O rekurzi. Typické ATD. Domácí úloha

Programovací techniky

doc. Ing. Jiří Rybička, Dr.
ústav informatiky
PEF MENDELU v Brně
rybicka@mendelu.cz

Iterace

Iterace

- Definice iterativní úlohy

Iterace

- Definice iterativní úlohy
- Vstup: řada čísel, výstup: součet vstupní řady.

$$S = a_1 + a_2 + \cdots + a_N$$

Iterace

- Definice iterativní úlohy
- Vstup: řada čísel, výstup: součet vstupní řady.

$$S = a_1 + a_2 + \cdots + a_N$$

- Řešení vede na použití cyklu:

```
S:=0;
while {není konec dat} do begin
   {získej hodnotu a}
   S:=S+a
end;
```

Iterace

- Definice iterativní úlohy
- Vstup: řada čísel, výstup: součet vstupní řady.

$$S = a_1 + a_2 + \cdots + a_N$$

- Řešení vede na použití cyklu:

```
S:=0;
while {není konec dat} do begin
   {získej hodnotu a}
   S:=S+a
end;
```

- Spotřeba paměti nezávisí na počtu vstupních dat

Rekurze

Rekurze

- Jiný způsob zadání téže úlohy:

$$\begin{aligned} S_i &= S_{i-1} + a_i \\ S_0 &= 0 \end{aligned}$$

Rekurze

- Jiný způsob zadání téže úlohy:

$$\begin{aligned} S_i &= S_{i-1} + a_i \\ S_0 &= 0 \end{aligned}$$

- Řešení vede na rekurzivní podprogram:

```
function S:real;
 var a: real;
 begin if {není konec dat} then begin
                read(a);
                S:=S+a
          end else S:=0
 end;
```

Vlastnosti rekurze

Vlastnosti rekurze

- Volání podprogramu – použití systémového zásobníku (návratové adresy, parametry volané hodnotou, lokální proměnné)

Vlastnosti rekurze

- Volání podprogramu – použití systémového zásobníku (návratové adresy, parametry volané hodnotou, lokální proměnné)
- Spotřeba paměti – závisí na počtu dat

Vlastnosti rekurze

- Volání podprogramu – použití systémového zásobníku (návratové adresy, parametry volané hodnotou, lokální proměnné)
- Spotřeba paměti – závisí na počtu dat
- Jiná úloha – obrácení posloupnosti vstupních řetězců:

```
procedure Obrat;
 var X: string;
 begin if not eof do begin
          readln(X);
          Obrat
       end;
       writeln(X)
 end;
```

Vlastnosti rekurze

- Volání podprogramu – použití systémového zásobníku (návratové adresy, parametry volané hodnotou, lokální proměnné)
- Spotřeba paměti – závisí na počtu dat
- Jiná úloha – obrácení posloupnosti vstupních řetězců:

```
procedure Obrat;
 var X: string;
 begin if not eof do begin
          readln(X);
          Obrat
       end;
       writeln(X)
 end;
```

- Objasnění činnosti – trasování a vykreslení stavu paměti

Význačné abstraktní datové typy

Význačné abstraktní datové typy

- Seznam

Význačné abstraktní datové typy

- Seznam
  - Fronta (Prioritní fronta)

Význačné abstraktní datové typy

- Seznam
  - Fronta (Prioritní fronta)
  - Zásobník

Význačné abstraktní datové typy

- Seznam
  - Fronta (Prioritní fronta)
  - Zásobník
- Stromové struktury

Význačné abstraktní datové typy

- Seznam
  - Fronta (Prioritní fronta)
  - Zásobník
- Stromové struktury
- Vyhledávací tabulka

Význačné abstraktní datové typy

- Seznam
  - Fronta (Prioritní fronta)
  - Zásobník
- Stromové struktury
- Vyhledávací tabulka
- Graf

Význačné abstraktní datové typy

- Seznam
  - Fronta (Prioritní fronta)
  - Zásobník
- Stromové struktury
- Vyhledávací tabulka
- Graf
- Řídké pole

Stromy

Stromy

- nepravidelné

Stromy

- nepravidelné
- pravidelné

Stromy

- nepravidelné
- pravidelné
  - n-ární

Stromy

- nepravidelné
- pravidelné
  - n-ární
  - ternární

Stromy

- nepravidelné
- pravidelné
  - n-ární
  - ternární
  - binární

Stromy

- nepravidelné
- pravidelné
  - n-ární
  - ternární
  - binární
  - binární uspořádané

Stromy

- nepravidelné
- pravidelné
  - n-ární
  - ternární
  - binární
  - binární uspořádané
- **Operace:**

Stromy

- nepravidelné
- pravidelné
  - n-ární
  - ternární
  - binární
  - binární uspořádané
- **Operace:**
  - Inicializace

Stromy

- nepravidelné
- pravidelné
  - n-ární
  - ternární
  - binární
  - binární uspořádané
- **Operace:**
  - Inicializace
  - Vložení do stromu

Stromy

- nepravidelné
- pravidelné
  - n-ární
  - ternární
  - binární
  - binární uspořádané
- **Operace:**
  - Inicializace
  - Vložení do stromu
  - Zjištění prázdnosti

Stromy

- nepravidelné
- pravidelné
  - n-ární
  - ternární
  - binární
  - binární uspořádané
- **Operace:**
  - Inicializace
  - Vložení do stromu
  - Zjištění prázdnosti
  - Zjištění přítomnosti prvku

Stromy

- nepravidelné
- pravidelné
  - n-ární
  - ternární
  - binární
  - binární uspořádané
- **Operace:**
  - Inicializace
  - Vložení do stromu
  - Zjištění prázdnosti
  - Zjištění přítomnosti prvku
  - Průchod stromem – výpis prvků do lineární struktury

Implementace stromu

Implementace stromu

- B-strom (speciální binární strom s nejvýše 2 následníky uzlů)

Implementace stromu

- B-strom (speciální binární strom s nejvýše 2 následníky uzlů)
- Datové prvky:

```
type TypData = longint;

     UkUzel = ^Uzel;
     Uzel = record
       Data: TypData;
       Vlevo, Vpravo: UkUzel;
     end;

     Tree = UkUzel;
```

Implementace stromu

```
procedure Init(var T: Tree);
 begin T:=nil
 end;

procedure Insert(var T: Tree; E: TypData);
 begin
       if T=nil then begin
          new(T);
          T^.Data:=E;
          T^.Vlevo:=nil;
          T^.Vpravo:=nil
       end else
          if E<=T^.Data then Insert(T^.Vlevo, E)
                        else Insert(T^.Vpravo, E)
 end;
```

Implementace stromu

```
procedure InOrder(T: Tree);
 begin if T<>nil then begin
         InOrder(T^.Vlevo);
         writeln(T^.Data);
         InOrder(T^.Vpravo)
       end;
 end;
var S: Tree;
    C: longint;

begin Init(S);
      while not eof do begin
         readln(C);
         Insert(S,C)
      end;
      InOrder(S);
end.
```

Komplexní domácí úloha

Komplexní domácí úloha

- Implementujte následující abstraktní datové typy, a to:

Komplexní domácí úloha

- Implementujte následující abstraktní datové typy, a to:
  1. v „obyčejném“ strukturovaném tvaru

Komplexní domácí úloha

- Implementujte následující abstraktní datové typy, a to:
  1. v „obyčejném“ strukturovaném tvaru
  2. v objektovém tvaru

Komplexní domácí úloha

- Implementujte následující abstraktní datové typy, a to:
  1. v „obyčejném“ strukturovaném tvaru
  2. v objektovém tvaru
  3. s obecnými datovými položkami

Komplexní domácí úloha

- Implementujte následující abstraktní datové typy, a to:
  1. v „obyčejném“ strukturovaném tvaru
  2. v objektovém tvaru
  3. s obecnými datovými položkami
  4. vždy v programové jednotce (modulu)

Komplexní domácí úloha

- Implementujte následující abstraktní datové typy, a to:
  1. v „obyčejném“ strukturovaném tvaru
  2. v objektovém tvaru
  3. s obecnými datovými položkami
  4. vždy v programové jednotce (modulu)
  5. s variantami konkrétních typů (pole, dyn. struktury, soubory bez udání typu).

Průpravný příklad

Průpravný příklad

- Průpravný příklad – implementace obyčejného lineárního dynamického jednosměrného seznamu s operacemi:

Průpravný příklad

- Průpravný příklad – implementace obyčejného lineárního dynamického jednosměrného seznamu s operacemi:
  - vložení prvku (na začátek, za/před vybraný prvek, na konec)

Průpravný příklad

- Průpravný příklad – implementace obyčejného lineárního dynamického jednosměrného seznamu s operacemi:
  - vložení prvku (na začátek, za/před vybraný prvek, na konec)
  - odstranění libovolného prvku

Průpravný příklad

- Průpravný příklad – implementace obyčejného lineárního dynamického jednosměrného seznamu s operacemi:
  - vložení prvku (na začátek, za/před vybraný prvek, na konec)
  - odstranění libovolného prvku
  - zjištění počtu prvků, výpisy, prohledávání

Abstraktní typy

Abstraktní typy

- **Seznamy:**

Abstraktní typy

- **Seznamy:**
  - zásobník

Abstraktní typy

- **Seznamy:**
  - zásobník
  - fronta

Abstraktní typy

- **Seznamy:**
  - zásobník
  - fronta
  - prioritní fronta

Abstraktní typy

- **Seznamy:**
  - zásobník
  - fronta
  - prioritní fronta
  - obecný seznam (jednosměrný, dvousměrný, cyklický)

Abstraktní typy

- **Seznamy:**
  - zásobník
  - fronta
  - prioritní fronta
  - obecný seznam (jednosměrný, dvousměrný, cyklický)
  - seznam s aktivním prvkem

Abstraktní typy

- **Seznamy:**
  - zásobník
  - fronta
  - prioritní fronta
  - obecný seznam (jednosměrný, dvousměrný, cyklický)
  - seznam s aktivním prvkem
- **Stromy:**

Abstraktní typy

- **Seznamy:**
  - zásobník
  - fronta
  - prioritní fronta
  - obecný seznam (jednosměrný, dvousměrný, cyklický)
  - seznam s aktivním prvkem
- **Stromy:**
  - obecný strom

Abstraktní typy

- **Seznamy:**
  - zásobník
  - fronta
  - prioritní fronta
  - obecný seznam (jednosměrný, dvousměrný, cyklický)
  - seznam s aktivním prvkem
- **Stromy:**
  - obecný strom
  - binární strom (uspořádaný ls<=o<ps, uspořádaný o>ls,ps)

Abstraktní typy

Abstraktní typy

- **Vyhledávací tabulka:**

Abstraktní typy

- **Vyhledávací tabulka:**
  - obecná (konstruovaná různými jinými typy)

Abstraktní typy

- **Vyhledávací tabulka:**
  - obecná (konstruovaná různými jinými typy)
  - tabulka s rozptýlenými hesly

Abstraktní typy

- **Vyhledávací tabulka:**
  - obecná (konstruovaná různými jinými typy)
  - tabulka s rozptýlenými hesly
  - stromové hledání

Abstraktní typy

- **Vyhledávací tabulka:**
  - obecná (konstruovaná různými jinými typy)
  - tabulka s rozptýlenými hesly
  - stromové hledání
- **Graf:**

Abstraktní typy

- **Vyhledávací tabulka:**
  - obecná (konstruovaná různými jinými typy)
  - tabulka s rozptýlenými hesly
  - stromové hledání
- **Graf:**
  - obecný graf

Abstraktní typy

- **Vyhledávací tabulka:**
  - obecná (konstruovaná různými jinými typy)
  - tabulka s rozptýlenými hesly
  - stromové hledání
- **Graf:**
  - obecný graf
  - orientovaný graf

Abstraktní typy

- **Vyhledávací tabulka:**
  - obecná (konstruovaná různými jinými typy)
  - tabulka s rozptýlenými hesly
  - stromové hledání
- **Graf:**
  - obecný graf
  - orientovaný graf
- **Řídké pole:**

Abstraktní typy

- **Vyhledávací tabulka:**
  - obecná (konstruovaná různými jinými typy)
  - tabulka s rozptýlenými hesly
  - stromové hledání
- **Graf:**
  - obecný graf
  - orientovaný graf
- **Řídké pole:**
  - obecné

Abstraktní typy

- **Vyhledávací tabulka:**
  - obecná (konstruovaná různými jinými typy)
  - tabulka s rozptýlenými hesly
  - stromové hledání
- **Graf:**
  - obecný graf
  - orientovaný graf
- **Řídké pole:**
  - obecné
  - trojúhelníková matice