

Część VIII C++

Przedrostkowa i przyrostkowa inkrementacja i dekrementacja

W poprzednim ćwiczeniu operatory inkrementacji i dekrementacji występowały w wersji przyrostkowej.

Istnieje inny sposób zapisu zwiększania i zmniejszania o 1 tak zwany przedrostkowy, polegający na tym że operator ++ lub -- występuje przed nazwą zmiennej.

Jego działanie nie zmienia się, zmienia się sposób wykonywania tej modyfikacji.

Ćwiczenie 1 – spróbuj wykonać samemu poniższe instrukcje, jeśli nie potrafisz lub pojawia się błąd podczas kompilowania skorzystaj z zrzutu kodu zamieszczonego poniżej instrukcji

1. Skopiuj na pulpit folder o nazwie **nazwisko 23** i zmień jego nazwę na **nazwisko24**

2. Usuń plik exe, otwórz plik z rozszerzeniem **.dev**

3. Zmodyfikuj program tak aby zamiast wykonywać instrukcje inkrementacji i dopiero potem wyświetlać zmienną do standardowego strumienia wyjściowego (cout), przekaz od razu wyrażenie x++ i dla porównania wyrażenie ++x

4. Kod programu powinien wyglądać tak:

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int x;
    cout << "Podaj dowolna liczbe: ";
    cin >> x;
    cout << "Liczba wynosi :" << x++ << endl;
    cout << "Liczba wynosi :" << x << endl;
    cout << endl;
    cout << "Podaj ta sama iczbe:";
    cin >> x;
    cout << "Liczba wynosi :" << ++x << endl;
    cout << "Liczba wynosi :" << x << endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

5. Skompiluj i uruchom program

6. Wprowadź z klawiatury pierwszą liczbę i naciśnij Enter

7. Wprowadź tą samą liczbę i naciśnij Enter

Wyświetlona na ekranie wartość nie będzie zwiększona o 1. Ponieważ do strumienia cout najpierw została przekazana zmienna x z klawiatury a dopiero potem zadziałał operator inkrementacji zwiększając wartość o 1..

```
Podaj dowolna liczbe: 20
Liczba wynosi :20
Liczba wynosi :21

Podaj ta sama iczbe:20
Liczba wynosi :21
Liczba wynosi :21
Aby kontynuować, naciśnij do
```

Jeśli zastosujemy wersje przedrostkową na ekranie pojawia się od razu zwiększona o 1 wartość zmiennej x. Dzieje się tak dlatego że zmienna najpierw została poddana inkrementacji i do strumienia cout trafia zmieniona wartość zmiennej.

Arytmetyczne operatory przypisania – ułatwiają życie programiście i upraszczają pracę kompilatora. Są to konstrukcje pozwalające za pomocą jednego symbolu zmienić wartość i przypisać ją na przykład do zmiennej.

▼ Arytmetyczne operatory przypisania			
operator	nazwa	przykład	równoważny zapis
+=	przypisanie z dodawaniem	x += 11	x = x + 11
-=	przypisanie z odejmowaniem	y -= x	y = y - x
*=	przypisanie z mnożeniem	x *= y - 2	x = x * (y - 2)
/=	przypisanie z dzieleniem	z /= x	z = z / x
%=	przypisanie reszty dzielenia	x %= 7	x = x % 7

Ćwiczenie 2

1. Utwórz na pulpicie folder o nazwie **nazwisko25**
2. Utwórz nowy projekt w Dev C++ i zapisz go w folderze nazwisko25
3. Wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej
4. Skompiluj i uruchom program
5. Przeanalizuj program

Działanie zapisu $x = x + 20$ jest takie same jak $x += 20$

```

#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int x;
    cout << "Podaj x: ";
    cin >> x;
    cout << "Zmienna x wynosi: " << x << endl;
    x = x + 20;
    cout << "Teraz zmienna wynosi: " << x << endl;
    cout<< endl;
    /*.....*/
    cout<< "Teraz drugi sposob liczenia za pomoca arytmetycznego operatora przypisania"<< endl;
    cout<< endl;

    cout << "Podaj ta sama wartosc x: ";
    cin >> x;
    x += 20;
    cout << "Teraz zmienna wynosi: " << x << endl;

    cout << endl << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

Przypisanie ze zmienna wartości o 1

Zobaczmy jak zachowa się kompilator w przypadku kiedy użyjemy zapisu $y = x++$

Ćwiczenie 3

1. Utwórz na pulpicie folder o nazwie **nazwisko26**
2. Utwórz nowy projekt w Dev C++ i zapisz go w folderze nazwisko26
3. Wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej
4. Skompiluj i uruchom program

```

#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int x, y;

    x = y = 1;
    y = x++;
    cout << "x = " << x << endl << "y = " << y << endl << endl;

    x = y = 1;
    y = ++x;
    cout << "x = " << x << endl << "y = " << y << endl;

    cout << endl << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

Przeanalizujmy powyższy program

W wierszu `y = x ++` przypisano wartość zmiennej `x` do zmiennej `y`.

Następnie zmienna `x` zostaje zwiększona o 1.

W wyniku tego będzie wartość 1 zapisana w zmiennej `y` i wartość 2 zapisana w zmiennej `x`

```

x = 2
y = 1

x = 2
y = 2

Aby kontynuować, naciśnij dowolny klawisz .

```

W drugiej części instrukcja zostanie wykonana w ten sposób że:

Najpierw zostanie zwiększona zmienna `x` o 1, a następnie zmieniona wartość zostanie przypisana do zmiennej `y`

Odejmowanie a znak minusa

Operator odejmowania ma drugie znaczenie, może oznaczać liczbę ujemną i należy zapisywać go w nawiasach `(-x)`

Relacje między zmiennymi

Operatory relacji służą do sprawdzania czy wartość zmiennej jest większa niż inna liczba lub wartość innej zmiennej

▼ Operatory relacji			
operator	nazwa	przykład	wynik
==	równe	$12 == (10 + 2)$	1 (prawda)
!=	różne od	$7 != (9 - 2)$	0 (fałsz)
<	mniejsze	$3 < 9$	1 (prawda)
>	większe	$4 > (2 + 7)$	0 (fałsz)
<=	mniejsze lub równe	$5 <= (2 + 3)$	1 (prawda)
>=	większe lub równe	$4 >= 5$	0 (fałsz)

Zamiast liczb 1 i 0 możemy operować pojęciami prawda i fałsz. W programowaniu to bardzo częsta interpretacja. Warto wiedzieć, że system dwójkowy przedstawia się również jako „jest sygnał”(1) i brak sygnału(0)

Ćwiczenie 4

1. Utwórz na pulpicie folder o nazwie **nazwisko27**
2. Utwórz nowy projekt w Dev C++ i zapisz go w folderze nazwisko27
3. Wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej
4. Skompiluj i uruchom program

```

#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int x, y, z;
    x = 5;
    y = 10;
    z = 15;

    cout << "wynikiem relacji x < y jest " << (x < y) << endl;
    cout << "wynikiem relacji x > y jest " << (x > y) << endl;

    cout << "wynikiem relacji x == y jest " << (x==y) << endl;
    cout << "wynikiem relacji z == 15 jest " << (z==15) << endl;

    cout << endl << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

Przeanalizujmy wynik działania programu!!!

W wierszu pierwszym oraz ostatnim jest podana **prawdziwa relacja** czyli **1**

W wierszu 2 i 3 **falszywą relacje** reprezentuje liczba **0**

```

wynikiem relacji x < y jest 1
wynikiem relacji x > y jest 0
wynikiem relacji x == y jest 0
wynikiem relacji z == 15 jest 1

Aby kontynuować, naciśnij dowolny klawisz . . .

```

ZADANIE KONTROLNE

1. Utwórz na pulpicie folder o nazwie **nazwisko28**

2. Zmodyfikuj poprzedni program w ten sposób by najpierw po kolei można było wprowadzić z klawiatury zmienna x , y , z a następnie program wykonał poprzednie operacje.

3. Skompiluj i uruchom program

Wszystkie pliki z nazwiskiem i kolejnym numerem umieszczamy w swoim folderze nazwiskocplusplus na serwerze.