

Instrukcje pętli. Typ tablicowy

Instrukcje pętli omawiane tutaj dotyczą sytuacji, gdy te same operacje mają być wykonane wielokrotnie. Niezbędnym elementem instrukcji pętli jest podanie warunku:

- albo warunku powtarzania pętli
- albo liczby powtórzeń pętli

Pętla z warunkiem powtarzania - "do while"

```
1 do {instrukcja} while (wyrażenie)
```

Instrukcja jest zawsze wykonywana co najmniej raz, a następnie za każdym razem, gdy wartość wyrażenia jest różna od 0. Np, wypisanie wielokrotności liczby "k" mniejszych niż wartość zmiennej *ograniczenie*:

```
1 do {  
2     cout << k << "\n" ;  
3     k = k*3 ;  
4 } while (k < ograniczenie) ;
```

Niezależnie od wartości zmiennej *ograniczenie* pętla zostanie wykonana co najmniej raz.

Pętla z warunkiem powtarzania - "while"

```
1 while (wyrażenie) {instrukcja}
```

Wykonanie "dopóki" polega na wykonywaniu instrukcji za każdym razem, gdy wartość wyrażenia jest różna od 0 (czyli ma wartość logiczną PRAWDA). Instrukcja nie jest wykonywana ani razu, gdy na samym początku wartością wyrażenia jest 0 (FAŁSZ).

```
1 while (k < ograniczenie)  
2 {  
3     cout << k << "\n" ;  
4     k = k*3 ;  
5 }
```

Jeśli początkowa wartość zmiennej *k* nie będzie mniejsza od wartości zmiennej *ograniczenie*, to pętla nie zostanie wykonana ani razu.

Pętla z liczbą powtórzeń - "for"

```
1 for (instr_ini; wyraz_warun ; instr_krok) treść_pętli
```

instr_ini - instrukcja wykonywana tuż przed rozpoczęciem pętli

wyraz_warun - wyrażenie obliczane przed każdym obiegiem pętli. Pętla jest wykonywana, gdy jest różne od zera.

instr_krok - instrukcja wykonywana na zakończenie każdego obiegu pętli

Wystąpienie poszczególnych elementów nie jest obowiązkowe. === Przykłady pętli "for" === Wypisanie 10 pierwszych liczb naturalnych:

```
1 for(i=0 ; i < 10 ; i=i+1)
2 {
3     cout << i
4 }
```

Pętla nieskończona:

```
1 for( ; ; )
2 {
3 }
```

Ingerowanie w pętle - Instrukcje break i continue

Instrukcja **break**, wykonywana w pętlach **while**, **for**, **do/while**, (oraz instrukcji **switch**) powoduje bezpośrednie wyjście.

```
1 int a;
2 for(a=1; a<=20; a++)
3 {
4     if(a==10) break;
5     cout << a << " ";
6 }
```

Gdy wartość zmiennej **a** osiągnie wartość **10**, pętla for zostanie przerwana.

Wyrażenie **continue**, kiedy jest wykonywane w pętlach **while**, **for**, **do/while**, (oraz instrukcji **switch**) pomija pozostałe instrukcje i przechodzi do następnego powtórzenia pętli.

```
1 int a;
2 for(a=1; a<=20; a++)
3 {
4     if(a==10) continue;
5     cout << "a=" <<a << "\t";
6 }
7 Zostaną wypisane wszystkie liczby od '1' do '20' za
wyjątkiem '10'.
```

Definiowanie tablicy

Przykłady

```
1 int a[20] ;           ///'a' jest tablicą 20 liczb całkowitych
2 char zdanie[80] ;     ///'zdanie' jest tablicą 80 elementów
   typu 'char'
3 int dd [4] [2] ;      ///'d' jest tablicą dwuwymiarową. Można
   to interpretować tak: 'dd' jest tablicą 4 elementów, z których
   każdy jest tablicą dwuelementową */
```

Inicjalizowanie tablicy

W momencie zdefiniowania tablicy można od razu nadać początkowe wartości jej elementom:

```
1 int t[4] = { 17, 5, 4, 200 } ;           ///'t' jest
   tablicą 4 elementową
```

Odwoływanie się do elementów tablicy

Pierwszy element tablicy ma indeks zerowy.

Wypisanie pierwszego elementu tablicy *t*:

```
1 cout << t[0] ;
```

Zastosowanie pętli dla wczytania wartości tablicy:

```
1 int t[4] ;
2 int i ;
3 for ( i = 0 ; i < 4 ; i++)
4     cin >> t[i] ;
```

Wniosek 1: Elementy tablicy należy wczytywać i zapisywać element po elemencie. Wniosek 2: Jeśli tablica ma rozmiar n , to nie istnieje element o indeksie n - największy dostępny indeks to $n - 1$