

Алгоритмизация и программирование

Программирование на C/C++
(ч.4 – операторы циклов)

Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
<http://berkut.homelinux.com>

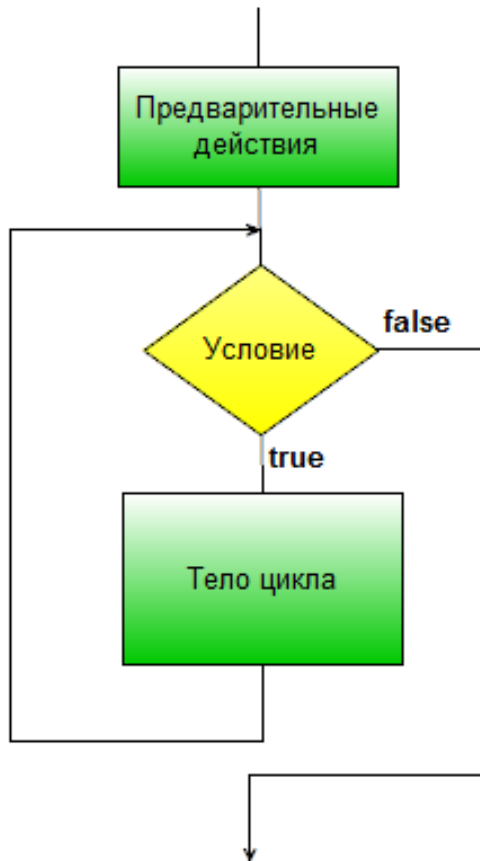
Циклы в C/C++

- Цикл — многократное прохождение по одному и тому же коду программы.
- Циклы необходимы программисту для многократного выполнения одного и того же кода, пока истинно какое-то условие.
- Если условие всегда истинно, то такой цикл называется бесконечным, у такого цикла нет точки выхода.

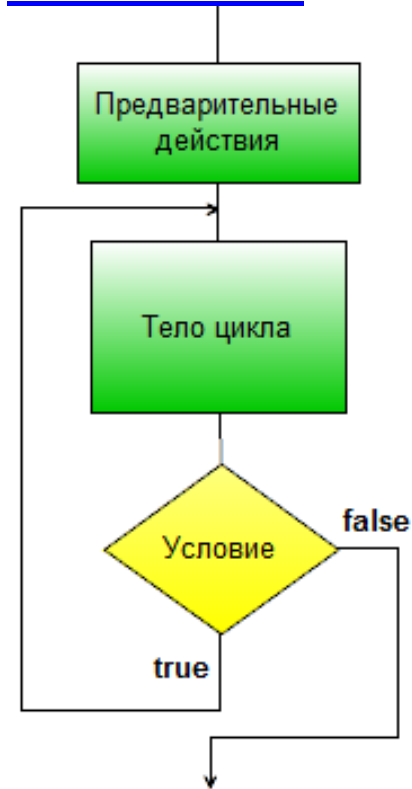
ЦИКЛЫ В С/С++

- Без блок-схем не обойтись 😊

while



do while



Циклы в C/C++

В C/C++ есть три вида циклов:

- `while` – цикл с предусловием
- `do ... while` – цикл с постусловием
- `for` – цикл с параметром



Цикл `while`

Оператор цикла `while` или цикл `while` - цикл, повторяющий одно и то же действие, пока условие продолжения цикла `while` остаётся ИСТИННЫМ.

```
// форма записи цикла while
while (/*условие продолжения цикла while*/)
{
    /*блок операторов*/;
    /*управление условием*/;
}
```

Цикл `while`

Пример использования `while`

```
// Пока скорость движения автомобиля меньше 60 км/ч,  
// продолжать наращивать скорость.  
int speed = 5; // начальная скорость автомобиля  
while ( speed < 60 ) // заголовок цикла while  
{  
    speed += 10; // тело цикла  
}
```



Демонстрація



Цикл **do ... while**

- Цикл **do while** отличается от цикла **while** тем, что в **do while** сначала выполняется тело цикла, а затем проверяется условие продолжения цикла.
- Цикл **do while** называют циклом с постусловием.
- Таким образом, если условие **do while** заведомо ложное, то хотя бы один раз блок операторов в теле цикла **do while** выполнится.

[Блок-схема](#)

Цикл do ... while

Оператор цикла `while` или цикл `while` - цикл, повторяющий одно и то же действие, пока условие продолжения цикла `while` остаётся **ИСТИННЫМ**.

```
// форма записи оператора цикла do while:  
do // начало цикла do while  
{  
    /*блок операторов*/;  
} while (/*условие выполнения цикла*/);  
// конец цикла do while
```

Цикл do ... while

```
#include <iostream>
#include <ctime>
using namespace std;

int main()
{
    srand(time(0));
    int balance = 8; // баланс
    do // начало цикла do while
    {
        cout << "balance = " << balance << endl;
        int removal = rand() % 3;
        cout << "removal = " << removal << endl;
        balance -= removal;
    }
    while ( balance > 0 ); // конец цикла do while
    return 0;
}
```



Демонстрація (Гра «Угадай число»)



Цикл for

// форма записи оператора цикла for:

```
for ( /*выражение1*/; /*выражение2*/; /*выражение3*/ )  
{  
    /*один оператор или блок операторов*/;  
}
```

Выражение 1 - объявление (и) или инициализация, ранее объявленной, переменной-счетчика, которая будет отвечать за истинность условия в цикле

```
int counter = 0;
```

```
counter = 9;
```

```
int counter;
```

Цикл for

// форма записи оператора цикла for:

```
for ( /*выражение1*/; /*выражение2*/; /*выражение3*/ )  
{  
    /*один оператор или блок операторов*/;  
}
```

Выражение 2 - это условие продолжения цикла for, оно проверяется на истинность.

```
counter < 10; // условие истинно пока count меньше 10
```

Цикл for

// форма записи оператора цикла for:

```
for ( /*выражение1*/; /*выражение2*/; /*выражение3*/ )  
{  
    /*один оператор или блок операторов*/;  
}
```

Выражение 3 изменяет значение переменной-счетчика. Без него цикл считается бесконечным, так как изменение содержимого переменной count выполняться не будет, и если изначально условие было истинным, то цикл будет бесконечным, иначе программа даже не войдет в цикл.

```
counter++ //
```

Цикл for

```
for ( int counter = 0; counter < 15; counter++)  
// выполняется приращение переменной counter  
// с шагом 1 от 0 до 15
```

Шаг в цикле for может быть отличным от единицы,
а точнее, любым целым(!) числом





Демонстрація





Спасибо!
Вопросы?



Алгоритмизация и программирование

Программирование на C/C++
(ч.4 – операторы циклов)

Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
<http://berkut.homelinux.com>