Vyšší programovací jazyky

<small>Z ČWUT</small>

Obsah

Jednoduché a strukturované typy dat

Jednoduché - specifikují množiny hodnot, které jsou z hlediska operací dále nedělitelné. Operaci specifikuje tzv. *signatura*. Např. +: integer × integer → integer

- číselné typy - integer, real, ..
- výčtové typy - enum
- logické typy - boolean
- znakové typy - char

Strukturované - datové typy skládající se z dílčích datových objektů (složek, prvků)

- statické datové struktury - s pevným počtem složek
- dynamické datové struktury - s proměnným počtem složek (zásobník, fronta,..)

- homogenní datové struktury - pole
- heterogenní datové struktury - záznam

Operace nad datovými strukturami

- selektory - operace kterými se vybírají složky datové struktury
- operace s celou dat. struk.
- vložení / odebrání složky
- vytvoření / zrušení dat.struk.

Abstraktní datový typ (ADT) umožnuje skrýt implementaci jednotlivých operací nad daným datovým typem a způsob fyzického uložení informace v proměnné daného datového typu. Hlavním cílem je zjednodušit a zpřehlednit program, který provádí operace s daným datovým typem. ADT umožnuje vytvářet i složitější datové typy, např. operace s ADT typu zásobník, fronta a pole. Všechny ADT lze realizovat pomocí základních algoritmických operací (přiřazení, sčítání, násobení, podmíněný skok,...).

Ukazatele a dynamické proměnné

Ukazatel = datový typ, jehož hodnota identifikuje jiný datový typ. (hodnota = adresa)
Základní operace nad ukazateli:

- vytvoření (new, malloc)
- dereference
- přiřazení mezi ukazateli stejného doménového typu

- operace porovnání ukazatelů (=,!=)
- nulární operace (null)
- zrušení (free, delete)
- spojové seznamy

Signatura operací ukazatelové aritmetiky jazyka C:

- +: T* × int → T*
- -: T* × int → T*
- -: T* × T* → int //rozdíl ukazatelů je rozdíl indexů prvků pole, které tyto ukazatele identifikovaly

Dynamické proměnné
Rušení dynamických objektů:

- explicitně - programátorem
- automaticky = Garbage Colllection

Druhy GC

- počítání referencí
- značkování a zametání
- kopírování

Jednoduché a strukturované příkazy

Jednoduché

- příkazy pro manipulaci s datovými objekty - např. přiřazení hodnoty proměnné
- příkazy pro řízení výpočtu - např. goto

Strukturované

- složený příkaz = posloupnost příkazů v bloku
- podmínka - if-then-else
- přepínač - switch
- cykly..

Procedury a funkce

- funkce mají obvykle návratovou hodnotu, procedury ne (co já vím tak v Céčku třeba procedury nejsou)
- občas je výhodné členit program do funkcí a procedur
- funkce a procedury jsou dobré třeba pro rekurze
- při skoku programu do procedury (funkce) se ukládá na zásobník stav proměnných
- proměnné mohou být volány/předávány
    - odkazem (pokud změníme hodnotu proměnné uvnitř funkce, změní se i "venku")
    - nebo hodnotou (pokud změním hodnotu takové proměnné, při návratu bude opět načtena původní hodnota ze zásobníku)
- objektový programátoři jsou velmi hákliví na připodobňování metod a funkcí, podle mě je to totéž, pouze interpretace se liší

Bloková struktura programu

- ve mně známých jazycích se bloky ohraničují složenými závorkami { } nebo dvojicí begin - end
- deklarace provedené uvnitř bloku (třeba for, to je jedno) mimo blok ztrácí platnost
- bloky lze (neomezeně) vnořovat, přičemž věc deklarovaná v bloku je viditelná ve vnitřnějším bloku také

Modulární struktura programu

Modul = programová jednotka, která narozdíl od bloku umožňuje, aby v ní deklarované proměnné a metody byly použitelné v jiných programových jednotkách.
Skládají se ze dvou částí:

- specifikační - zde jsou veřejné deklarace (vyvážené)

- implementační (může obsahovat i neveřejné deklarace)

Zpracování výjimek

```
try {
   return null;
   // hlídaný blok
} catch( třídaVýjimek1 jménoProměnné1 ) {
   // ošetření výjimky
} catch( třídaVýjimek2 jménoProměnné2 ) {
   // ošetření výjimky
} finally {
   // tohle se provede vždy
   return "Bagr";
}
```

- Zvláštní postavení mezi výjimkami mají tzv. runtime výjimky - instance třídy RuntimeException a jejích potomků. Tyto výjimky nemusí být v metodách zachycovány ani deklarovány. (ArrayIndexOutOfBoundsException nebo ArithmeticException)
- Propagace výjimek se provádí (aspoň tedy v Javě) příkazem throw
- Blok finally se provede vždy -- příkládek tedy vrátí "Bagr", nikoliv null

Objektově orientované prostředky

Stránka byla naposledy editována v 20:20, 7. 2. 2007.
