

# Instrukcja warunkowa IF, operatory relacji i relacje między zmiennymi

1. Zapis operatora przypisania
2. Zapis operatora relacji porównania
3. Kiedy używa się którego operatora
4. Co zwraca operator relacji porównania
5. Co jest prawdą a co fałszem

Instrukcję warunkową można ogólnie zapisać w postaci:

**if ( warunek ) instrukcja;**

Działanie instrukcji **if** można odczytać następująco:

Jeśli warunek jest **prawdziwy**, wykonaj instrukcję.

Gdy warunek jest **fałszywy**, nic nie rób.

Stosuj poprawny zapis nawiasów klamrowych gdy chcesz użyć **więcej niż jednej instrukcji**:

**if ( warunek )**

```
{  
    1 instrukcja;  
    2 instrukcja;  
    n instrukcja;  
}
```

Zwróć **szczególną uwagę**, że **po if (warunek) nie piszemy średnika**.

Bezpośrednio po `if ( . . . )` znajduje się instrukcja, bądź blok instrukcji do wykonywania warunkowo.

Jeśli napiszesz średnik po `if ( . . . )`, warunkowo wykonywana będzie **instrukcja pusta!**

## Zapis sprawdzający kolejne warunki

**if ( warunek 1 )**

```
{  
    1 instrukcja;  
    2 instrukcja;  
}
```

**if ( warunek 2 )**

```
{  
    1 instrukcja;  
    2 instrukcja;  
}
```

# Operator przypisania =

Do tej pory poznałeś operator **przypisania**, który oznaczony jest znakiem równości = Służy do przypisywania wartości do zmiennej np.

```
string imie = „Jan”;
```

Powyższy zapis oznacza że ciąg znaków „Jan” został **przypisany** do zmiennej imie.

# Operator relacji porównania ==

Aby sprawdzić czy dwie zmienne są równe, należy posłużyć się operatorem porównania, który zapisuje się dwoma znakami równości ==

**Przykład1:** dla sprawdzenia czy dwie liczby są równe należy posłużyć się zapisem

```
12 == 12   wynik relacji to prawda (1)
```

**Przykład2:** dla sprawdzenia czy dwie liczby są równe należy posłużyć się zapisem

```
2 == 4    wynik relacji to fałsz (0)
```

**Przykład3:** dla sprawdzenia czy dwie zmienne są równe należy posłużyć się zapisem

```
zmienna1 == zmienna2
```

**W zależności od przechowywanych wartości zmiennych relacja zwróci wartość logiczną: **prawdę (1)** lub **fałsz (0)****

Operatory relacji nazywają się tak ponieważ **między dwoma argumentami zachodzi relacja** która w wyniku zwraca **wartość logiczną**, określającą spełnienie bądź nie spełnienie reprezentowanej przez ten operator relacji zachodzącej między podanymi argumentami.

## Operatory relacji „większy lub równy” i „mniejszy lub równy”

Te operatory zapisuje się w taki sam sposób jak w Excel.

$\geq$  to operator „większy lub równy”

$\leq$  to operator „mniejszy lub równy”

**Ważne:** zawsze najpierw zapisuje się znak większości i mniejszości, a za nim znak równości.

**Operatory błędnie zapisane to:**

$=<$

$=>$

Oprócz wyżej wymienionych operatorów relacji są też znane z matematyki

większy  $>$

i mniejszy  $<$

## Relacje między zmiennymi

Wynikiem operatora relacji jest wartość logiczna: **prawda** albo **falsz**, czyli warunek jest spełniony albo nie. Słowa kluczowe, które opisują te stany w języku C++ to odpowiednio **true** i **false**. Tak samo jak w przypadku operacji arytmetycznych, operacje porównania możemy wykonywać bezpośrednio na wartościach jak i zmiennych.

**WARTOŚCIĄ LOGICZNĄ JEST PRAWDA LUB FAŁSZ**

**PRAWDA TO LICZBA RÓŻNA OD ZERO**

**FAŁSZ TO ZERO**

Link do kompilatora C++ online [https://www.onlinegdb.com/online\\_c++\\_compiler](https://www.onlinegdb.com/online_c++_compiler)

**Zadanie 1.** Napisz program z przykładu poniżej, który przedstawi zachodzące relacje między zmiennymi. Program przedstawia kiedy relacja między zmiennymi zwraca prawdę (1) a kiedy fałsz (0).

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int x=5;
8      int y=10;
9      int z=15;
10
11     cout<<"Wynikiem relacji x < y jest " << (x < y) <<endl;
12     cout<<"Wynikiem relacji x > y jest " << (x > y) <<endl;
13
14     cout<<"Wynikiem relacji x == y jest " << (x == y) <<endl;
15     cout<<"Wynikiem relacji z == 15 jest " << (z == 15) <<endl;
16
17     return 0;
18 }
```

Wyniki są następujące:

```
Wynikiem relacji x < y jest 1
Wynikiem relacji x > y jest 0
Wynikiem relacji x == y jest 0
Wynikiem relacji z == 15 jest 1
```

## Co jest **prawdą**, a co **fałszem**?

Reguła w C++ jest następująca:

**jeśli wartość jest różna od zera to **prawda** - w przeciwnym wypadku **fałsz****

**Zadanie 2.** Dopisz trzy instrukcje do programu z zadania 1.

instrukcje:

2 >= y

y <= y

x = y

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int x=5;
8     int y=10;
9     int z=15;
10
11     cout<<"Wynikiem relacji x < y jest " << (x < y) <<endl;
12     cout<<"Wynikiem relacji x > y jest " << (x > y) <<endl;
13     cout<<"Wynikiem relacji x == y jest " << (x == y) <<endl;
14     cout<<"Wynikiem relacji z == 15 jest " << (z == 15) <<endl;
15
16     cout<<"Wynikiem relacji 2 >= y jest " << (2 >= y) <<endl;
17     cout<<"Wynikiem relacji y <= y jest " << (y <= y) <<endl;
18     cout<<"Wynikiem relacji x = y jest " << (x = y) <<endl;
19
20     return 0;
21 }
```

input

```
Wynikiem relacji x < y jest 1
Wynikiem relacji x > y jest 0
Wynikiem relacji x == y jest 0
Wynikiem relacji z == 15 jest 1
Wynikiem relacji 2 >= y jest 0
Wynikiem relacji y >= y jest 1
Wynikiem relacji x = y jest 10
```

**Wyjaśnij wyniki instrukcji!**

**Zadanie 3.** Napisz program, który sprawdzi czy jedziesz samochodem z poprawną szybkością.

Program ma zapytać użytkownika z jaką szybkością jedzie i wypisać odpowiedni komunikat.

Jeżeli jedzie 50 km/h lub mniej to niech pojawi się komunikat „Jedziesz prawidłowo”.

Jeżeli jedzie powyżej 50 km/h to niech pojawi się komunikat „Przekroczyles predkosc!”

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      float predkosc;
7
8      cout << "Podaj predkosc samochodu ";
9      cin >> predkosc;
10
11
12     if ( predkosc < 51 )
13     {
14         cout << "Jedziesz prawidłowo" << endl;
15     }
16
17     if ( predkosc >= 51 )
18     {
19         cout << "Przekroczyles predkosc!" << endl;
20     }
21
22     return 0;
23 }
```

**Zadanie 4.** Czy pamiętasz Co jest **prawdą**, a co **falszem**?

Reguła w C++ jest następująca:

jeśli wartość jest różna od zera to **prawda** - w przeciwnym wypadku **falsz**

Zamień w poprzednim programie wyważenia:

***predkosc < 51***

i

***predkosc >= 51***

na dowolne liczby. Wypróbuj też zero.

Jakie uzyskałeś wyniki działania programu?

Czy liczby rzeczywiste, dodatnie i ujemne zwracają prawdę?

**Zadanie 5.** Dopisz do każdej instrukcji warunkowej IF nowe instrukcje. W pierwszym przypadku instrukcja ma obliczyć o ile kierowca przekroczył prędkość. W drugim przypadku dopisz instrukcje która obliczy i wypisze do konsoli o ile kierowca może przyspieszyć, aby jechać zgodnie z przepisami.

Przykład rozwiązania poniżej

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      float predkosc;
7
8      cout << "Podaj predkosc samochodu ";
9      cin >> predkosc;
10
11
12     if ( predkosc < 51 )
13     {
14         cout << "Jedziesz prawidlowo" << endl;
15         cout << "Mozesz przyspieszyc o " << 50 - predkosc << endl;
16     }
17
18     if ( predkosc >= 51 )
19     {
20         cout << "Przekroczyles predkosc!" << endl;
21         cout << "Zwolnij o " << predkosc - 50 << endl;
22     }
23
24     return 0;
25 }
```

input

```
Podaj predkosc samochodu 45
Jedziesz prawidlowo
Mozesz przyspieszyc o 5
```




**Zadanie 6.** Dopisz do programu trzecią instrukcję warunkową, która poinformuje użytkownika o jeździe po autostradzie jeśli podał prędkość większą od 90 km/h

Przykład rozwiązania poniżej

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      float predkosc;
7
8      cout << "Podaj predkosc samochodu ";
9      cin >> predkosc;
10
11
12     if ( predkosc < 51 )
13     {
14         cout << "Jedziesz prawidlowo" << endl;
15         cout << "Mozesz przyspieszyc o " << 50 - predkosc << endl;
16     }
17
18     if ( predkosc >= 51 )
19     {
20         cout << "Przekroczyles predkosc!" << endl;
21         cout << "Zwolnij o " << predkosc - 50 << endl;
22     }
23
24     if ( predkosc > 90 )
25     {
26         cout << "Jedziesz po autostradzie!" << endl;
27     }
28
29     return 0;
30 }
31
```

input

```
Podaj predkosc samochodu 105
Przekroczyles predkosc!
Zwolnij o 55
Jedziesz po autostradzie!
```



**Zauważ, że program w zależności od podanej szybkości niepotrzebnie podaje wartość z drugiej i trzeciej instrukcji warunkowej if.**

**Możesz to poprawić stosując zagnieżdżanie pętli lub operator łączenia (zad. 7)**

**Zadanie 7.** Dopisz do programu przedział szybkości samochodu za pomocą polecenia łączenia && (7 + shift)

```
if ( predkosc >= 51 && predkosc < 90 )
```