



Nazwa implementacji: Nauka języka C – zmienne

Autor: Piotr Fiorek

Opis implementacji: Poznanie struktury programu, pojęcia zmiennych oraz podstawowych typów danych, jakie zmienne mogą przybierać.

Wstęp

Język C jest jednym z najstarszych używanych języków programowania i jednym z najpopularniejszych. Powstał w 1972 w Bell Laboratories na potrzeby nowo tworzonego systemu UNIX. Dzięki temu C nadaje się praktycznie do każdego zastosowania - od tworzenia systemów operacyjnych, aż po zadania multimedialne. Warto poznać ten język, ponieważ na C opiera się większość nowoczesnych języków oraz dlatego, że dobre poznanie C pozwala lepiej zrozumieć jak działa komputer oraz programy.

Implementacja

Programy operują na danych. Tak naprawdę programowanie nie jest niczym innym niż tylko przetwarzaniem danych według określonych schematów nazywanych algorytmami. Dlatego właśnie naukę C zaczniemy od poznania i omówienia typów zmiennych, których C są tylko cztery: char, int, float i double. Pierwsze dwie służą do przechowywania liczb całkowitych, a pozostałe dwie do danych zmiennoprzecinkowych, czyli z ułamkiem. W ramach par typy te różnią się wielkością.

char (od angielskiego „character” czyli „znak”) jest najmniejszym typem danych i zajmuje tylko jeden bajt, czyli osiem bitów. Pozwala to na przechowywanie zmiennych o wartościach od -128 do 127 lub jeśli użyjemy zmiennej bez znaku od 0 do 255.

int

(od angielskiego „integer” czyli „liczba całkowita”) jest trochę większy i z reguły ma 4 bajty, co pozwala na przechowywanie wartości od -2,147,483,648 do 2,147,483,647 lub od 0 do 4,294,967,295 co z reguły wystarcza do większości zastosowań.

Zmienne typu float są zmiennoprzecinkowe pojedynczej precyzji, a double, jak nazwa sugeruje, podwójnej. W praktyce oznacza to, że double pozwala zapisywać zmienne ułamkowe z większą precyzją.

Nazwy zmiennych mogą się składać z liter (ich wielkość ma znaczenie) i cyfr. Zanim zaczniemy używać zmiennej trzeba ją najpierw zadeklarować. Wygląda to tak:

```
int zmienna1;
```

Taki zapis oznacza, że stworzyliśmy zmienną o nazwie „zmienna1” i że ta zmienna ma być typu całkowitego. Średnik na końcu jest konieczny i oznacza koniec instrukcji. Można też w jednej linii zadeklarować kilka zmiennych, oddzielając je od siebie przecinkami. Wszystkie zmienne zadeklarowane razem są tego samego typu:

```
int zmienna1, zmienna2, x, Y, z;
```





Zmiennym można też nadawać wartości od razu podczas deklaracji:

```
int zmienna1=6, zmienna2=-2, x, Y, z=42;  
  
float i=1.5, j=0.0, pi=3.1415;
```

Przypisanie zmiennym wartości początkowej jest bardzo dobrym pomysłem, ponieważ w C zmienne nie mają określonej wartości początkowej. Zmienna zaraz po utworzeniu ma taką wartość jak dane, które były w pamięci w miejscu, gdzie system operacyjny przeznaczył na tą zmienną.

Te wszystkie zmienne są zmiennymi typu „signed” czyli ze znakiem (mogą przyjmować zarówno wartości dodatnie i ujemne) ponieważ jest to domyślny sposób deklarowania zmiennych. Jednak jeśli wiemy, że jakaś zmienna będzie przyjmowała tylko wartości dodatnie i chcemy to wykorzystać, żeby zwiększyć jej pojemność, to możemy zmienną zadeklarować jako zmienną bez znaku. W tym celu nazwę typu poprzedzamy słowem „unsigned”:

```
unsigned int zmienna1=0;  
  
int zmienna2=-2, x, Y, z=42;
```

Na zmiennych można wykonywać standardowe operacje arytmetyczne jak dodawanie, odejmowanie, mnożenie, dzielenie i inne:

```
int zmienna1=0, zmienna2, zmienna3, x=3, Y=4, z=42;  
  
zmienna1 = x + Y;  
zmienna2 = z - x;  
zmienna3 = Y * z;
```

Skoro już poznaliśmy typy zmiennych oraz podstawowe operacje na nich czas na pierwszy program:

```
#include <stdio.h>  
  
int main(void)  
{  
  
int zmienna1=42, zmienna2=54, dodawanie, odejmowanie;  
  
dodawanie = zmienna1 + zmienna2;
```





```
odejmowanie = zmienna2 - zmienna1;

printf("Wynik dodawania to %d\na odejmowania to %d\n", dodawanie, odejmowanie);

return 0;

}
```

Program zapisujemy przy użyciu edytora tekstu w pliku o dowolnej nazwie koniecznie z rozszerzeniem „.c”. Po zapisaniu programu musimy go jeszcze skompilować, czyli kazać komputerowi przetłumaczyć go z kodu, który napisaliśmy na formę zrozumiałą przez procesor. Programy w C kompiluje się w konsoli za pomocą polecenia:

```
gcc <nazwa-pliku> -o program1
```

a skompilowany program wykonujemy, wpisując w konsoli:

```
./program1
```

Napisałmy pierwszy działający program - czas omówić co w zasadzie napisaliśmy. Nasz program zaczyna się od instrukcji „#include”, która mówi kompilatorowi, że do naszego programu ma być dołączony inny plik. Nazwa tego pliku jest podana jako parametr instrukcji „#include”. Zapisujemy ją pomiędzy znakami „<”, „>” co oznacza, że kompilator ma szukać pliku w standardowym miejscu w systemie.

Kolejna linia z kodem to deklaracja głównej funkcji programu. Każdy program w C musi zawierać dokładnie raz funkcję „main”, ponieważ to od niej zawsze zaczyna się wykonanie programu. Co oznaczają instrukcje w tej linii, wytłumaczone będzie w rozdziale poświęconym funkcjom. Na razie po prostu zapamiętajcie, że w każdym programie kod musi być zapisany w funkcji, która tak się zaczyna.

Klamra w następnej linii to klamra otwierająca treść funkcji, czyli to, co nasz program ma zrobić. W C bloki kodu jak np. treść funkcji, czy pętli zapisuje się pomiędzy dwoma klamrami „{” i „}”. Klamry zawsze muszą występować parami, czyli dla każdej klamry otwierającej musi być klamra zamykająca blok kodu.

Kolejne linie to deklaracja zmiennych oraz przypisanie wartości do utworzonych zmiennych.

Następnie widzimy wywołanie funkcji „printf”. Funkcja ta pochodzi z pliku, który załączaliśmy w pierwszej linii naszego kodu i jest to funkcja służąca do wypisywania rzeczy na ekranie. Parametry, czyli dane, na których funkcje mają operować, przekazuje się wewnątrz nawiasów po nazwie funkcji i oddziela od siebie przecinkami. W naszym programie do funkcji „printf” podajemy trzy parametry. Pierwszym jest tzw. łańcuch znakowy, czyli po prostu napis, który zawiera tekst do wyświetlenia oraz kilka magicznych znaków specjalnych, a pozostałe dwa parametry to zmienne, których wartości chcemy wyświetlić.





Magiczne znaki w naszym napisie to „%d” i „\n”. Pierwszy mówi funkcji „printf”, że w tym miejscu należy wstawić wartości zmiennych (po kolei tak jak występują po napisie), a drugi, że w tym miejscu należy przejść do nowej linii. Jeśli uruchomicie program, to zobaczycie, że dokładnie tak to działa i, że znaki specjalne zostały zastąpione. Do drukowania zmiennych typu float służy inna kombinacja znaków – „%f”.

Przedostatnia linia w programie to wartość, którą funkcja zwraca na koniec działania. To również zostanie szerzej omówione w rozdziale o funkcjach. Zwyczaj mówi, że funkcja „main” zwraca na koniec zero, jeśli nie wydarzyło się nic niezwykłego.

Ostatnia linia to klamra zamykająca nasz program.

