# Лабораторная работа №3

**Тема:** Разработка программ с использованием операторов выбора, цикла и передачи управления.

**Цель:** Научиться разрабатывать программы с использованием операторов выбора, цикла, передачи управления.

**Техническое оснащение:** Ноутбук Dell Latitude 13 9390, ОС Windows 10, Microsoft Word 2019, MS Visio 2016, MS VS 2019 CE.

**Выполнения работы**

**Вариант 18**

## Пример 1:

Проверка числа на чётность/нечётность

Листинг программы

using System;

using System.Windows.Forms;

using Microsoft.VisualBasic;

class OddEvenDemo

{

static void Main()

{

// Целочисленные переменные:

int number, reminder;

// Считывание целого числа:

number = Int32.Parse(

Interaction.InputBox(

// Текст в окне:

"Введите целое число:",

// Название окна:

"Проверка")

);

// Вычисляется остаток от деления на 2:

reminder = number % 2;

string txt = "Bы ввели ";

// Использован тернарный оператор:

txt += (reminder == 0 ? "четное" : "нечетное") + " число!";

MessageBox.Show(txt);

}

}

Запуск программы (рисунок 1).

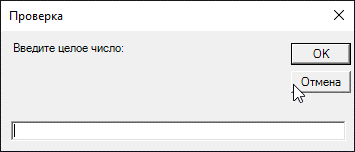


Рисунок 1 – Запуск программы примера 1

Вводим значение 100 и нажимаем на кнопку и получаем результат (рисунок 2).

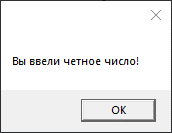


Рисунок 2 – Результат работы программы

## Пример 2:

Использование условного оператора

Листинг программы

using Microsoft.VisualBasic;

class UsingIDemo

{

static void Main()

{

MessageBoxIcon icon;

string msg, title, name;

name=Interaction.InputBox(

"Как Вас зозут?",

"Знакомимся");

if (name == "")

{

icon = MessageBoxIcon.Error;

msg = "Очень жаль, что мы не познакомились!";

title = "3накомство не состоялось";

}

else

{

icon=MessageBoxIcon.Information;

msg = "Очень приятно, " + name + "!";

title = "Знакомство состоялось";

}

MessageBox.Show(msg, title, MessageBoxButtons.OK, icon);

}

}

Запуск программы (рисунок 3).

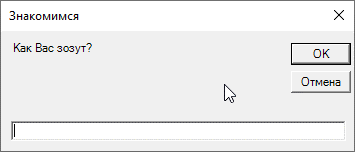


Рисунок 3 – Запуск программы примера 2

Вводим значение и нажимаем на кнопку ОК (рисунок 4).

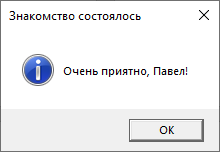


Рисунок 4 – Результат работы программы

## Пример 3:

Даны три числа: *a, b, c*. Могут ли эти числа быть длинами сторон треугольника? Если «да», то вычислить его площадь, используя формулу Герона.

Листинг программы

using System;

namespace Prim\_IF\_2

{

class Program

{

public static void Main(string[] args)

{

double a, b, c, p, s;

Console.Write("a=");

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

Console.Write("b=");

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

Console.Write("c=");

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

if (a + b > c && b + c > a && a + c > b)

{

Console.WriteLine("Треугольник существует");

p = (a + b + c) / 2;

s = Math.Sqrt(p \* (p - a) \* (p - b) \* (p - c));

Console.WriteLine("s={0}", s);

}

else

Console.WriteLine("Треугольник не существует");

Console.Write("Press any key to continue . . . ");

Console.ReadKey(true);

}

}

}

Запуск программы примера 3 (рисунок 5)

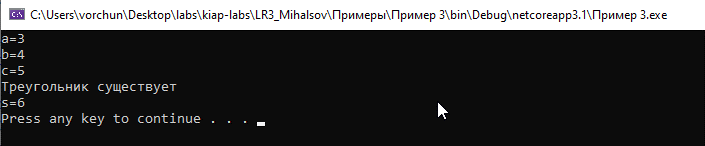


Рисунок 5 – Запуск программы примера 5

## Пример 4:

По номеру дня недели вывести на экране монитора название этого дня недели.

Листинг программы

using System;

namespace Prim\_switch

{

class Program

{

public static void Main(string[] args)

{

Console.WriteLine("Задайте номер недели:");

int n = int.Parse(Console.ReadLine());

switch (n)

{

case 1:

Console.WriteLine("Понедельник");

break;

case 2:

Console.WriteLine("Вторник");

break;

case 3:

Console.WriteLine("Среда");

break;

case 4:

Console.WriteLine("Четверг");

break;

case 5:

Console.WriteLine("Пятница");

break;

case 6:

Console.WriteLine("Суббота");

break;

case 7:

Console.WriteLine("Воскресенье");

break;

default:

Console.WriteLine("Неверный номер для дня недели: {0}", n);

break;

}

Console.Write("Press any key to continue . . .");

Console.ReadKey(true);

}

}

}

Запуск программы примера 4 (рисунок 6).

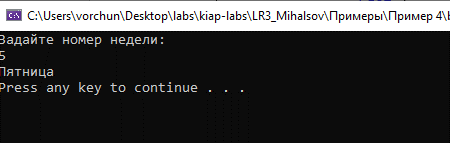


Рисунок 6 – Работа программы примера 4

## Пример 5:

Найти сумму квадратов первых *n* натуральных чисел (Цикл while)

Листинг программы

using System;

namespace Prim\_while

{

class Program

{

public static void Main(string[] args)

{

int i, n, s;

Console.Write("n=");

n = int.Parse(Console.ReadLine());

s = 0;

i = 1;

while (i <= n)

{

s += i \* i;

i++;

}

Console.WriteLine("s={0}", s);

Console.Write("Press any key to continue . . . ");

Console.ReadKey(true);

}

}

}

Работа программы примера 5 (рисунок 7).

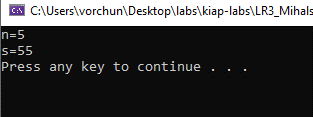


Рисунок 7 – Работа программы примера 5

## Пример 6:

Найти сумму квадратов первых *n* натуральных чисел (Цикл do)

Листинг программы

using System;

namespace Prim\_do

{

class Program

{

public static void Main(string[] args)

{

int i, n, s;

do

{

Console.WriteLine("Задайте n>0:");

n = int.Parse(Console.ReadLine());

}

while (n < 1);

s = 0;

i = 1;

do

{

s += i \* i;

i++;

}

while (i <= n);

Console.WriteLine("s={0}", s);

Console.Write("Press any key to continue . . . ");

Console.ReadKey(true);

}

}

}

Работа программы примера 6 (рисунок 8).

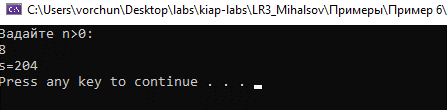


Рисунок 8 – Проверки работы примера 6

## Пример 7:

Найти сумму квадратов первых *n* натуральных чисел (Цикл for)

Листинг программы

using System;

namespace Prim\_for

{

class Program

{

public static void Main(string[] args)

{

int i, n, s;

do

{

Console.WriteLine("Задайте n>0:");

n = int.Parse(Console.ReadLine());

}

while (n < 1);

for (i = 1, s = 0; i <= n; i++)

s += i \* i;

Console.WriteLine("s={0}", s);

Console.Write("Press any key to continue . . . ");

Console.ReadKey(true);

}

}

}

Проверка работы программы (рисунок 9).

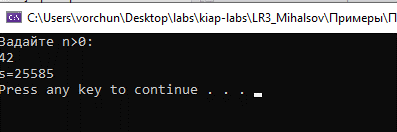


Рисунок 9 – Проверка программы примера 7

## Задание 1:

Составить графическую схему алгоритма и написать программу вычисления выражения приведенного выражения. В программе предусмотреть вывод значений x и y в консоль.

Блок схема алгоритма программы (рисунок 10).



Рисунок 10 – Блок схема алгоритма программы

Листинг программы

using System;

namespace Задание\_1

{

class Program

{

static void Main(string[] args)

{

float x;

double y;

Console.Write("Введиете X = ");

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

if (x <= 0)

{

y = Math.Exp(2.0 \* x);

Console.WriteLine("Выбрано условие: x <= 0");

}

else

{

if (0.0 < x && x < 7.0)

{

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

Console.WriteLine("Выбрано условие: 0 < x < 7");

}

else

{

y = x / 2.0 - Math.Pow(x, 2);

Console.WriteLine("Выбрано условие: Во всех остальных случаях");

}

}

Console.WriteLine("y = {0}", y);

}

}

}

Проверка работы программы

Проверка условия номер один (рисунок 11).

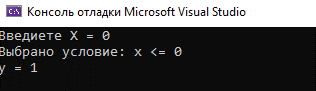


Рисунок 11 – Проверка работы по первому условию

Проверка условия номер два (рисунок 12).

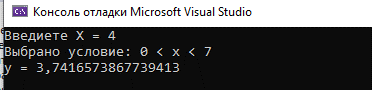


Рисунок 12 – Проверка работы по второму условию

Проверка условия номер три (рисунок 13).

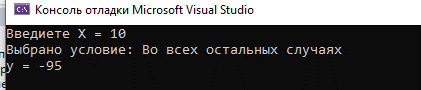


Рисунок 13 – Проверка программы по третьему условию

## Задание 2:

Создать реализацию Задания 1, с тем условиям что данные необходимо вводить с формы с компонентами.

Создание прототипа формы (рисунок 14).

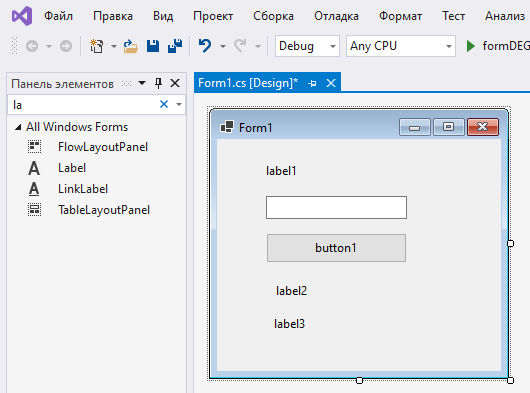


Рисунок 14 – Прототип формы программы

На данной форме присутствуют следующие компоненты: label1 – указывающий что необходимо ввести в текстовое поле; textbox1 – необходимый для ввода значения x; label2 – необходимый для вывода выбранного метода расчета значения переменной; label3 – необходимый для вывода значения переменной y.

Преобразованная форма (рисунок 15).

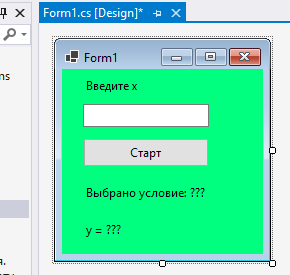


Рисунок 15 – Преобразованная форма программы

Листинг программы

using System;

using System.Windows.Forms;

namespace \_1\_WindowsFormsApp

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

float x;

double y;

x = float.Parse(textBox1.Text);

if( x <= 0)

{

y = Math.Exp(2.0 \* x);

label2.Text = "Выбрано условие: x <= 0";

}

else

{

if (0.0 < x && x < 7.0)

{

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

label2.Text = "Выбрано условие: 0 < x < 7";

}

else {

y = x / 2.0 - Math.Pow(x, 2);

label2.Text = "Выбрано условие: \nВо всех остальных случаях";

}

}

label3.Text = "y = " + y.ToString();

}

}

}

Проверка программы

Проверка первого условия (рисунок 16).

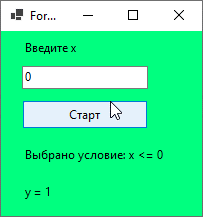


Рисунок 16 – Проверка программы по первому условию

Проверка второго условия (рисунок 17).

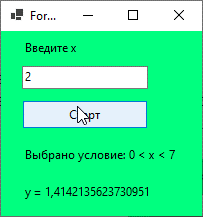


Рисунок 17 – Проверка программы по второму условию

Проверка третьего условия (рисунок 18).

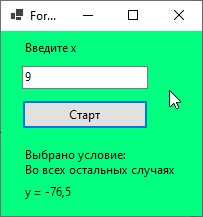


Рисунок 18 – Проверка программы по третьему условию

## Задание 3:

В соответствии с приведенной функцией вычислить значения функции y=f (x, a, b) для значений аргумента x, изменяющегося в интервале от xнач до xкон с шагом ∆x, и заданных констант a и b. Исходные данные вводить с консоли.

Блок схема программы (рисунок 19).



Рисунок 19 – Блок схема работы программы

Листинг программы

using System;

namespace \_2\_ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

float x, dx, x0, xn, a, b;

double y;

Console.WriteLine("Введите x0 = ");

x0 = float.Parse(Console.ReadLine());

Console.WriteLine("Введите xn = ");

xn = float.Parse(Console.ReadLine());

Console.WriteLine("Введите dx = ");

dx = float.Parse(Console.ReadLine());

Console.WriteLine("Введите a = ");

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

Console.WriteLine("Введите b = ");

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

for(x = x0; x <= xn; x += dx)

{

y = Math.Sqrt((a + b\*x)/(Math.Pow(Math.Log(x),2)));

Console.WriteLine("y = {1},\tx = {0}", x, y);

}

}

}

}

Проверка работы программы (рисунок 20)

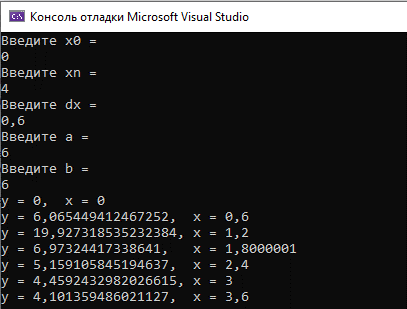


Рисунок 20 – Проверка работы программы

## Вывод:

В результате выполнения лабораторной работы были получены практические навыки разработки программ на языке C# с использованием операторов выбора и циклов в программе.