

SEMINAR 8

Aplicație Gestionare Produse Dulci (extindere aplicație anterioară pentru a lucra cu mai multe entități).

Product: id, denumire, tara_origine

Shop: id, nume, locatie

SaleItem: id, produs, shop, pret, stoc

PREVIOUSLY:

Clasa Product + gestiune lista produse

ADDED (BY DEFAULT):

Clasa Shop + gestiune lista magazine

Custom Exceptions:

- CandyStoreException (base class)
- Derived classes: ValidationException, Repository Exception (DuplicateID Exception, ProductAlreadyAssigned Exception), ProductNotFoundException, ShopNotFoundException, SaleItemNotFoundException

TO DO:

1. Să se implementeze clasa ProductFileRepo care lucrează cu produse dintr-un fișier dat (încarcă și salvează). Fiecare produs este salvat pe o linie din fișier, de forma: *id;denumire, tara_origine*. Nu se va folosi moștenire.

2. Să se implementeze clasa SaleItem și să se ofere posibilitatea de a crea un SaleItem pentru un produs și un magazin dat.

- Modalitati de implementare SaleItem/asociere Product & Shop
 - Clasa SaleItem, obiect Product, obiect Shop, float pret, int stoc
 - ◆ Similar cu varianta 2, mai convenabil pentru moment pentru a avea toate campurile (altfel ar trebui sa folosim findProduct, findShop de fiecare data)
 - Clasa SaleItem, product_id, shop_id, float pret, int stoc
 - ◆ In fisier sale_items.txt (pt. un SaleItemsRepoFile ulterior) oricum vom salva doar ID-uri, pret, stoc
 - Liste de Products/Shops
 - ◆ Probleme la salvarea in fisier (e.g. salvare entitati de mai multe ori)
 - ◆ Statistici care depind de istoric/urmarirea evolutiei legaturilor (e.g. stergere produs, pt. Problema carte - inchiriere - client, daca returnez cartea, o sterg din lista clientului - nu ramane urma de inchiriere - raport cele mai inchiriate carti corect?)

STATISTICI/RAPOARTE

3. Să se afișeze locațiile cu cel mai mic preț pentru un produs dat (primele 3). Se va afișa numele produsului, numele magazinului, locația acestuia și prețul.

4. Să se afișeze numărul de produse vândute în fiecare magazin.