



**«Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)


О т ч е т

по лабораторной работе № 3

**Название лабораторной работы: Программирование циклического
процесса. Типы циклов.**

Дисциплина: Алгоритмизация и программирование

Студент гр. ИУ6-15Б


(Подпись, дата)

В.А. Бирюков
(И.О. Фамилия)

Преподаватель


(Подпись, дата)

А.А. Веселовский
(И.О. Фамилия)

Москва, 2024

Цель: изучить основные принципы работы с циклами в языке C++

Задание: В заданном целом числе определить количество нечетных цифр.

Рассмотреть решение предложенной задачи с использованием всех трех видов циклов. Отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

Напишем программу на языке C++:

```
1  #include <iostream>
2  #include <Windows.h>
3  #include <cmath>
4
5  int main() {
6      SetConsoleOutputCP(WC_UTF8); //для русского языка в терминале
7      int x, num, len = 0, count_while = 0, count_for = 0, count_do = 0, k_while = 0, k_do = 0, k_for = 0;
8      std::cout << "Введите число: " << std::endl;
9      std::cin >> num;
10
11     x = num;
12     while (fabs(x) > 0) {
13         if ((x%10)%2!=0)
14             count_while++;
15         x /= 10;
16         k_while++;
17     }
18
19     x = num;
20     while (fabs(x) > 0) {
21         x /= 10;
22         len++;
23         k_for++;
24     }
25     x = num;
26     for (int i = 0; i < len; i++) {
27         if ((x%10)%2 != 0)
28             count_for++;
29         x /= 10;
30         k_for++;
31     }
32
33     x = num;
34     do {
35         if ((x%10)%2!=0)
36             count_do++;
37         x /= 10;
38         k_do++;
39     } while (fabs(x)>0);
40
41     std::cout << "Количество нечётных цифр: " << count_while << std::endl;
42     std::cout << "Количество итераций через цикл while: " << k_while << "\n" << std::endl;
43     std::cout << "Количество нечётных цифр: " << count_for << std::endl;
44     std::cout << "Количество итераций через цикл for: " << k_for << "\n" << std::endl;
45     std::cout << "Количество нечётных цифр: " << count_do << std::endl;
46     std::cout << "Количество итераций через цикл do: " << k_do << "\n" << std::endl;
47     return 0;
48 }
```

Рисунок 1 – код программы

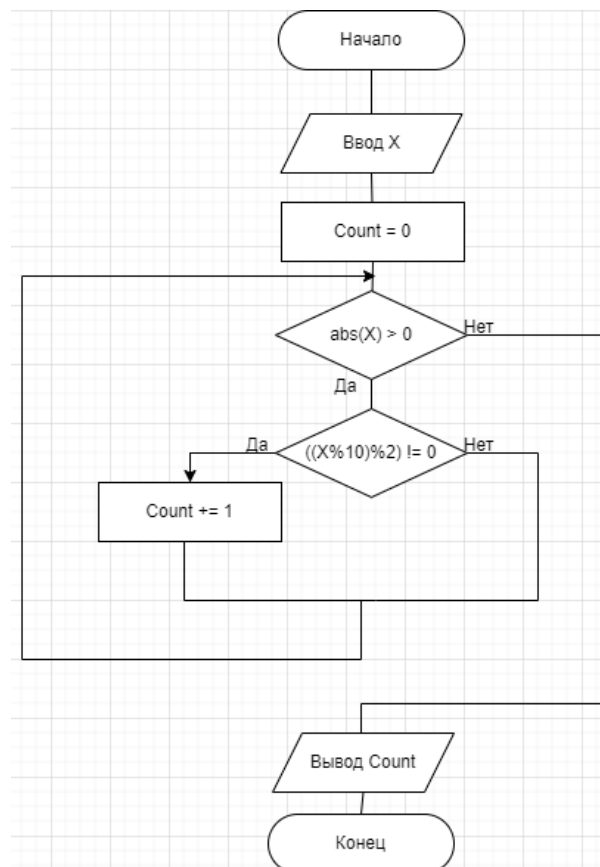


Рисунок 2 – схема алгоритмов для решения через цикл While

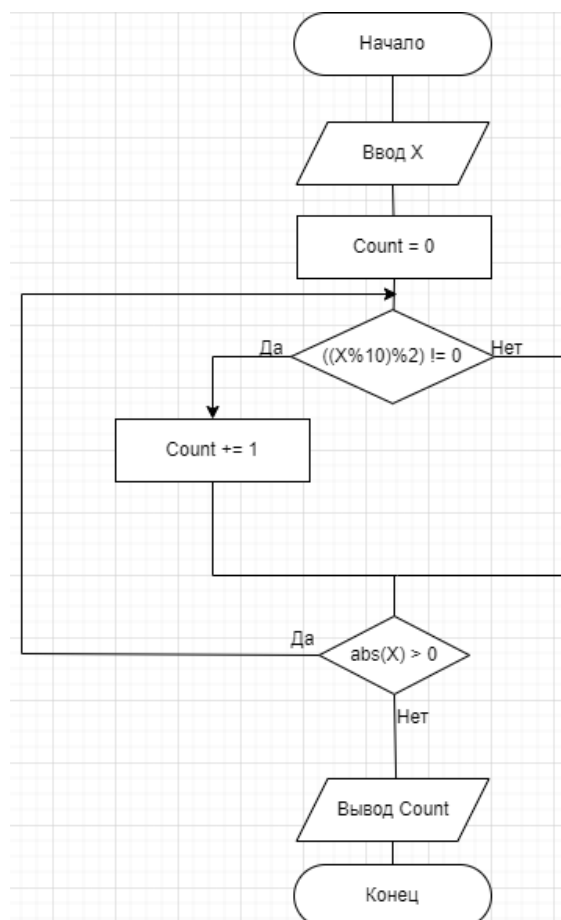


Рисунок 3 – схема алгоритмов для решения через цикл Do

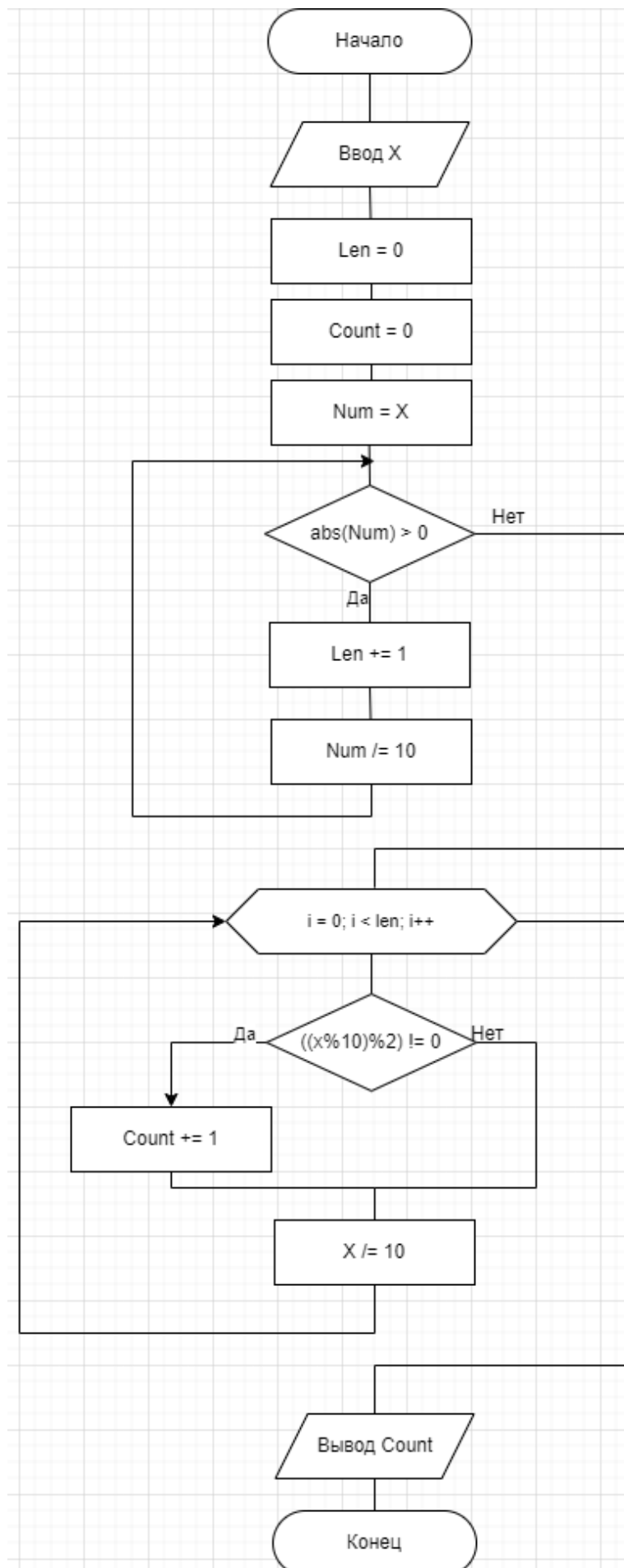


Рисунок 4 – схема алгоритмов для решения через цикл `For`

Запустим программу и посмотрим какой цикл проходит наименьшее количество итераций, чтобы определить самый эффективный цикл для этой задачи

```
Введите число:
-123567709
Количество нечётных цифр: 6
Количество итераций через цикл while: 9

Количество нечётных цифр: 6
Количество итераций через цикл for: 18

Количество нечётных цифр: 6
Количество итераций через цикл do: 9

Process finished with exit code 0
|
```

Рисунок 5 – результат работы программы

Заметим, что цикл for имеет в два раза больше итераций, так как для использования цикла for необходимо определить длину числа, используя вспомогательный цикл while

Цикл While и цикл Do всегда проходят одинаковое количество итераций, кроме случая, когда пользователь изначально вводит 0

```
"C:\Users\uprtrk\OneDrive\Рабочий стол\Лабы\АиП\1
Введите число:
0
Количество нечётных цифр: 0
Количество итераций через цикл while: 0

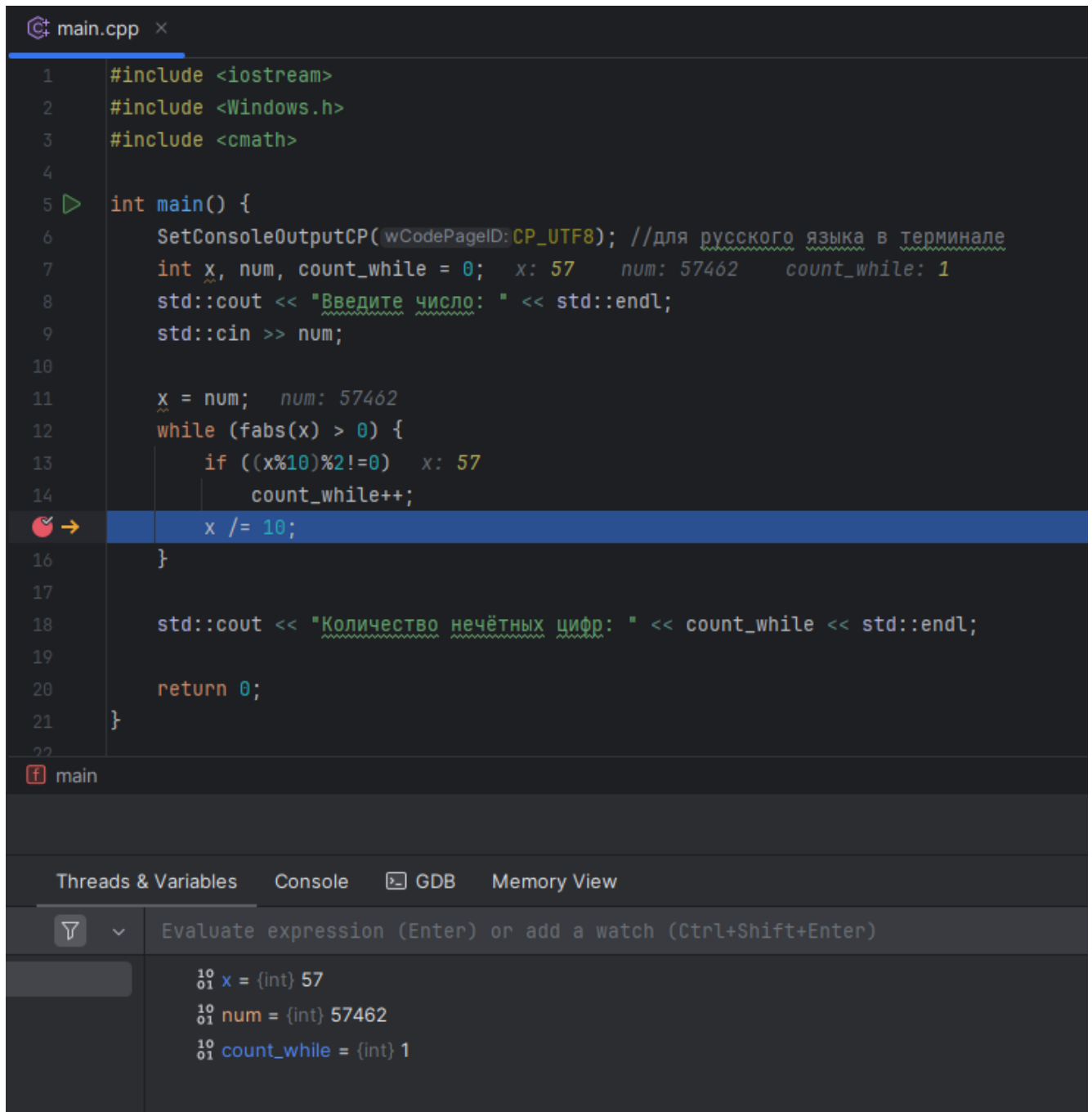
Количество нечётных цифр: 0
Количество итераций через цикл for: 0

Количество нечётных цифр: 0
Количество итераций через цикл do: 1

Process finished with exit code 0
|
```

Рисунок 6 – результат работы программы при вводе «0»

Исходя из результатов выше, можно сделать вывод, что использование в данной задаче цикла `while` – самое рациональное решение. Отладим программу с ЭТИМ ЦИКЛОМ



```
1  #include <iostream>
2  #include <Windows.h>
3  #include <cmath>
4
5  int main() {
6      SetConsoleOutputCP(WCodePageID: CP_UTF8); //для русского языка в терминале
7      int x, num, count_while = 0;  x: 57  num: 57462  count_while: 1
8      std::cout << "Введите число: " << std::endl;
9      std::cin >> num;
10
11     x = num;  num: 57462
12     while (fabs(x) > 0) {
13         if ((x%10)%2!=0)  x: 57
14             count_while++;
15         x /= 10;
16     }
17
18     std::cout << "Количество нечётных цифр: " << count_while << std::endl;
19
20     return 0;
21 }
```

main

Threads & Variables Console GDB Memory View

✓ Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)

10	x	=	{int} 57
10	num	=	{int} 57462
10	count_while	=	{int} 1

Рисунок 7 – отладка программы с циклом `while`

Вывод: в ходе лабораторной работы научился работать с циклами в языке C++