Министерство науки и ВЫСШЕГО образования   
Российской Федерации

Федеральное автономное образовательное учреждение высшего образования  
«Санкт-Петербургский государственный политехнический   
университет Петра Великого»  
**Институт среднего профессионального образования**

**Лабораторная работа № 3  
Тема:** Определение и вызов функций. Передача параметров и возврат результатов.

МДК.01.01 Разработка программных модулей

**Вариант 9**

Выполнил: студент группы 32919/8  
Крупина Елена

Проверил: преподаватель   
Прокофьев А.А.

Санкт-Петербург  
2024

**Цель работы:** изучить правила определения и вызова функций в языке C#, получить практические навыки разработки спецификаций функций по обработке простых типов данных и функций по обработке структурированных типов данных, и создания программ, использующих функции программиста.

*Указания:*

Требуется разработать 2 проекта, каждая программа  − консольное приложение на языке C#, в среде программирования Visual Studio, использующая дополнительные функции.

**Задание 1**: Разработать подпрограмму для определения, пересекаются ли две окружности на плоскости. Окружности задаются координатами центра и радиусом.

**Блок-схема:**



**Блок-схемы (подпрограммы):**



**Код программы:**

using System;

class Program

{

static double Input(string message)

{

Console.Write(message);

string input = Console.ReadLine();

// Проверка на корректность ввода

if (!double.TryParse(input, out double result))

{

Console.WriteLine("Некорректный ввод. Пожалуйста, введите число.");

Console.WriteLine();

return Input(message); // вызов метода для повторного ввода

}

return result;

}

static double InputRadius(string message)

{

Console.Write(message);

// Проверка на корректность ввода

if (!double.TryParse(Console.ReadLine(), out double result) || result <= 0)

{

Console.WriteLine("Некорректный ввод. Пожалуйста, введите число.");

Console.WriteLine();

return InputRadius(message); // вызов метода для повторного ввода

}

return result;

}

static void Intersection(double r1, double r2, double x1, double x2, double y1, double y2)

{

double distance = Math.Sqrt(Math.Pow(x1 - x2, 2) + Math.Pow(y1 - y2, 2));

if (distance < r1 + r2 && distance > Math.Abs(r1 - r2))

{

Console.WriteLine("Окружности пересекаются.");

}

else { Console.WriteLine("Окружности не пересекаются."); }

}

public static void Main()

{

double R1, R2;

double x1, x2;

double y1, y2;

Console.WriteLine("Заполните данные первой окружности: ");

R1 = InputRadius("Введите радиус: ");

x1 = Input("Введите координату x центра:");

y1 = Input("Введите координату y центра:");

Console.WriteLine();

Console.WriteLine("Заполните данные второй окружности: ");

R2 = InputRadius("Введите радиус: ");

x2 = Input("Введите координату x центра:");

y2 = Input("Введите координату y центра:");

Console.WriteLine();

Intersection(R1, R2, x1, x2, y1, y2);

}

}

**Результат работы:**

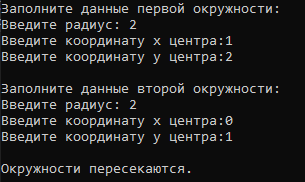
****

Рис.1 – Результат

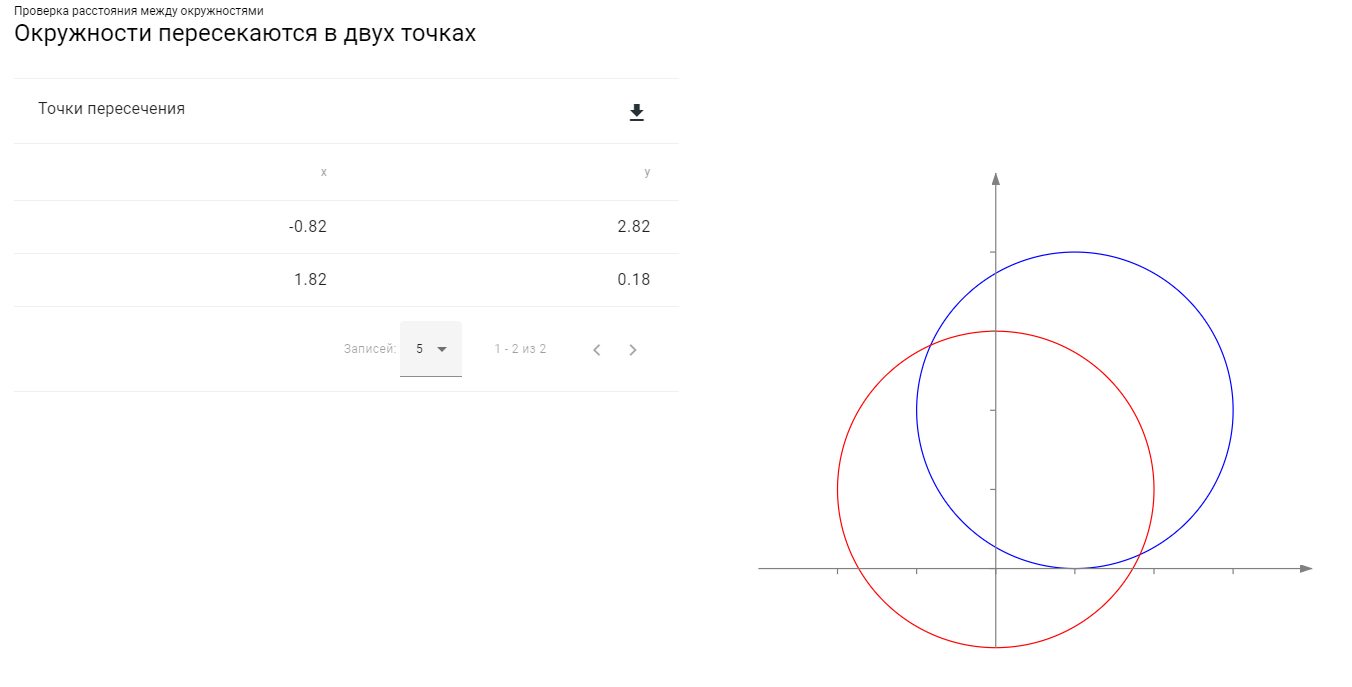


Рис.2 – Проверка результата

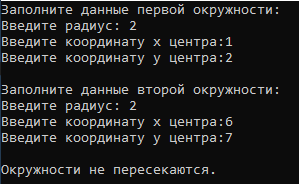


Рис. 3 – Результат

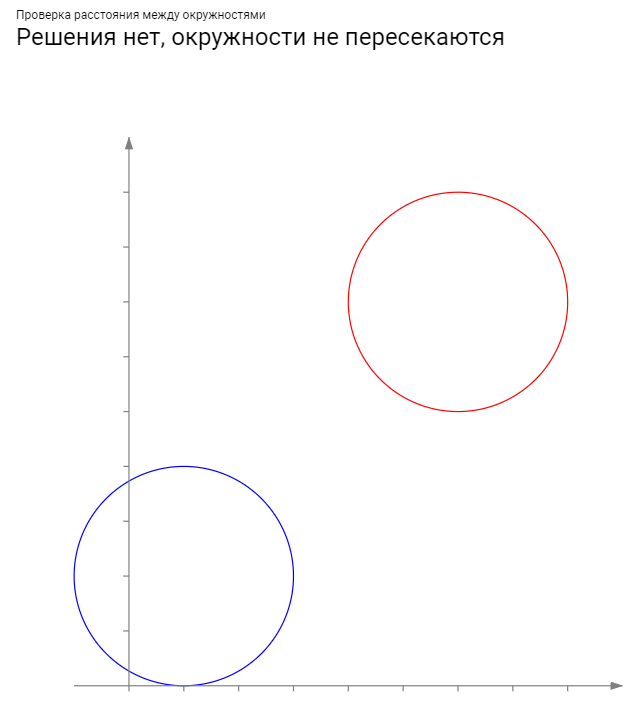


Рис.4 – Проверка результат

**Задание 2:**

* Заданы два одномерных массива разной размерности. Организовать ввод и вывод на экран массивов с использованием функций программиста;
* Найти среднее геометрическое элементов каждого массива, используя функцию программиста;
* В том массиве, где среднее геометрическое больше всего, поменять местами первый чётный и последний чётный элемент массива. Обмен осуществить при помощи функции.

**Блок-схемы:**



**Код программы:**

using System;

class Program

{

static int Input(string message)

{

Console.Write(message);

string input = Console.ReadLine();

// Проверка на корректность ввода

if (!int.TryParse(input, out int result) || result <= 0)

{

Console.WriteLine("Некорректный ввод. Пожалуйста, введите число.");

Console.WriteLine();

return Input(message); // вызов метода для повторного ввода

}

return result;

}

static void Output(int[] nums)

{

for (int i = 0; i < nums.Length; i++)

{

Console.Write("[" + i + "]" + "\t");

}

Console.WriteLine();

for (int i = 0; i < nums.Length; i++)

{

Console.Write(nums[i] + "\t");

}

Console.WriteLine();

Console.WriteLine();

Console.WriteLine();

}

static int GeometricMean(int[] nums)

{

int GeometricMean = 1;

for (int i = 0; i < nums.Length; i++)

{

GeometricMean \*= nums[i];

}

return Math.Abs(GeometricMean \* nums.Length);

}

static void Parity(ref int[] nums)

{

int first = -1, last = -1; // инициализируем -1, если четных элементов нет

if (nums.Length > 1)

{

for (int i = 0; i < nums.Length; i++)

{

if (nums[i] % 2 == 0)

{

first = i;

break;

}

}

for (int i = nums.Length - 1; i >= 0; i--)

{

if (nums[i] % 2 == 0)

{

last = i;

break;

}

}

int n = nums[first];

nums[first] = nums[last];

nums[last] = n;

}

//Console.WriteLine(first + "\t" + last);

}

public static void Main()

{

int len1 = Input("Введите размер первого массива: ");

int len2 = Input("Введите размер второго массива: ");

Console.WriteLine();

int[] nums1 = new int[len1];

int[] nums2 = new int[len2];

Random rnd = new Random();

for (int i = 0; i < len1; i++)

{

nums1[i] = rnd.Next(-10, 10);

}

for (int i = 0; i < len2; i++)

{

nums2[i] = rnd.Next(-10, 10);

}

Console.WriteLine("Первый массив:");

Output(nums1);

Console.WriteLine("Второй массив:");

Output(nums2);

Console.WriteLine();

Console.WriteLine($"Среднее геометрическое первого массива: {GeometricMean(nums1)}");

Console.WriteLine($"Среднее геометрическое второго массива: {GeometricMean(nums2)}");

Console.WriteLine();

if (GeometricMean(nums1) >= GeometricMean(nums2))

{

Console.WriteLine("Измененный первый массив: ");

Parity(ref nums1);

Output(nums1);

}

else

{

Console.WriteLine("Измененный второй массив: ");

Parity(ref nums2);

Output(nums2);

}

}

}

**Результат работы:**

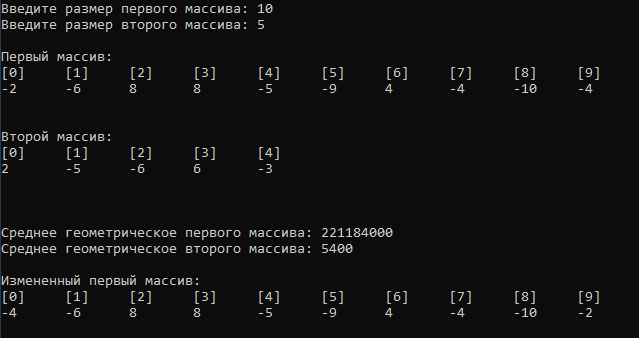
****

Рис. 5 – Результат

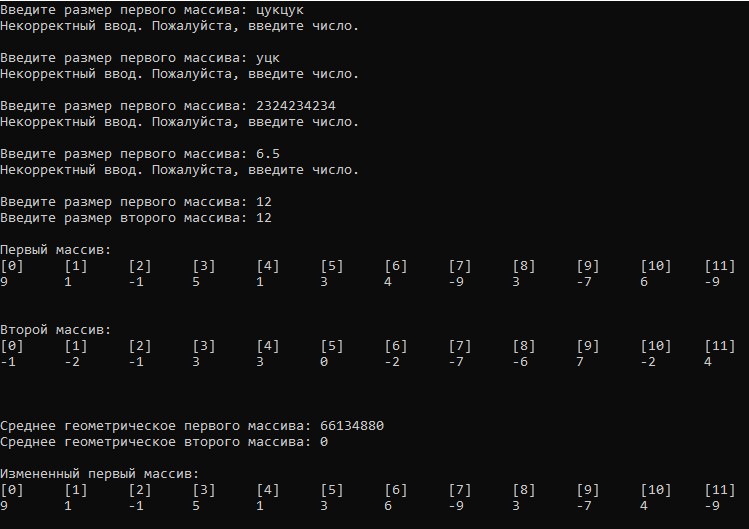


Рис. 6 - Результат