

Instrukcja warunkowa pozwala wykonywać pewne instrukcje przy spełnionym bądź nie spełnionym **warunku** dzięki niej program nie musi zawsze biegać tą samą ścieżką, lecz w zależności od zastanej sytuacji może on wybierać właściwą drogę. Bez instrukcji warunkowej komputery byłyby niczym więcej niż kalkulatorami. Dzięki niej mogą rozwiązywać dowolnie skomplikowane problemy. Instrukcja warunkowa posiada w języku C++ następującą składnię:

```
if(warunek) instrukcja1; else instrukcja2;
```

warunek - jest wyrażeniem logicznym, które może przyjmować wartość false albo true. W zależności od jego wartości wykonywana jest jedna z instrukcji:

warunek = **true** - wykonywana jest *instrukcja1*, a *instrukcja2* zostaje pominięta

warunek = **false** - pominięta zostanie *instrukcja1*, a wykonana *instrukcja2*.

Jeśli warunkowo wykonujemy tylko jedną instrukcję, to człon z else można pominąć:

```
if(warunek) instrukcja;
```

Teraz instrukcja będzie wykonana, jeśli warunek jest prawdziwy. Jeśli jest on fałszywy, to program pominie instrukcję i przejdzie do dalszej swej części.

Jeśli w ramach instrukcji warunkowej chcemy wykonywać więcej niż jedną instrukcję, to stosujemy tzw. instrukcję złożoną, czyli dowolnie dużą grupę instrukcji ujętych w klamery:

Instrukcja warunkowa w języku C++

Wpisany przez Ilona Simek

wtorek, 14 kwietnia 2020 14:58 - Poprawiony wtorek, 14 kwietnia 2020 15:05

```
if(warunek)
{
    ciąg instrukcji wykonywanych, gdy warunek jest prawdziwy
}
else
{
    ciąg instrukcji wykonywanych, gdy warunek jest fałszywy
}
```

Zwróć uwagę, że po klamerkach obejmujących grupę instrukcji nie stosujemy już średnika.

warunek może być również wyrażeniem arytmetycznym. W takim przypadku wartość **zero** jest traktowana jak

false

, a wartość

różna od zera

jest traktowana jak

true

- niekiedy to podejście upraszcza zapis programu.

W warunkach można stosować operatory porównań - wynikiem działania takiego operatora jest

true

, jeśli relacja jest spełniona i

false

w przypadku jej niespełnienia. Mamy następujące operatory porównań

(a i b to wyrażenia arytmetyczne, niekoniecznie zmienne!)

:

Operator

Przykład

==

Instrukcja warunkowa w języku C++

Wpisany przez Ilona Simek

wtorek, 14 kwietnia 2020 14:58 - Poprawiony wtorek, 14 kwietnia 2020 15:05

a	==	b
---	----	---

!=

a	!=	b
---	----	---

<

a	<	b
---	---	---

<=

a	<=	b
---	----	---

>

a	>	b
---	---	---

>=

Instrukcja warunkowa w języku C++

Wpisany przez Ilona Simek

wtorek, 14 kwietnia 2020 14:58 - Poprawiony wtorek, 14 kwietnia 2020 15:05

a	>=	b
---	----	---

Należy uważać z operatorem równości ==, gdyż pominięcie jednego znaku = robi z niego instrukcję przypisania:

if(a == 10) b = 0; // zeruje zmienną b tylko wtedy, gdy zmienna a jest równa dziesięć.

if(a = 10) b = 0; // zmienna b będzie zawsze zerowana, dodatkowo zmienna a przyjmie wartość 10

W drugiej wersji zamiast warunku porównania `a == 10` mamy zwykle przypisanie `a = 10`. Instrukcja przypisania posiada wartość równą przypisywanemu wyrażeniu - w tym przypadku 10

. Ponieważ jest to wartość różna od zera, to instrukcja `if` zinterpretuje ją jako `true` i wykona przypisanie

`b`

`=`

`0`

. Widzimy zatem, że bez względu na wartość zmiennej

`a`

, przypisanie

`b`

`=`

`0`

będzie wykonywane zawsze. Dodatkowo zawartość zmiennej

`a`

zostanie stracona i zastąpiona liczbą

10

- a o to przecież nam nie chodziło.