Colecţia

COLLECTION, BAG, MULTI-SET

Observaţii

- **Colecţie** ("**bag**") este un container, o grupare finită de elemente.
- Într-o colecţie elementele nu sunt distincte (nu există o singură instanţă a unui element).
  - Din această cauză colecţia mai e cunoscută sub numele de **multi-mulţime** ("**multi-set**").
  - Operaţii specifice pe o colecţie sunt: adăugarea, ştergerea, căutarea unui element într-o colecţie.
    - Ca urmare tipul elementelor din colecţie, **TElement** ar trebui să suporte cel puţin operaţiile de: atribuire (←) şi testarea egalităţii (=).
- Caracteristică a colecţiei - nu contează ordinea elementelor.
  - Spre exemplu, o colecţie de numere întregi ar putea fi: ***c***={1, 2, 3, 1, 3, 2, 4, 2, 2}.

În continuare, vom prezenta specificaţia Tipul Abstract de Date **Colecţie** (interfața minimală).

domeniu

$\mathcal{Col}$={**col | col** este o colecţie cu elemente de tip **TElement**}

operaţii (**interfaţa** TAD-ului Colecţie)

creează(c)

pre: -

post: c∈$\mathcal{Col}$, c este colecţia vidă (fără elemente)

adaugă(c, e)

pre: c∈$\mathcal{Col}$, e∈**TElement**

post: c'∈$\mathcal{Col}$, c'=c+{e}

{s-a adăugat elementul în colecţie}

şterge(c, e)

pre: c∈$\mathcal{Col}$, e∈**TElement**

post: c'∈$\mathcal{Col}$, c'=c-{e}

{s-a eliminat o apariţie a elementului din colecţie}
{se poate returna adevărat dacă elementul a fost șters}

caută(c, e)

pre: c∈$\mathcal{Col}$, e∈**TElement**

post: ***caută*** = adevărat dacă e∈c

fals în caz contrar

dim(c)

pre: c∈ $\mathcal{Col}$,

post: ***dim*** = dimensiunea colecţiei c (numărul de elemente) ∈ $\mathcal{N}$

vidă(c)

pre: c∈ $\mathcal{Col}$

post: ***vidă***= adevărat în cazul în care c e colecţia vidă
fals în caz contrar

iterator(c, i)

pre: c∈ $\mathcal{Col}$

post: i∈ $\mathcal{I}$, i este un iterator pe colecţia c

distruge(c)

pre: c∈ $\mathcal{Col}$

post: colecţia c a fost 'distrusă' (spaţiul de memorie alocat a fost eliberat)

Menţionăm că pot fi definite în interfaţa Tipului Abstract de Date Colecţie şi operaţii specifice cum ar fi: reuniunea, intersecţia, diferenţa a două colecţii.

Deoarece colecţia are o operaţie care furnizează un iterator pe elementele sale, subalgoritmul care va tipări elementele unei colecţii ***c*** poate fi descris sub forma:

Subalgoritmul *tipărire*(c) **este**
{pre: c este o colecţie}
{post: se tipăresc elementele colecţiei}
iterator(c,i) {colecţia îşi construieşte iteratorul}
CâtTimp valid(i) **execută** {cât timp iteratorul e valid}
element(i, e) {se obţine elementul curent din iteraţie}
@ tipăreşte e {se tipăreşte elementul curent}
următor(i) {se deplasează iteratorul}
SfCâtTimp
SfTipărire

Complexitatea-timp a subalgoritmului de tipărire este θ (|c|), unde prin |c| am notat dimensiunea colecţiei c.

Ca şi modalităţi de reprezentarea ale unei colecţii, avem cel puţin următoarele posibilităţi:

- se reprezintă toate elementele colecţiei : 1, 2, 1, 4, 3, 1, 4, 2, 5;
- se reprezintă colecţia sub forma unor perechi de forma (e_1, f_1), (e_2, f_2),..,(e_n, f_n), unde $e_1, e_2,...,e_n$ reprezintă elementele distincte din colecţie, iar $f_1,f_2,...,f_n$ reprezintă frecvenţele de apariţie (numărul

de apariţii în colecţie) a elementelor corespunzătoare: spre exemplu colecţia anterioară s-ar reprezenta sub forma perechilor (1, 3), (2, 2), (4, 2), (3, 1), (5, 1).

Observaţie. În cazul în care elementele din colecție sunt de tip **TComparabil**, elementele pot fi memorate în ordine (în raport cu o anumită relație de ordine, de ex. ≤), pentru a reduce complexitatea timp a unor operații.

Modalităţi de implementare ale colecţiilor ar putea fi folosind:

- tablouri (dinamice);
- liste înlănţuite;
- tabele de dispersie;
- arbori binari (de căutare echilibrați).

În directorul TAD Colecție (pe pagina cursului, curs 3) găsiți implementarea parțială, în limbajul C++, a containerului **Colecție**, reprezentare secvențială pe vector dinamic (se memorează toate elementele colecției în vector).