lProblema 21:

TAD CoadaCuPrioritati – implementare folosind un arbore binar de cautare.

Containerul este CoadaCuPrioritati.

Interfata si Specificatiile containerului :

* domeniu:

CP = {cp | cp este o coada cu prioritati cu elemente (e, p) de tip TElement x TPrioritate, avand relatia de ordine R intre prioritati}

* Creeaza(cp)

{creeaza o coada cu prioritati vida}

Pre : -

Post :cp

* Adauga(cp,e)

{se adauga un element in coada cu prioritati}

Pre : cp,e

Post : cp’ , cp’=cp

* Element(cp,e)

{se acceseaza elementul « cel mai prioritar »}

Pre : cp

Post :cp’=cp, e ,e este « cel mai prioritar element din cp »

* Vida(cp)

Pre : cp

Post :

* Plina(cp)

Pre :

Post : :

* Distruge(cp)

{destructor}

Pre :

Post :cp a fost « distrusa »(spatiul de memorie alocat a fost eliberare)

{Coada cu Prioritati nu are iterator}

Reprezentarea Cozii cu Prioritat : ARBORE BINAR DE CAUTARE

Am ales ca arborele binar de cautare sa fie reprezentat inlantuit pe noduri.

* Structura Nodului :

Nod :

Info :TElement{ aceasta este informatia propriu-zisa a nodului}

fiuMare :{acesta este fiul mai mare al unui nod parinte(din punct de vedere al prioritatii)}

fiuMic :{acesta este fiul mai mic al unui nod parinte(din punct de vedere al prioritatii)}

parinte :{parintele nodului curent}

* Structura elementului generic TElement

TElement :

R :relatie {R ⊆ T Prioritate x T Prioritate, R relatie de ordine}

{am ales aceasta reprezentare deoarece este mai generala}

Prioritate :intreg {aceasta este prioritatea elementului,ea trebuie setata de utilizator}

Info :TElem{acea informatie,care trebuie returnata,de tip generic}

* Structura Cozii cu Prioritati :

CP :

Dim :integer {ma va ajuta sa vad daca CP este vida sau nu}

prim :{nodul radacina al arborelui}

* **Creeaza(cp) :**

cp.prim=aloca() ;

cp.dim=0 ;

* sfarsit Subalgoritm

**Adauga(cp,e) {complexitate**

;

;

;

* Daca (cp.prim=0 )

Cp.[prim].info=e ;

Cp.dim=cp.dim+1;

* Sfarsit daca

q=cp.prim ;

ant=q ;

* Cattimp(q){

Ant=q ;

i=[q].info.relatie([q].info,e) ;{aceasta relatie imi returneaza 1,daca primul element are prioritatea mai mare si 2 daca al doilea element are prioritatea mai mare}

* + daca(i=1) q=[q].fiuMic ;

altfel q=[q].fiuMare ;

* + sfarsit daca
* sfarsit cattimp
* daca (i=1) [ant].fiuMic=aloca() ;

[ant].[fiuMic].parinte=ant ;

[ant].info=e ;

altfel

[ant].fiuMare=aloca() ;

[ant].[fiuMare].parinte=ant ;

[ant].info=e ;

* Sfarsit daca

Cp.dim=cp.dim+1 ;

* Sfarsit subalgoritm

**Element(cp,e) {complexitate**

* daca(cp.dim=0) e=cp.[prim].info ;

Altfel

* Cattimp(q){

Ant=q ;

q=[q].fiuMare ;

* sfarsit cattimp

e=[ant].info ;

* sfarsit daca
* sfarsit Subalgoritm

**sterge(cp,e) {complexitate**

* daca(cp.dim=0) e=cp.[prim].info ;

dealoca(cp.prim) ;

Altfel

ant

* Cattimp(q)

ant=q ;

* q=[q].fiuMare ;
* sfarsit cattimp

e=[q].info ;

tmp=q ;

daca([q].fiuMic≠NIL)q=[q].fiuMic ;

{s-ar putea sa am un fiu « mai mic »,acesta trebuie sa fie acum,cel mai prioritate,prin urmare,il trec in locul nodului de sters}

dealoca(tmp) ;

* sfarsit daca
* sfarsit Subalgoritm
* plina(cp)

plina=false ;

* sfarsit Subalgoritm

* vida(cp) **{complexitate**

daca(cp.dim=0)vida=true

altfel vida=false

* sfarsit subalgoritm

**Distruge(cp) : {complexitate**

TElement e ;{datoriata faptului ca operatia de stergere imi returneaza un element,sunt obligat sa retin acest element,intr-o variabila}

Pentru(i=0,i<cp.dim ,i++)

Sterge(cp,e) ;

Sfarsit pentru

Sfarsit Subalgoritm

Aplicatia :

Intr-o institutie de invatamant,este o singura imprimanta,si mai multi utilizatori. Stiind ca si cadrele didactice au acces la aceasta imprimanta,dar cu grad de prioritate diferit,sa se creeze un program care afiseaza ordinea de listare a sarcinilor,tinand cont de urmatoarele grade de prioritate :

Elev-grad de prioritate 0 ;

Profesor –grad de prioritate 1 ;

Secretariat-grad de prioritate 2 ;

Director-grad de prioritate 3 ;

SarcinaNoua :

Nume :string

Info :TElement

Subalgoritm sarcinaNoua(cp,e)

{Preconditii :cp este o coada cu prioritati

Postconditii :e este de tip sarcinaNoua}

Daca(e.nume= « Director ») atunci e.info.prioritate=3 ;

Altfel daca(e.nume= « Secretariat »)atunci e.info.prioritate=2 ;

Altfel daca(e.nume= « Profesor »)atunci e.info.prioritate=1 ;

Altfel e.info.prioritate=0 ;

Sfarsit daca

Sfarsit daca

Sfarsit daca

Adauga(cp,e.info) ;

Sfarsit subalgoritm

Subalgoritm Listare(cp)

{preconditii :cp este o coada cu prioritati

Postconditii :cp este o coada cu prioritati}

e :TElement ;

cattimp(vida(cp)=false) executa

Sterge(cp,e) ;

Sfarsit cattimp

Sfarsit daca

@afiseaza e {pentru a stii ce s-a listat}

Sfarsit subalgoritm

Subalgoritm principal

Creeaza(cp) ;

I intreg ;

I=0;

E :SarcinaNoua

@afiseaza meniul

@daca doriti adaugare apasati 1

@citesc pe i

Cattimp(i≠0)

@afisez : « Introduceti utilizatorul(Director/secretariat/profesor/elev)

@citesc e.nume

@afisez : « introduceti sarcina,numele fisierului »

@citesc e.info

SarcinaNoua(cp,e) ;

@afisez un meniu : « Daca doriti adaugarea unei noi sarcini apasati 1 »

@ « Daca doriti inceperea listarii apasati  0 »

Sfarsit cattimp ;

Listare(cp) ;

Sfarsit subalgortim