

## Structuri de regăsire la nivel de obiect

Asist. Marian DÂRDALĂ, asist. Felix-Titus FURTUNĂ,  
Catedra de Informatică Economică, A. S. E., București

*Articolul prezintă impactul proiectării și programării orientate obiect asupra structurilor de regăsire a datelor. Este abordată în detaliu structura de tip hcC (hierarchy class Chain), derivată din arbori B.*

*Cuvinte cheie: obiect, clasă, ierarhia claselor, arbore.*

### Introducere

Varietatea informațiilor economice și complexitatea conexiunilor în care se află, impun identificarea unor structuri de date adecvate. Evoluția lor se face în concordanță cu tehnicile ce le manipulează.

Creșterea impresionantă a capacității de stocare a datelor în memorie (internă/externă) a determinat în paralel perfecționarea tehnicilor de regăsire a informațiilor, astfel încât timpul de acces să fie cât mai mic.

O structură de date ce s-a dovedit a fi eficientă în stocarea informațiilor este cea de obiect. Acest lucru a creat un impact major în cadrul analizei și proiectării sistemelor informatice, iar folosirea conceptului de programare orientată obiect stă la baza sistemelor de gestiune a bazelor de date orientate obiect.

Lucrarea își propune să prezinte structura arborelui hcC (hierarchy class Chain) utilizată în regăsirea eficientă a obiectelor în funcție de valoarea atributului de indexare.

Acest tip de structură este de fapt o adaptare a arborilor  $B^+$  la tipul de dată obiect. Se vor prezenta tipurile de noduri folosite, structura arborelui în ansamblu și posibilitatea de a realiza funcții pentru exploatarea lui, iar în final anumite aplicații la care o astfel de structură ar putea fi utilizată cu succes.

### Tipuri de noduri și structura lor

*Structura nodului de identificare a obiectelor.* Aceste noduri sunt la ultimul nivel al unui arbore hcC. Nodurile de identificare a obiectelor sunt divizate în  $n+1$  lanțuri de noduri de identificare. Există câte un lanț de noduri de identificare, numit **lanțul clasei**, corespunzător fiecărei clase din ierarhie și un lanț de noduri de identificare pentru toate clasele, numit **lanțul ierarhiei**.

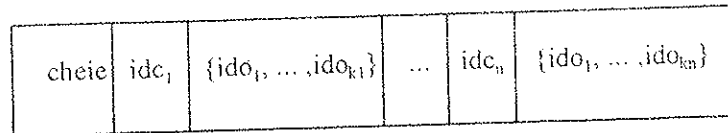
Din cele arătate mai sus se desprind două structuri și anume:

- lanțul clasei ce conține identificatorii obiectelor (*ido*) unei singure clase. Fiecare intrare în lanțul clasei are forma: <cheie,listă identificatori obiecte>:

Cheie	Nr. de identificatori obiecte	{ <i>ido</i> <sub>1</sub> , ..., <i>ido</i> <sub><i>n</i></sub> }
-------	-------------------------------------	---

Se poate observa deci, că ea conține valoarea cheii, numărul de identificatori obiecte și identificatorii obiectelor acelei clase care au aceeași valoare a cheii pentru atributul de indexare.

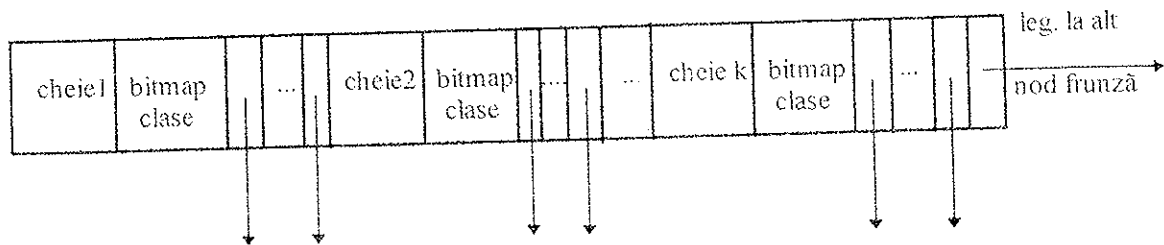
- lanțul ierarhiei conține identificatorii obiectelor aparținând tuturor claselor: <cheie, lista setului de identificatori de obiecte>



unde, *idc* este identificatorul clasei.  
Pentru o valoare dată a cheii, lanțul ierarhiei de identificatori obiecte conține *n* liste de identificatori. Fiecare listă cuprinde identificatorii obiectelor acelei clase care au o anumită valoare

pentru un atribut de indexare. Nodurile celor *n+1* lanțuri sunt legate împreună astfel încât să faciliteze traversarea de la stânga la dreapta.

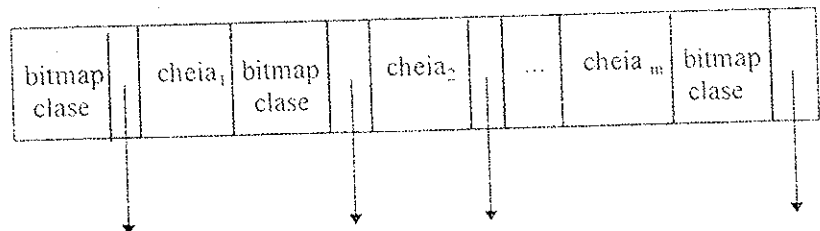
Structura nodului frunză este următoarea:



Fiecare intrare într-un nod frunză conține *k* chei, un bitmap și un set de pointeri.

Bitmap-ul conține *n* biți, câte un bit pentru fiecare clasă. El este setat dacă și numai dacă există cel puțin un obiect aparținând la acea clasă, având valoarea cheii, *k*.

Să presupunem că *m* biți din bitmap sunt setați pentru o valoare dată *k* a cheii. Setul de pointeri conține *m+1* valori, dintre care *m* referă nodurile din lanțul clasei și celălalt, un nod din lanțul ierarhiei. Nodurile punctate vor conține identificatorii obiectelor *ε* a căror atribut de indexare are valoarea *k*.  
Structura nodului intern este:



Structura lui este similară cu a unui nod al arborelui  $B^+$  în care sunt *m* chei și *m+1* pointeri la nodurile nivelului inferior. Există informații suplimentare stocate în nodul intern al arborelui hcC. Se asociază celor *m+1* intervale (determinate de cele *m* chei), din nodul intern, *n* biți pe interval, formând un bitmap. Un bit corespunde unei clase din ierarhia de clase. Pentru un interval el este 0 dacă și numai dacă nu există obiecte aparținând la acea clasă, cu valoarea atributului de indexare ce se află în acel interval.

### Descrierea structurii arborescente hcC

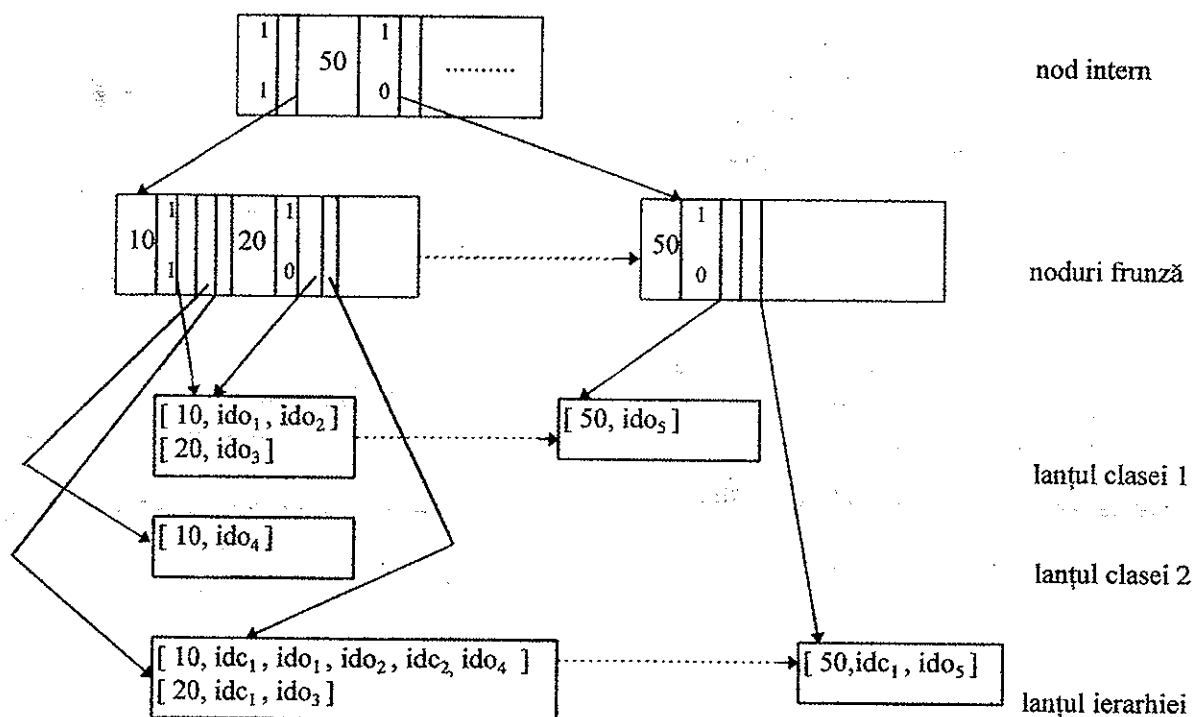
Structura arborelui hcC va fi descrisă folosind un exemplu. Vom considera un astfel de arbore, parțial populat. Nodul rădăcină este un nod intern. Sunt două noduri frunză la nivelul inferior și restul noduri de identificare a obiectelor. În figura de mai jos, ierarhia clasei este reprezentată prin două clase. Nodul rădăcină are numai o cheie și două bitmap-uri. Al doilea este {1, ...} ceea ce înseamnă că în intervalul având valori ale atributului de indexare *r*

mari de 50, există obiecte aparținând clasei 1 și nu există obiecte aparținând clasei 2.

În primul nod frunză (cu cheile 10 și 20), pentru cheia 10, primul pointer punctează la nodul care are identificatorul obiectului aparținând clasei 1. Al doilea pointer punctează la nodurile care au obiecte aparținând clasei 2 și au tot valoarea 10 pentru atributul de indexare. Aceștia sunt pointeri la lanțul clasei. Al treilea pointer, pentru cheia 10 punctează la

nodul care stochează identificatorii obiectelor ambelor clase (1 și 2). Acesta este pointerul la lanțul ierarhiei. Deoarece nu sunt obiecte aparținând clasei 2 cu cheia de valoare 20, bitul ei este 0 și nu există pointer la lanțul clasei respective.

În figura de mai jos, se observă că mai mulți pointeri dintr-un nod frunză pot puncta la același nod de identificare a obiectelor. Pentru cheile 10 și 20, primii pointeri referă același nod de identificare a obiectelor.



Pe această structură se pot defini operații de bază cum ar fi: căutarea, inserarea și ștergerea nodurilor. Un interes deosebit prezintă operația de căutare ce poate fi făcută atât punctual, cât și pe domeniu, pentru o anumită clasă sau pentru toate clasele din ierarhie.

### Aplicații

Sistemele de informare geografică (GIS), prin eterogenitatea informației manipulate, prin volumul mare al acestei informații și printr-o diversitate

de interogări, se pretează foarte bine la utilizarea unor structuri arborescente utilizate în regăsirea obiectelor.

Obiectul este preferat ca structură de stocare a informației în cadrul GIS. Aceste sisteme informatice combină în aplicații obiectele grafice deja cunoscute de anumite medii de programare cu altele construite de programatori, descrise prin intermediul claselor. GIS-urile folosesc pe lângă structuri de regăsire a obiectelor și structuri de date adaptate la stocarea și manipularea informației grafice cum ar fi: masive bidimensionale ce au elementele puse în corespondență cu

pixeli unei zone de pe ecran, quadtree ce este o arborescență construită în scopul divizării unei imagini în mai multe cadrane pentru a fi mai ușor prelucrată și vizualizată.

Modelele de date spațiale descriu formal semantici ale conceptelor din modele de date geometrice. Apelând la concepte de tipul topologie metrică și ordonare sunt modelate informații geografice pentru a răspunde în timp real la întrebări specifice GIS. Relațiile de tip incluziune / includere pot fi exprimate în mai multe moduri dar cu performanțe diferite în regăsire.

Rezolvarea problemelor de determinare a intersecției a două obiecte din model, studiul adiacenței, analiza suprapunerilor parțiale este direct dependentă de modul de manipulare a obiectelor.

### Concluzii

În literatura de specialitate mai sunt consacrate încă două structuri de indexi pentru obiecte și anume arbori CH (Class Hierarchy) și H (Hierarchy). Studiile făcute asupra acestora au demonstrat că structura arborescentă hcC este mai eficientă din punct de

vedere al necesarului de memorie în concordanță cu facilitățile pe care le oferă, cât și a vitezei de prelucrare.

### Bibliografie

1. Egenhofer, M.J., Herring, J.R., - *High - Level Spatial Data Structures for GIS*, Proceedings of AUTOCARTO GIS, Uniwn, 1990.
2. Franklin, W.M.R., - *Computer System and Low - Level Data Structures for GIS*, Proceedings of AUTOCARTO GIS, Uniwn, 1990.
3. Kim, W., Kim, K.C., Dale, A., - *Indexing Techiques for Object Oriented Databases*, Addison Wesley, 1989.
4. Low, C.C., Ooi, B.C., Lu, H., - *H - trees : A Dynamic Associative Search Index for OOB*. In Proceedings of the 1992 ACM - SIGMOD Conference on the Management of Data, June 1992.
5. Svenoth, B., Seshardi, S. - *The hcC - tree : An Efficient Index Structure for Object Oriented Databases*, Proceedings of 20- th Very Large Data Bases Conference, Santiago, Chile, September, 1994.