

РАЗРАБОТКА НА C++

Урок 4. Цикл “while”

«while» — повторяющийся «if»

Инструкции внутри if выполняются только один раз, и это происходит только если условие в if истинно (true). Если условие ложно (false), инструкции внутри if пропускаются.

Пример:

```
int x = 5;
if (x > 0) {
    std::cout << "Это выполнится 1 раз" << std::endl;
}
```

«while» — повторяющийся «if»

Цикл «while» выполняет инструкции внутри себя многократно до тех пор, пока условие, заданное после ключевого слова while, остается истинным, что делает его похожим на "повторяющийся if". Количество выполнений инструкций внутри цикла зависит от того, как долго условие остается истинным.

Пример:

```
int count = 5;
while (count > 0) {
    cout << "Это выполняется " << count << " раз." << endl;
    count--; // Уменьшаем счетчик на 1 при каждой итерации
}
```

В этом примере цикл while выполняет инструкции внутри него 5 раз, потому что count начинается с 5 и уменьшается на каждой итерации до 0.

Условие в цикле «while»

Цикл while в C++ принимает условие, которое должно быть либо истинным (true), либо ложным (false). Булевы значения (true или false) могут быть результатом выражений с логическими операторами: <, >, ==, !=.

Бесконечный цикл - это цикл, который выполняется бесконечно, так как условие всегда истинно. Цикл while может стать бесконечным, если вы зададите условие, которое всегда остается истинным.

Пример бесконечного цикла:

```
while (true) {  
    cout << "Это бесконечный цикл!" << endl;  
}  
// Код после бесконечного цикла  
cout << "Этот код никогда не выполнится." << endl;
```

В этом примере условие true всегда истинно, поэтому цикл будет выполняться бесконечно. Внимательно следите за условиями в циклах, в случае создания бесконечного цикла может произойти зависание программы.

Счетчик и условие в «while»

Цикл «while» часто используется с переменной-счетчиком и условием, зависящим от этой переменной. Счетчик обычно увеличивается или уменьшается на каждой итерации цикла, и выполнение цикла зависит от того, соответствует ли счетчик заданному условию.

Пример счетчика и условия в while:

```
int n = 5; // Выбранное число для таблицы умножения
```

```
int i = 1; // Счетчик итераций
```

```
cout << "Таблица умножения для " << n << ":" << endl;
```

```
while (i <= 9) {  
    int result = n * i;  
    cout << n << " * " << i << " = " << result << endl;  
    i++; // Увеличиваем счетчик перед следующей итерацией  
}
```

В этом примере программа выводит таблицу умножения для числа n, в данном случае, для n = 5. Цикл while выполняется 10 раз (от 1 до 9), и на каждой итерации счетчик i увеличивается на 1. Результат умножения выводится на экран.

Цикл «do-while»

Цикл «do-while» в C++ представляет собой конструкцию, которая позволяет выполнять набор инструкций как минимум один раз или более раз, если это соответствует условию. Основное отличие «do-while» от «while» состоит в том, что в «do-while» инструкции выполняются по крайней мере один раз, даже если условие ложное. Это делает «do-while» полезным, когда необходимо гарантировать выполнение определенных действий хотя бы один раз.

Синтаксис цикла «do-while»

Синтаксис:

```
do {  
    // Инструкции, которые нужно выполнить  
} while (условие);
```

- do: ключевое слово, обозначающее начало цикла do-while;
- { ... }: блок инструкций, который будет выполняться;
- while: ключевое слово, указывающее начало условия;
- (условие): условие, которое определяет, будет ли выполняться цикл.

Логика работы цикла «do-while»

Пример:

```
int count = 1;
```

```
do {  
    cout << "Этот цикл выполнится 1 раз." << endl;  
    count++; // Увеличиваем счетчик на 1 при каждой итерации  
} while (count == 0); // Это условие выдаст false, цикл прервется
```

Работа цикла «do-while»:

1. Сначала инструкции внутри блока «do» выполняются по крайней мере один раз, независимо от того, истинно ли начальное условие.
2. После выполнения блока инструкций проверяется условие, указанное после «while».
3. Если условие истинное, цикл «do-while» продолжает выполняться, и инструкции в блоке «do» выполняются снова.
4. Если условие ложное, выполнение цикла завершается, и программа продолжает выполнение инструкций, идущих после цикла.

Перевод операторов «while» и «do-while»

while (с англ. - "пока")

Цикл while - это конструкция, которая выполняет определенные инструкции, **пока** заданное условие истинно.

do-while (с англ. - "делай, пока")

Цикл do-while можно коротко описать словами «делай» и «пока». Благодаря команде «делай» (do) цикл безоговорочно выполняет инструкции, вложенные в себя, как минимум 1 раз. А далее вступает в действие принцип «пока» (while), при котором инструкции продолжают выполняться, **пока** заданное условие истинно.

Операторы управления циклом

Оператор «break» (с англ. - "прервать")

Оператор «break» используется для **прерывания** выполнения цикла. Когда «break» выполняется, программа выходит из ближайшего цикла и продолжает выполнять инструкции, стоящие после цикла.

Не используйте этот оператор, если без него можно обойтись!

Пример использования «break»:

```
int count = 1;

while (count <= 10) {
    if (count == 5) {
        cout << "Достигнуто значение 5, прерываем цикл." << endl;
        break; // Прерываем выполнение цикла
    }
    cout << "Значение count: " << count << endl;
    count++;
}
```

```
int count = 1;

while (count <= 5) {
    cout << "Значение count: " << count <<
endl;
    count++;
}
```

Операторы управления циклом

Оператор «continue» (с англ. - "продолжить")

Оператор «continue» используется для пропуска текущей итерации цикла и перехода к следующей. Он позволяет пропустить часть инструкций в текущей итерации и перейти к следующей итерации цикла.

Пример использования «continue»:

В США число 9 считается несчастливым. В многоквартирных домах в городах США нет 9-го этажа. Американские технологические компании, которые нумеруют свои продукты числами по порядку, пропускают цифру 9 при именовании продуктов в своем модельном ряду.

Давайте рассмотрим пример программы, которая выводит номера этажей 25-этажного дома, исключая 9-й этаж с помощью оператора «continue»:

```
int number = 1;

while (number <= 25) {
    if (count == 9) {
        cout << "Пропускаем значение 9 и переходим к следующей итерации." << endl;
        number++;
        continue; // Пропускаем остаток текущей итерации и переходим к следующей
    }
    cout << "Этаж № " << number << endl;
    count++;
}
```

Содержимое цикла «while»

Циклы могут содержать внутри себя различные инструкции. Даже такие сложные, как условные конструкции «if». Это позволяет выполнять определенные действия в зависимости от условий внутри цикла.

Пример использования «if» в цикле «while»:

```
int count = 1;

while (count <= 5) {
    if (count % 2 == 0) {
        cout << count << " - это четное число." << endl;
    } else {
        cout << count << " - это нечетное число." << endl;
    }
    count++;
}
```

Вычисления с большим кол-вом элементов в «while»

Циклы «while» в C++ удобно использовать для проведения математических операций над элементами. Для этого обычно используется переменная, которая аккумулирует значения элементов, удовлетворяющих условию.

Пример вычисления суммы четных чисел от 1 до 10 с использованием цикла «while»:

```
int count = 1;
int sum = 0;

while (count <= 10) {
    if (count % 2 == 0) {
        sum += count; // Добавляем четное число к сумме
    }
    count++;
}

cout << "Сумма четных чисел от 1 до 10: " << sum << endl;
```