

Оператори за преход

Операторите за преход предават безусловно управлението в някаква точка на програмата. В C++ такива операторите са `break`, `continue` и `goto`. Следващите раздели разглеждат тези оператори.

1. Оператор `break`

а. Синтаксис

`break:`

б. Семантика

Прекъсва изпълнението на `while`, `for`, `do` и `switch`. Ако няколко от изброените по-горе оператори са вложени един в друг, ще се прекрати изпълнението на този, който съдържа `break` непосредствено в тялото си. Изпълнението продължава от първия оператор след `break`.

Пример:

```
#include <iostream.h>
#include <conio.h>

int main()
{
    int M;
    cout<<"M=";
    cin>>M;
    switch (M)
    {
        case 1:
        case 3:
            return 0;
        case 5:
        case 7:
        case 9: cout<<"числото е нечетно";break;
        case 2 :
        case 4:
        case 6:
        case 8: cout<<"числото е четно";break;
        default :cout<<"Не сте въвели едноцифрено
число!\n";
    }
    getch();
}
```

2. Оператор **continue**

а. Синтаксис

continue;

- б. Семантика - Оператор `continue` прекратява текущото изпълнение (итерация) на `while`, `for` или `do`. При влягане `continue` прекратява текущата итерация на оператора, в чието тяло непосредствено се намира. Ако `continue` е в тялото на `while` или `do`, изпълнението продължава с изчисление на израза-условие. Ако операторът е `for`, се изчислява `израз_2`. За разлика от `break`, който прекратява целия цикъл, `continue` прекратява само текущата итерация

Пример: Да се намери произведението на целите числа от -2 до 2

```
int i, P=1;
for(i=-2; i<=2; i++)
{
if (i==0) then continue
P=P*i;
}
```

Намира произведението на числата различни от 0. В противен случай текущата итерация завършва.

3. Оператор **goto**

Ако `break` и `continue` не съществуваха, щеше да има нужда от оператор, чрез който да се излиза от тялото на `for`, `while`, `do` и `switch`. Оператор `goto` реализира тази необходимост, но той се използва рядко в програмирането на C++. `goto` предава управлението в точка от програмата безусловно. Точката, в която се предава управлението, се означава с етикет. Този етикет и `goto` трябва да са в една и съща функция.

а. Синтаксис:

```
goto етикет;
```

Като етикет може да използва произволен идентификатор. Етикетът трябва да се среща само веднъж в съответната функция и да завършва с ":". Той не може да стои непосредствено пред дясна затваряща скоба "}". Това ограничение се преодолява, като след етикета се постави празен оператор. Например:

```
end; // празен оператор
}
```