

Segona pràctica d' Estructures de dades

Empresa turística

Presentació del problema

Disposem de les dades de les excursions que ofereix una empresa turística amb les activitats que les componen i dels turistes que han contractat alguna d'aquestes excursions. Els dos fitxers estan en format CSV i tenen el següent contingut:

Fitxer 1: Conté un llistat de totes les excursions disponibles, i dins de cada excursió totes les activitats de que està composta i el preu de cadascuna.

codi excursió; nom excursió; codi activitat 1; nom activitat1; preu activitat 1;.... ; codi activitat N; nom activitat N; preu activitat N

Fitxer 2: Conté un llistat on cada línia és una excursió contractada per un turista. Penseu que un turista pot aparèixer en diferents línies si contracta varies activitats.

codi excursió; nom excursió; identificador turista; nom turista, cognom turista

En aquesta pràctica el que heu de fer és dissenyar els següents TADs per a guardar la informació de la millor forma i que donin resposta a les necessitats que es descriuen més endavant. Els 5 TADs que us demanem són els següents:

- TAD Activitat
- TAD Excursió
- TAD ContingutExcursió
- TAD Turista
- TAD ExcursióContractada

Vosaltres heu de decidir quines operacions constructores i consultores són necessàries en cadascun dels TADs, i posar totes les especificacions dels mateixos a la vostra documentació.

Implementació dels TADs i de l'aplicació

Per a realitzar la implementació de la pràctica heu de seguir les següents indicacions:

- Cal crear un **TAD LlistaGenerica** que faran servir alguns dels altres TADs que teniu que implementar. La Llista Genèrica es compondrà a partir d'una interfície i **DUES implementacions** obligatòries (recordeu posar sempre els comentaris adients):

1. memòria dinàmica, **llista amb punters ordenada**.
2. Utilitzar una estructura de les *java.util.collection*.

El criteri d'ordre per al cas de les activitats és el codi, pel cas de les excursions també és el codi, i per part dels turistes és el cognom i després pel nom.

- Cal crear també un **TAD MultillistaGenerica** mitjançant una interfície i fer **UNA implementació** obligatòria utilitzant memòria dinàmica doblement encadenada amb punters als dos elements als que correspon la relació.
- Es disposarà d'una classe amb el programa principal que permeti fer les proves pertinents amb qualsevol de les implementacions disponibles. El programa principal ha de:
 - Preguntar quina de les 2 implementacions es vol fer servir. Totes les llistes genèriques que es creïn en el programa faran servir el mateix tipus d'implementació.
 - Preguntar el nom dels dos fitxers a llegir.
 - Omplir l'estructura de dades a partir del contingut dels fitxers. Mostrar per pantalla el temps que ha tardat en executar-se la càrrega de les dades.
 - Capturar i gestionar correctament tots els errors existents com a excepcions (duplicats, errors de codis que no existeixin, etc.), 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 i mostrar el temps que tarda en executar cadascuna d'elles:
 - A partir del codi d'una excursió mostrar per pantalla quines activitats inclou i quin és l'import total de l'excursió (la suma dels preus de les activitats que la componen).
 - A partir del codi d'una activitat mostrar per pantalla quines excursions ofereixen l'activitat i quin és l'import total de cada una de les excursions.
 - A partir del codi d'una activitat mostrar per pantalla aquelles que l'han realitzat menys de X turistes. El valor de X es demanarà per teclat.
 - A partir del codi d'una excursió mostrar per pantalla quins turistes l'han realitzat, sumar-los i mostrar el nombre total de turistes.
 - A partir de l'identificador d'un turista, mostrar per pantalla quines excursions ha contactat.
 - A partir de l'identificador d'un turista, mostrar per pantalla l'import total de les excursions que ha realitzat.
 - Permetre consultar les següents dades: el nombre total d'activitats introduïdes en el sistema, el nombre total de turistes introduïts en el sistema, i el nombre total d'excursions contractades.

Informe de la pràctica

Es demana un informe que contingui:

- Portada amb les dades que identifiquen el treball (títol de la pràctica i nom de l'assignatura), el nom de l'alumne i la data d'entrega
- Índex
- Especificacions de tots els TADs que heu creat per a fer la vostra pràctica.
- Disseny amb
 - una breu explicació de com s'ha dissenyat la solució utilitzant els diferents TADs (dades que es guarden en els TADs, relacions entre els TADs, etc.)
 - anàlisi de cost de les operacions en les diferents implementacions
- Joc de proves comparant els temps d'execució de totes les consultes descrites anteriorment usant les diferents implementacions.
- Recursos utilitzats. Referències (bibliogràfiques i pàgines web), programari, hardware, etc...

Instruccions per al desenvolupament de la pràctica

- Aquesta pràctica es farà individualment.
- Heu de crear el *Java Project* d'Eclipse amb el nom, **ED_PRAC2_COGNOM_NOM1**, corresponents al nom de l'alumne. **NO ES CORREGIRAN PRÀCTIQUES QUE NO COMPLEIXIN AQUEST CRITERI.**
- Recordeu que totes les classes han d'estar ben documentades i utilitzar l'eina JavaDoc per a generar la documentació pertinent.
- Lliurareu els codi 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_PRAC2_COGNOM_NOM.zip**
- El termini de lliurament és el divendres **13 d'abril de 2018 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 implementació on es demana explícitament.