

1. Циклы в C++

Циклы в C++ позволяют многократно повторять один и тот же фрагмент кода. Это нужно для выполнения итераций - повторяющихся шагов обработки данных.

Благодаря циклам можно, например, вывести элементы массива, подсчитать сумму чисел, получить данные из файла - выполнив при этом одни и те же действия для каждого элемента.

Основные виды циклов в C++:

- цикл `for` - используется, когда известно количество итераций
- цикл `while` - повторяется пока выполняется условие
- цикл `do-while` - гарантированно выполнится хотя бы 1 раз

Циклы могут быть вложенными, то есть содержать в теле другие циклы.

2. Цикл for

Цикл for используется, когда нужно выполнить код фиксированное число раз. Он имеет следующий синтаксис:

```
for (инициализация; условие; итерация) {  
    // код цикла  
}
```

Здесь:

- инициализация - код, который выполняется один раз перед стартом цикла
- условие - проверяется перед каждой итерацией
- итерация - код, который выполняется после каждой итерации

Пример цикла for для вывода чисел от 1 до 5:

```
for (int i = 1; i <= 5; i++) {  
    cout << i << endl;  
}
```

Этот цикл выведет числа 1 2 3 4 5, инкрементируя переменную i от 1 до 5.

3. Цикл `while`

Цикл `while` повторяется, пока заданное условие истинно. Синтаксис:

```
while (условие) {  
    // тело цикла  
}
```

В отличие от цикла `for`, здесь нет отдельных блоков инициализации и итерации.

Например, цикл для считывания данных пока не будет введен 0:

```
while(cin >> x && x != 0) {  
    // обработка данных  
}
```

4. Цикл **do-while**

Цикл **do-while** сначала выполняет тело, а затем проверяет условие:

```
do {  
  // тело цикла  
} while (условие);
```

Такой цикл гарантированно выполнится минимум один раз, вне зависимости от условия.

Такая структура часто используется, когда нужно выполнить цикл хотя бы один раз.

5. Пример программы

Цикл for:

```
// Цикл выводит числа от 1 до 5

for(int i = 1; i <= 5; i++) {

    // вывод числа i на экран
    cout << i << endl;

    // Увеличение счетчика i на 1
    i++;

}
```

Описание работы:

Здесь для цикла объявляется счетчик i , который инициализируется значением 1.

Проверяется условие $i \leq 5$ - пока это истина, цикл повторяется.

После каждой итерации i увеличивается на 1. Таким образом, цикл выведет числа от 1 до 5.

Цикл `while`:

```
// Цикл считывает числа пока не будет введен 0

int x;

while (cin >> x && x != 0) {

    // Обработка числа x

    cout << x << endl;

}
```

Описание работы:

Здесь из потока ввода считывается число в переменную `x`.

Проверяется условие входа и `x != 0`. Цикл работает, пока вводятся не нулевые числа.

После ввода 0, условие станет ложным и цикл завершится.

Цикл do-while:

```
// Цикл вывода чисел от 5 до 1  
  
int i = 5;  
  
do {  
  
    cout << i << endl;  
  
    i--; // уменьшение счетчика  
  
} while (i > 0);
```

Описание работы:

Сначала выводится число 5, затем i уменьшается на 1.

Проверяется условие $i > 0$, пока это истина - цикл повторяется.

Цикл гарантированно выполнится хотя бы 1 раз и выведет числа от 5 до 1.

6. Номера задач по теме

- Задача №333. Четные числа
- Задача №334. Остаток
- Задача №335. Квадраты
- Задача №339. Минимальный делитель
- Задача №340. Делители числа
- Задача №341. Количество делителей
- Задача №342. Сумма ста
- Задача №343. Сумма чисел
- Задача №345. Нули
- Задача №346. Подсчет чисел
- Задача №348. Уравнение по возрастанию
-
- Задача №113. Список квадратов
- Задача №3058. Минимальный делитель
- Задача №3059. Список степеней двойки
- Задача №3060. Точная степень двойки
- Задача №3061. Двоичный логарифм
- Задача №3062. Утренняя пробежка
- Задача №3063. Банковские проценты
- Задача №3074. Числа Фибоначчи
- Задача №3075. Номер числа Фибоначчи
- Задача №3076. Исполнитель Раздвоитель
- Задача №3082. Исполнитель Водолей

Сайт с задачами: <https://informatics.msk.ru>