

UPR

Cvičení 6

Struktury

- Známe datové typy pro uložení základních datových typů (celá čísla, desetinná čísla, znaky, řetězce)
- Představme si, že vytváříme hru, kde si budeme chtít pamatovat žebříček nejlepších hráčů. Pro každého hráče si tedy budeme pamatovat

```
char* name;
```

```
int high_score;
```

- Co když máme víc hráčů?

Struktury

```
char* name1;  
int high_score1;  
char* name2;  
int high_score2;  
...
```

Nebo pole?

- Museli bychom vytvořit zvlášť pole pro jména a pole pro skóre
- Ke každému hráči navíc přidáme i věk – opět je třeba vytvořit nové pole

Struktury

- Řešení: struktura
 - Seskupíme více datových typů a vytvoříme tak nový datový typ

```
struct Player {  
    char* name;  
    int high_score;  
};
```

a ve funkci `main` vytvoříme proměnnou typu `Player` jedním ze způsobů:

```
struct Player player;  
struct Player player = {"Joe", 5};  
struct Player player = {.name="Joe", .high_score=5};
```

Struktury

- Přístup k atributům pomocí tečky

```
player.name = "Joe";
```

```
player.high_score = 10;
```

- Můžeme předat jako parametr funkce

```
void player_info(struct Player p) {...}
```

- Je zvykem psát funkce ve tvaru:

```
nazevStruktury_nazevFunkce
```

Struktury

- Úkol č.1

Vytvořte funkci `player_new_score`, která nastaví nové maximální skóre u hráče, pokud je větší než aktuální skóre hráče

Struktury

- Předání struktury do funkce ukazatelem
- Zápis `(*p).high_score` lze nahradit `p->high_score`
- Je nutné psát všude `struct Player` ?
 - alias pro datový typ:

```
typedef <datovy typ> <alias>
```

```
typedef struct Player Player
```

Struktury

- Úkol č.2

Vytvořte pole hráčů, naplňte ho daty a vypište hráče s nejvyšším skóre.


Třídění

- Chceme vytvořit žebříček hráčů
- Potřebujeme seřadit hráče podle skóre od největšího po nejmenší
 - Můžeme si napsat vlastní algoritmus
 - Nebo použít funkci `qsort` (součást `stdlib.h`)

Třídění

```
void qsort (void* base, size_t num,  
           size_t size,  
           int (*compar) (const void*, const void*));
```

```
qsort (pole, pocet_prvku, velikost_prvku,  
      porovnavaci_funkce);
```



Tato funkce porovná dva prvky a na základě toho, co vrátí (kladné nebo záporné číslo) tyto prvky seřadí

Struktury + třídění

- Úkol č.3

Vytvořte pole hráčů, naplňte ho daty a vypište hráče seřazené podle skóre od největšího po nejmenší