03

Języki programowania 1

Podstawy programowania w C

Zakres:

- stałe i zmienne,
- wyrażenia,
- instrukcje,
- bloki instrukcji,
- typy i nazwy funkcji w języku C,
- argumenty funkcji,
- ciało funkcji,
- wywołanie funkcji.

Podstawowe elementy programów w C

Podobnie jak każda złożona część składa się z poszczególnych elementów, tak i struktura języka C składa się z danych wyrażeń, instrukcji, bloków instrukcji i bloków funkcji.

Stałe i zmienne

Stała: jak wynika z jej samej nazwy, nigdy nie zmienia swojej wartości. Przykładem mogą być poszczególne liczby, które zawsze mają swoją (określoną) wartość.

Zmienna: może być użyta do reprezentowania różnych wartości.

Np.

i=4; (4 jest stałą; wartość 4 przypisujemy do i)

i=8; (8 jest stałą; wartość 8 przypisujemy znów do i; widać, że i jest zmienną.

Wyrażenia

Wyrażenie to kombinacja stałych, zmiennych i operatorów stosowanych do zapisu operacji matematycznych.
Przykładowo:
(3+2)*6
wyrażenie, które w wyniku da 30, a składa się z dodania dwóch stałych i następnie pomnożenia przez 6.

Wyrażenie	Opis
5	Stała
i	Zmienna
5+i	Dodawanie stałej do zmiennej
exit(0)	Wyrażenie – wywołanie funkcji

Operatory arytmetyczne

W wyrażeniach mogą występować następujące symbole: +, -, *, /, %.
W języku C symbole te nazwane są operatorami arytmetycznymi.

Symbol	Znaczenie
+	Dodawanie
-	Odejmowanie
*	Mnożenie
/	Dzielenie
%	Reszta z dzielenia (6%4=2)

Pośród wszystkich operatorów arytmetycznych:

- * i / mają najwyższy priorytet,
- + i – mają najniższy priorytet.

Przykładowo:
2+3*10
będzie 32 (a nie 50 !!!)

Identyfikatory

Oprócz stałych numerycznych oraz operatorów matematycznych język C może zawierać również słowa (łańcuchy znaków alfanumerycznych), które nazywają się identyfikatorami. Identyfikatorami są:

- nazwy funkcji bibliotecznych (np. exit),
- nazwy zmiennych (np. i, dana),
- słowa kluczowe języka C.

Do tworzenia identyfikatorów dozwolone jest korzystanie z następujących znaków alfanumerycznych:

- litery A…Z oraz a…z,
- cyfry 0…9 (cyfra nie może być pierwszym znakiem identyfikatora),
- znak podkreślenia _.

Znaki zakazane, których niw można używać w obrębie identyfikatorów:

- znaki operatorów arytmetycznych: +, -, /, *, %,
- kropka .,
- apostrof i cudzysłów ‘ ”,
- pozostałe znaki specjalne: @, $, #, ?, !, itp.

Instrukcje

W języku C instrukcja to polecenie zakończone średnikiem. W wielu przypadkach wyrażenie jest już instrukcją. Przykładowo:

```
i=(2+3)*10;
dana=i+2*10;
return 0;
exit(0); etc.
```

Bloki instrukcji

Grupa instrukcji może tworzyć blok instrukcji, który zaczyna się od nawiasu klamrowego { klamrowego kończy się nawiasem klamrowym}. Taki blok traktowany jest przez kompilator jak pojedyncza instrukcja. Przykładowo:

```
for(……)
{
 dana=x1+x2;
 suma=dana-10;
 wynik=suma%2;
}
```

Blok instrukcji to sposób połączenia wielu instrukcji prostych tak, by zawsze były wykonywane razem, jak jedna instrukcja.

Funkcja w języku C

O funkcjach w języku C można powiedzieć, że są to takie poszczególne elementy, z których buduje się całe programy. Prócz standardowych funkcji

bibliotecznych w programach można budować własne funkcje. Standardowo każda funkcja składa się z sześciu części:

- typu funkcji,
- nazwy funkcji,
- argumentów funkcji,
- nawiasu otwierającego,
- ciała funkcji,
- nawiasu zamykającego.

Przykładowo:

```
int dodawanie(int x, int y)
{
 int wynik;
 wynik=x+y;
 return wynik;
}
```

Typ funkcji określa rodzaj wartości (wyniku), który funkcja zamierza zwrócić po wykonaniu się. W języku C istniej kilka typów danych (o nich może trochę później). W ww. przykładzie dana funkcja zwraca wartość typu int (integer, całkowity). Rezultat zwracany przez funkcję może być wykorzystywany w programie, bez konieczności zapamiętywania tej zmiennej. Przykładowo:

```
int x;  /*deklaracja zmiennej*/
x=funkcja(…); /*wartość zwrócona przez daną funkcję została przypisana do
                zmiennej x*/
lub też
x=(funkcja(…)+5)*6;
```

Nazwa funkcji powinna zostać nadawana w sposób ułatwiający jej identyfikację. Z racji tego, że funkcja jest identyfikatorem, zatem jej nazwy też powinny spełniać wymagania identyfikatorów. Należy również pamiętać o tym, że różnym funkcją nie wolno nadawać tej samej nazwy.

Argumenty przekazywane do funkcji są pewnymi danymi, które przekazywane są do funkcji w celu prawidłowego jej wykonania. Argumenty umieszczane są w nawiasach okrągłych tuż za nazwą funkcji. Ilość argumentów w funkcji nie istotna i zależy tylko od tego ile dana funkcja będzie ich potrzebować. W przypadku większej ilości argumentów oddzielamy je przecinkami. Jeśli funkcja nie wymaga żadnych argumentów to za nazwą funkcji pojawia się tylko para nawiasów.

Początek i **koniec** funkcji oznacza się poprzez parę nawiasów klamrowych {…}.

Ciało funkcji to miejsce w funkcji, w którym znajdują się deklaracje zmiennych i instrukcje języka C. Dla poprawności składni języka C należy pamiętać o tym, że na samym początku należy zadeklarować wszystkie zmienne wykorzystywane w funkcji (w C++ nie trzeba).

Przykładowy program z wykorzystanie funkcji, *funkcja.c.*
