Nazwa implementacji: Nauka języka C – Struktury i unie

Autor: Piotr Fiorek

Opis implementacji: Nauka tworzenia oraz praktycznego korzystania ze struktur oraz unii

Czasami zdarza się, że chcemy zgrupować razem zmienne, ponieważ, aby opisać jakiś pomysł, potrzebne jest więcej danych, niż może pomieścić jedna zmienna. W takich sytuacjach używa się struktur, czyli właśnie kontenerów grupujących razem kilka zmiennych.

Deklaracja struktury wygląda tak:

```
struct nazwa-struktury

{

zmienna1;

zmienna2;

...

} nazwa;
```

Ostatni parametr „nazwa” jest opcjonalny i z reguły pomijany. Jego zdefiniowanie tworzy zmienna typu naszej struktury o nazwie „nazwa”.

Stwórzmy sobie przykładową strukturę do przetrzymywania danych o psie:

```
struct pies

{

char rasa[100];

char imie[100];

char masc[100];

int wiek;

char plec;

};
```

Dzięki takiej strukturze możemy opisać psa. Zawiera ona tablice do przetrzymywania napisów definiujących imię, rasę i maść, zamienna liczbową na wiek oraz zmienną znakową na płeć: p – pies, s – suczka. Mając taką strukturę, napiszmy program, który będzie pobierał od użytkownika dane o psie, a następnie wyświetlał je na ekranie:

```
#include <stdio.h>

struct pies

1
```

```
{
char rasa[100];
char imie[100];
char masc[100];
int wiek;
char plec;
};
int main(void)
{
struct pies piesek;
printf("Podaj imie psa: ");
scanf("%s", piesek.imie);
printf("Podaj rase: ");
scanf("%s", piesek.rasa);
printf("Jakiej masci jest pies: ");
scanf("%s", piesek.masc);
printf("Ile lat ma pies: ");
scanf("%d", &piesek.wiek);
printf("Jakie plci jest pies - (p)ies lub (s)uczka: ");
getchar();
scanf("%c", &piesek.plec);
if(piesek.plec == 'p')
{
printf("Pies ma na imie %s. Jest rasy %s, w kolorze %s i ma %d lat(a).\n", piesek.imie, piesek.rasa, piesek.masc, piesek.wiek
}
else
{
printf("Suczka ma na imie %s. Jest rasy %s, w kolorze %s i ma %d lat(a).\n", piesek.imie, piesek.rasa, piesek.masc, piesek.wie
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
}

return 0;

}
```

Sam program powinien być bardzo prosty do zrozumienia. Jedyne co jest nowe to nasza struktura. Na początku definiujemy ją tak jak wcześniej. Później w funkcji „main” tworzymy zmienną „piesek”, która jest typu naszej nowo stworzonej struktury – „struct pies”. Do elementów struktury zawsze odwołujemy się poprzez ”nazwa-zmiennej.pole-w-strukturze”.

Struktura po zdefiniowaniu staje się typem zmiennej tak samo jak „int” czy „char”, a więc można też deklarować tablice o typie struktur. Deklaracja takiej tablicy wygląda tak samo jak deklaracja normalnej tablicy:

```
struct pies tablica[10];

a np. odwołanie się do pola „imie” z trzeciego elementu:

tablica[2].imie;
```

Struktury są bardzo często wykorzystywaną konstrukcją, ponieważ pozwalają porządkować program, aby był logicznie bardziej przejrzysty, dzięki czemu łatwiej uniknąć błędów.

Przy okazji struktur warto też powiedzieć o uniach. Jest to konstrukcja rzadko używana, ale i tak warto poznać jak działa. Unie definiuje się tak samo jak struktury, tylko zamiast słowa kluczowego

„struct”

używamy

„union”

. Różnica między unią a strukturą polega na tym, że mimo iż unia tak samo jak struktura ma zdefiniowane wiele pól, to pozwala na użycie tylko jednego naraz. W ten sposób, jeśli nie wiemy jaka zmienna będzie potrzebna, a chcemy oszczędzić miejsce zajmowane przez nasz program w pamięci operacyjnej (miejsce zajmowane przez unie jest równe wielkości jej największego elementu) skorzystamy właśnie z unii. Przykład może wyglądać tak:

```
#include <stdio.h>

union liczba

{

int calkowita;

float zmiennoprzecinkowa;

};

int main(void)

{

union liczba zmienna;
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
FWIOO
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
char wybierz;

printf("Jakiego typu zmienna chcesz wykorzystac? (c)alkowita / (z)miennoprzecinkowa: ");

scanf("%c", &wybierz);

printf("Podaj liczbe: ");

if(wybierz == 'c')

{

scanf("%d", &zmienna.calkowita);

}

else

{

scanf("%f", &zmienna.zmiennoprzecinkowa);

}

return 0;

}
```

Po przypisaniu wartości do jednej ze zmiennych w unii, pozostałe stają się niedostępne i odwołanie się do nich spowoduje błąd.

Zadania:

Stwórz strukturę opisującą człowieka, wypełnij ją danymi i wyświetl.

Stwórz struktury do zaimplementowania listy łączonej oraz drzewa binarnego.

- **Napisz program który wyświetli bieżącą datę i czas. Użyj funkcji time (man 2 time) która pobiera bieżący czas, a następnie użyj funkcji localtime (man 3 localtime) która wypełnia strukturę reprezentującą czas w formie zrozumiałej dla ludzi.**

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.