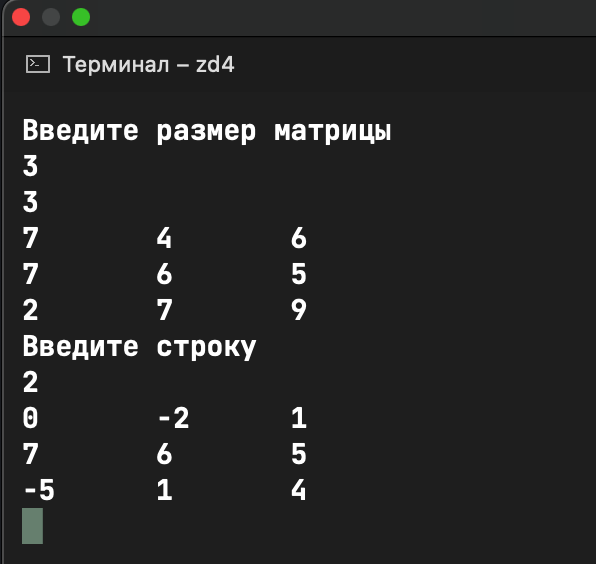


Рисунок А.4 - Результат работы программы из задания №4

Задание №5. Описать класс «студенческая группа». Предусмотреть возможность работы с переменным числом студентов, поиска студента по какому-либо признаку (например, по фамилии, имени, дате рождения), добавления и удаления записей, сортировки по разным полям, доступа к записи по номеру. Написать программу, демонстрирующую все разработанные элементы класса.

Листинг программы:

StudentGroup group = new StudentGroup("32 программисты");

student st = new student("Иванов", "Иван", "Иванович");

for (int i = 0; i < 10; i++)

group.AddStudent(st);

student s = new student("Сидоров", "Сидр", "Сидорович");

group.AddStudent(s);

group.ShowGroup();

student s\_find = group.FindStudent("Сидоров");

Console.WriteLine("Найденный студент: " + s.\_name);

Console.WriteLine("Название группы:" + group.GroupName);

Console.ReadLine();

struct student

{

public string \_fam;

public string \_name;

public string \_otch;

public student(string fam, string name, string otch)

{

\_fam = fam;

\_name = name;

\_otch = otch;

}

}

class StudentGroup

{

string group\_name;

List<student> students;

public StudentGroup(string groupName)

{

group\_name = groupName;

students = new List<student>();

}

public void AddStudent(student newStudent)

{

students.Add(newStudent);

}

public string GroupName

{

get { return group\_name; }

set { group\_name = value; }

}

public void RemoveStudent(int index)

{

if (index >= students.Count)

throw new Exception("Студента с таким индексом не существует.");

students.RemoveAt(index);

}

public void ShowGroup()

{

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

Console.WriteLine("ФИО: " + students[i].\_fam + " " + students[i].\_name + " " + students[i].\_otch);

}

public student FindStudent(string key)

{

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

if (students[i].\_fam == key || students[i].\_name == key || students[i].\_otch == key)

return students[i];

return new student("", "", "");

}

}

Таблица А.5 – Входные и выходные данных задание №5

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

Анализ результатов:

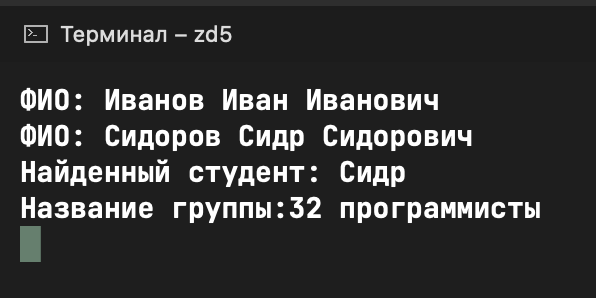


Рисунок А.5 - Результат работы программы из задания №5