

[C/C++] Wszystko co chcielibyście wiedzieć, ale boicie się zapytać (sheadovas/artykuly/programowanie/cc-wszystko-co-chcielibyście-wiedzieć-ale-boicie-się-zapytać/)

Gru 09, 2014 / Programowanie (sheadovas/category/artykuly/programowanie/)

„Wszystko co chcielibyście wiedzieć o programowaniu w c/cpp, ale boicie się zapytać”, czyli artykuł poświęcony temu co warto wiedzieć o języku c/c++, a nie w każdym poradniku o tym języku jest powiedziane, to właśnie tutaj sobie o paru takich trickach, ciekawostkach opowiemy.

Chciałbym zaznaczyć, że tytuł tego artykułu może wyraziłem nieco zbyt poetycko, bo na pewno nie znajdziecie tutaj wszystkich tego typu sztuczek, które mogą okazać się przydatne. Ten artykuł ma raczej na celu pokazanie że język c ma wiele niespodzianek, a tutaj znajdziesz zestawienie tych, z którymi się spotkałem podczas mojej przygody z programowaniem. Nie ma sensu więcej przedłużać tego wstępu, zaczynamy!

/ dodam jeszcze że kolejność ciekawostek jest zupełnie przypadkowa */*

#1 Warunki są wykonywane jako instrukcje

Oznacza to nie mniej, nie więcej że jeżeli damy do np. if’a instrukcję przypisania (i=1) to zostanie to wyrażenie potraktowane jako instrukcja, czyli do i zostanie przypisana wartość 1, a taki kod sprawi, że bez napisania ani jednego średnika wczytamy 2 liczby i wyświetlimy ich sumę:

```
1  #include <iostream>
2
3  int main()
4  {
5      if(int a=1)
6      {
7          if(int b=1)
8          {
9              if(std::cin >> a)
10             {
11
12             }
13             if(std::cin >> b)
14             {
15
16             }
17             if(std::cout<<a+b)
18             {
19
20             }
21         }
22     }
23 }
```

#2 Warunki złożone nie są zawsze sprawdzane są w całości

Tutaj jako „warunek złożony” rozumiem warunek połączony spójnikami AND lub OR. Zilustrujmy sobie to na 2 przykładach, jeżeli mamy warunki połączone tylko za pomocą AND, to kolejne warunki są sprawdzane do napotkania pierwszego fałszywego warunku, bo wtedy całe wyrażenie jest fałszywe więc nie ma sensu go sprawdzać.

Odwrotnie jest w przypadku OR, warunki są sprawdzane do napotkania pierwszego warunku zwracającego prawdę.

To spostrzeżenie może nam pomóc w projektowaniu kodu, ponieważ możemy w ten sposób w jakiś sposób zwiększyć szybkość działania programu, jeżeli w odpowiedni sposób rozmieścimy warunki w if'ie.

C++

```
1 #include <iostream>
2
3 using namespace std;
4
5 bool f1()
6 {
7     cout << "Wykonano f1!\n";
8     return true;
9 }
10
11 bool f2()
12 {
13     cout << "Wykonano f2n";
14     return false;
15 }
16
17 int main()
18 {
19     cout << "Test dla &&:\n";
20     if (f2() && f1())
21     {
22         // &&
23     }
24
25     cout << "Test dla ||:\n";
26     if (f1() || f2())
27     {
28
29     }
30     cin.get();
31 }
32
33 /* OUTPUT:
```

#3 Instrukcja warunkowa

(warunek ? jezeli_true : jezeli_false)

Jest to bardzo fajna choć już nieco zapomniana rzecz jeszcze z czasów języka C. W miejsce warunku wstawiamy jakieś zdanie logiczne, następnie po ? możemy podać 2 wartość, w przypadku gdy warunek jest spełniony lub po : piszemy co ma zostać zwrócone w przypadku fałszu. Ta instrukcja przydaje się w przypadkach gdy istnieją tylko 2 przypadki, a my jedynie chcemy ustawić odpowiednią wartość, np.:

C++

```
1 // zamiast takiego kodu:
2 if(i < 10)
3     i = i+1;
4 else
5     i = 0;
6
7 // możemy napisać to tak:
8 i = (i<10 ? i+1 : 0);
```

#4 Operatory

Z pewnością znacie wiele podstawowych operatorów (np. *,/,+,-), ale istnieje wiele, o których się nawet nie wie że istnieją.

C++

```
1 | // do operacji na bitach:
2 |
3 | | // OR
4 | & // AND
5 | ^ // XOR
6 | ~ // NOT
7 | >> // przesunięcie bitów w prawo
8 | << // przesunięcie bitów w lewo
```

W zasadzie poprzedni podpunkt #3 też zalicza się do operatorów, ale moim zdaniem wygodniej jest o tym myśleć jak o instrukcji.

#5 STL

Jest to zestaw standardowych szablonów do języka c++, znajdziesz tutaj vectory (dynamiczne tablice), talie, listy, etc. Po więcej informacji odsyłam tutaj (<http://cpp0x.pl/kursy/Kurs-STL-C++/114>).

#6 „Konwersja liczb”

Jest to kolejna rzecz, którą można podczepić pod operatory, ale myślę że fajnie to wyodrębnić. Otóż castowanie (<http://www.cplusplus.com/doc/tutorial/typcasting/>) pozwala nam na wymuszenie konwersji z 1 typu liczb na drugi, tzn jeśli mamy zmienną int, ale chcemy aby została potraktowana przez chwilę jako float (bo np. musimy dostarczyć ją do funkcji, która nie przyjmuje int), to możemy zrobić coś takiego:

C++

```
1 | int i = 5;
2 | float(i); // teraz jest traktowana jako float
3 |
4 | 5.2; // taki zapis jest traktowany jakby to była liczba double
5 | 5.2f; // liczba float
```

#7 Przydatne funkcje

Jeżeli poszukujemy algorytmów do sortowania, podmieniania wartości 2 zmiennych,etc. to istnieje biblioteka algorithm (<http://www.cplusplus.com/reference/algorithm/>), która nam to umożliwia.

```
#include <algorithm>
```

To tyle, jeżeli chodzi o zestaw ciekawostek na ten moment, od razu mówię że ten post będzie aktualizowany co jakiś czas. Jeżeli chcecie podzielić się jakimiś ciekawostkami/trikami tego typu, czyli rzeczami, które po prostu warto wiedzieć w programowaniu nawet na początku zapraszam do komentarzy, a obok ciekawostki w tym poście pojawi się wasz nick.

Tak samo jeżeli znalazłeś błąd lub chcesz się po prostu podzielić swoją opinią zapraszam do systemu komentarzy.

Code ON!