**Процедуры и функции – методы класса**

Задание№1. 

Листинг программы

using System;

class Program

{

static double F(double x)

{

return Math.Pow(x, 3.0) - Math.Sin(x);

}

static void Main(string[] args)

{

Console.WriteLine("Введите а: ");

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

Console.WriteLine("Введите b: ");

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

Console.WriteLine(F(a) > F(b) ? "a" : "b");

Console.ReadKey(true);

}

}

Анализ программы

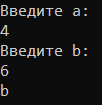
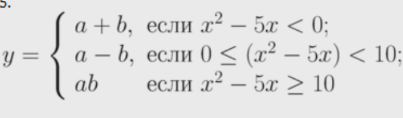


Рисунок 1.1 - Анализ программы

Источник – собственная разработка

Задание №2.



Листинг программы:

using System;

namespace ZADACHA1

{

class Class1

{

static double F(double x, double z, double r)

{

double y;

if (Math.Pow(x, 2) - 5 \* x < 0)

{

y = z + r;

}

else if ((0 <= Math.Pow(x, 2) - 5 \* x) && (Math.Pow(x, 2) - 5 \* x < 10))

{

y = z - r;

}

else

{

y = z \* r;

}

return y;

}

static void Main()

{

Console.WriteLine("a=");

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

Console.WriteLine("b=");

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

Console.WriteLine("x=");

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

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

Console.WriteLine("f({0:f2})={1:f4}", i, F(x: i, r: b, z: a));

}

}

}

Анализ программы:

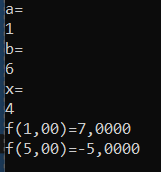
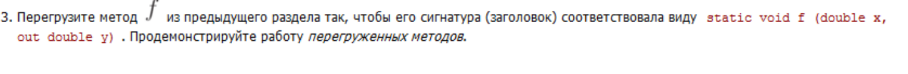


Рисунок 1.2 - Анализ программы

Источник – собственная разработка

Задание №3.



Листинг программы:

using System;

namespace ZADACHA1

{

class Class1

{

static void F(double x, double z, double r, out double y)

{

if (Math.Pow(x, 2) - 5 \* x < 0)

{

y = z + r;

}

else if ((0 <= Math.Pow(x, 2) - 5 \* x) && (Math.Pow(x, 2) - 5 \* x < 10))

{

y = z - r;

}

else

{

y = z \* r;

}

}

static void Main()

{

Console.WriteLine("a=");

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

Console.WriteLine("b=");

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

Console.WriteLine("x=");

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

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

{

F(x: i, r: b, z: a, y: out double y);

Console.WriteLine("f({0:f2})={1:f4}", i, y);

}

}

}

}

Анализ программы:

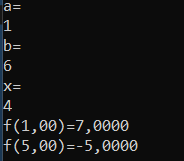


Рисунок 1.3 - Анализ программы

Источник – собственная разработка

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

Листинг программы:

public class Matrix

{

private double[,] \_numbers;

private int \_rowCount;

private int \_columnCount;

public int RowCount => \_rowCount;

public int ColumnCount { get { return \_columnCount; } }

public double this[int i, int j]

{

get

{

if (i >= 0 && i < \_rowCount & j >= 0 & j < \_columnCount)

{

return \_numbers[i, j];

}

throw new IndexOutOfRangeException();

}

set

{

if (i >= 0 && i < \_rowCount && j >= 0 && j < \_columnCount)

{

\_numbers[i, j] = value;

}

else

{

throw new IndexOutOfRangeException();

}

}

}

private Matrix() { }

private Matrix(int rowCount, int columnCount)

{

\_rowCount = rowCount;

\_columnCount = columnCount;

\_numbers = new double[rowCount, columnCount];

}

public static Matrix Generate(int rowCount, int columnCount)

{

Matrix matrix = new Matrix(rowCount, columnCount);

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

{

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

{

matrix[i, j] =Random.Shared.NextDouble() \* Random.Shared.Next(3, 20);

}

}

return matrix;

}

internal void Substruction(int rowNumber)

{

if (rowNumber<0 || rowNumber>=\_rowCount)

{

throw new IndexOutOfRangeException($"RowNumber is out of {typeof(Matrix)} range");

}

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

{

if (i == rowNumber)

{

continue;

}

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

{

\_numbers[i, j] = \_numbers[i, j] - \_numbers[rowNumber, j];

}

}

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите размер матрицы (после каждого введённого значения прожимаем Enter)");

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

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

Matrix matrix = Matrix.Generate(n, m);

Console.WriteLine("Generated matrix");

PrintMatrix(matrix);

var isCorectRowNumber=false;

while (!isCorectRowNumber)

{

try

{

Console.WriteLine("Введите индекс строки (номер строки - 1)");

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

matrix.Substruction(rowNumber);

isCorectRowNumber=true;

}

catch (IndexOutOfRangeException ex)

{

Console.WriteLine(ex.Message);

}

}

Console.WriteLine("Modified matrix");

PrintMatrix(matrix);

Console.WriteLine("New matrix");

Matrix matrix1 = Matrix.Generate(n,m);

PrintMatrix(matrix1);

Console.ReadLine();

}

private static void PrintMatrix(Matrix matrix)

{

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

{

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

{

Console.Write($"{matrix[i,j]:F2} ");

}

Console.WriteLine();

}

}

}

Анализ программы:

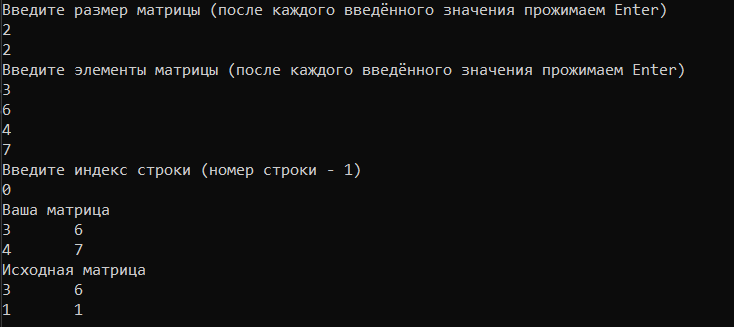


Рисунок 1.4 - Анализ программы

Источник – собственная разработка

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:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication45

{

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("", "", "");

}

}

class Program

{

static void Main(string[] args)

{

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.ReadKey();

}

}

}

Анализ программы:

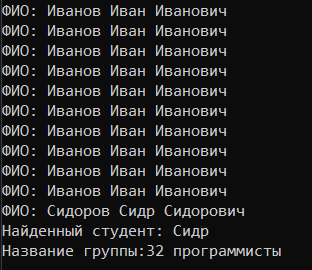


Рисунок 1.5 - Анализ программы

Источник – собственная разработка