**Лабораторная работа №4**

**РАЗРАБОТКА ПРОГРАММ, РЕАЛИЗУЮЩИХ ВЕТВЛЕНИЕ: ПРОСТЕЙШИЕ СЛУЧАИ**

**Цели:**

1. Научиться создавать консольные приложения, реализующие ветвление.

**Приложение Lab04\_01. Определение четности числа.**

***Задание*:**

Разработать консольное приложение, определяющее, является ли введенное число четным. Результатом работы программы должен быть вывод одного из двух сообщений: «Введено четное число» либо «Введено нечетное число».

***Константы:*** отсутствуют.

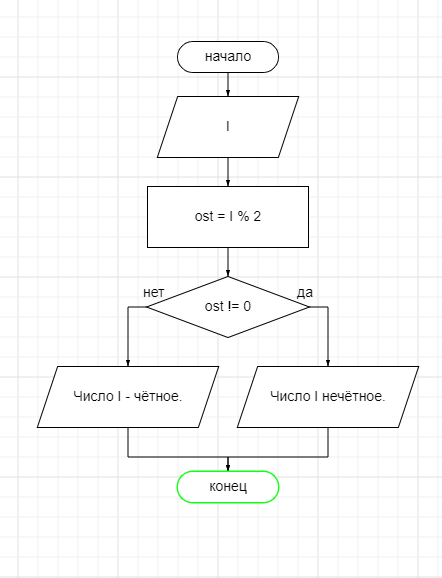
***Переменные:*** ost, I – переменная типа int.

***Исходные данные:*** I – число целого типа, чётность которого нужно определить.

***Результат:***

Программа определяет чётное число или нет.

***Блок-схема:***



***Код приложения:***

**Листинг 1 – Код консольного приложения Lab04\_01**

namespace PuzanovVE.OP.Lab04\_01

{

class Program

{

static void Main(string[] args)

{

//Стандартная шапка

Console.WriteLine("Лабороторная работа №4");

Console.WriteLine("Приложение Lab04\_01");

Console.WriteLine("Разроботчик: Пузанов В. Е.");

Console.WriteLine("------------------------------------------------------");

//информация о задаче

Console.WriteLine("Определение четности числа");

Console.WriteLine("------------------------------------------------------");

//основной код программы

int I, ost;

Console.WriteLine("Вас приветсвует программа по определению, чётное число или нет.\nВведите целое число:");

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

ost = I % 2;

if (ost != 0)

{

Console.WriteLine("Число " + I + " - нечётное.");

}

else

{

Console.WriteLine("Число " + I + " - чётное.");

}

Console.ReadLine();

}

}

}

**Тестирование:**

Контрольный пример 1:

Исходные данные:

I = 5;

Результат:

Число 5 – нечётное;

Результат работы программы для указанных исходных данных приведён на рисунке 1.1

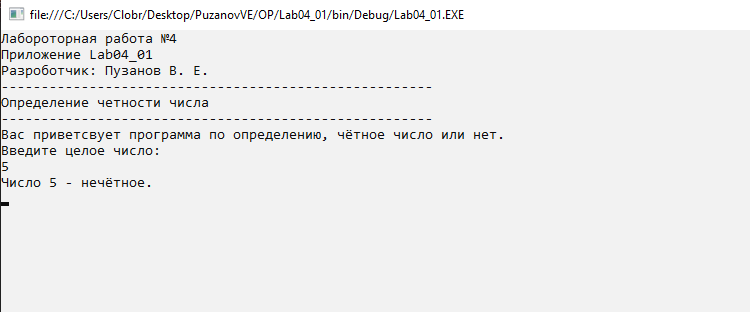


Рисунок 1.1 –Результат работы программы Lab04\_01.

Контрольный пример 2:

Исходные данные:

I = 22;

Результат:

Число 22 – чётное;

Результат работы программы для указанных исходных данных приведён на рисунке 1.2

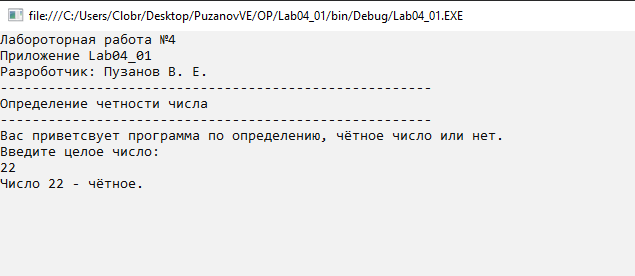


Рисунок 1.2 –Результат работы программы Lab04\_01.

Контрольный пример 3:

Исходные данные:

I = -5;

Результат:

Число -5 – нечётное;

Результат работы программы для указанных исходных данных приведён на рисунке 1.3

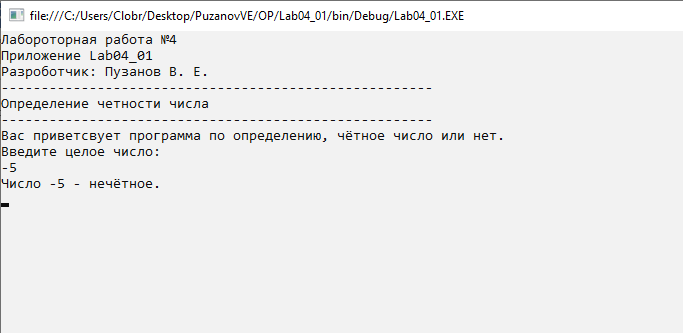


Рисунок 1.3 –Результат работы программы Lab04\_01.

**Приложение Lab04\_02. Модифицированный алгоритм определения четности числа.**

***Задание*:**

Модифицировать алгоритм определения четности числа.

***Константы:*** отсутствуют.

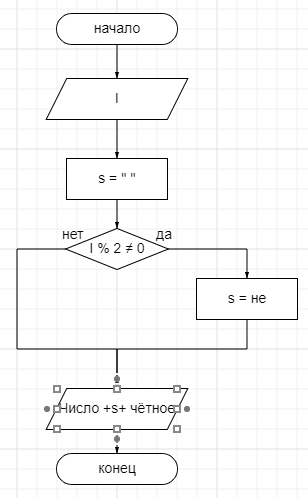
***Переменные:*** ost, I – переменная типа int; s – переменная типа string.

***Исходные данные:*** I – число целого типа, чётность которого нужно определить.

***Результат:***

Программа определяет чётное число или нет.

***Блок-схема:***



***Код приложения:***

**Листинг 2 – Код консольного приложения Lab04\_02**

namespace PuzanovVE.OP.Lab04\_02

{

class Program

{

static void Main(string[] args)

{

//Стандартная шапка

Console.WriteLine("Лабороторная работа №4");

Console.WriteLine("Приложение Lab04\_02");

Console.WriteLine("Разроботчик: Пузанов В. Е.");

Console.WriteLine("------------------------------------------------------");

//информация о задаче

Console.WriteLine("Модифицированный алгоритм определения четности числа");

Console.WriteLine("------------------------------------------------------");

//основной код программы

int I;

string s = "";

Console.WriteLine("Вас приветсвует программа по определению, чётное число или нет.\nВведите целое число:");

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

if (I % 2 != 0)

{

s = "не";

}

Console.WriteLine("Число " + I + " - " + s +"чётное.");

Console.ReadLine();

}

}

}

**Тестирование:**

Контрольный пример 1:

Исходные данные:

I = 5;

Результат:

Число 5 – нечётное;

Результат работы программы для указанных исходных данных приведён на рисунке 2.1

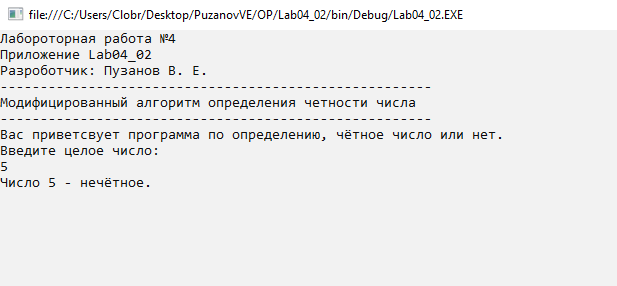


Рисунок 2.1 –Результат работы программы Lab04\_02.

Контрольный пример 2:

Исходные данные:

I = 22;

Результат:

Число 22 – чётное;

Результат работы программы для указанных исходных данных приведён на рисунке 2.2

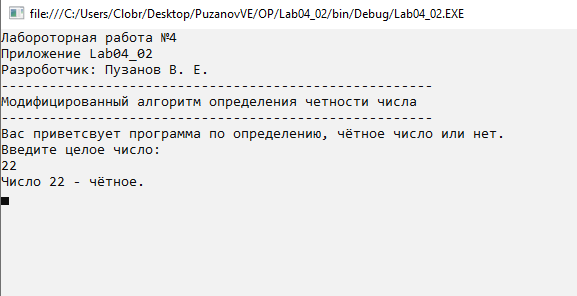


Рисунок 2.2 –Результат работы программы Lab04\_02.

Контрольный пример 3:

Исходные данные:

I = -5;

Результат:

Число -5 – нечётное;

Результат работы программы для указанных исходных данных приведён на рисунке 2.3

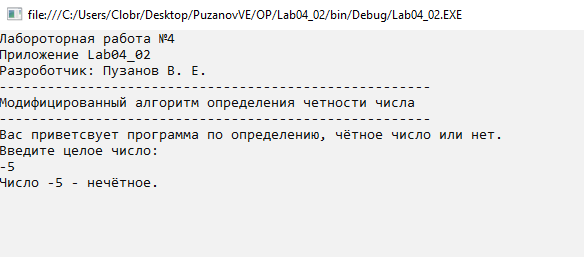


Рисунок 2.3 –Результат работы программы Lab04\_02.

**Приложение Lab04\_03.Определение существования действительных корней квадратного уравнения**

***Задание*:**

Разработать консольное приложение, определяющее, существуют ли действительные корни у квадратного уравнения 0 2 ax + bx + c = , где значения коэффициентов a , b и c вводятся пользователем. Результатом работы программы должен быть вывод одного из двух сообщений: «Уравнение не имеет действительных корней» либо «Уравнение имеет действительные корни».

***Константы:*** отсутствуют.

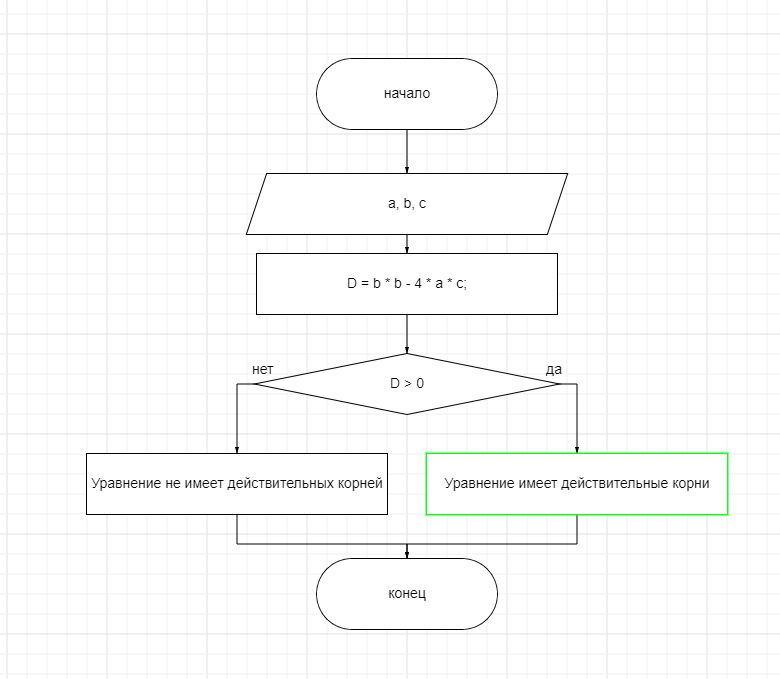
***Переменные:*** a, b, c, D – переменные типа double.

***Исходные данные:*** a, b, c – коифиценты уравнения.

***Результат:***

Программа определяет существуют ли корни у уравнения или нет.

***Блок-схема:***



***Код приложения:***

**Листинг 3 – Код консольного приложения Lab04\_03**

namespace PuzanovVE.OP.Lab04\_03

{

class Program

{

static void Main(string[] args)

{

//Стандартная шапка

Console.WriteLine("Лабороторная работа №4");

Console.WriteLine("Приложение Lab04\_03");

Console.WriteLine("Разроботчик: Пузанов В. Е.");

Console.WriteLine("------------------------------------------------------");

//информация о задаче

Console.WriteLine("Определение существования действительных корней квадратного уравнения ");

Console.WriteLine("------------------------------------------------------");

//основной код программы

double a, b, c, D;

Console.WriteLine("Вас приветствует программа, по определению имеет ли уравнение << ax^2 + bx +c >> действительные корни или нет.\nВведите коэфиценты.");

Console.WriteLine("a:");

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

Console.WriteLine("b:");

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

Console.WriteLine("c:");

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

D = b \* b - 4 \* a \* c;

if (D > 0)

{

Console.WriteLine("Уравнение имеет действительные корни");

}

else

{

Console.WriteLine("Уравнение не имеет действитель\_ных корней");

}

Console.ReadLine();

}

}

}

**Тестирование:**

Контрольный пример 1:

Исходные данные:

a = 5;

b = 6;

c = 2;

Результат:

Уравнение не имеет действительных корней;

Результат работы программы для указанных исходных данных приведён на рисунке 3.1

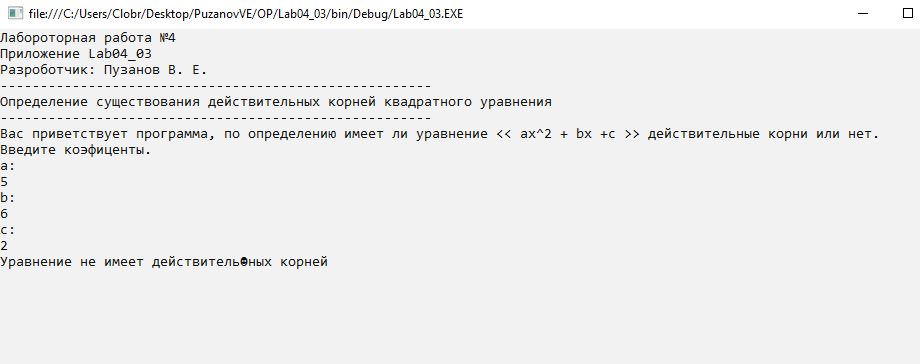


Рисунок 3.1 –Результат работы программы Lab04\_03.

Контрольный пример 2:

Исходные данные:

a = -22;

b = 3;

c = 3;

Результат:

Уравнение имеет действительные корни.

Результат работы программы для указанных исходных данных приведён на рисунке 3.2

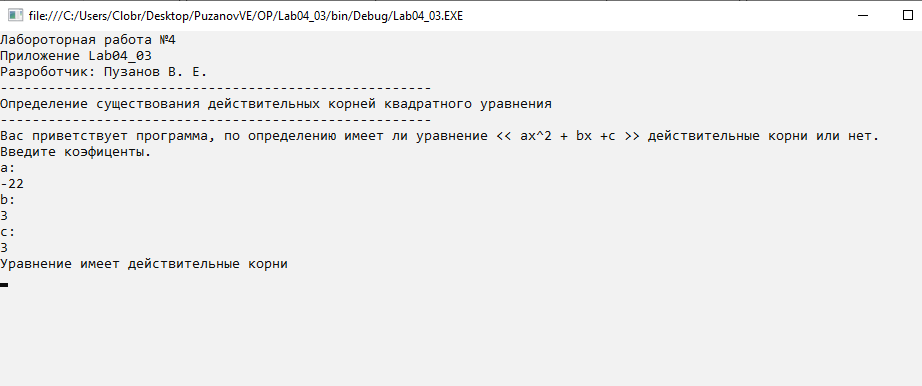


Рисунок 3.2 –Результат работы программы Lab04\_03.

Контрольный пример 3:

Исходные данные:

a = 8,6;

b = 3;

c = 3;

Результат:

Число -5 – нечётное;

Результат работы программы для указанных исходных данных приведён на рисунке 3.3

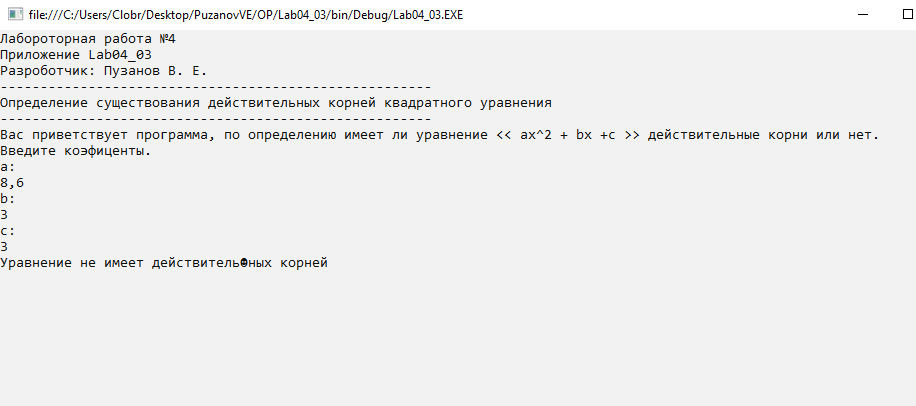


Рисунок 3.3 –Результат работы программы Lab04\_03.

**Приложение Lab04\_04.** **Вычисление сопротивления участка электрической цепи**

***Задание*:**

Разработать консольное приложение, реализующее алгоритм вычисления сопротивления участка электрической цели, состоящего из двух резисторов с известными сопротивлениями. Пользователю должна быть предоставлена возможность выбора способа соединения резисторов: последовательное либо параллельное

***Константы:*** отсутствуют.

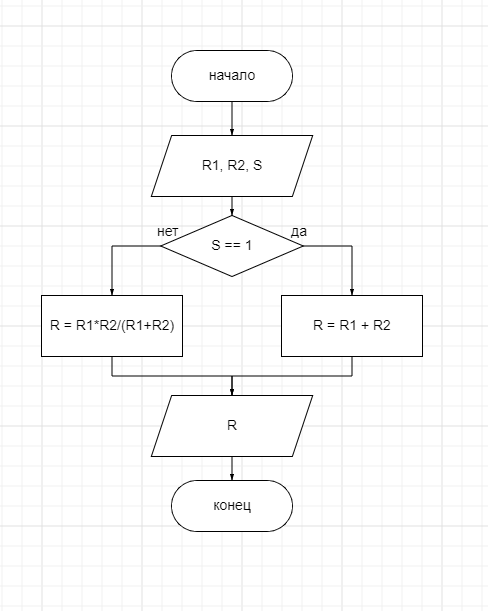
***Переменные:*** R, R1, R2, S – переменные типа double.

***Исходные данные:*** R1, R2 – сопротивления 1го и 2го резисторов; S – переменная, по значению которой определяется тип сопротивления.

***Результат:***

Программа высчитывает напряжение по заданным данным.

***Блок-схема:***



***Код приложения:***

**Листинг 4 – Код консольного приложения Lab04\_04**

namespace PuzanovVE.OP.Lab04\_04

{

class Program

{

static void Main(string[] args)

{

//Стандартная шапка

Console.WriteLine("Лабороторная работа №4");

Console.WriteLine("Приложение Lab04\_04");

Console.WriteLine("Разроботчик: Пузанов В. Е.");

Console.WriteLine("------------------------------------------------------");

//информация о задаче

Console.WriteLine("Вычисление сопротивления участка электрической цепи");

Console.WriteLine("------------------------------------------------------");

//основной код программы

double R1, R2, S, R;

Console.WriteLine("Вас приветствует программа по вычислению сопротивления 2x резисторов в сети.\nВведите сопротивление резисторов.");

Console.WriteLine("R1");

R1 = double.Parse(Console.ReadLine());

Console.WriteLine("R2");

R2 = double.Parse(Console.ReadLine());

Console.WriteLine("Выберите тип соеденения:\nПоследовательное - 1\nПаралельное - 2");

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

if (S == 1)

{

R = R1 + R2;

Console.WriteLine("Сопротивление в цепи равно - " + R);

}

else if(S == 2)

{

R = R1 \* R2 / (R1 + R2);

Console.WriteLine("Сопротивление в цепи равно - " + R);

}

else

{

Console.WriteLine("Выбран неправильный тип соеденения");

}

Console.ReadLine();

}

}

}

**Тестирование:**

Контрольный пример 1:

Исходные данные:

R1 = 5;

R2 = 6;

S = 1;

Результат:

Сопротивление цепи равно - 11;

Результат работы программы для указанных исходных данных приведён на рисунке 4.1

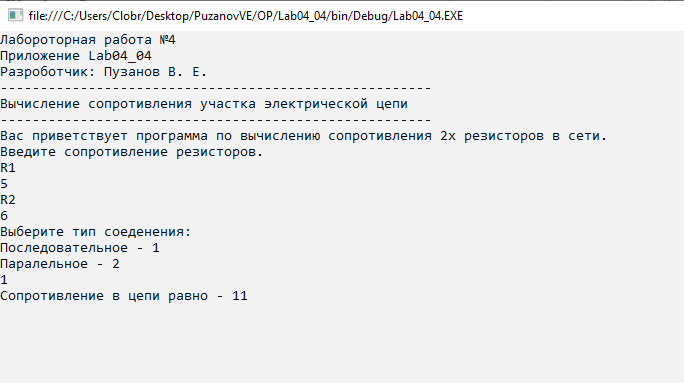


Рисунок 4.1 –Результат работы программы Lab04\_04.

Контрольный пример 2:

Исходные данные:

R1 = 21;

R2 = 43;

S = 2;

Результат:

Сопротивление в цепи равно - 14,109375

Результат работы программы для указанных исходных данных приведён на рисунке 4.2

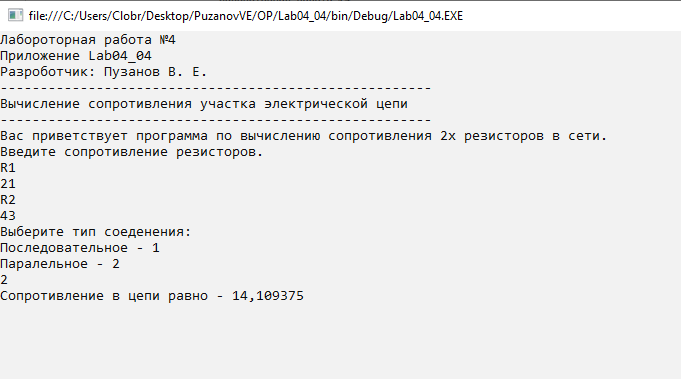


Рисунок 4.2 –Результат работы программы Lab04\_04.

Контрольный пример 3:

Исходные данные:

R1 = 1;

R2 = 4;

S = 5;

Результат:

Выбран неправильный тип соеденения

Результат работы программы для указанных исходных данных приведён на рисунке 4.3

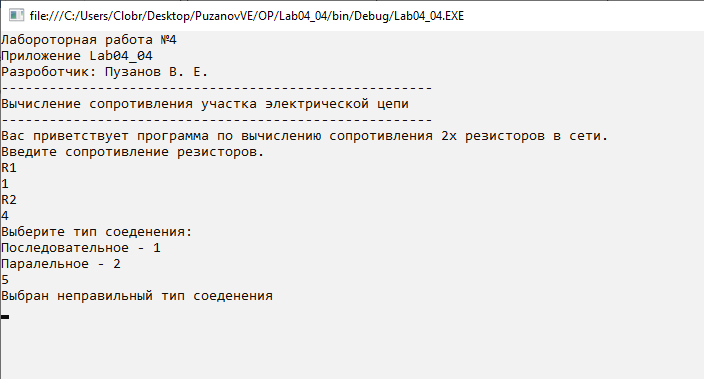


Рисунок 4.3 –Результат работы программы Lab04\_04.

**Приложение Lab04\_05.** **Вычисление суммы и процентной ставки годового налога.**

***Задание*:**

Разработать консольное приложение, вычисляющее сумму годового налога и общий налог в процентах.

***Константы:*** отсутствуют.

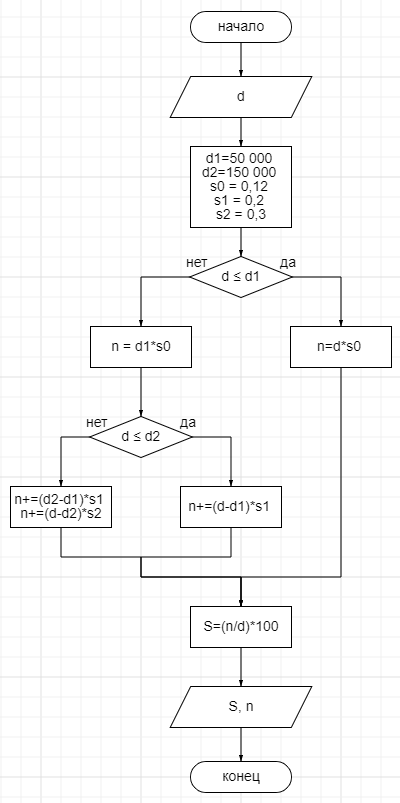
***Переменные:*** d, d1, d2, S0, S1, S2, S, n – переменные типа double.

***Исходные данные:*** d – размер годового налога.

***Результат:***

Программа высчитывает сумму налога и подходящей процентной ставки налога на годовой доход.

***Блок-схема:***



***Код приложения:***

**Листинг 5 – Код консольного приложения Lab04\_05**

namespace PuzanovVE.OP.Lab04\_05

{

class Program

{

static void Main(string[] args)

{

// Стандартная шапка

Console.WriteLine("Лабораторная работа №4");

Console.WriteLine("Приложение Lab04\_05");

Console.WriteLine("Разработчик: Пузанов В.Е.");

Console.WriteLine("------------------------");

// Информация о задаче

Console.WriteLine("Вычисление суммы годового налога");

Console.WriteLine("------------------------");

// Основной код программы

Console.WriteLine("Введите размер годового налога:");

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

double d1 = 50000;

double d2 = 150000;

double s0 = 0.12;

double s1 = 0.2;

double s2 = 0.3;

double n, S;

if (d <= d1)

{

n = d \* s0;

}

else

{

n = d1 \* s0;

if (d <= d2)

{

n += (d - d1) \* s1;

}

else

{

n += (d2 - d1) \* s1;

n += (d - d2) \* s2;

}

}

S = (n / d) \* 100;

Console.WriteLine("Сумма годового налога равна " + n);

Console.WriteLine("Общий налог равен " + S + " %");

Console.ReadLine();

Console.ReadLine();

}

}

}

**Тестирование:**

Контрольный пример 1:

Исходные данные:

d = 40000;

Результат:

Сумма годового налога равна 4800

Общий налог равен 12 %

Результат работы программы для указанных исходных данных приведён на рисунке 5.1

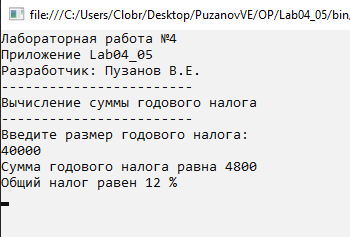


Рисунок 5.1 –Результат работы программы Lab04\_05.

Контрольный пример 2:

Исходные данные:

d = 100000;

Результат:

Сумма годового налога равна 16000

Общий налог равен 16 %

Результат работы программы для указанных исходных данных приведён на рисунке 5.2

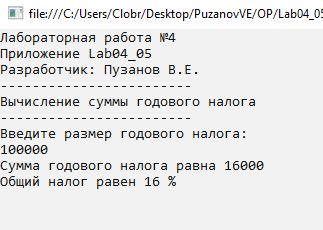


Рисунок 5.2 –Результат работы программы Lab04\_05.

Контрольный пример 3:

Исходные данные:

d = 200000;

Результат:

Сумма годового налога равна 41000

Общий налог равен 20,5 %

Результат работы программы для указанных исходных данных приведён на рисунке 5.3

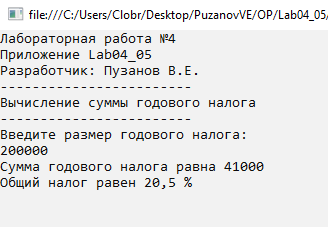


Рисунок 5.3 –Результат работы программы Lab04\_05.

Выполнил студент Пузанов В. Е., ФИТУ 010304-КМСб-о22

Проверил ст. преподаватель каф. ПМ Черноиван Д.Н.