Programare Orientată pe Obiecte

13.01.2021

Produse și Oferte (Grupa 30224)

Construiți o aplicație care să gestioneze produse și oferte. Principalele funcționalități vor fi crearea, ștergerea și căutarea de produse și oferte. Produsele vor avea o denumire și un preț, iar o ofertă va avea o listă de produse și un preț total.

Observație: Aplicația nu are nevoie de o bază de date, dar toate obiectele ce ar trebui salvate în baza de date vor fi salvate în structurile de date descrise mai jos.

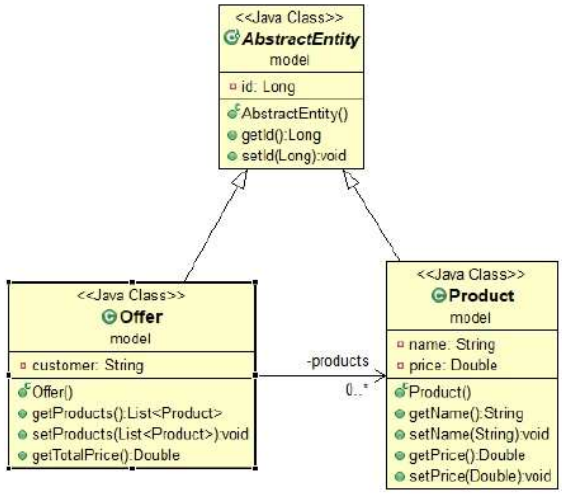

Figura 1: Diagrama UML modele

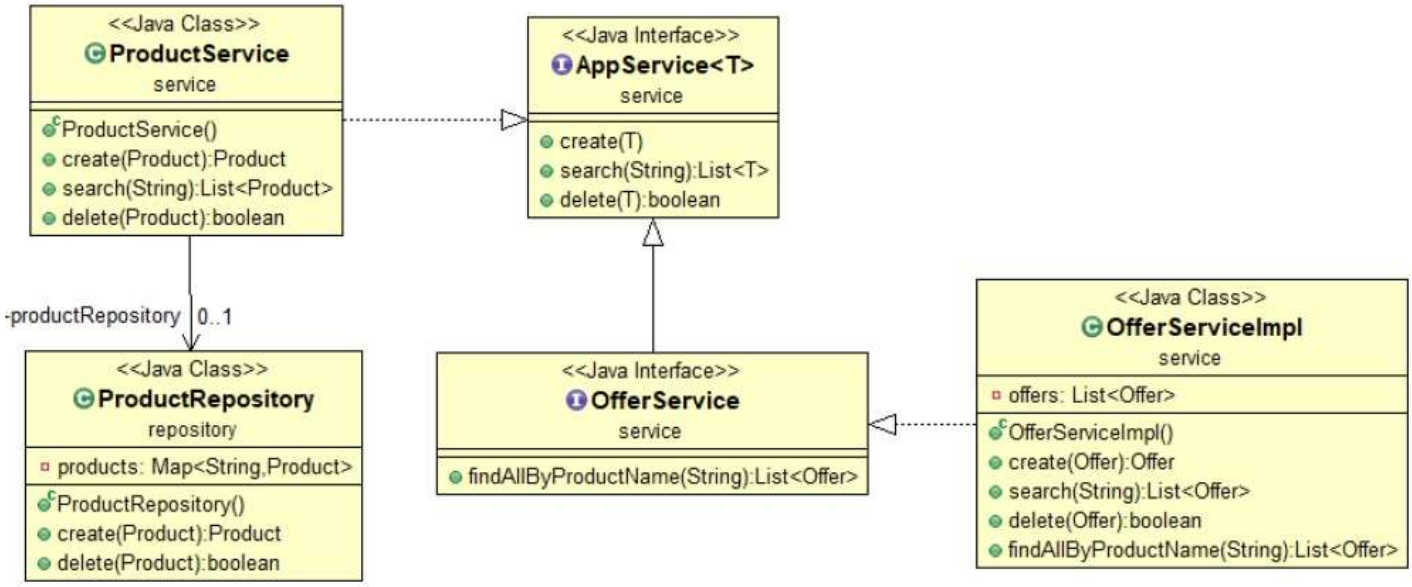

Figura 2: Diagrama UML servicii

Programare Orientată pe Obiecte

13.01.2021

Observație: Citiți cu atenție și înțelegeți cerința și diagramele UML. Fiți atenți la legăturile din diagrame pentru că fiecare dintre ele are o anumită semnificație. Dacă aveți nevoie de modificări de structură puteți să le faceți (metode/atribute suplimentare).

1. Creați modelele ce vor fi folosite în aplicație după diagrama din prima figură.
2. Creați și implementați clasa ***ProductRepository*** după cum urmează:
    - metoda *create* ar trebui să adauge un produs în tabelul de dispersie din cadrul clasei (atributul product). Cheia din tabel reprezintă denumirea produsului (ignorați diferența dintre litere mari/mici, adică produsul "lapte" e același cu "Lapte").
    - metoda *delete* va încerca să șteargă un produs din structura de date dacă acesta există. Dacă produsul nu există atunci se va arunca excepția *ProductNotFoundException* (excepție verificată [checked])
3. Creați clasa ***ProductService*** urmărind diagrama din figura 2.
    - în metoda *create* trebuie să verificați, înainte de a salva produsul folosing clasa *ProductRepository*, diverse aspecte ce țin de produsul pe care doriți să-l salvați. Produsul trebuie să aibă o denumire și un preț valid. În caz contrar se va arunca o excepție *ValidationFailedException* (de tipul excepție neverificată [unchecked]).
    - în interiorul metodei *delete* tratați excepția pe care ar putea să o arunce metoda din *ProductRepository* folosind blocul try-catch.
    - metoda *search* trebuie să returneze o lista de produse a căror denumire începe cu șirul de caractere pe care metoda îl primește ca parametru. (de ex. pentru "la" rezultatul poate să fie ["lapte", "lămâie"]; același rezultat îl va întoarce și căutarea după șirul "La").
4. Creați și implementați ***OfferService*** și ***OfferServiceImp***l după diagrama din figura 2 și următoarele completări:
    - în metoda *create* o oferta este considerată valida dacă are numele unui client setat și o lista de produse ce conține cel puțin un produs. Aruncați excepția *ValidationFailedException* dacă oferta de adăugat nu respectă aceste condiții
    - metoda *search* va returna o lista de oferte al căror client începe cu șirul de caractere căutat
    - la *delete* aruncați excepția *OfferNotFoundException* (excepție neverificată) dacă oferta pe care doriți să o ștergeți nu se regăsește în lista de oferte din clasa OfferServiceImpl
    - *findAllByProductName* trebuie să returneze o lista de oferte care conțin cel puțin un produs care are exact aceeași denumire cu textul după care se face căutarea.
5. Creați o clasă în care să simulați o secvență logică de operații care pot fi efectuate în cadrul aplicației de gestionare a produselor și a ofertelor (creare, ștergere și adăugare de produse și oferte) și afișați rezultatele intermediare.
6. Creați o clasă *FileReader* care conține o metodă statică *readProducts(String filePath)* care citește din fișier datele despre produse (numele și prețul). Datele despre fiecare produs se află pe câte o linie în fișier sub forma: numeProdus, prețProdus.
7. Scrieți niște teste unitare (folosind *JUnit*) care să arate buna funcționare a metodei *create* din *OfferServiceImpl*. Demonstrați printr-un test buna funcționare a metodei *search* din cadrul clasei *ProductService*.