

Sam dla siebie C++
każdy programować może

sobota, dnia 13.09. 2025 r.

Pętle w C++ nr 006/25 *część - 1*

Pętla for

Pętla w języku C++ , jak mi wiadomo jest fragmentem programu mianowicie instrukcją lub blokiem instrukcji, umożliwiającym powtarzania się tego samego fragmentu kody tak długo, dopóki spełniony jest określony warunek. Pętla w języku C++ , z własnej praktyki hobbistycznej - realizuje automatyczne powtarzanie się funkcji, zadań, wyświetlania wiele razy informacji, lub są zdolne by modernizować, usprawniać, przetwarzać, modyfikować, zarządzać dane.

W języku C++, jak spostrzegłem, jest trzy podstawowe główne pętle, do których należą: **for**, **while** i **do-while**.

Przykładami mogą być np., gry, w których są powtarzane wielokrotnie czynności.

Pętle są kluczowym elementem programowania w C++ i pozwalają na tworzenie bardziej złożonych algorytmów.

Pętla for

- Służy do wykonania kodu określoną liczbą ilość razy.

Składnia pętli for:

```
for ( int i = 0; i < 10; i++ ) {
    ..... // kod do wykonania 10 razy
}
```

Prostym językiem mówiąc, powyższa składnia ma następujący wydźwięk:

```
dla ( stanPoczątkowy; warunekOkreślony, zmiany) {
    lista instrukcji do wykonania
}
```

lub też można by powyższą składnię przedłożyć na język ludzki w następującej formie:

```
for ( inicjalizacja; warunek; inkrementacja/dekrementacja ) {
    ..... // kod do wykonania
}
```

- inicjalizacja – ustawia wartość początkową zmiennej sterującej,

np.:

```
int i = 0
```

- warunek – wyrażenie logiczne, które jest sprawdzane przed każdą iteracją, pętla działa dopóki warunek jest prawdziwy,

np.:

```
i < 10
```

- inkrementacja – instrukcja wykonywana po każdej iteracji,

np.:

```
i++ lub ++i
```

- dekrementacja – instrukcja wykonywana po każdej iteracji,

np.:

```
i-- lub --i
```

Przykłady pętli for

Pętla for wypisuje liczbę 10 razy, od 0 do 9.

```
/* pętla for1 – przykład pierwszy */  
  
#include <iostream>  
#include <conio.h>  
  
using namespace std;  
  
int i, a;  
  
cout << " Funkcja oblicza wartosc a dla rownania a = 3*i zmieniającego się od 1 do 10." << endl;  
  
for ( i = 0; i < 10; i++ ) {  
  
    a = 3 * i;  
  
    cout << " i = " << i << "\t" << " a = " << a << endl;  
  
    getch ();  
    return 0;  
}
```

Oto wynik działania programu:

```
Funkcja oblicza wartosc a dla rownania a = 3*i zmieniajacego sie od 0 do 10.
i = 0   a = 0
i = 1   a = 3
i = 2   a = 6
i = 3   a = 9
i = 4   a = 12
i = 5   a = 15
i = 6   a = 18
i = 7   a = 21
i = 8   a = 24
i = 9   a = 27
```

Następny przykład:

```
/* petla for2 – przykład drugi */

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

using namespace std;

double i, a;

cout << " Funkcja oblicza wartosc a dla rownania a = 3*i zmieniajacego sie od 1 do 10." << endl;

for ( i = 0; i < 10; i-- ) {

    a = 3 * i;

    cout << " i = " << i << '\t' << " a = " << a << endl;

    getch ();
    return 0;
}
```

Wyniku działania tego programu nie uwiecznię pod treścią programu z powodu takiego, że przedział liczbowy $i < 10$ jest przedziałem ograniczony z prawej strony liczbą 10 a z lewej strony dąży do nieskończoności - do nieskończoności mniejszej od liczby 10. Program liczy i liczy i nowe wyniki wypisuje na monitorze. Będzie wypisywać tak długo dopóki wynik będzie się mieścić w zmiennej double. Jeśli czytasz lub przeglądasz moje zaangażowanie poszerzaniem umiejętności hobbistycznej w C++ , śmiało możesz ten program skopiować na swój komputer. Zatrzymać program można najechemieniem ikoną myszki na czarne pole wyświetlanego wyniku programu i kliknięciem lewym klawiszem myszki. Program się zatrzyma. Ten program związany z dekrementacją ($i--$) zrobiłem sobie w celu testowania zmiennej. Polecam.

Następny program. Tylko że tym razem zmieniający wartość liczby a od 10 do 1.

```
/* petla for3 - przykład trzeci */  
  
#include <iostream>  
#include <conio.h>  
  
using namespace std;  
  
int i, a;  
  
cout << " Funkcja oblicza wartosc a dla rownania a = 3 * i zmieniającego się od 1 do 10." << endl;  
  
for ( i = 10; i > 0; i-- ) {  
  
    a = 3 * i;  
  
    cout << "i=" << i << '\t' << "a=" << a << endl;  
  
    getch ();  
    return 0;  
}
```

I mam wynik działania programu, Wynik jest odwrotny do wyniku pierwszego programu.

```
Funkcja oblicza wartosc a dla rownania a = 3 * i zmieniającego się od 10 do 1.  
i = 10    a = 30  
i = 9     a = 27  
i = 8     a = 24  
i = 7     a = 21  
i = 6     a = 18  
i = 5     a = 15  
i = 4     a = 12  
i = 3     a = 9  
i = 2     a = 6  
i = 1     a = 3
```

Następny przykład programu for. Tym razem ze zmienną double dla równania $a = 3 / i$.

```
/* petla for4 - przykład czwarty */  
  
#include <iostream>  
#include <conio.h>  
  
using namespace std;  
  
double i, a;  
  
cout << " Funkcja oblicza wartosc a dla rownania a = 3 / i zmieniającego się od 10 do 1." << endl;  
  
for ( i = 10; i > 0; i-- ) {  
    a = 3 / i;  
  
    cout << "i=" << i << '\t' << "a=" << a << endl;  
  
    getch ();  
    return 0;  
}
```

Oto wynik

```
Funkcja oblicza wartosc a dla rownania a = 3 / i zmieniającego się od 10 do 1.  
i = 10    a = 0.3  
i = 9     a = 0.333333  
i = 8     a = 0.375  
i = 7     a = 0.428571  
i = 6     a = 0.5  
i = 5     a = 0.6  
i = 4     a = 0.75  
i = 3     a = 1  
i = 2     a = 1.5  
i = 1     a = 3
```

Następny przykład programu for. Tym razem ze zmienną float dla równania $a = 3 / i$.

```
/* petla for5 - przykład piąty */  
  
#include <iostream>  
#include <conio.h>  
  
using namespace std;  
  
float i, a;  
  
cout << " Funkcja oblicza wartosc a dla rownania a = 3 / i zmieniającego się od 10 do 1." << endl;  
  
for ( i = 10; i > 0; i-- ) {  
  
    a = 3 / i;  
  
    cout << " i = " << i << '\t' << " a = " << a << endl;  
  
    getch ();  
    return 0;  
}
```

Wynik taki sam jak w przykładzie czwartym.

```
Funkcja oblicza wartosc a dla rownania a = 3 / i zmieniającego się od 10 do 1.  
i = 10    a = 0.3  
i = 9     a = 0.333333  
i = 8     a = 0.375  
i = 7     a = 0.428571  
i = 6     a = 0.5  
i = 5     a = 0.6  
i = 4     a = 0.75  
i = 3     a = 1  
i = 2     a = 1.5  
i = 1     a = 3
```

A następny przykład programu for - tym razem ze zmienną int dla równania $a = 3 / i$.

```

/* petla for6 - przykład szusty */

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

using namespace std;

int i, a;

cout << " Funkcja oblicza wartosc a dla rownania a = 3 / i zmieniającego się od 10 do 1." << endl;

for ( i = 10; i > 0; i-- ) {

    a = 3 / i;

    cout << " i = " << i << '\t' << " a = " << a << endl;

    getch ();
    return 0;
}

```

Hmmm . . . I wynik mam ciekawy.

Funkcja oblicza wartosc a dla rownania $a = 3 / i$ zmieniającego się od 10 do 1.

```

i = 10    a = 0
i = 9     a = 0
i = 8     a = 0
i = 7     a = 0
i = 6     a = 0
i = 5     a = 0
i = 4     a = 0
i = 3     a = 1
i = 2     a = 1
i = 1     a = 3

```

Jak widać w powyższych przykładach, wyniki działania programu są uzależnione od zmiennej użytej do utworzenia programu.

Krótką uwaga.:

Niniejsze opracowanie, jak i następne opracowania celem nauki własnej którą kontynuuję, opieram na lekturach takich jak:

1. Język C++ Szkoła programowania Stephen Prata
2. Programowanie w języku C++ Wiesław Porębski

3. Podstawy języka C++ Stanley B. Lippman
4. C++ Zadania z programowania z przykładami rozwiązaniami – Mirosław J Kubiak *WYDANIE III*
5. Symfonia C++ standard Jerzy Grębosz
6. PRZEWODNIK DLA POCZĄTKUJĄCYCH C++ ALEX ALLAIN
7. i internet