

Języki programowania 1

Instrukcje sterujące przebiegiem wykonania programu

Pewną grupą instrukcji w języku C są tzw. instrukcje warunkowe. Zaliczane są do nich:

- if (jeżeli),
- if – else (jeżeli-jeżeli nie),
- switch (przełącz),
- break (przerwij),
- continue (kontynuuj),
- goto (idź do).

Instrukcja warunkowa if

Jedną z popularniejszych instrukcji warunkowych w języku C jest instrukcja if. Służy ona do sprawdzania, czy warunek jest spełniony i podjęcia decyzji, czy kontrolowany przez nią blok instrukcji powinien zostać wykonany, czy też nie. Ogólny format tejże instrukcji jest następujący:

```
if(wyrażenie)  
{  
    instrukcja_1;  
    instrukcja_2;  
    ...  
}
```

Wyrażenie w tym zapisie stanowi tzw. kryterium warunkowe. Jeżeli wyrażenie to zwraca wartość logicznej prawdy (TRUE) lub też inaczej warunek jest spełniony, bądź też wyrażenie to zwraca dowolną wartość niezerową, wówczas instrukcja lub blok instrukcji (wówczas ujęty w nawiasy {} zostaną wykonane. Jeżeli natomiast wyrażenie to zwróci wartość zero lub też wyrażeni dane nie jest spełnione, wówczas instrukcja lub ich blok zostanie pominięty.

Przykład zastosowania instrukcji warunkowej - jeżeli *if.c*.

Instrukcja warunkowa if – else

W instrukcji if jeśli warunek jest spełniony, komputer wykonuje instrukcję kontrolowaną przez tę instrukcję sterującą. Jeżeli natomiast wyrażenie to jest FALSE, następuje przeskok do następującej instrukcji w programie. Czasami jednak programiście chodzi o to, że gdy warunek nie jest spełniony, to program ma wykonać również jakieś instrukcje. Do tego celu służy kolejna instrukcja warunkowa if – else. Ogólna jej postać jest następująca:

```
if(wyrażenie)  
{  
    instrukcja_1;  
    instrukcja_2;  
    ...  
}  
else  
{  
    instrukcja_A;  
    instrukcja_B;  
    ...  
}
```

Podobnie jak w poprzedniej instrukcji wyrażenie jest kryterium warunkowym. Jeżeli jest ono spełnione to wówczas zostają wykonane instrukcje występujące „zara po”. Natomiast, jeżeli dany warunek nie jest spełniony, to wówczas zostaje wykonana instrukcja lub blok instrukcji występujący po else (jeżeli nie). Przykład zastosowania tej instrukcji – program *if_else.c*.

Zagnieżdżone instrukcje warunkowe if-else-if

Często pisząc program istnieje potrzeba podejmowania wielu decyzji i dokonywania wyboru. Można zrobić to za pomocą tzw. instrukcji zagnieżdżonych if.

Przykład zagnieżdżania instrukcji if zawarty jest w programie *if_else_if.c*.

Instrukcja switch

Inną możliwością podjęcia decyzji (zwłaszcza przy wielu wariantach) jest możliwość wykorzystania instrukcji switch (przełącz). Przy pomocy tej instrukcji można dokonywać wyborów spośród nieograniczonej ilości wariantów posługując się wartością wyrażenia warunkowego i wariantami case.

Ogólna składnia tej instrukcji jest następująca:

```
switch(wyrażenie_sterujące)  
{  
    case wyrażenie_1:instrukcja_1;  
    case wyrażenie_2:instrukcja_2;  
    ....  
    default: instrukcja domyślna;  
}
```

Na samym początku jest sprawdzana wartość zwracana przez wyrażenie_sterujące. Jeżeli wartość zwracana z danego wyrażenia jest równa wartości zwróconej przez którekolwiek wyrażenie (wyrażenie_1,

wyrażenie_2...etc.), to wówczas następuje wykonanie odpowiadającej mu instrukcji (instrukcja_1, instrukcja_2, etc.). Wyrażenia: wyrażenie_1, wyrażenie_2.... są wyrażeniami stałowartościowymi (powinny być wzajemnie się wykluczające).

Jeżeli natomiast zwrócona wartość z wyrażenia_sterującego nie będzie równa żadnej wartości wyrażen stałowartościowych, to wówczas zostanie wykonana instrukcja domyślna umieszczona po słowie default.

W instrukcji możliwy jest przypadek, że wielu wariantom, odpowiada to samo działanie:

```
switch(x)
{
    case '1':
    case '2':
        printf("Wybrana została opcja 1 lub 2");
    case '3':
        printf("Teraz trójeczka");
    default:
        printf("Pozostałe opcje");
}
```

Przykład zastosowania instrukcji switch pokazany w programie **switch.c**.

Instrukcja break

Wstawienie tej instrukcji powoduje przerwanie wykonywania się danego bloku (instrukcji, pętli, etc..) Przykładowo w instrukcji switch wstawiając po wykonaniu danego wyrażenia, instrukcję break spowodujemy pominięcie wykonania się pozostałych wyrażen, przerywając tym samym instrukcję switch. (patrz przykład switch.c).

Instrukcja `break` może również przerywać nieskończone pętle (`for(;;)` oraz `while(1)....`).

Instrukcja `continue`

Czasami zamiast całkiem przerywać instrukcję, wystarczy tylko pominąć niektóre wyrażenia, a sama instrukcja ma się wykonywać dalej. Do tego celu służy właśnie instrukcja `continue`. Powoduje ona natychmiastowy skok na początek pętli i podjęcie próby ponownego wykonania pętli od początku.

Przykład zastosowania instrukcji `continue` – program ***`continue.c`***.