

Údajový typ štruktúra

Predstavte si, že by sme potrebovali spraviť úplne nový typ. Napríklad premennú, do ktorej by sme vedeli uložiť meno, priezvisko, vek, váhu a výšku človeka. T.j potrebovali by sme skupinu pomenovaných premenných rôzneho typu, ku ktorým by sme pristupovali ako k celku. Takéto premenné sa vytvárajú pomocou štruktúry (struct).

Vytvorenie štruktúry sa robí 2 spôsobmi.

1. spôsob

```
struct názov_štruktúry_nepovinné
{
    typ_premenná1;
    ...
} struk_prem1_nepovinné, ...;
```

Definovanie premenných typu štruktúra:

```
struct názov_štruktúry struk_prem1, ...;
```

Vyzerá to napríklad takto:

Vytvoríme si štruktúru s názvom **clovek**, ktorá má 5 položiek: meno, priezvisko, vek, vaha, vyska.

```
struct clovek
{
    char meno[20];
    char priezvisko[20];
    int vek;
    int vaha;
    float vyska;
};
```

Na inom mieste v programe si vytvoríme premennú **ziak**, ktorá bude typu **struct clovek**, teda bude mať všetkých 5 položiek:

```
struct clovek ziak;
```

Jednotlivé položky štruktúry sú sprístupňované prostredníctvom operátora **•** (**bodka**).

Ak sa chceme dostať k položke **vaha** premennej **ziak**, napíšeme **ziak.vaha**, ak sa chceme dostať k položke **meno** premennej **ziak**, napíšeme **ziak.meno**, atď.

Použitie v programe:

```
#include <stdio.h>
#include <string.h>

struct clovek
{
    char meno[20];
    char priezvisko[20];
    int vek;
    int vaha;
    float vyska;
};

int main()
{
    struct clovek ziak;

    printf("Zadaj meno: ");
    scanf("%s", ziak.meno);
    printf("Zadaj priezvisko: ");
    scanf("%s", ziak.priezvisko);
    printf("Zadaj vek: ");
    scanf("%d", &ziak.vek);
    printf("Zadaj hmotnosť: ");
    scanf("%d", &ziak.vaha);
    printf("Zadaj výšku: ");
    scanf("%f", &ziak.vyska);

    printf("%s %s Vek:%d Vaha:%d Vyska:%g\n", ziak.meno, ziak.priezvisko, ziak.vek, ziak.vaha, ziak.vyska);
}
```

Dúfam, že si pamätáte, že reálne čísla používajú desatinnú bodku, nie čiarku.

2. spôsob

2. spôsob je spojený s vytvorením nového typu pomocou **typedef**

```
typedef struct název_struktury_nepovinný
{
    typ_proměnná1;
    ...
} NÁZEV_TYPU;
```

Je nepísané pravidlo, že keď vytvárame nový vlastný typ, do názvu dáme **_t**. Predchádzajúci príklad teda môžeme zapísať:

```
typedef struct clovek
{
    char meno[20];
    char priezvisko[20];
    int vek;
    int vaha;
    float vyska;
} CLOVEK_t;
```

Na inom mieste v programe si vytvoríme premennú **ziak**:

```
CLOVEK_t ziak;
```

K položkám štruktúry prístupujeme ako predtým pomocou operátora **•** (**bodka**).

Použitie v programe:

```
#include <stdio.h>
#include <string.h>

typedef struct clovek
{
    char meno[20];
    char priezvisko[20];
    int vek;
    int vaha;
    float vyska;
} CLOVEK_t;

int main()
{
    CLOVEK_t ziak;

    printf("Zadaj meno: ");
    scanf("%s", ziak.meno);           //polozka meno je pole, pole bez[] je pointer a teda napiseme &
    printf("Zadaj priezvisko: ");
    scanf("%s", ziak.priezvisko);     //polozka priezvisko je pole, pole bez[] je pointer a teda napiseme &
    printf("Zadaj vek: ");
    scanf("%d", &ziak.vek);
    printf("Zadaj hmotnost: ");
    scanf("%d", &ziak.vaha);
    printf("Zadaj vysku: ");
    scanf("%f", &ziak.vyska);

    printf("%s %s Vek:%d Vaha:%d Vyska:%g\n", ziak.meno, ziak.priezvisko, ziak.vek, ziak.vaha, ziak.vyska);
}
```

Po vyskúšaní vidíte, že oba programy pracujú rovnako.

Iný príklad:

Vytvorme štruktúru zodpovedajúcu bodu súradnicovej sústavy, ktorý ma položky X-súradnicu a y-súradnicu. Naplňte pole 3 bodov hodnotami a vypíšte ho na obrazovku, potom zmeňte súradnice 1 bodu a opäť vypíšte body na obrazovku.

```
#include <stdio.h>
typedef struct bod
{
    int x;
    int y;
} SURADNICE_t;

int main()
{
    SURADNICE_t bod[3] = {{1, 1}, {3, 3}, {7, 5}};
    int i;
    printf("Zadane body:\n");
    for (i = 0; i < 3; i++)
        printf("Bod [%d] ma suradnice [%d,%d]\n", i, bod[i].x, bod[i].y);
    bod[0].x = 30;
    bod[0].y = 20;

    printf("Body po zmene:\n");
    for (i = 0; i < 3; i++)
        printf("Bod [%d] ma suradnice [%d,%d]\n", i, bod[i].x, bod[i].y);
}
```

K jednotlivým položkám štruktúry sa opäť pristupuje s pomocou bodky. Celé pole môžeme naplniť (ale len pri deklarácii) spôsobom, ktorý je vidieť vyššie. Mimo deklarácie môžeme pole naplniť len cez cyklus.

V prípade, že máme pointer na nejakú štruktúru, môžeme k jednotlivým položkám pristupovať cez tento pointer. V takom prípade sa nepoužíva na sprístupnenie položiek bodka, ale šípka **->**, zložená zo znakov „mínus“ a „väčšie“.

Keďže vieme, že názov poľa bez hranatých zátvoriek je pointer, ktorý ukazuje na prvú položku poľa (teda na začiatok poľa), môžeme predošlý program napísať trochu iným spôsobom:

```
#include <stdio.h>
typedef struct bod
{
    int x;
    int y;
} SURADNICE_t;

int main()
{
    SURADNICE_t bod[3] = {{1, 1}, {3, 3}, {7, 5}};
    int i;
    printf("Zadane body:\n");
    for (i = 0; i < 3; i++)
        printf("Bod [%d] ma suradnice [%d,%d]\n", i, (bod+i)->x, (bod+i)->y);
    bod->x = 30;    // (bod+0)->x
    bod->y = 20;    // (bod+0)->y

    printf("Body po zmene:\n");
    for (i = 0; i < 3; i++)
        printf("Bod [%d] ma suradnice [%d,%d]\n", i, (bod+i)->x, (bod+i)->y);
}
```

Pre zopakovanie:

```
int a[10];
```

a[i] zapíšeme cez pointery ***(a+i)**

a[3] zapíšeme cez pointery ***(a+3)**

a[0] zapíšeme cez pointery ***a**

teda, ak:

```
SURADNICE_t a[10];
```

a[i].x zapíšeme cez pointer **(a+i)->x**

a[i].y zapíšeme cez pointer **(a+i)->y**

a[3].x zapíšeme cez pointer **(a+3)->x**

a[0].x zapíšeme cez pointer **a->x** alebo **(a+0)->x**

Zadanie č. 1:

Všetky programy si prepíšete do vývojového prostredia, ktoré používate napr. CodeBlocks a otestujete, či pracujú správne.

Zadanie č. 2 verzia a:

- Vytvorte 2. spôsobom nový typ, ktorý bude obsahovať značku, model a rok výroby auta. (napr.: značka = skoda, model = fabia, rok = 2008)
- Vytvorte pole 6 áut a naplňte ho (spôsob naplnenia si môžete vybrať).
- Vypíšte údaje o prvých troch autách priamo s použitím hranatých zátvoriek.
- Vypíšte údaje o druhých troch autách cez pointery (bez hranatých zátvoriek).

Zadanie č. 2 verzia b:

- Vytvorte 2. spôsobom nový typ, ktorý bude obsahovať názov tovaru, kód tovaru a cenu tovaru. (napr.: názov = kreslo otoman, kód = 1235897, cena = 45.00).
- Vytvorte pole 6 tovarov a naplňte ho (spôsob naplnenia si môžete vybrať).
- Vypíšte údaje o prvých troch tovaroch priamo s použitím hranatých zátvoriek.
- Vypíšte údaje o druhých troch tovaroch cez pointery (bez hranatých zátvoriek).