KURS C++

Wprowadzenie do pętli w C++

Przygotował: Kamil Feliszewski

Spis treści

1 Co to są pętle?

W językach programowania stosuje się instrukcje, które mają za zadanie powtarzać określoną czynność. W języku C++ może to być wyświetlanie jakiś elementów, np. liczby, znaki, obiekty itd. Aby pętle (instrukcje) wykonywały odpowiednią sekwencję powtarzania bądź wyświetlania muszą być spełnione określone warunki. W języku C++ stosuje się najczęściej instrukcje `for, while, do ... while`.

1.1 Pętla for

Składnia pętli `for` wygląda następująco:

```
for (instrukcja1; warunek; instrukcja2) {
  //instrukcje;
}
```

Na początku jest wykonywana instrukcja1. Następnie sprawdzany jest warunek. Jeżeli jest spełniony to wykonywane są instrukcje zawarte w bloku, a jeśli nie to pętla kończy swoje działanie. Po wykonaniu tych instrukcji (jeśli pętla się nie zakończyła) wykonywana jest instrukcja2. Po której ponownie jest sprawdzany warunek.

Interpretacja pętli for:
instrukcja1 wykonywana jest zawsze tylko raz na samym początku. Następnie sprawdzamy jest warunek i wykonywane są instrukcje zawarte w bloku pętli (warunek musi być spełniony). Instrukcja2 wykonywana jest zawsze po wykonaniu się instrukcji z bloku ale przed ponownym sprawdzeniem warunku. Wykonuje się zatem tyle razy ile wykonane zostały instrukcje z bloku pętli chyba, że pętla została przerwana wewnątrz bloku (break, goto, return). Jeśli wewnątrz bloku wystąpi instrukcja continue to instrukcje w tym bloku występujące dalej zostaną pominięte i wykonana zostanie instrukcja2, po której sprawdzony zostanie ponownie warunek.

1.2 Pętla while

Składnia pętli `while` wygląda następująco:

```
while (warunek) {
  //instrukcje;
}
```

Po słowie kluczowym `while` w nawiasach podajemy warunek, a następnie instrukcje zawarte w bloku. Na początku sprawdzany jest warunek jeżeli jest on spełniony to wykonywane są instrukcje zawarte w bloku. Następnie ponownie jeśli warunek jest spełniony wykonywane są te same instrukcje z bloku pętli. Warunek nie jest spełniony wtedy i tylko wtedy, gdy jego wartość logiczna wynosi FALSE, co liczbowo odpowiada 0 (zeru), w przeciwnym wypadku warunek jest spełniony. Pętla kończy się jeżeli warunek nie zostanie spełniony (możliwe jest to, że instrukcje nie zostaną wykonane ani razu, gdyż warunek nie został spełniony już przy pierwszym sprawdzaniu).

1.3 Pętla do ... while

Składnia pętli `do ... while` wygląda następująco:

```
do {
  //instrukcje;
} while (warunek);
```

Po nawiasach, w których jest zawarty warunek koniecznie musimy postawić średnik ";".
Pętla `do ... while` działa niemal tak samo jak pętla `while` z ta różnicą, że instrukcje zawarte w bloku tej pętli są wykonywane zawsze minimum jeden raz i dopiero po ich wykonaniu sprawdzany jest warunek. Zależnie od jego spełnienia pętla ponownie wykonuje dane instrukcje bądź jest przerywana.

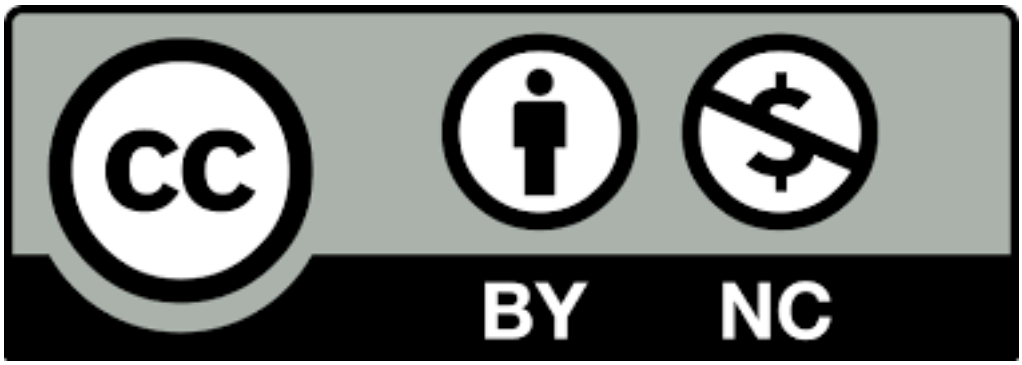
1.4 Sposoby przerywania pętli

Jeżeli wewnątrz bloku pętli występuje słowo kluczowe `break` to kolejne instrukcje zawarte w tym bloku nie zostaną już wykonane i pętla zakończy swoje działanie.
Instrukcja `goto` zawarta w bloku pętli może spowodować skok do innej części programu znajdującego się poza blokiem pętli, tym samym powodując jej przerwanie.
Wewnątrz bloku pętli może znajdować się także instrukcja `continue`, która co prawda nie powoduje natychmiastowego przerwania pętli ale sprawia, że kolejne instrukcje zawarte w bloku pętli nie zostają już wykonane i nastąpi ponowne sprawdzenie warunku pętli.
Słowo kluczowe `return` przerywa wykonywanie pętli.

1.5 Pętle zagnieżdżone

Podobnie jak do instrukcji warunkowych w pętlach też występuje zagnieżdżenie, czyli wywołanie jednej pętli wewnątrz drugiej. Przykład:

```
#include<iostream>
using namespace std;
int main() {
  for(int i=1; i<=10; i++) {
    for(int j=1; j<=10; j++) {
      cout<<i+j<<" ";
    }
    cout<<"\n"
  }
  return 0;
}
```


BY
NC