**Coada de așteptare**

Regulile de laborator (Secțiunea IX din documentul disponibil în

General → Files → Class Materials → SDA Descriere&Cerinte.pdf

rămân valabile

* în cazul în care se întârzie un laborator, nota finală se înmulțește cu 0.8; nu se acceptă întârzieri mai mari de un laborator (2 săptămâni).

1. Reguli de desfășurare a laboratorului

* Tema cu termen de predare în acest laborator este L4. Se pot preda și L3, respectiv L1, cu întârziere, însă în condițiile în care se predau maxim 2 teme de laborator.
* La acest laborator se va primi funcționalitate nouă pentru L4 și pentru L3 .
  + Funcționalitatea nouă care va trebui implementată, pentru fiecare problemă, va fi înscrisă în secțiunea 4. Funcționalități extra la începutul orei de laborator. Ea este dependentă de TAD-ul pe care l-ați implementat.
  + Pentru a implementa funcționalitatea nouă, aveți la dispoziție 30 minute de la momentul începerii activității de laborator. După (cel mult) 30 de minute, numele tuturor studenților care doresc să predea în activitatea de laborator curentă trebuie să apară în tabel. Fiecare student se va trece în tabel imediat după finalizarea noii cerințe.
  + Se vor indica complexitățile (caz favorabil, caz defavorabil, complexitatea totală) pentru funcționalitatea nouă.
  + Se va scrie, sub forma de comentariu, descrierea Pseudocod pentru funcționalitatea nouă.
  + Trebuie implementat cod pentru a verifica corectitudinea funcționalității noi.
* Predarea temei de laborator se va efectua într-o conversație individuală cu cadrul didactic
  + fiecare student are alocate maxim 5 minute (pentru a asigura posibilitatea de predare pentru toți studenții din semigrupă);
  + în momentul finalizării discuției, fiecare student va marca cu X în ultima coloană a tabelului
* În cazul în care timpul permite (după predarea temelor tuturor studenților care doresc să predea), la finalul laboratorului, se mai pot adresa eventuale întrebări.

2. Anunțuri

*-------------------------------------------------------------------------------------------------*

*(Aici se vor face eventuale anunțuri (de interes general) pe perioada laboratorului.)*

*Vă rog să vă asigurați că ați scris, sub formă de comentarii, complexitățile (totale ale)tuturor operațiilor (implementate ca temă).*

*Dacă sunt studenți care doresc să predea exclusiv L1 (pentru aceasta temă nu se va primi funcționalitate extra), îi rog pe aceștia să se treacă în tabel imediat după începerea acțivității de laborator.*

*Va rog sa notati cerinta pentru fiecare laborator in proiect, pentru a o avea la indemana.*

*V-am asingnat temele pentru L6. Termenul de predare este saptamana 11. In urmatoarea activitate de laborator, se pot preda temele in avans, optional.*

*-------------------------------------------------------------------------------------------------*

3. Tabel predare

|  |  |  |  |
| --- | --- | --- | --- |
|  | Nume | Doresc să prezint L1 și/sau L3 (a se marca cu X) | Am finalizat  discuția (a se marca cu X) |
| 1 | Ciupeiu Diana | x |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |
| 11 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 |  |  |  |
| 17 |  |  |  |

4. Funcționalități extra

* Coadă

Această operație nu ar trebui să fie adăugată în clasa Coadă. Ar trebui să fie implementată în fișierul corespunzător programului principal, folosind operațiile din interfața TAD Coadă, deci fără a modifica clasa Coadă și fără a accesa reprezentarea.

// tipărește conținutul cozii. La sfârșitul operațieii, coada conține toate elementele pe care le conținea înainte.

void tipăreșteCoada(Coadă& c);

* Coadă cu priorități

// modifică prioritatea unui element

// returnează prioritatea veche a elementului *elem* sau, dacă *elem* nu este în CP, returnează -1

TPrioritate CP::schimbaPrioritate(TElem elem, TPrioritate prioritateNoua);

* Colecție

TAD Colecție

// elimină *nr* apariții ale elementului *elem*. În cazul în care elementul apare mai puțin de *nr* ori, toate aparițiile sale vor fi eliminate.

// returnează numărul de apariții eliminate ale elementului.

// aruncă excepție în cazul în care este *nr* este negativ.

int eliminaAparitii(int nr, TElem elem);

* Dicționar (indiferent dacă este ordonat sau nu)

// returnează un vector cu toate valorile dicționarului

vector<TValoare> colecțiaValorilor() const;

* Listă (indiferent dacă este ordonată sau nu)

TAD Listă

// elimină din k în k elemente din listă. Dacă lista conține mai puțin de k elemente, lista nu este modificată

// returnează numărul de elemente eliminate

// aruncă excepție în cazul în care k este negativ sau 0

int eliminaDinKInK(int k);

* Matrice

// determină suma elementelor de sub diagonala principală

TElem Matrice::sumaSubDiagonalaPrincipala ();

* Multidicționar (indiferent dacă este ordonat sau nu)

// returneaza valoarea care apare cel mai frecvent în multidicționar. Dacă mai multe valori apar cel mai frecvent, se returnează una (oricare) dintre ele.

// Dacă multidicționarul este vid, operația returnează NULL\_TVALOARE

TValoare ceaMaiFrecventaValoare() const;

* Mulțime

// returnează diferența dintre valoarea maximă și cea minimă (presupunem valori întregi)

// dacă mulțimea este vidă, se returnează -1

int diferențaMaxMin() const;

* Stivă

Transformarea într-o Stivă de capacitate fixă. Efectuați următoarele modificări în proiectul:

* Constructorul clasei Stivă primește ca parametru un număr întreg reprezentând capacitatea stivei. În cazul în care acest număr este 0 sau negativ, se aruncă excepție.
* Operația de adăugare aruncă excepție în cazul în care stiva este plină.
* Adăugați o operație estePlină (similar cu esteGoală/vidă) care returnează true dacă stiva este plină și false în caz contrar.

Obs: Aceste modificări se vor efectua pe o copie a proiectului, pentru a vă asigura că testele vor rula în continuare pe proiectul inițial.