Podstawy programowania komputerów

Wykład 4: „Instrukcje proste i złożone w ANSI C”

swalover@ie.tu.koszalin.pl

Instrukcję proste i złożone

- Instrukcja prosta składa się z identyfikatorów i operatorów:

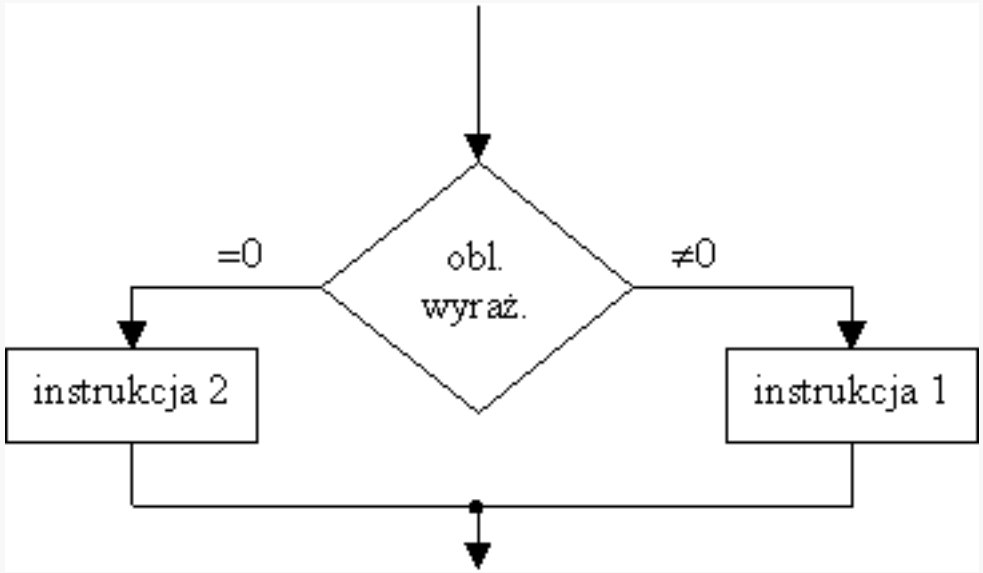
```
y=x*x;
i++;
```

- Instrukcja złożona powstaje po zgrupowaniu instrukcji prostych przy użyciu nawiasów klamrowych:

```
{//wymiana wartości a i b
    c=a;
    a=b;
    b=c;
} //nie występuje średnik
```

Instrukcja warunkowa

```
if (<wyrażenie>) <instr_1> else <instr_2>;
```



można stosować
skróconą instrukcję
bez else

Wynikiem <wyrażenia> jest wartość logiczna:

<prawda> ⇨ *<true>* ⇨ „wartość nie zerowa”

<nieprawda> ⇨ *<false>* ⇨ „wartość zerowa”

Operator warunkowy

Język C udostępnia skrótowy sposób wyrażenia jednego z wariantów instrukcji if else:

```
<wyrażenie_1>?<wyrażenie_2>:<wyrażenie_3>;
```

Przykłady:

```
x = (y<0) ? -y : y;
```

Zmiennej x przypisywana wartość bezwzględna y

```
printf("Masz %d %s !\n", IleDni, IleDni==1 ?
"dzień" : "dni");
```

Końcówka zdania dopasowana do sytuacji

Instrukcja przełączająca

```
switch (<wyrażenie>)
{       case <stała 1>:
                { <instr_złożona_1>
        break;}
        case <stała 2>:
                { <instr_złożona_2>
        break;}
        case <stała 3>:
                { <instr_złożona_3>
        break;}
        ...
        default:
                { <instr_złożona_0> }
}//Instrukcja przełącza instrukcje
```

Wynik wyrażenia jest elementem listy numerowanej.

Jeśli break opuścić to wykonywane są wszystkie case od wchodzenia i do końca.

Odpowiada wynikowi nie przypisanemu do żadnego case.

Instrukcja pętli while

```
while (<wyrażenie>) <instrukcja_złożona>
```

- Instrukcja jest powtarzana dopóki wartość wyrażenia jest prawdą.
- Łatwo tworzy się pętle sterowane przez użytkownika.
- Jest to często pętla nieokreślona bo liczby wykonanych powtórzeń nie można przewidzieć.

```
np. while (num==1) {suma=suma++;
                    num=scanf("%ld", &num)}
```

Instrukcja pętli for

Dowolne wyrażenie można pominąć, ale średnik należy pozostawić

```
for(<wyraż_1>;<wyraż_2>;<wyraż_3>) <instr.złożona>
```

Inicjalizacja licznika

warunek

aktualizacja zmiennych

- Stosuje się do tworzenia pętli liczących, w których liczba powtórzeń z góry określona.
- Elastyczność pętli ma swoje źródło we wszechstronności trzech wyrażeń sterujących.
- Operator przecinkowy < , > umożliwia umieszczenie kilku wyrażeń w obrębie jednego wyrażenia sterującego.

Instrukcja pętli do while

- Pętle while i for mają warunek wejścia, który sprawdza się przed każdym powtórzeniem pętli.
- Pętla do while sprawdza warunek wyjścia (exit condition) po każdej iteracji, co gwarantuje przynajmniej jedno wykonanie zawartości pętli.

Przykład:

```
do
{
   scanf("%c", &znak);
   printf("%c", znak);
}
while (znak != '$');
```

Instrukcje przerwania i skoku

- Instrukcja przerwania break powoduje "wyskok" z pętli, a także z instrukcji switch.
  - Jeżeli jest umieszczona w pętli zagnieżdżonej, to wyskok jest realizowany tylko dla najbardziej zagnieżdżonej pętli.
- Instrukcja skoku continue powoduje skok do końca pętli.
  - Pętla jest w dalszym ciągu wykonywana.
- Instrukcja skoku goto powoduje skok do instrukcji poprzedzonej etykietą.
  - Skok może być realizowany tylko wewnątrz funkcji