Министерство образования Республики Беларусь

Учреждение образования

«Гомельский государственный университет им. Ф. Скорины»

Физический факультет

Кафедра общей физики

«Операторы цикла»

**Отчёт**

Выполнил: Никитин Т.И

Проверил: Шамына А.А.

Гомель 2022

Лабораторная работа №3

# Операторы цикла

**Цель работы:** усвоение принципов работы операторов цикла; построение программ, содержащих операторы цикла.

# Краткие теоретические сведения

**Цикл *for***

Синтаксис:

**for** ( *Инициализация; Условие\_Выполнения; Изменение )*

{

*//* Здесь инструкции цикла (тело цикла)

}

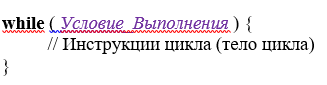
*Инициализация* — инструкция инициализации счетчика циклов.

*Условие\_Выполнения —* выражение, значение которого определяет условие выполнения инструкций цикла. Инструкции цикла выполняются до тех пор, пока *Условие\_Выполнения* истинно, т. е. не равно нулю.

*Изменение* — инструкция изменения параметра цикла. Как правило, эта инструкция изменяет значение переменной, которая входит в *Условие\_Выполнения.*

* инструкция **for** используется для организации циклов с фиксированным, известным во время разработки программы, числом повторений;
* количество повторений цикла определяется начальным значением переменной-счетчика и условием завершения цикла;
* переменная-счетчик должна быть целого *(****int****)* типа и может быть объявлена непосредственно в инструкции цикла

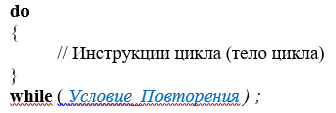
# Цикл while



Сначала проверяется значение выражения *Условие\_Вылолнения.* Если оно не равно нулю, т.е. условие истинно, то выполняйте инструкции цикла (тело цикла). Затем снова проверяется значение выражения *Условие\_Вылолнения*, и если оно не равно нулю, инструкции цикла выполняются еще раз. И так до тех пор, пока значение выражения *Условие\_Вылолнения* не станет равным нулю.

* число повторений инструкций цикла **while** определяется ходом выполнения программы;
* инструкции цикла **while** выполняются до тех пор, пока значение выражения, записанного после слова **while,** не станет ложным (равным нулю);
* после слова **while** надо записывать условие выполнения инструкций цикла;
* для завершения цикла **while** в теле цикла обязательно должны быть инструкции, выполнение которых влияет на условие завершения цикла;
* цикл **while** — это цикл с предусловием, т. е. возможна ситуация, при которой инструкции тела цикла ни разу не будут выполнены;
* цикл **while,** как правило, используется для организации приближенных вычислений, в задачах поиска и обработки данных, вводимых с клавиатуры или из файла.

**Цикл *do…while***



Сначала выполняются инструкции цикла (тело цикла), затем проверяется значение выражения *Условие\_Повторения,* и если условие истинно, не равно нулю, то инструкции цикла выполняются еще раз. И так до тех пор, пока *Условие\_Повторения* не станет ложным, т. е. равным нулю.

* + число повторений инструкций цикла **do…while** определяется ходом выполнения программы;
  + инструкции цикла **do…while** выполняются до тех пор, пока значение выражения, записанного после слова **while,** не станет ложным (равным нулю);
  + после слова **while** надо записывать условие выполнения инструкций

цикла;

# Практическая часть

## Упражнение 1

## Сумма 10000 руб. положена в сберегательный банк под 3% годовых (процент капитализированный). Составить алгоритм, определяющий через какой промежуток времени первоначальная сумма увеличится в 2 раза.

## C:\Users\Admin\Downloads\diagram (10).png

## Листинг программы:

## #include<stdio.h>

## void main()

## {

## int a,b,c;

## b=10000\*0.03;

## c=0;

## 

## printf("Primary amount - 10000 RUB.\n");

## while(a<=20000) {

## a+=b;

## c++;

## }

## printf("After ");

## printf("%d", c);

## printf(" year, the original amount will have doubled.");

## }

## Упражнение 2

## Найти натуральное число, состоящее из трёх цифр, с возрастающими слева направо цифрами, являющееся полным квадратом. Число является полным квадратом, если квадратный корень из него есть простое число (число 121 – полный квадрат, т.к. 121=11\*11, а 11 – простое число)

## Листинг программы:

## #include <stdio.h>

## #include <stdbool.h>

## #include <math.h>

## bool isPrime(int s)

## {

## for (int i = 2; i < s; i++)

## if (s % i == 0)

## return false;

## 

## return true;

## }

## int main()

## {

## int n, m, s, a, b, c;

## 

## for (int m = 100; m <= 999; m++) {

## 

## s = sqrt(m);

## 

## c = m % 10;

## b = m / 10 % 10;

## a = m / 100 % 10;

## 

## if (s\*s==m && isPrime(s) == true && c>b && b>a)

## printf("%d\n", s\*s);

## }

## }

## C:\Users\Admin\Downloads\diagram (13).png