

Segona pràctica d' Estructures de dades

Matrícula en el GEI

Presentació del problema

Disposem de les dades de matrícula dels estudiants d'informàtica en un curs. Tenim un fitxer en format csv amb el següent contingut:

codi assignatura; nom assignatura; crèdits; curs; quadrimestre; codi alumne; nom alumne

Es tracta de dissenyar els TADs per a guardar la informació de la millor forma i que donin resposta a les següents necessitats.

Alumne

Operació constructora:

Alumne: String String => Alumne

Operació que crea un objecte a partir del (1) codi de l'alumne i (2) el seu nom

Operacions consultores:

getCodi: Alumne => String

getNom: Alumne => String

Operacions que permeten consultar els valors dels diferents camps

Assignatura

Operació constructora:

Assignatura: Integer String Integer Integer Integer => Assignatura

Operació que crea un objecte a partir del (1) codi de l'assignatura, (2) el seu nom, (3) els crèdits que té, (4) el curs en que s'imparteix dins el pla d'estudis i (5) el quadrimestre

Operacions consultores:

getCodi: Assignatura => Integer

getNom: Assignatura => String

getCredits: Assignatura => Integer

getCurs: Assignatura => Integer

getQuadrimestre: Assignatura => Integer

Operacions que permeten consultar els valors dels diferents camps

TAD Matricula

Operació constructora:

Matricula: Integer String => Matricula

Operació que crea una relació entre el (1) codi de l'assignatura i (2) el codi de l'alumne

Operacions consultores:

getCodiAssignatura: Matricula => Integer

getCodiAlumne: Matricula => String

Operacions que permeten consultar els valors dels diferents camps

Implementació dels TADs i de l'aplicació

En aquest exercici pràctic s'ha d'implementar el següent:

- Implementar els TADs indicats anteriorment.
- Cal definir l'especificació del **TAD LlistaGenerica** mitjançant una interfície i fer **DUES implementacions** obligatòries (recordeu posar sempre els comentaris adients):

1. memòria estàtica, un vector ordenat. **Implementació amb cursors i gestionar els espais lliures.**
2. memòria dinàmica, **llista amb punters ordenada.**
3. (opcional) Utilitzar una estructura de les *java.util.collection*.

El criteri d'ordre per al cas dels Alumnes és el nom, per al cas de les assignatures és el curs, quadrimestre i nom (és a dir, les assignatures s'ordenen per curs, dins el mateix curs per quadrimestre i dins el quadrimestre per nom de l'assignatura).

- Cal definir l'especificació del **TAD Multillista** mitjançant una interfície i fer **UNA implementació** obligatòria utilitzant memòria dinàmica doblement encadenada amb punters a l'alumne i l'assignatura a la que correspon la relació.

- Es disposarà d'una classe amb el programa principal que permeti fer les proves pertinents amb qualsevol de les implementacions, és a dir, important els TADs definits anteriorment de forma correcta. El programa principal ha de:

- Preguntar quina de les 2 implementacions (3 si es fa l'opcional) es vol fer servir.
- Preguntar el nom del fitxer a llegir.
- Construir l'estructura de dades a partir del contingut del fitxer. Mostrar per pantalla el temps que ha tardat en executar-se.
- Capturar i gestionar correctament tots els errors existents com a excepcions, evitant que el programa es bloquegi en cap moment.
- Implementar els mètodes toString, compareTo, equals, clone en les classes base i crear iterators per a les llistes.
- Permetre les següents consultes:
 - A partir d'un codi d'alumne, mostrar per pantalla les dades de les assignatures que ha matriculat i a quants crèdits corresponen en total. Mostrar el temps que ha tardat en executar-se.
 - A partir d'un codi d'assignatura, mostrar per pantalla quins alumnes l'han matriculat. Sumar el número total d'alumnes i mostrar-lo per pantalla. Mostrar el temps que ha tardat en executar-se.

- Mostrar per pantalla les dades dels alumnes que han matriculat X crèdits o menys. El valor de X es demanarà per teclat. Mostrar el temps que ha tardat en executar-se.
- Mostrar per pantalla les dades de les assignatures que tenen un mínim de Y estudiants matriculats. El valor de Y es demanarà per teclat. Mostrar el temps que ha tardat en executar-se.

Informe de la pràctica

Es demana un informe que contingui:

- una breu explicació de com s'ha dissenyat la solució utilitzant els diferents TADs
- l'especificació dels TADs
- anàlisi de cost de les operacions en les diferents implementacions
- anàlisi dels temps d'execució dels diferents jocs de proves comparant les diferents implementacions.

Instruccions per al desenvolupament de la pràctica

- Aquesta pràctica es farà en grups de dos persones.
- Heu de crear el *Java Project* d'Eclipse amb el nom, **ED_PRACX_COGNOM1_NOM1_COGNOM2_NOM2**, corresponents als noms dels dos integrants del grup.
- Recordeu que totes les classes han d'estar ben documentades i utilitzar l'eina JavaDoc per a generar la documentació pertinent.
- Lliurareu els codis fonts (fitxer Java), la documentació generada amb l'eina JavaDoc i un informe en pdf segons les indicacions que hi ha en aquest enunciat. Tota aquesta informació es guardarà com a projecte exportat de l'Eclipse (fitxer ZIP) i es penjarà a la tasca corresponent del "moodle". El nom del fitxer serà **ED_PRACX_COGNOM1_NOM1_COGNOM2_NOM2.zip**
- El termini de lliurament és el divendres **07 d'abril de 2017 a les 21h**.
- En cap cas es consideraran lliurades les pràctiques que no arribin amb el procés indicat (codi + informe).
- No s'acceptaran pràctiques que (a) No estiguin acabades. (b) No facin el que es demana a l'enunciat. (c) No compilin i executin. o (d) No compleixin amb la nomenclatura d'entrega.
- No es poden fer servir col·leccions de Java (java.util.*) excepte en la part opcional.