

C++ ЦИКЛИ

Андрій Янковський

2020

Цикли

З їх допомогою ми можемо виконати одну чи кілька команд **багато разів**.

Наприклад, давайте виведемо слово “Ya mozhu” п'ять разів.

```
for ( int i = 0; i < 5; i++ ){  
    cout << "Ya mozhu."  
}
```

Виведеться:

Ya mozhu.

Ya mozhu.

Ya mozhu.

Ya mozhu.

Ya mozhu.



Цей цикл називається **for** (фор)

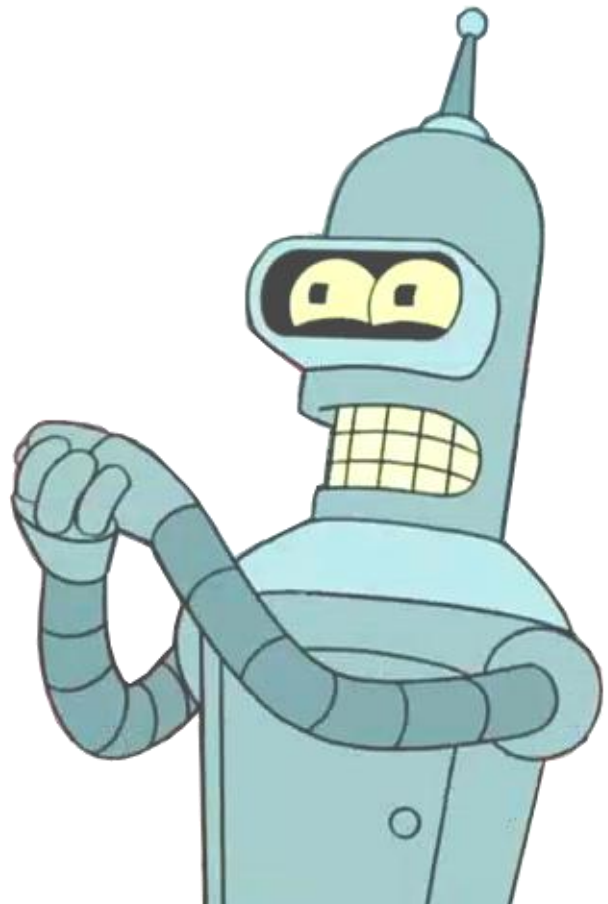
В середині циклу об'являється змінна **i**, яка буде доступна тільки всередині фігурних дужок **{}** циклу **for**

Цю змінну можна вивести на екран:

```
for ( int i = 0; i < 5; i++ ){  
    cout << i << endl;  
}
```

Виведеться:

0
1
2
3
4



Всередині циклу ми об'явили змінну `i` та присвоїли їй значення `0`.

Саме тому при **першому** запуску цикл вивів на екран значення `0`, потім спрацювала команда `i++` (збільшити змінну `i` на 1)

`i` на **другому** запуску цикл вже вивів `1`, на **третьому** - `2`, на **четвертому** - `3`, на **п'ятому** - `4`, а коли змінна `i` стала дорівнювати `5`, умова `i < 5` (`5 < 5`) перестала виконуватися і цикл закінчив роботу.

Далі будуть виконуватися команди, що стоять після циклу.

А є команда `i--` ?

Звичайно!

Вона зменшує змінну `i` на 1

А якщо ми хочемо вивести на екран такі цифри: 1, 3, 5, 7, 9

```
for ( int i = 1; i < 10; i = i + 2 ){  
    cout << i << ", ";  
}
```

Виведеться:

1, 3, 5, 7, 9,

Цикл **for** складається з трьох частин, які розділяються між собою знаком **;**

Початкове значення: $i = 1$

Умова: $i < 10$

доки умова виконується - цикл працює

Крок: як буде змінюватися змінна i при наступному запуску циклу

А є інші цикли, не `for`?

Звісно є!

```
int a = 0;
while ( a < 5 )
{
    a++;
    cout << a << " ";
}

```

1 2 3 4 5

Цикл `while` більш компактний, якщо у вас вже об'явлена змінна (над циклом), яка буде бігати в циклі, а ще цю змінну можна використовувати і після того, як цикл завершить роботу.

Ще є цикл `do-while`, який використовується дуже рідко, його особливість в тому, що він спочатку робить команди в фігурних дужках `{}`, а вже потім перевіряє умову

```
int a = 10;  
  
do  
{  
    a++;  
    cout << a;  
}  
  
while ( a < 5 );
```

10



Тобто, якщо вам треба виконати якість команди мінімум один раз, незалежно від значення змінної **a**, тоді цикл **do-while** - те що вам потрібно!

А бувають безкінечні цикли?

Так звичайно!

Якщо замість умови написати:

true - правда

```
while ( true )
```

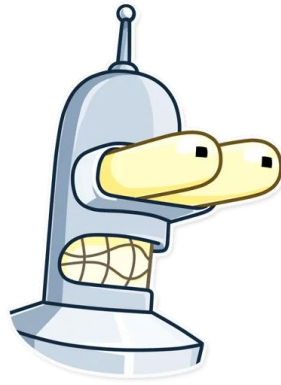
```
{
```

```
    cout << "0";
```

```
}
```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...



А як тоді вийти з безкінечного циклу?
Для цього є команда **break**;

```
int a = 0;
while ( true )
{
    cout << "0 ";
    a++;
    if ( a > 5 ){ break; }
}
```

0 0 0 0 0

А ми можемо в середині циклу сказати, а далі нічого не робити і перейти до наступного запуску.

Звісно! Для цього є команда `continue`

```
int a = 0;
while ( a < 7 )
{
    cout << "0";
    a++;
    if ( a > 3 ){ continue; }
    cout << "1 ";
}
```

01 01 01 0000

Бендер бажає тобі успіху!

