



Циклы

Что такое цикл

- Часто в программе возникает необходимость многократно повторять какие-то действия. Т.е. совершать циклические действия.
- Для того, что бы не повторять код многократно, в языке Си имеются средства для организации циклов.

Что такое цикл

- **Цикл** — специальная конструкция языка программирования, с помощью которой те или иные действия выполняются **нужное количество раз**, в зависимости от некоего условия.

Что такое цикл

- В языке Си существует **три** основных способа реализации цикла:
- **for**;
- **while**;
- **do while**;

Что такое цикл

- Каждое повторение цикла называется:
- **ШАГ ЦИКЛА** или **ИТЕРАЦИЯ**

Цикл с предварительным условием



Цикл **WHILE ()** цикл с предусловием

- Общий синтаксис:

```
while(утверждение)
{
    действия для повторения;
    //это тело цикла.
}
```


Цикл **WHILE** ()

while(**утверждение**)

- Сначала проверяется **утверждение**, если оно **истинно** выполняются действия в **теле цикла**. По завершении всех действий тела, условие проверяется повторно.
- Действия в теле цикла выполняются до тех пор, пока проверяемое **утверждение ИСТИННО**.
- Как только **утверждение** станет ЛОЖНЫМ повторение завершится.

Особенность цикла **WHILE ()**

while(**утверждение**)

Если утверждение в скобках ложно, то цикл **может ни разу не выполниться.**

Цикл **WHILE** ()

```
// объявление управляющей переменной
int counter=0;
while(counter<7) // проверка утверждения
{
    counter++; // изменение управляющей переменной
               // действие для повторения
    cout<<"Вы видите "<<counter<< " чудо света!!!\n";
}
cout<<«\n";
```

Цикл **WHILE** ()

```
// объявление управляющей переменной
int counter;
while(counter<7) // проверка утверждения
{
    counter++; // изменение управляющей переменной
    // действие для повторения
    cout<<"Вы видите "<<counter<<" чудо света!!!\n";
}
cout<<«\n";
```

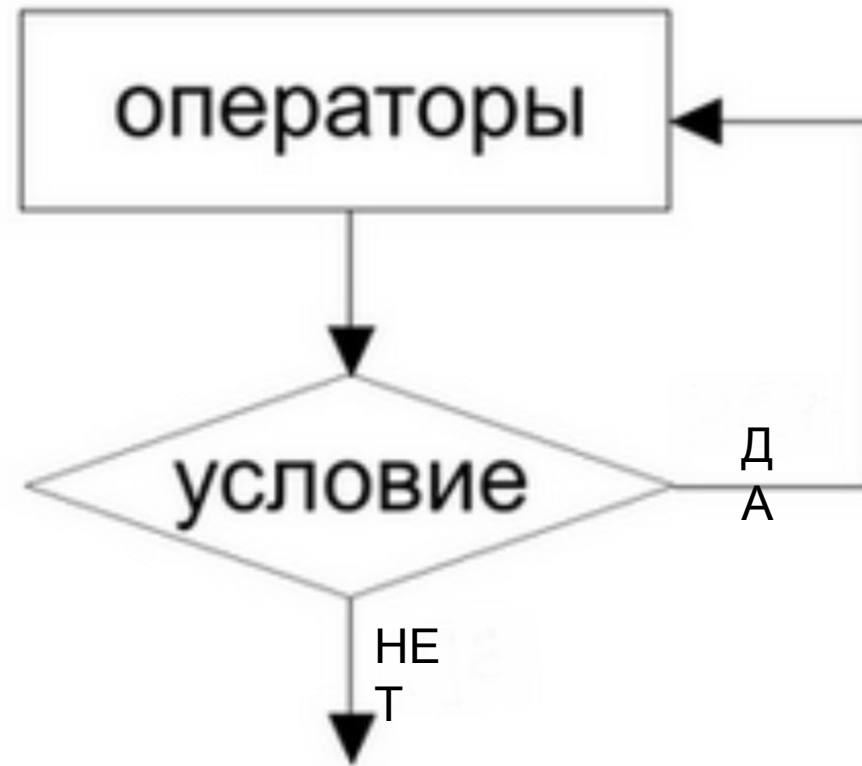
Цикл **WHILE** ()

```
int counter=0;
while(counter<7) // проверка утвердени
{
    // действие для повторения
    // вы увидели ... чудо света
    cout<<"Вы видите "<<++counter<<" чудо света!!!\n";
}
cout<<«\n";
```

Цикл **WHILE** ()

```
// объявление управляющей переменной
int counter=0;
while(counter<7) // проверка утверждения
{
    // действие для повторения
    // вы увидели ... чудо света
    cout<<"Вы видите "<<counter++<<" чудо света!!!\n";
}
cout<<"\n";
```

Цикл с последующим условием



ЦИКЛ **DO .. WHILE ()** ЦИКЛ С ПОСТУСЛОВИЕМ

do

{

 действие;

//это тело цикла.

}

while(условие);

Цикл DO .. WHILE ()

DO

{//это тело цикла. }

while(утверждение)

- Сначала выполняются действия в **теле цикла**. По завершении всех действий цикла, проверяется условие и если оно истинно, действия цикла выполняются повторно.
- Действия в теле цикла выполняются до тех пор, пока проверяемое **утверждение ИСТИННО**.
- Как только **утверждение** станет ЛОЖНЫМ повторение завершится.

Особенность цикла **DO .. WHILE ()**

DO

{//это тело цикла. }

while(утверждение)

Хотя бы **один раз действия этого цикла выполняться в любом случае**, даже если утверждение всегда ложно.

Цикл **DO .. WHILE ()**

Программа проверяющая ввод на соответствие каким-то условиям.

```
int userNumber;
do
{
    cout<<"Введите Ваше число, не меньше 10 ";
    cin >> userNumber;
}
while(userNumber <10) ;
cout<<"Ваше число, не меньше 10, принято!";
cout<<"\n";
}
```

Цикл **DO .. WHILE ()**

Программа проверяющая ввод на соответствие каким-то условиям.

// объявление управляющей переменной

Попробуйте написать предыдущую программу с помощью цикла

while(условие)

{

}

Цикл **DO .. WHILE ()**

Программа проверяющая ввод на соответствие каким-то условиям.

```
int userNumber;
cout<<"Введите Ваше число, не меньше 10 ";
cin >> userNumber;
while(counter<10)
{
    cout<<"Ваше число должно быть, не меньше 10 ";
    cin >> userNumber;
}

cout<<"Ваше число, не меньше 10, принято!";
cout<<"\n";
}
```

Сравнение циклов

WHILE ()

- Сначала **проверяется условие**, если оно верно выполняются действия в цикле.
- Действия могут ни разу не выполниться, если утверждение ложно.

DO .. WHILE ()

- Сначала **выполняются действия цикла**, затем проверяются условия.
- Хотя бы один раз действия выполняются при любом значении утверждения.



```
int const BEGIN=1; // начало диапазона суммируемых значений
int const END=5; // конец диапазона суммируемых значений
int sum=0; // переменная для накопления суммы
int i=BEGIN; // управляющая переменная цикла проверка условия
while(i <= END) // (сравнение управляющей переменной с окончанием
диапазона)
{
    sum += i; // накапливание суммы
    cout << "Итерация №" << i << " sum = " << sum << "\n";
    i++; // изменение управляющей переменной
}
cout << "Сумма всех чисел = " << sum << "\n\n"; // показ результата
```


Управление циклом

break – позволяет прервать цикл повторений независимо от условий цикла;

continue – позволяет прервать выполнение текущей итерации и начать следующую итерацию.

Управление циклом

```
int const BEGIN=1; // начало диапазона суммируемых значений
int const END=5; // конец диапазона суммируемых значений
int sum=0, test; // переменная для накопления суммы
int i=BEGIN; // управляющая переменная цикла проверка условия
while(i <= END) // (сравнение управляющей переменной с окончанием
диапазона)
{
    cout<< «Первая часть программы.» <<endl;
    i++; // изменение управляющей переменной
    cin>> test;
    if (test < END) break;
    cout<< "Вторая часть программы." <<endl;
    sum+=i; // накапливание суммы
    cout << i << endl;
}
cout<<" i = " << i << " Сумма = " << sum << "\n\n"; // показ результата
```

Управление циклом

```
int const BEGIN=1; // начало диапазона суммируемых значений
int const END=5; // конец диапазона суммируемых значений
int sum=0, test; // переменная для накопления суммы
int i=BEGIN; // управляющая переменная цикла проверка условия
while(i <= END) // (сравнение управляющей переменной с окончанием
диапазона)
{
    cout<< «Первая часть программы.» <<endl;
    i++; // изменение управляющей переменной
    cin>> test;
    if (test < END) continue;
    cout<< "Вторая часть программы." <<endl;
    sum+=i; // накапливание суммы
    cout << i << endl;
}
cout<<" i = " << i << " Сумма = " << sum << "\n\n"; // показ результата
```