

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Тульский государственный университет»

Институт прикладной математики и компьютерных наук

Кафедра информационной безопасности

Программирование

Лабораторная работа № 3

Тема

Изучение операторов цикла языка C++.

Выполнил

Студент гр. Б260221

Воробьёв А.А.

Проверила

Сафронова М.А.

Тула 2023 г.

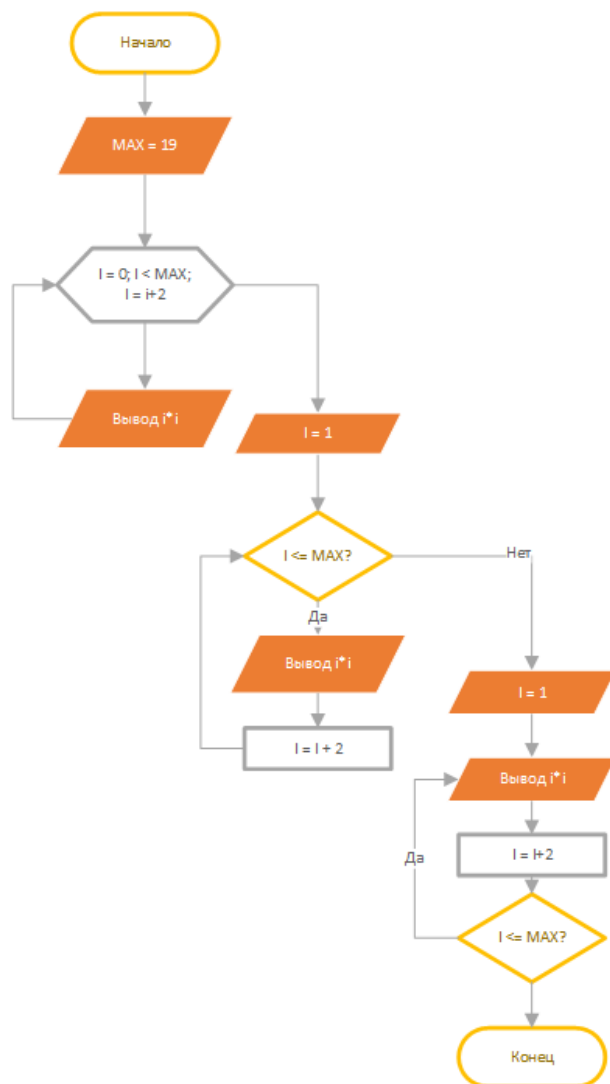
**Цель работы:** Изучение и понимание операторов цикла в языке C++. Операторы цикла позволяют выполнять повторяющиеся действия в программе, основываясь на определенных условиях. Они предоставляют эффективный способ автоматизировать выполнение однотипных задач и обработку данных.

## Задание 1. Операторы цикла. Вычисление значений функции

Написать 3 варианта программы с использованием операторов for, while, do.

Вывести на экран таблицу квадратов первых десяти целых нечётных положительных чисел.

**Блок-схема:**



### **Код программы:**

```
void Task1()
{
    cout << endl
        << endl
        << "Task 1" << endl;
    cout << "FOR" << endl;
    for (int i = 1; i <= 19; i += 2)
    {
        cout << i << " в квадрате: " << i * i << endl;
    }

    cout << "WHILE" << endl;
    int i = 1;
    while (i <= 19)
    {
        cout << i << " в квадрате: " << i * i << endl;
        i += 2;
    }

    cout << "DO WHILE" << endl;
    i = 1;
    do
    {
        cout << i << " в квадрате: " << i * i << endl;
        i += 2;
    } while (i <= 19);
}
```

### **Вывод программы:**

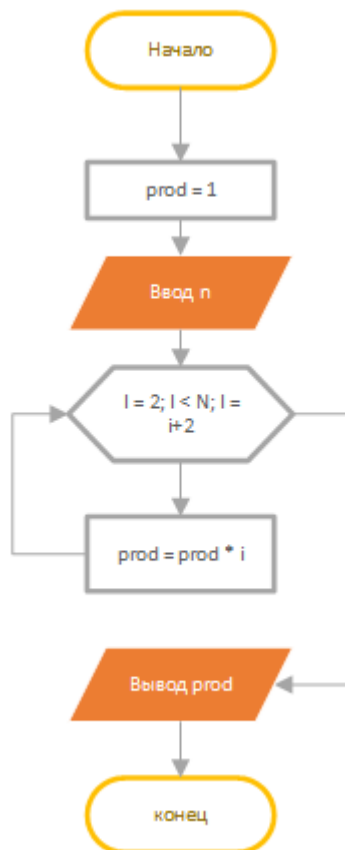
```
Run: CodeSolution x
Task 1
FOR
1 в квадрате: 1
3 в квадрате: 9
5 в квадрате: 25
7 в квадрате: 49
9 в квадрате: 81
11 в квадрате: 121
13 в квадрате: 169
15 в квадрате: 225
17 в квадрате: 289
19 в квадрате: 361
WHILE
1 в квадрате: 1
3 в квадрате: 9
5 в квадрате: 25
7 в квадрате: 49
9 в квадрате: 81
11 в квадрате: 121
13 в квадрате: 169
15 в квадрате: 225
17 в квадрате: 289
19 в квадрате: 361
DO WHILE
1 в квадрате: 1
3 в квадрате: 9
5 в квадрате: 25
7 в квадрате: 49
9 в квадрате: 81
11 в квадрате: 121
13 в квадрате: 169
15 в квадрате: 225
17 в квадрате: 289
19 в квадрате: 361
```

## **Задание 2. Операторы цикла. Последовательности натуральных чисел**

Разработать программу на языке C++ для следующих заданий:

Дано целое положительное число N. Вычислить произведение натуральных чётных чисел, не превышающих это число.

**Блок-схема:**



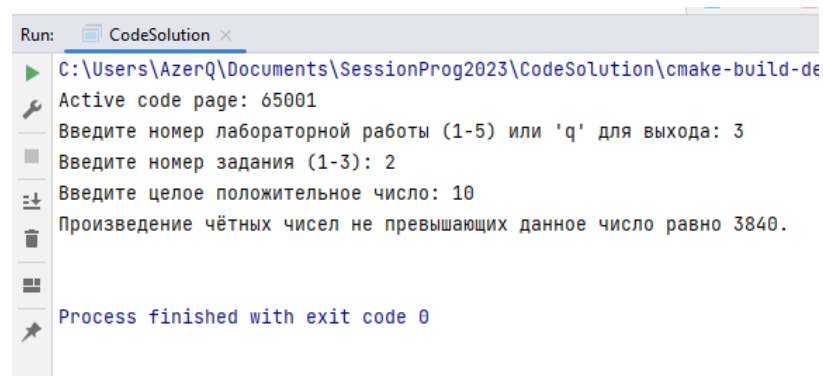
### Код программы:

```

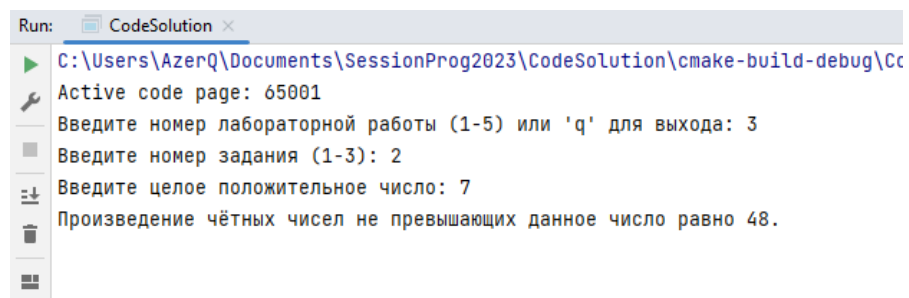
void Task2()
{
    int N;
    cout << "Введите целое положительное число: ";
    cin >> N;

    if(N < 2) {
        std::cout << "Произведение чётных чисел не превышающих
данное число равно 0.\n";
        return;
    }
    long long product = 1;
    for (int i = 2; i <= N; i += 2) {
        product *= i;
    }
    cout << "Произведение чётных чисел не превышающих данное
число равно " << product << ".\n";
}
  
```

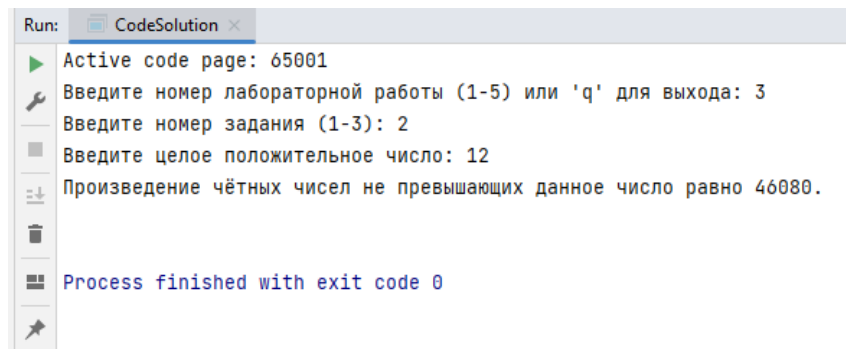
## Вывод программы:



```
Run: CodeSolution x
C:\Users\AzerQ\Documents\SessionProg2023\CodeSolution\cmake-build-de
Active code page: 65001
Введите номер лабораторной работы (1-5) или 'q' для выхода: 3
Введите номер задания (1-3): 2
Введите целое положительное число: 10
Произведение чётных чисел не превышающих данное число равно 3840.
Process finished with exit code 0
```



```
Run: CodeSolution x
C:\Users\AzerQ\Documents\SessionProg2023\CodeSolution\cmake-build-debug\Cc
Active code page: 65001
Введите номер лабораторной работы (1-5) или 'q' для выхода: 3
Введите номер задания (1-3): 2
Введите целое положительное число: 7
Произведение чётных чисел не превышающих данное число равно 48.
```



```
Run: CodeSolution x
Active code page: 65001
Введите номер лабораторной работы (1-5) или 'q' для выхода: 3
Введите номер задания (1-3): 2
Введите целое положительное число: 12
Произведение чётных чисел не превышающих данное число равно 46080.
Process finished with exit code 0
```

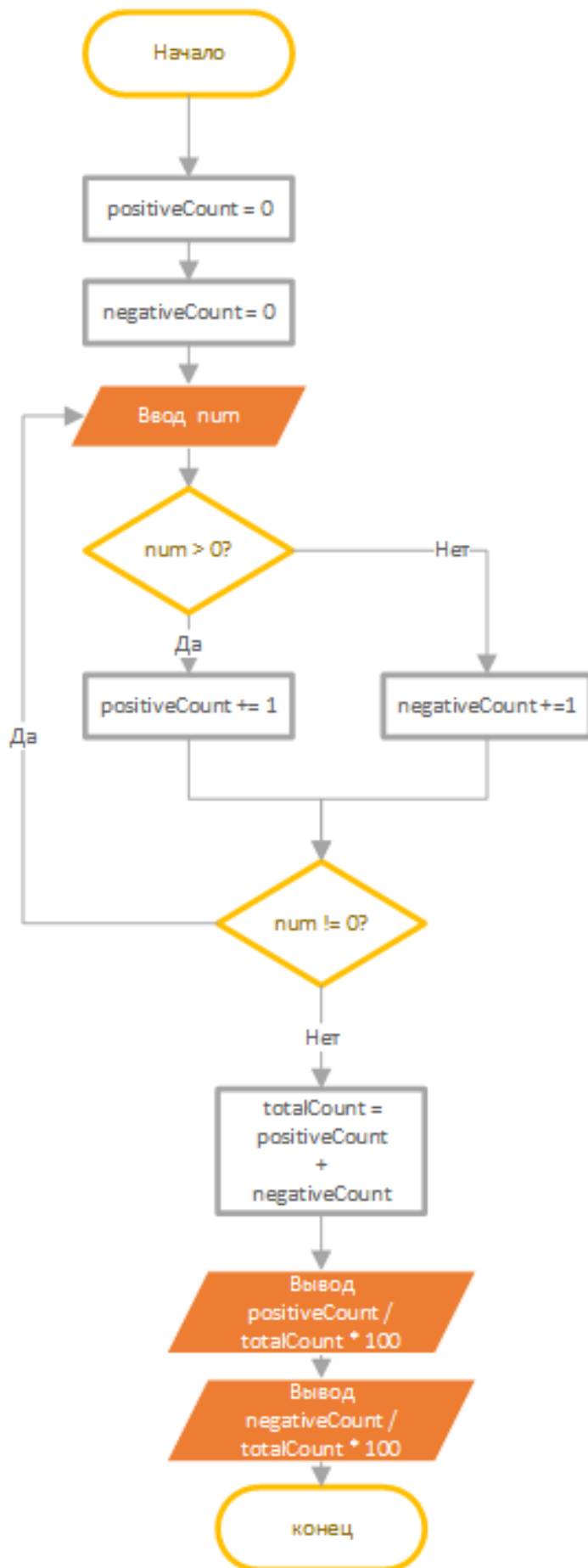
## Задание 3. Операторы цикла. Последовательности произвольных чисел

Разработать программу на языке C++ для следующих заданий:

Вводится последовательность ненулевых чисел, 0 — конец последовательности.

Подсчитать процент положительных и отрицательных чисел.

## Блок-схема:





### Код программы:

```
void Task3()
{
    cout << endl
        << endl
        << "Task 3" << endl;
    int positiveCount = 0, negativeCount = 0, num;
    do
    {
        cout << "Введите число (для завершения введите 0): ";
        cin >> num;
        if (num > 0)
        {
            positiveCount++;
        }
        else if (num < 0)
        {
            negativeCount++;
        }
    } while (num != 0);
    int totalCount = positiveCount + negativeCount;
    float positivePercentage = (float)positiveCount / totalCount * 100;
    float negativePercentage = (float)negativeCount / totalCount * 100;
    cout << "Процент положительных чисел: " << positivePercentage
        << "%" << endl;
    cout << "Процент отрицательных чисел: " << negativePercentage
        << "%" << endl;
}
```

## Вывод программы:

```
Run: CodeSolution x
Введите номер лабораторной работы (1-5) или 'q' для выхода: 3
Введите номер задания (1-3): 3

Task 3
Введите число (для завершения введите 0): 90
Введите число (для завершения введите 0): 1
Введите число (для завершения введите 0): -7
Введите число (для завершения введите 0): 23
Введите число (для завершения введите 0): 26
Введите число (для завершения введите 0): 76
Введите число (для завершения введите 0): 0
Процент положительных чисел: 83.3333%
Процент отрицательных чисел: 16.6667%
```

```
Run: CodeSolution x

Task 3
Введите число (для завершения введите 0): -8
Введите число (для завершения введите 0): -10
Введите число (для завершения введите 0): 23
Введите число (для завершения введите 0): 54
Введите число (для завершения введите 0): 12
Введите число (для завершения введите 0): -7
Введите число (для завершения введите 0): 12
Введите число (для завершения введите 0): 43
Введите число (для завершения введите 0): 0
Процент положительных чисел: 62.5%
Процент отрицательных чисел: 37.5%
```

## Вывод

После изучения операторов цикла языка C++, мы можем прийти к следующим выводам:

1. Циклы важны для программирования: Операторы цикла позволяют выполнять повторяющиеся действия в коде. Это делает возможным обработку больших наборов данных, создание повторяющихся действий и выполнение задач, которые требуют многократного повторения операций.
2. Различные типы циклов для различных ситуаций: C++ предлагает несколько типов циклов, каждый из которых подходит для разных ситуаций. Циклы `for` часто используются, когда известно, сколько раз цикл должен выполняться. Циклы `while` и `do-while` обычно используются, когда число итераций неизвестно заранее.
3. Циклы можно вкладывать: Один цикл может быть помещен внутри другого для создания более сложных структур повторения. Это часто используется при обработке многомерных структур данных, таких как массивы.
4. Управление циклами: С помощью операторов `break` и `continue` можно контролировать выполнение циклов. Оператор `break` позволяет немедленно прекратить выполнение цикла, а оператор `continue` пропускает оставшуюся часть текущей итерации и переходит к следующей.
5. Безопасность циклов: Необходимо аккуратно обращаться с условиями циклов, чтобы избежать бесконечных циклов, что может привести к зависанию программы.
6. Роль циклов в алгоритмах: Циклы играют ключевую роль во многих алгоритмах и структурах данных, таких как поиск, сортировка, итерация по элементам массива или списка и т.д.
7. Таким образом, операторы цикла в C++ являются мощным инструментом, позволяющим создавать эффективные и компактные программы. Их изучение и понимание важно для любого, кто хочет стать эффективным программистом на C++.