**Лабораторная работа №7**

**Итерационные циклические вычислительные процессы с управлением по функции.**

**Цель:** научиться итерационным циклическим процессам с управлением по функции средствами компилятора free Pascal.

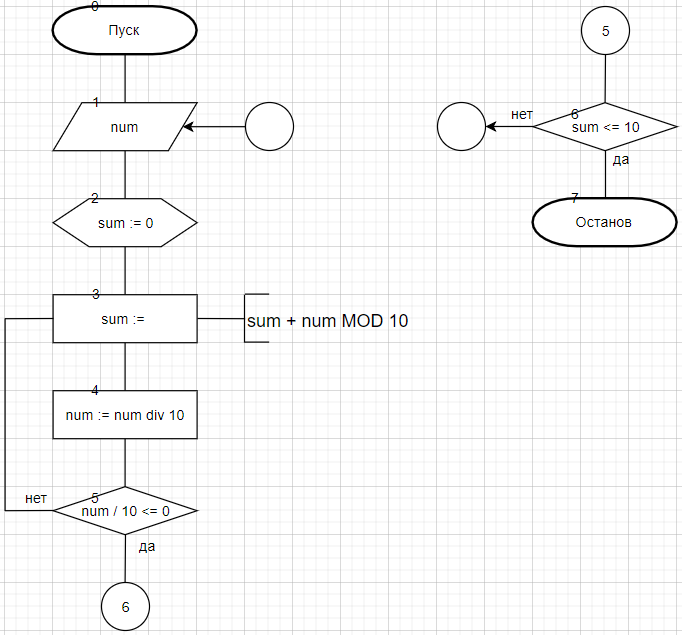
**Используемое оборудование:** пк, PascalABC.

**Задание 1:**

**4.1)** Реализовать программу, которая принимает трехзначное число, после чего проверяет сумму цифр этого числа, и если эта сумма больше 10, то вводиться следующее трехзначное число и так далее, иначе программа завершается.

**5.1)**  sum = sum + num MOD 10, num = num div 10

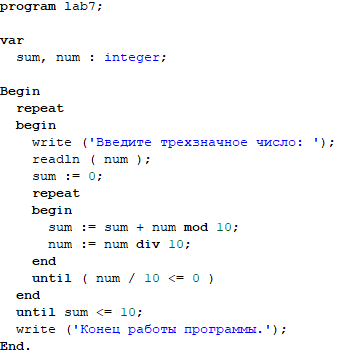
**6.1)**



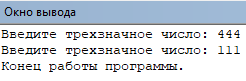
**7.1)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| **sum** | **Переменная, сумма цифр трехзначного числа “num”** | **Int.** |
| **num** | **Переменная, трехзначное число** | **Int.** |

**8.1)**



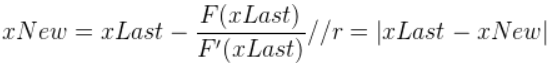
**9.1)**



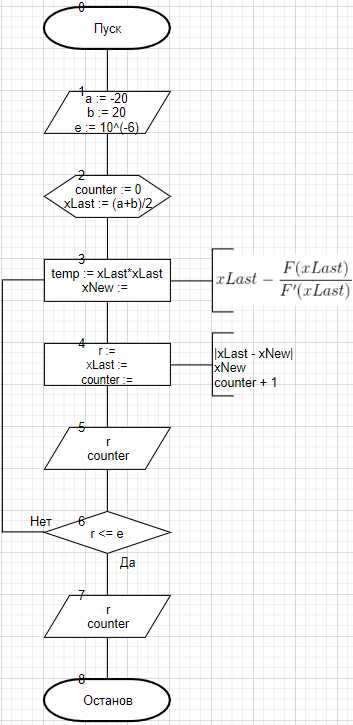
**10.1**) Задача решена при помощи основных математических операций, а также двух циклов конструкций while и repeat until. Сначала работы программы мы попадаем в цикл с постусловием, repeat until, там пользователь вводит трехзначное число “num”, после чего обнуляется переменная суммы “sum”, а дальше попадаем в следующий цикл. В цикле while задается условие, num / 10 > 0, пока выражение принимает истину, цикл считает сумму цифр трехзначного числа ”num”. После выхода из цикла while мы проверяем условие первого цикла, sum > 10, если ложь, то цикл возвращается в начало, иначе конец программы.

**Задача 2:**

**4.2)** Решить нелинейное уравнение методом Ньютона из индивидуального задания.

**5.2)** , xLast = (a+b)/2

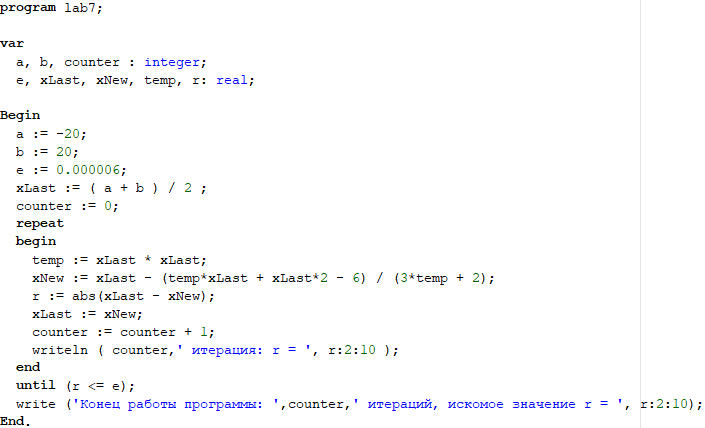
**6.2)**



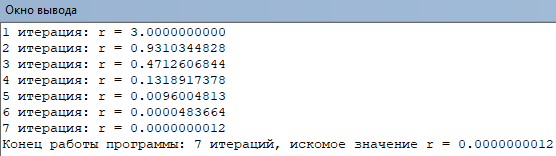
**7.2)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| **a** | **Переменная, левая граница криволинейной функции** | **Int.** |
| **b** | **Переменная, правая граница криволинейной функции** | **Int.** |
| **counter** | **Переменная, счетчик итераций.** | **Int.** |
| **e** | **Переменная, заданная точность** | **Real.** |
| **xLast** | **Переменная, некая точка на оси абсциссы** | **Real.** |
| **xNew** | **Переменная, некая точка на оси абсциссы** | **Real.** |
| **r** | **Перменная, погрешность** | **Real** |
| **temp** | **Промежуточная переменная, содержит квадрат значения xLast** | **Real** |

**8.2)**



**9.2)**



**10.2)** Задача была выполнена при помощи основных математических операций, функции abs() и цикла с постусловием , until repeat. Начальная точка вычислений определяется формулой, (a + b) / 2, после обнуляется счетчик итераций, дальше мы попадаем в цикл с постусловием. Для упрощения вычислений машине в переменной “temp” содержится квадрат значения xLast, в переменной “xNew” вычисляется следующее значение. Следующим шагом, находится погрешность по формуле |xLast – xNew|, которую записываем в переменную ”r”. После увеличения счетчика на 1, проверяется условие, если r <= e истина, то программа выходит из цикла и завершает работу, в ином случае цикл возвращается в начала с новыми значениями.

**11)** Мы научились реализовывать алгоритмы итерационных циклических процессов с управлением по функции средствами компилятора free Pascal.