

PJC
Cvičení #3

Strukturované datové typy

Strukturované datové typy I.

- Slouží k agregaci různých dat a různých funkcí do jednoho objektu
- Neznámější a nepoužívanější jsou `class` a `struct`
- Oba typy se liší pouze ve výchozích přístupových právech
- `struct` je sice převzaté z jazyka C, ale byla přidána další funkčnost
- `struct` se z historických důvodů používá většinou jen pro data (viz další sekce slidů)

Strukturované datové typy II.

- Datové složky jsou v `class` nebo `struct` uloženy přímo za sebou
- Použijeme-li na proměnnou typu `struct` a `class` operátor `sizeof()`, vrátí nám součet velikostí všech datových složek
- Jaké to má výhody?
- Kód metod je v přeložené aplikaci uložen zvlášť, nejčastěji společně s kódem funkcí
- Kód metod se za běhu programu vyskytuje jen jednou bez ohledu na počet instancí objektu `class/struct`
- Ve zkompilovaném programu se metody a funkce příliš neliší

Struktura v jazyce C

- Struktura se v jazyce C používá jinak než v C++
- Neobsahuje metody, pouze data
- Deklaruje se stejně, jen je nutné před jejím názvem používat stále klíčové slovo struct

```
struct osoba_t {  
    int vek;  
    char jmeno[20]; }  
  
struct osoba_t osoba; // osoba_t osoba;  
osoba.vek = 25;  
strcpy(osoba.jmeno, "Pepa z depa");
```

Deklarace typedef

- Klíčovým slovem `typedef` zavádíme nové pojmenování pro datový typ
- Slouží především pro zpřehlednění kódu:

```
typedef struct osoba_t {  
    int vek;  
    char jmeno[20]; } Osoba;  
Osoba osoba;  
  
typedef map<int *, vector<int *> > Table;  
// map<int *, vector<int *> > table;  
Table table;
```

Historické datové typy

- Jazyk C++ je zdědil z jazyka C
- Příliš se nepoužívají, ale moderní překladače je stále tolerují
- Jedná se o unii a bitové pole

Unie – union I.

- Deklaruje se klíčovým slovem `union`
- Obsahuje více datových složek, ale v jeden okamžik může být použita pouze jedna z nich
- Vznikla kvůli úspoře paměti, datové složky jsou v paměti uloženy “přes sebe”
- Její velikost v paměti je dána největším uloženým prvkem
- V C++ byla rozšířena o metody, ale oproti třídám má značné omezení

Unie – union II.

- Ukázka:

```
union Soubor {  
    char jmeno_souboru_dos[12];  
    char jmeno_souboru_unix[50];  
}  
  
...  
  
Soubor soubor;  
  
char jmeno_souboru[50];  
  
cin >> jmeno_souboru;  
  
if (strlen(jmeno_souboru) <= 12) {  
    strcpy(soubor.jmeno_souboru_dos, jmeno_souboru);  
}  
else {  
    strcpy(soubor.jmeno_souboru_unix, jmeno_souboru);  
}
```


Bitové pole I.

- Primárně se příliš neliší od obyčejné struktury
- Slouží opět k úspoře paměti
- Jeho velikost je zaokrouhlována např. podle “bitového čísla” procesoru
- Každé datové složce určuje velikost v bitech
- Nelze na něj vytvořit odkaz

Bitové pole II.

- Ukázka:

```
struct datum {  
    int den_v_tydnu      :5;  
    int mesic            :4;  
    int rok              :12;  
}
```

Použité zdroje

- Jazyky C a C++ podle normy ANSI/ISO; D. Luis, P. Mejzlík, M. Virius, GRADA, 1999

Úloha 1

- Doplňte kód Uloha1.cpp tak, aby šel přeložit a aby fungoval dle instrukcí
- Instrukce jsou uvedeny přímo ve zdrojovém kódu

Domácí cvičení 1

- Rozdělte úlohu 1 do tří souborů:
 - main.cpp – obsahuje funkci main() a testovací kód
 - Number.h – obsahuje deklaraci třídy Number a jejích metod
 - Number.cpp – obsahuje kód metod třídy Number