

Sam dla siebie C++
każdy programować może

piątek, dnia 05.09. 2025 r.

Operatory matematyczne w C++ nr 004/25

Dodawanie, odejmowanie, mnożenie, dzielenie i dzielenie modulo. Typy zmiennej.

Przystępując do zagłębiania się w programowaniu w języku C++, zauważyłem, że bez operatorów arytmetycznych, operatorów przypisania, operatorów relacji i operatorów logicznych, nie ma mowy o nabywaniu umiejętności programowania w/w języku C++. Bowiem projektując program, projektuje go do wykonania zadania które chcę aby zostało wykonane. Czy to zadanie matematyczne, czy gra, czy instrument który chcę sobie użyć, bądź robot, czy w obsłudze maszyny. Do tego właśnie są programy.

Zapewne się dziwisz czytając mojego bloga. Ale prawda jest taka. Zaczynj programować a się przekonasz jak programowanie zmienia usposobienie do życia. Pół żartem pół serio, taka jest prawda.

Dlatego też, swoja wiedzę zanurzam w operatorach arytmetycznych powyżej wskazanych, jak i w operacjach matematycznych.

Do tych **operatorów arytmetycznych** należą:

- + dodawanie
- odejmowanie
- * mnożenie
- / dzielenie
- % dzielenie modulo (reszta)

Poniżej przedstawiam:

symbol operatora	funkcja	zastosowanie
+	dodawanie	wyrażenie + wyrażenie
-	odejmowanie	wyrażenie - wyrażenie
*	mnożenie	wyrażenie * wyrażenie
/	dzielenie	wyrażenie / wyrażenie
%	operator reszty z dzielenia	wyrażenie % wyrażenie

Do wykonania powyższych zadań, potrzebne są operatory relacji.

Do **operatorów relacji** należą:

- < mniejszy

- <= mniejszy lub równy
- > większy
- >= większy lub równy
- == równy
- != nie równy

Oto zestawienie operatorów relacji:

symbol operatora	funkcja	zastosowanie
<	mniejszy	wyrażenie < wyrażenie
<=	mniejszy lub równy	wyrażenie <= wyrażenie
>	większy	wyrażenie > wyrażenie
>=	większy lub równy	wyrażenie >= wyrażenie
==	równy	wyrażenie == wyrażenie
!=	nierówny	wyrażenie != wyrażenie

Niemniej, bez operatorów logicznych, nie ma co podchodzić do programowania. A więc do operatorów logicznych należą:

- ! negacja
- && koniunkcja
- || alternatywa

Ich składnię przedstawiam:

symbol operatora	funkcja	składnia
!	negacja	!wyrażenie
&&	koniunkcja	wyrażenie && wyrażenie
	alternatywa	wyrażenie wyrażenie

Oprócz wyżej przedstawionych operatorów aby nie dochodziło do omyłki pisarskiej lub błędów w tworzenie programu przedstawię jeszcze operatory bitowe logiczne - które interpretują operand(y) jako uporządkowany ciąg bitów. I tych operatorach napisze sobie później.

Bitowe operatory logiczne to:

- & bitowa koniunkcja
- | bitowa alternatywa
- ^ bitowa różnica symetryczna
- << przesunięcie w lewo
- >> przesunięcie w prawo

~ bitowa negacja

W zestawieniu wygląda następująco:

symbol operatora	funkcja	składnia
&	bitowa koniunkcja	wyrażenie & wyrażenie
	bitowa alternatywa	wyrażenie wyrażenie
^	bitowa różnica symetryczna	wyrażenie ^ wyrażenie
<<	przesunięcie w lewo	wyrażenie << wyrażenie
>>	przesunięcie w prawo	wyrażenie >> wyrażenie
~	bitowa negacja	wyrażenie ~ wyrażenie

I do potrzeb programowania na początek, jeszcze zapiszę operatory przypisania.

Oto zestawienie operatorów przypisania:

symbol operatora	zapis skrócony	zapis rozwinięty
+=	a += b	a = a + b;
-=	a -= b	a = a - b;
*=	a *= b;	a = a * b;
/=	a /= b;	a = a / b;
%=	a %= b	a = a % b;
<<=	a <<= b;	a = a << b
>> =	a >>= b;	a = a >>= b
&=	a &= b;	a = a & b;
=	a = b;	a = a b;
^=	a ^= b;	a = a ^ b;

W rzeczy samej to nie są wszystkie operator. Bowiem jest ich więcej. Ale jak na początek to mi wystarczy.

Abym mógł podjąć się programowania, jak na początek małych programów potrzebne mi są typy zmiennych.

Nadmieniam, że na początek programuje małe programy, bo jak sam się już przekonałem wielkie programy właśnie są suma małych programów. Stąd też sobie zapisuję wszystko to co jest mi potrzebne do tworzenia małych programów.

Jak już wiadomo, **zmienna** to obiekt przechowująca informację w programie. Dlatego też, są zmienne globalne i zmienne lokalne. Zmienna globalna jest to zmienna dla całego programu, a zmienna lokalna tylko i wyłącznie dla wyodrębnionej części wręcz fragmentu programu.

W związku z powyższym, przystępując do programowania, należy zmienną zadeklarować jeszcze przed pierwszym użyciem.

Deklaracja zmiennej polega na przedstawieniu typu zmiennej i nadanie nazwy zmiennej.

To jest:

typ_zmiennej nazwa_zmiennej

Deklarując zmienną, informuje kompilator o tym że zmienna która używam w programie, jest zmienną jest obiekt określonego typu i posiada nazwę.

Na przykład:

```
int a
```

```
char znak
```

Natomiast definicja zmiennej jest zarezerwowaniem dla niej określonego miejsca w pamięci wraz z przypisaniem jej wartości.

Na przykład:

```
int a = 8
```

```
char znak = „R”
```

Podstawowe typy zmiennej które używam w programowaniu to:

int – liczba całkowita

double – liczba rzeczywista podwójnej precyzji

char – znak

bool - zmienna logiczna

string – tekst

Są jeszcze inne typy zmiennych. Ale użyje je później.

A teraz pora na ćwiczenia w programowaniu.

Dodawanie, ćwiczenie 1.

```
/* zadanie pierwsze dodawanie */  
  
#include <iostream>  
#include <math.h>  
#include <string>  
#include <conio.h>  
  
using namespace std;  
  
int liczba1, liczba2;  
  
int main() {  
  
cout << "Podaj liczbe:" << endl;  
cin >> liczba1;  
  
cout << "Podaj nastepna liczbe:" << endl;
```

```
cin >> liczba2;  
  
cout << "liczba1 + liczba2 = " << liczba1 + liczba2 << endl;  
  
getch ();  
  
return 0;  
  
}
```

Na ekranie komputera pojawi się:

Podaj liczbę:

—

k

Więc podajemy liczbę z klawiatury, np. 6. I klikamy na enter Wtedy z ekranu odczytamy:

Podaj liczbę:

6

Podaj następną liczbę:

—

Więc podajemy następną liczbę, np. 12. Po dodaniu liczby 12 na monitorze pojawia się to co następuje:

Podaj liczbę:

6

Podaj następną liczbę:

12

—

Klikamy na enter. Program wykonuje zadanie i na ekranie odczytujemy:

Podaj liczbę:

6

Podaj następną liczbę:

12

liczba1 + liczba2 = 18

Program został wykonany.

Tak samo jest z odejmowaniem, mnożeniem i dzieleniem.

Ja wszystko wykonuje na Windows10 i starym kompilatorze. Dlatego jest trochę inaczej.

Zamiast `cin.get ()`; mam wstawione `getch ()`;

Operatory które sobie wypisałem, będę używał w następnych własnych programach.

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. Symfonia C++ standard Jerzy Grębosz
5. PRZEWODNIK DLA POCZĄTKUJĄCYCH C++ ALEX ALLAIN
6. i internet