СОДЕРЖАНИЕ

[Лабораторная работа №1 3](#_Toc485292038)

[Лабораторная работа №2 6](#_Toc485292039)

[Лабораторная работа №3 10](#_Toc485292040)

[Лабораторная работа №4 18](#_Toc485292041)

[Лабораторная работа №5 24](#_Toc485292042)

# Лабораторная работа №1

**Тема: «Линейные программы»**

**Цель работы: «Изучить основы линейных программ на языке C#»**

Вариант 5

**Теоретическая часть:**

Pow - математическая функция в С# которая возвращает значение, то есть возводит число x в степень a.

Класс Math - предоставляет константы и статические методы для тригонометрических, логарифмических и иных общих математических функций.

Ключевое слово double обозначает простой тип, используемый для хранения 64-разрядных значений с плавающей запятой. В следующей таблице представлен приблизительный диапазон и точность для типа double.

**Практическая часть:**

z1= 1-1/4\*(sin(2\*a))^2+cos(2\*a)

z2= (cos(a))^2+(cos(a))^4

Блок-схема:



Код программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

double z1, z2, a;

Console.Write("Введите а=");

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

z1 = 1 - (1.0/4) \* Math.Pow(Math.Sin(2\*a), 2) + Math.Cos(2 \* a);

z2 = Math.Pow(Math.Cos(a),2);

z2 = z2 + z2 \* z2;

Console.WriteLine("z1=" +z1);

Console.WriteLine("z2=" +z2);

Console.ReadKey(true);

}

}

}

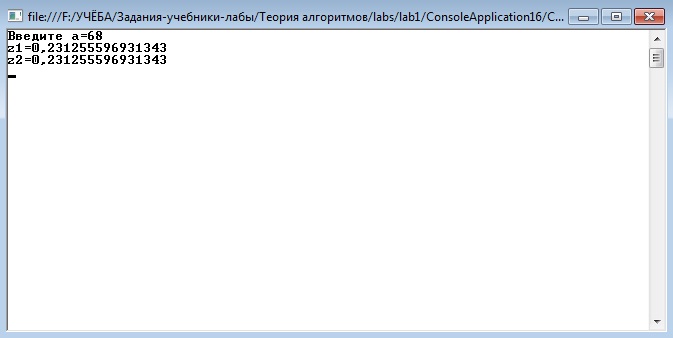


Рисунок 1 – Выполнение программы

**Вывод:** в процессе выполнения лабораторной работы была создана программа, вычисляющая два уравнения.

# Лабораторная работа №2

**Тема: «Разветвляющиеся вычислительные процессы»**

**Цель работы:** «Изучение операторов ветвлений на языке C#»

Вариант 5

**Теоретическая часть:**

using System - область имен с системными базовыми классами.

using System.Collections.Generic - дженерики и коллекции.

using System.Text - для работы с текстом.

System.Linq - содержит классы и интерфейсы, которые поддерживают LINQ.

System.Threading.Tasks - пространство имен предоставляет типы, которые упрощают задачу написания параллельного и асинхронного кода.

name space - пространство имен.

static void Main() статик говорит, что майн - локальная функция для .cpp файла и вне файла к ней получить доступ просто так не получится. В скобках пишутся аргументы функции.

Double - Обозначает простой тип, используемый для хранения 64-разрядных значений с плавающей запятой.

Console.Write - Записывает текстовое представление заданного значения или значений в стандартный выходной поток.

Console.WriteLine - Записывает текущий признак конца строки в стандартный выходной поток.

Console.ReadLine - Считывает следующую строку символов из стандартного входного потока.

Класс Convert - Преобразует значение одного базового типа данных к другому базовому типу данных.

Convert.ToDouble - Преобразует заданное строковое представление числа в эквивалентное число с плавающей запятой двойной точности.

Конструкция if/else проверяет истинность некоторого условия и в зависимости от результатов проверки выполняет определенный код. После ключевого слова if ставится условие. И если это условие выполняется, то срабатывает код, который помещен далее в блоке if после фигурных скобок.

Класс Math - Предоставляет константы и статические методы для тригонометрических, логарифмических и иных общих математических функций.

Math.PI - Представляет отношение длины окружности к ее диаметру, определяемое константой π.

Math.Sqrt - Возвращает квадратный корень из указанного числа.

Math.Sin - Возвращает синус указанного угла.

Math.Pow - Возвращает указанное число, возведенное в указанную степень.

Console.ReadKey - Получает следующий нажатый пользователем символ или функциональную клавишу.

Значение true, чтобы не отображать нажатую клавишу  в окне консоли. В противном случае — значение false.

**Практическая часть:**

Задание 1. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика. Параметр R вводится с клавиатуры.

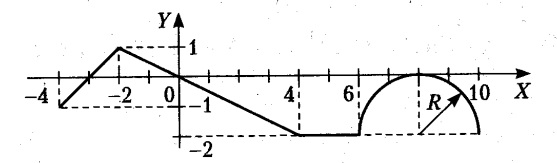


Рисунок 1 – График функции

Задание 2. Написать программу, которая определяет, попадает ли точка с заданными координатами в область, закрашенную серым цветом.

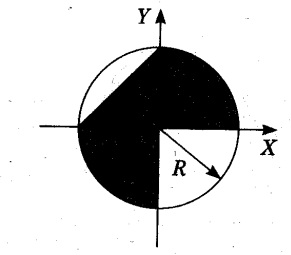


Рисунок 2 – Закрашенная область

Блок-схема:



Код программы:

double R=0, x, y = 0;

// Задание 1

Console.WriteLine("Первое задание");

Console.Write("x=");

x = Convert.ToDouble(Console.ReadLine());

Console.Write("R=");

R = Convert.ToDouble(Console.ReadLine());

if (-4 <= x && x < -2)

y = 1 \* x + 3;

else if (-2 <= x && x < 4)

y = 0.5 \* x;

else if (4 <= x && x < 6)

y = -2;

else if (6 <= x && x < 10)

y = Math.Sqrt(Math.Pow(R, 2) - Math.Pow(x - 8, 2)) - 2;

else

Console.WriteLine("Неопределена");

Console.WriteLine("y=" + y);

Console.ReadKey(true);

// Задание 2

Console.WriteLine("\nЗадание 2");

Console.Write("x=");

x = Convert.ToDouble(Console.ReadLine());

Console.Write("y=");

y = Convert.ToDouble(Console.ReadLine());

if (-R<=x&&x<=0&&y>=-Math.Sqrt(Math.Pow(R, 2) - Math.Pow(x, 2))&&y<=x+R)

Console.WriteLine("Пара (x,y) принадлежит заданной области");

else if (R >= x && x >= 0 && y <= Math.Sqrt(Math.Pow(R, 2) - Math.Pow(x, 2)) && y >=0)

Console.WriteLine("Пара (x,y) принадлежит заданной области");

else

Console.WriteLine("Пара (x,y) не принадлежит заданной области");

Console.ReadKey(true);

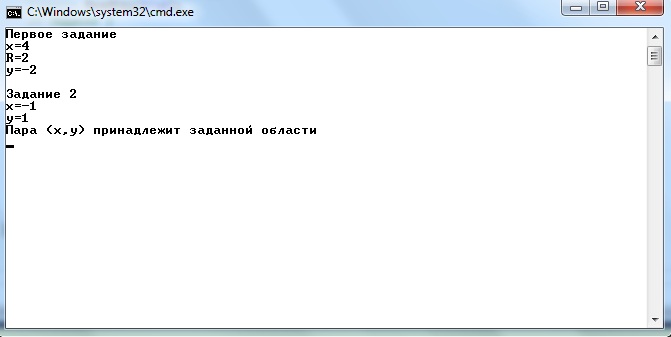


Рисунок 3 – Выполнение программы

**Вывод:** были изучены операторы ветвлений на языке C# и написаны программы вычисляющие значение функции и определяющие нахождение точки в графике.

# Лабораторная работа №3

**Тема: «Организация классов»**

**Цель работы:** «Изучение организация классов на языке C#»

Вариант 5

**Теоретическая часть:**

Класс Math- Предоставляет константы и статические методы для тригонометрических, логарифмических и иных общих математических функций.

Math.PI- Представляет отношение длины окружности к ее диаметру, определяемое константой π.

Math.Sqrt- Возвращает квадратный корень из указанного числа.

Math.Sin- Возвращает синус указанного угла.

Math.Pow- Возвращает указанное число, возведенное в указанную степень.

Console.ReadKey- Получает следующий нажатый пользователем символ или функциональную клавишу.

Значение true, чтобы не отображать нажатую клавишу  в окне консоли. В противном случае — значение false.

Begin- начать.

End- конец.

While- оператор , выполняет оператор или блок операторов, пока определенное выражение не примет значение false.

**Практическая часть:**

1. Таблица значений функции. Вычислить и вывести на экран в виде таблицы значения функции, заданной графически, на интервале от х нач. до х кон. с шагом dx. Интервал и шаг задать таким образом, чтобы проверить все ветви программы. Таблицу снабдить заголовком и шапкой.

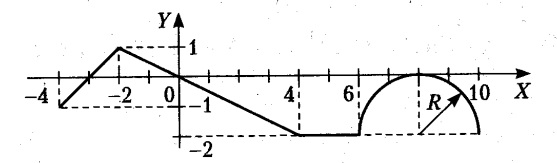


Рисунок 1 - График функции

1. Серия выстрелов по мишени. Для десяти выстрелов, координаты которых задаются с клавиатуры, вывести текстовые сообщения о попадании в мишень из задания 2 лабораторной работы 2.

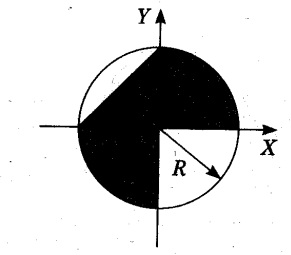


Рисунок 2 - Закрашенная область

1. Ряды Тейлора. Вычислить и вывести на экран в виде таблицы значения функции, заданной с помощью ряда Тейлора, на интервале от х нач. до х кон. с шагом dx с точкой "е". Таблицу снабдить заголовком и шапкой. Каждая строка таблицы должна содержать значения аргумента, значение функции и количество просуммированных членов ряда.

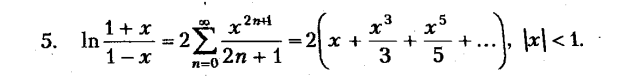


Рисунок 3 - Функция

Блок-схема:







Код программы:

double x=-4, y=-1, dx=1;

Console.WriteLine("Вариант 5");

//Задание 1

Console.WriteLine("Первое задание");

Console.WriteLine("Значения кусочной функции:");

Console.WriteLine(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.WriteLine(" | X | Y |");

Console.WriteLine(" |\_\_\_\_\_\_\_|\_\_\_\_\_\_\_|");

for(dx = 1; x < 11; x = x + dx)

{

if (-4 <= x && x < -2)

y = x + 3;

else if (-2 <= x && x < 4)

y = -0.5 \* x;

else if (4 <= x && x < 6)

y = -2;

else if (6 <= x && x < 10)

y = Math.Sqrt(Math.Pow(2, 2) - Math.Pow(x - 8, 2)) - 2;

Console.WriteLine(" |" + String.Format("{0,4:0.#}", x) + " |" + String.Format("{0,5:0.##}", y) + " |");

}

Console.WriteLine(" |\_\_\_\_\_\_\_|\_\_\_\_\_\_\_|");

//Задание 2

Console.WriteLine("Второе задание");

Console.WriteLine();

double R = 8, a, b=0;

Console.WriteLine("Серия выстрелов по мишени:");

Console.WriteLine("R=8");

for (int i = 0; i < 8; i++)

{

Console.Write("a=");

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

Console.Write("b=");

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

if (-R <= a && a <= 0 && b >= -Math.Sqrt(Math.Pow(R, 2) - Math.Pow(a, 2)) && b <= a + R)

Console.WriteLine("Принадлежит заданной области");

else if (R >= a && a >= 0 && b <= Math.Sqrt(Math.Pow(R, 2) - Math.Pow(a, 2)) && b >= 0)

Console.WriteLine("Принадлежит заданной области");

else

Console.WriteLine("Не принадлежит заданнй области");

}

//Задание 3

Console.WriteLine("Третье задание");

Console.WriteLine("\n\tРяд Тейлора для функции loG");

Console.WriteLine(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ");

Console.WriteLine(" | x | loG | n | log |");

Console.WriteLine(" |\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|");

Double x\_end=0.9, e = 0.001, log, loG;

dx = 0.1;x = -0.9;

while (x<=x\_end)

{

Double last=0, sum=0;

int n = 0;

do

{

last = Math.Pow(x, 2 \* n + 1) / (2 \* n + 1);

sum += last;

n++;

} while (Math.Abs(last) > e);

loG = 2 \* sum;

log = Math.Log((1+x) / (1-x));

Console.WriteLine(" |" + String.Format("{0,5:0.#}", x) + "|" + String.Format("{0,8:0.####}", loG) + "|" + String.Format("{0,5:0}", n) + "|" + String.Format("{0,9:0.####}", log) + "|");

Console.WriteLine(" |\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|");

x = x + dx;

}

Console.ReadKey(true);

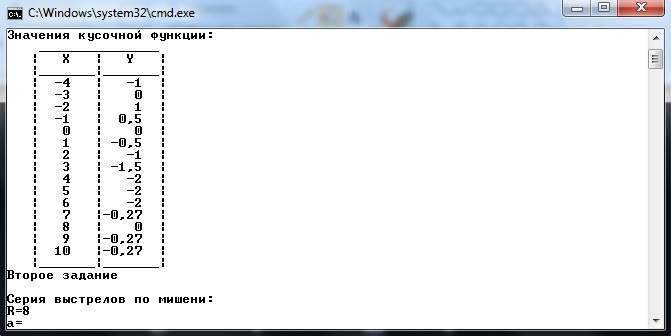


Рисунок 1 – Первое задание

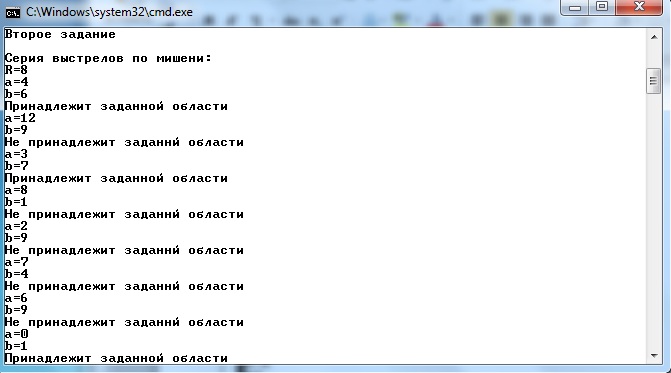


Рисунок 2 – Второе задание

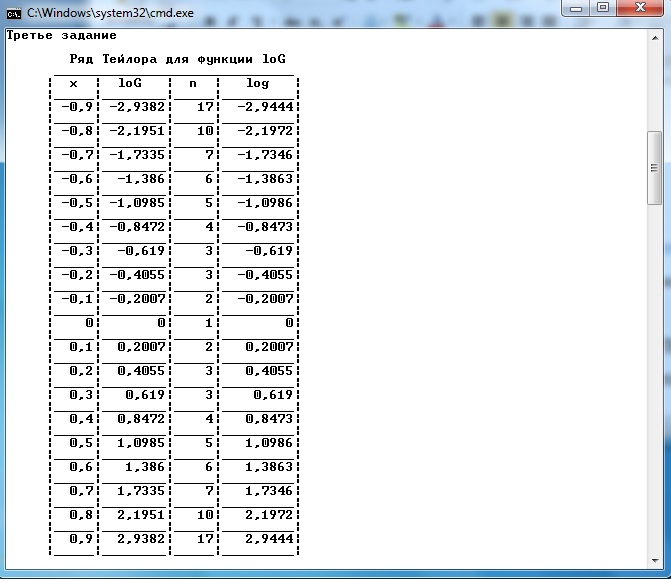


Рисунок 3 – Третье задание

**Вывод:** получены навыки написания программ с помощью организация классов на языке C# в среде программирования Microsoft Visual Studio.

# Лабораторная работа №4

**Тема: «Одномерные массивы»**

**Цель работы:** Научиться работать с одномерными массивами на языке C#.

**Задание:** Вариант 5. В одномерном массиве, состоящем из n вещественных элементов, вычислить:

**-** максимальный элемент массива;

**-** сумму элементов массива, расположенных до последнего положительного элемента.

Сжать массив, удалив из него все элементы, модуль которых находится в интервале [a, b]. Освободившиеся в конце массива элементы заполнить нулями.

**Краткая теория**

В C# массивы фактически являются объектами, а не только адресуемыми областями непрерывной памяти, как в C и C++. Array является абстрактным базовым типом всех типов массивов. Можно использовать свойства и другие члены класса, которые имеет Array. В примере используется свойство Length для получения длины массива. В следующем коде длина массива numbers, равная 5, присваивается переменной lengthOfNumbers:

int[] numbers = { 1, 2, 3, 4, 5 };

int lengthOfNumbers = numbers.Length;

Класс Array позволяет использовать много других полезных методов и свойств для выполнения сортировки, поиска и копирования массивов.

Инициализация массива

Массив можно инициализировать при объявлении. В этом случае спецификация ранга не нужна, поскольку она уже предоставлена по числу элементов в списке инициализации. Примеры.

int[] array1 = new int[] { 1, 3, 5, 7, 9 };

Строковый массив можно инициализировать таким же образом. Ниже приведено объявление строкового массива, в котором каждый элемент инициализируется названием дня:

string[] weekDays = { "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };

При инициализации массива при объявлении можно использовать следующие сочетания клавиш:

int[] array2 = { 1, 3, 5, 7, 9 };

string[] weekDays2 = { "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };

Можно объявить переменную массива без инициализации, но при присвоении массива этой переменной нужно использовать оператор new. Примеры.

int[] array3;

array3 = new int[] { 1, 3, 5, 7, 9 }; // OK

//array3 = {1, 3, 5, 7, 9}; // Error

**Практическая часть**

Блок-схема:







Код программы:

double[] ar = new double[10];

Random rand = new Random();

double max = 0.0;

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

{

ar[i] = Math.Round(rand.NextDouble() \* 100 - 50, 1);

}

foreach (var i in ar)

Console.WriteLine(i);

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

{

if (ar[i] > max)

max = ar[i];

}

Console.WriteLine("Максимальный элемент = {0}", max);

/\*

double[] ar = new double[10];

Random rand = new Random();

int lastPositive = 0;

double sum = 0.0;

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

{

ar[i] = Math.Round(rand.NextDouble() \* 100 - 50, 1);

}

foreach (var i in ar)

Console.WriteLine(i);

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

{

if (ar[i] > 0)

{

lastPositive = i;

break;

}

}

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

{

sum += ar[i];

}

Console.WriteLine("Сумма элементов до последнего положительного = {0}", sum);

/\*

int first = 5;

int last = 18;

int[] ar = new int[10];

int[] ar2 = new int[10];

Random rand = new Random();

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

{

ar[i] = rand.Next(-100, 101);

}

foreach (var i in ar)

Console.WriteLine(i);

Console.WriteLine();

for (int i = 0, j = 0; i < ar.Length; i++)

{

if (Math.Abs(ar[i]) < first || Math.Abs(ar[i]) > last)

{

ar2[j] = ar[i];

j++;

}

}

foreach (var i in ar2)

Console.WriteLine(i);

**Вывод:** изучена работа с одномерными массивами на языке C# в среде программирования Microsoft Visual Studio.

# Лабораторная работа №5

**Тема: «Двумерные массивы»**

**Цель работы:** Научиться работать с двумерными массивами на языке C#.

**Задание:** Вариант 5. Дана целочисленная квадратная матрица. Определить:

**-** сумму элементов в тех столбцах, которые не содержат отрицательных элементов;

**-** минимум среди сумм модулей элементов диагоналей, параллельной побочной диагонали матрицы.

**Краткая теория**

Простейшей формой многомерного массива является двумерный массив. Местоположение любого элемента в двумерном массиве обозначается двумя индексами. Такой массив можно представить в виде таблицы, на строки которой указывает один индекс, а на столбцы — другой.

// Объявляем двумерный массив

int[,] myArr = new int[a,b]; //a,b – высота и ширина массива

**Практическая часть**

Блок-схема:





Код программы:

namespace lab5

{

class Program

{

private static int sum1;

private static int sum2;

static void Main(string[] args)

{

int N = 3;

Random ran = new Random();

int[,] x = new int[3, 3];

int[,] z = new int[3, 3];

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

x[i, j] = ran.Next(-5, 5);

Console.Write("{0}\t", x[i, j]);

}

Console.WriteLine();

}

for (int j = 0; j < 3; j++)

{

int s = 0;

bool fl = true;

for (int k = 0; k < 3; k++)

{

s += x[k, j];

if (x[k, j] < 0) fl = false;

}

if (fl)

Console.WriteLine("Сумма элементов столбца [{0}] = " + s, j + 1);

}

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

{

for (int j = 0; j < N - i; j++)

{

z[i, j] = x[N - j - 1, N - i - 1];

}

}

for (int n = N; n > 0; n--)

{

for (int i = 0; i < N - 1; i++)

{

sum1 += Math.Abs(x[i, N - i - 2]);

sum2 += Math.Abs(z[i, N - i - 2]);

}

N--;

}

int min;

if (sum1 < sum2) min = sum1;

else min = sum2;

Console.WriteLine("Up: " + sum1 + " " + "Down: " + sum2 + "\n" + "\n");

Console.WriteLine("Минимальная сумма: " + min);

Console.ReadKey();

}

}

}

**Вывод:** изучена работа с двумерными массивами на языке C# в среде программирования Microsoft Visual Studio.