# Лабораторная работа № 4

**Тема:** Разработка и тестирование программного кода на языке С# с использованием циклов с параметром

**Цель работы:**

1. формирование навыков разработки программ на языке С# с использованием циклов с параметром;
2. формирование навыков работы в среде Visual Studio.Net;
3. изучение оператора цикла с параметром для описания циклических алгоритмов на С#

**Оборудование:**

1. Ноутбук MSI Modern
2. программное обеспечение: ОС Windows, среда Visual Studio.Net

**Ответы на контрольные вопросы:**

**1. Какие виды операторов цикла используются в языке С# ? (Написать название и общий вид оператора)**

В языке программирования C# используются следующие виды циклов:

1. Цикл for:

for (начальное значение; условие; инкремент)

{

// тело цикла

}

2. Цикл while:

while (условие)

{

// тело цикла

}

3. Цикл do-while:

do

{

// тело цикла

}

while (условие);

**2. Какой оператор С# используется для описания арифметических циклов?**

Оператор for используется для описания арифметических циклов в C#. Он позволяет выполнять повторяющийся код определенное количество раз, основываясь на числовых условиях (инициализация, условие продолжения и шаг итерации). Оператор for обеспечивает удобный способ управления итерациями, когда необходимо выполнить код заданное количество раз.

**3. Как выполняется оператор цикла с параметром в языке С# ?**

В C# оператор цикла с параметром представлен конструкцией for. Вот как он работает:

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

{

// тело цикла

}

При выполнении этого цикла происходят следующие действия:

1. Сначала выполняется инициализация, в данном случае int i = 0. Это делается только один раз, при входе в цикл.

2. Затем проверяется условие (i < 10). Если условие истинно, то управление передается внутрь цикла и выполняется код, находящийся в его теле.

3. После завершения каждой итерации цикла выполняется инкремент (i++), а затем снова проверяется условие. Если условие по-прежнему истинно, процесс повторяется.

4. Как только условие становится ложным, выполнение цикла прекращается и управление передается следующему оператору после цикла.

**4. Когда при использовании оператора цикла с параметром цикл будет бесконечным?**

Цикл будет бесконечным, если условие, управляющее выполнением цикла, всегда остается истинным. Например, при использовании оператора цикла while или do-while, если условие всегда остается истинным, цикл будет выполняться бесконечно.

Пример бесконечного цикла с использованием while:

while (true)

{

// Тело цикла

}

Пример бесконечного цикла с использованием do-while:

do

{

// Тело цикла

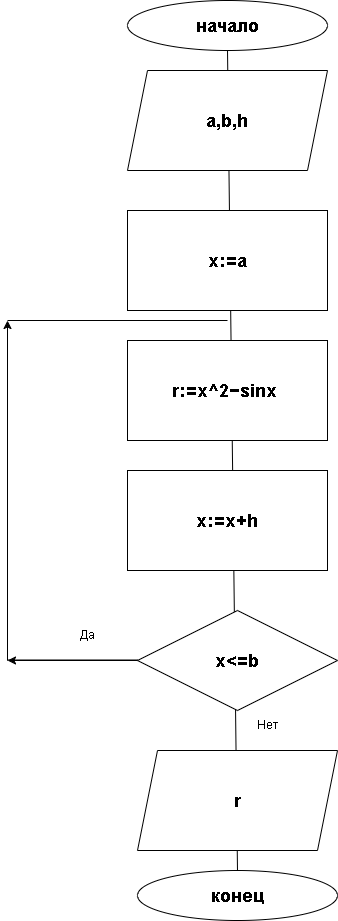
} while (true);

**Ход работы:**

1. Разработать алгоритм задачи и представить его в виде схемы программы
2. В Visual Studio.Net создать файл с программой
3. Используя средства Visual Studio.Net выполнить тестирование и отладку программу
4. Результаты представить в виде отчета
5. Сделать вывод о проделанной работе

**Задание на лабораторную работу:** вычислите значения функции f(x)=x^2−sinx на отрезке [a;b] с шагом h. Результат представить в виде таблицы.

**Схема программы**



**Листинг с исходным кодом**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp5

{

internal class Program

{

static void Main(string[] args)

{

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

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

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

Console.WriteLine("X\tF(X)");

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

{

double r = Math.Pow(x, 2) - Math.Sin(x);

Console.WriteLine($"{x}\t{r}");

}

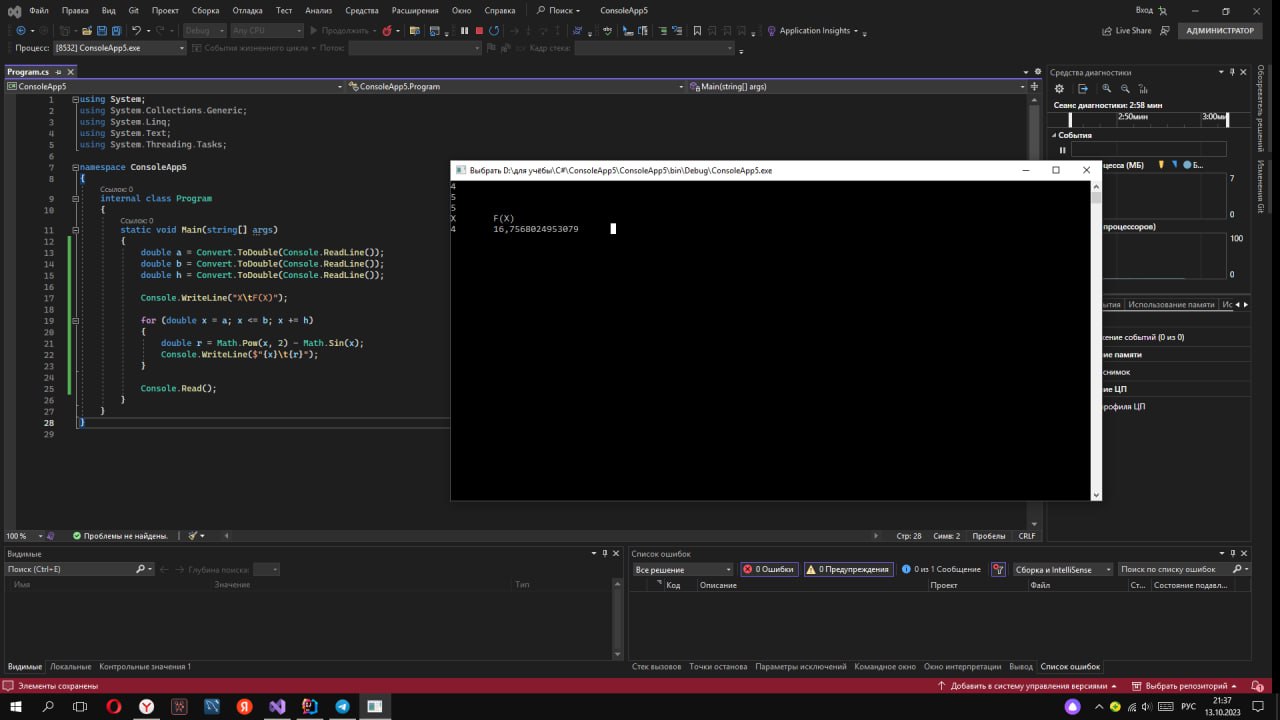
Console.Read();

}

}

}

**Результаты тестирования**



**Вывод по работе.** В ходе работы были изучены и применены разветвляющиеся алгоритмы в контексте программирования на языке C#. Эти алгоритмы, использующие условные операторы и циклы, являются фундаментальными элементами большинства программ. Разработка и тестирование кода на языке C# с использованием разветвляющихся алгоритмов помогает создавать эффективные и надежные приложения.