Documentatie

Order management



Tehnici de programare

Tema3

Realizat de Balc Horia-Ovidiu

Grupa 30227

Cuprins

1 . Cerinte Functionale

2 . Obiectivul temei

3 . Analiza problemei

4 . Proiectare (decizii de proiectare, diagrame UML, structuri de date, proiectare clase, interfete, relatii, packages, algoritmi, interfata utilizator)

5 . Implementare

6. Testare

7 . Concluzie

8 . Bibliografie

# Cerinte Functionale

Proiectati si implementati un model de aplicatie a unui magazine online, datele fiind stocate intr-o daza de date.

Programul:

-va avea o baza de date formata din cel putin 3 tabele (4 tabele: client, product, orders, si orderitem)

-folosirea operatiilor cu baze de date ( inserarea, stergerea si modificarea datelor ddin tabele)

- utilizarea unui jar file

- creearea pdf-urilor pentru fiecare table (in cazul tabelului OrderItem se va crea cate un pdf la fiecare comanda)

-creearea java doc-ului pentru fiecare clasa existent in proiect

-citirea din fisier a unor instructiuni si utilizarea lor

-impartirea proiectului pe pachete

# 2Obiectivul Temei

2.1 Obiectivul principal

Obiectivul temei este crearea, in limbajul JAVA, a unui model de magazine online cu ajutorul caruia se actualizeaza baza de date in functie de operatiile date din fisier .Aceasta tema are practicabilitate in viata reala si are multe obiective importante care stau la baza programarii in limbajul JAVA.

2.2 Obiectivul secundare

Obiectivele secundare sunt: - folosirea programarii obiectuale prin utilizarea incapsularii

- scrierea variabilelor in stilul Camel Case

-utilizarea metodelor(maxim 30 de linii) si claselor( maxim 300 de linii)

- citirea din fisier a comenzilor care indica ce operatie sa se execute in functie de

-utilizarea unei baze de date formata din cel putin 3 tabele (4 tabele: client, product, orders, si orderitem)

-folosirea operatiilor cu baze de date ( inserarea, stergerea si modificarea datelor ddin tabele)

- utilizarea unui jar file

- creearea pdf-urilor pentru fiecare table (in cazul tabelului OrderItem se va crea cate un pdf la fiecare comanda)

-creearea java doc-ului pentru fiecare clasa existent in proiect

-citirea din fisier a unor instructiuni si utilizarea lor

-impartirea proiectului pe pachete

-creearea a doua fisiere dump (pentru baza de date, unul inainte de popularea tabelelor si altul dupa completarea acestuia)

3.Analiza problemei, modelare, scenarii, cazuri de utilizare

3.1Analiza problemei

In aceasta sectiune vom trata anumite scenarii. Un exemplu ar fi inserarea in tabel a comenzii cdoar daca stocul este sufficient, vde asemenea cand stergem un client sau un produs, este nevoie mai intai sa stergem din tabelele orders si orderitem comanda care contine forein key-ul clientului sau a produsului.

4.Proiectare (decizii de proiectare, diagrame UML, structuri de date, proiectare clase, interfete, relatii, packages, algoritmi, interfata utilizator)

4.1Decizii de proiectare si Structuri de date

Am decis sa modelez acest proiect cu ajutorul a 5 pachete:

-start care are clasa principala (Start) si clasa Parsare (unde se citeste din fisier si se decide ce functii se apeleaza, adica stergere, inserare, update (modificare) sau creare de pdf (report))

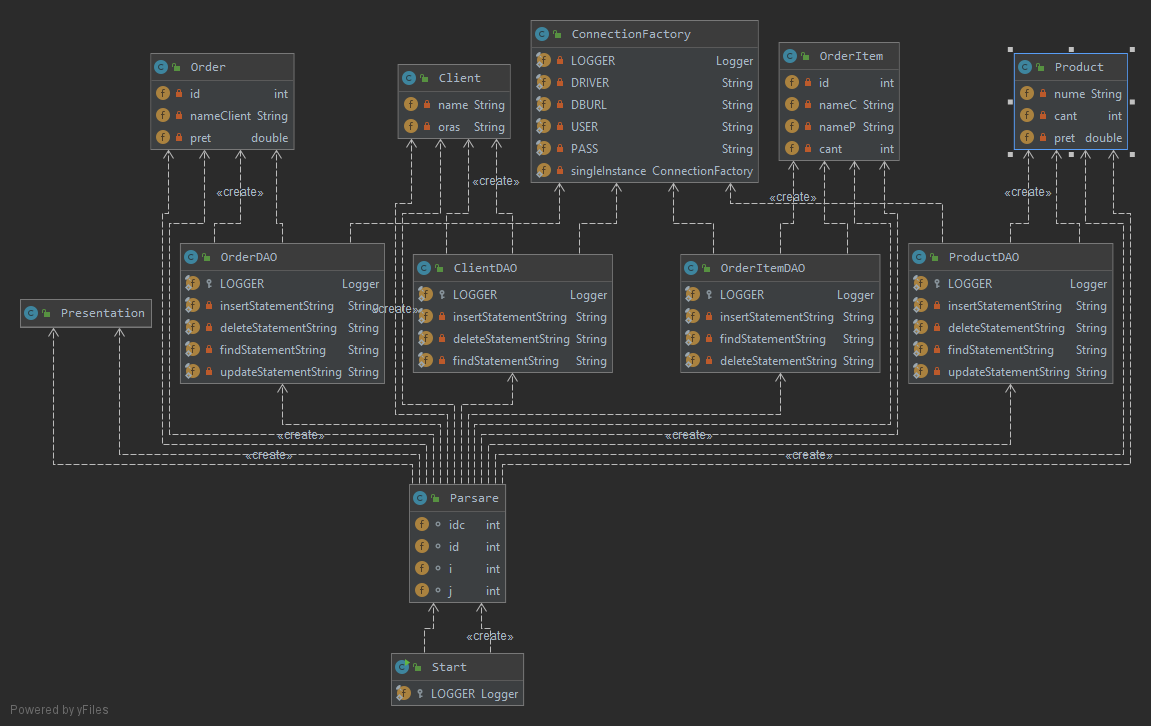
-model are 4 clase ce reprezinta cele 4 tabele existente in baza de date (client, product, orders si orderitem)

-connection are clasa ConnectionFactory care are rolul de a lega baza de date din mysql de programul scris in limbajul JAVA

-dao are 4 clase care fac cele legatura cu ajutorul pachetului connection se fac operatiile de insert delete si update din cele 4 tabele din baza de date

-Presentation are clasa Presentation care, dupa cum ii indica numele prezinta pdf-urile de la fiecare table existent in baza de date

4.2Diagrame UML



5.Implementare

\*Pachetul start

Clasa Start

In aceasta clasa se foloseste doar o metoda statica care arunca o exceptie de tip SQLException, in aceasta metoda se instantiaza un obiect de tip Parsare prin care se apeleaza functia citire din clasa Parsare, dand ca paramentru un string dat de la tastatura (din cmd). Aceste doua instructiuni sunt puse intr-un bloc try catch pentru a prinde exceptia posibila de tip SQLException.

Clasa Parsare

In aceasta clasa se utilzeaza mai multe metode care practic formeaza o singura metoda ce citeste din fisier si in functie de cuvintele gasite se apeleaza diferite metode.

Metoda citire arunca exceptii de tipul DocumentExcepton, SqlException, ClassNotFoundException, se deschide un fisier text f cu numele dat de parametrul dat la metoda si se initializeaza un reader de tip scanner de tip dupa obiectul f prins intr-un bloc try catch (exceptia FileNotFoundException).Cat timp readerul mai are linie necititase citeste linia si se stocheaza intr-un string ,dupa care cu ajutorul metodei predefinite replaceAll se inlocuieste “,” cu stringul “” care reprezinta stringul null. De asemenea se creeaza un sir de stringuri splitString care stocheaza stringurile despartite de “ ” (tasta spatiu) cu ajutorul metodei default split.In urma impartirii creem 4 obiecte (unul de tip ClientDAO, altul ProductDAO,altul OrderDAO si OrderItemDAO). Daca primul string din sirul de stringuri este “Insert” se apeleaza metoda inserare din aceeasi clasa, daca primul string din sirul de stringuri este”Delete” se apeleaza metoda stergere din aceeasi clasa, daca primul string din sirul de stringuri este “Order:” se apeleaza metoda modifica din aceeasi clasa si daca primul string din sirul de stringuri este “Report” se apeleaza metoda report din aceeasi clasa,

Metoda inserare primeste ca parametrii sirul de stringuri un obiect de tip ClientDAO si altul de tip ProductDAO. Daca al doilea string este “client:” atunci se apeleaza metoda insert din clasa ClientDAO (avand ca parametru un obiect de tip client cu numele (SplitString[2]+” ”+ SplitString[3]) si orasul (SplitString[4]) ), daca al doilea string este “product::” atunci cu ajutorul metodei findById se decide daca se apeleaza metoda insert din clasa ProductDAO(ClientDAO (avand ca parametru un obiect de tip client cu numele (SplitString[2]) cantitate(SplitString[3] convertit in int) si prêt(SplitString[4] convertit in double) , sau se apeleaza metoda update din clasa ProductDAO.

Metoda stergere primeste ca parametrii sirul de stringuri un obiect de tip ClientDAO, tip OrderItemDAO, tip OrderDAO si altul de tip ProductDAO. Daca al doilea string este “client:” atunci se apeleaza metoda delete din clasa OrderDAO, din clasa Order si din clasa ClientDAO (avand ca parametru un obiect de tip client cu numele (SplitString[2]+” ”+ SplitString[3]) si orasul (SplitString[4]) ), daca al doilea string este “product::” atunci se apeleaza metoda delete din clasa OrderDAO si din clasa ProductDAO(ClientDAO (avand ca parametru un obiect de tip client cu numele (SplitString[2]) cantitate(SplitString[3] convertit in int) si prêt(SplitString[4] convertit in double) .

Metoda modificare primeste ca parametrii sirul de stringuri un obiect de tip OrderItemDAO, tip OrderDAO si altul de tip ProductDAO. Se initializeaza un obiect orderItem cu id( calculat pe loc prin incrementare cu o unitate), nume (SplitString[2]+” ”+ SplitString[3]) cantitate (SplitString[4] convertit in int) si prêt(SplitString[5]) convertit in double). De asemenea se initializeaza un obiect de tip Presentation. In aceasta metoda folosim metoda insert din clasa OrderItemDAO si in cazul in care este suficient stoc se creeaza un pdf cu datele comenzii (se apeleaza pdf3), si se modifica cantitatea din tabela product cu ajutorul metodei update din clasa ProductDAO altfel se creeaza alt pdf(se apeleaza pdf4) cu mesajul stoc insufficient pe langa datele comenzii.

Metoda report primeste ca parametru sirul de stringuri. Aici se instantiaza un obiect de tip Presentation prin intermediul careia se apeleaza metoda pdf daca al doilea string din sir este “client”, pdf1 daca al doilea string din sir este “product”, pdf2 daca al doilea string din sir este “order”.

\*Pachetul dao

Clasa ClientDAO

Metoda findByName dupa cum ii spune si numele cauta in tabelul din baza de date in coloana name numele dat ca parametru. Cu ajutorul unui obiect de tip connection se apeleaza metoda prepareStatement care cauta statementul respectiv, dupa care se retin variabilele in 2 string-uri si re returneaza clientul corespunzator. In cazul in care nu se gaseste in tabel un client cu numele dat se returneaza null.

Metoda insert insereaza cu ajutorul unui obiect de tip connection se apeleaza metoda prepareStatement un client nou in tabelprepareStatement, dupa care se seteaza variabilele in table cu cele pe care le are obiectul de tip client primit ca parametru.

Metoda delete sterge cu ajutorul unui obiect de tip connection se apeleaza metoda prepareStatement un obiect de tip client primit ca parametru gasit in tabel.

Clasa ProductDAO

Metoda findById cauta in tabelul din baza de date in coloana name numele dat ca parametru. Cu ajutorul unui obiect de tip connection se apeleaza metoda prepareStatement care cauta statementul respectiv, dupa care se retin variabilele intr-un string, un int si un double si returneaza product-ul corespunzator. In cazul in care nu se gaseste in tabel un product cu numele dat se returneaza null.

Metoda insert insereaza cu ajutorul unui obiect de tip connection se apeleaza metoda prepareStatement un product nou in table, dupa care se seteaza variabilele in table cu cele pe care le are obiectul de tip product primit ca parametru.

Metoda delete sterge cu ajutorul unui obiect de tip connection se apeleaza metoda prepareStatement un obiect de tip product primit ca parametru gasit in tabel.

Metoda update modifica cu ajutorul unui obiect de tip connection se apeleaza prepareStatement un obiect de tip product modificand doar cantitatea acestuia adunand doar valoarea celui de-al doilea parametru, cantitatea.

Clasa OrderDAO

Metoda findByName cauta in tabelul din baza de date in coloana name numele dat ca parametru. Cu ajutorul unui obiect de tip connection se apeleaza metoda prepareStatement care cauta statementul respectiv, dupa care se retin variabilele intr-un string, un int si un double si returneaza Orderul-ul corespunzator. In cazul in care nu se gaseste in tabel un order cu numele dat se returneaza null.

Metoda insert insereaza cu ajutorul unui obiect de tip connection se apeleaza metoda prepareStatement un product nou in tabel, dupa care se seteaza variabilele in table cu cele pe care le are obiectul de tip order primit ca parametru.

Metoda delete sterge cu ajutorul unui obiect de tip connection se apeleaza metoda prepareStatement un obiect de tip Order primit ca parametru gasit in tabel.

Metoda update modifica cu ajutorul unui obiect de tip connection se apeleaza prepareStatement un obiect de tip order modificand doar pretul total al acestuia (coloana pret).

Clasa OrderItemDAO

Metoda findByName cauta in tabelul din baza de date in coloana name numele dat ca parametru. Cu ajutorul unui obiect de tip connection se apeleaza metoda prepareStatement care cauta statementul respectiv, dupa care se retin variabilele 2 string-uri, un int si un double si returneaza OrderulItemul-ul corespunzator. In cazul in care nu se gaseste in tabel un order cu numele dat se returneaza null.

Metoda insert insereaza cu ajutorul unui obiect de tip connection se apeleaza metoda prepareStatement un orderitem nou in tabel, dupa care se seteaza variabilele in table cu cele pe care le are obiectul de tip order primit ca parametru.

Metoda delete sterge cu ajutorul unui obiect de tip connection se apeleaza metoda prepareStatement un obiect de tip OrderItem primit ca parametru gasit in tabel.

\*Pachetul Presentation

Clasa Presentation

In aceasta clasa sunt create functiile ce construiesc pdf-urile cu tabele.

Metoda pdf creeaza un pdf in care se va afisa un tabel avand ca si cap de tabel numele coloanelor din baza de date si restul datelor existente in tabela Client.Acesta s-a creat cu ajutorul instantierii unui obiect de tip Document , unuia de tip PdfTable si unuia de tip PdfCell.Cat timp gasim un nou rand se adayga in tabel un nou client.

Metoda pdf1 creeaza un pdf in care se va afisa un tabel avand ca si cap de tabel numele coloanelor din baza de date si restul datelor existente in tabela Product.Acesta s-a creat cu ajutorul instantierii unui obiect de tip Document , unuia de tip PdfTable si unuia de tip PdfCell.Cat timp gasim un nou rand se adayga in tabel un nou product.

Metoda pdf2 creeaza un pdf in care se va afisa un tabel avand ca si cap de tabel numele coloanelor din baza de date si restul datelor existente in tabela Order.Acesta s-a creat cu ajutorul instantierii unui obiect de tip Document , unuia de tip PdfTable si unuia de tip PdfCell.Cat timp gasim un nou rand se adayga in tabel un nou order.

Metoda pdf3 creeaza cate un pdf in care se va afisa un tabel avand ca si cap de tabel numele coloanelor din baza de date si restul datelor existente in tabela OrderItem.Acesta s-a creat cu ajutorul instantierii unui obiect de tip Document , unuia de tip PdfTable si unuia de tip PdfCell.Cat timp gasim un nou rand se adayga in tabel un nou OrderItem cu stoc suficient.

Metoda pdf4 creeaza cate un pdf in care se va afisa un tabel avand ca si cap de tabel numele coloanelor din baza de date si restul datelor existente in tabela OrderItem.Acesta s-a creat cu ajutorul instantierii unui obiect de tip Document , unuia de tip PdfTable si unuia de tip PdfCell.Cat timp gasim un nou rand se adayga in tabel un nou OrderItem cu stoc insuficient.

\*Pachetul connection

Clasa ConnectionFactory

Aceasta clasa are rolul de a face conexiunea cu baza de date.Aici se initializeaza variabilele string Driver, Dburl, User, si Pass.

Metoda createConnection creeaza conexiunea la baza de date cu ajutorul variabilelor mentionate mai sus.

Metoda getConnection este un getter

Metoda close are rolul de a inchide conexiunea. De asemenea mai avem o metoda close care o suprascrie pe precedent, aceasta din urma inchide statementul. Totodata mai avem o metoda close care o suprascrie pe precedent, aceasta din urma inchide resultSet.

\*Pachetul model

Clasa Client

Aceasta clasa contine doua variabile private care reprezinta coloanele tabelului Client. De asemenea, clasa are getteri pentru cele doua variabile. Metoda toString are rolul de a afisa datele din aceasta clasa.

Clasa Product

Aceasta clasa contine trei variabile private care reprezinta coloanele tabelului Product. De asemenea, clasa are getteri pentru cele trei variabile. Metoda toString are rolul de a afisa datele din aceasta clasa.

Clasa Order

Aceasta clasa contine trei variabile private care reprezinta coloanele tabelului Order. De asemenea, clasa are getteri si setteri pentru cele trei variabile. Metoda toString are rolul de a afisa datele din aceasta clasa.

Clasa OrderItem

Aceasta clasa contine patru variabile private care reprezinta coloanele tabelului OrderItem. De asemenea, clasa are getteri si setteri pentru cele patru variabile. Metoda toString are rolul de a afisa datele din aceasta clasa.

6.Testare

Programul a fost testat cu urmatorul text in ficierul de intrare:

Insert client: Ion Popescu, Bucuresti

Insert client: Luca George, Bucuresti

Report client

Insert client: Sandu Vasile, Cluj-Napoca

Report client

Delete client: Ion Popescu, Bucuresti

Report client

Insert product: apple, 20, 1

Insert product: peach, 50, 2

Insert product: apple, 20, 1

Report product

Delete Product: peach

Insert product: orange, 40, 1.5

Insert product: lemon, 70, 2

Report product

Order: Luca George, apple, 5

Order: Luca George, lemon, 5

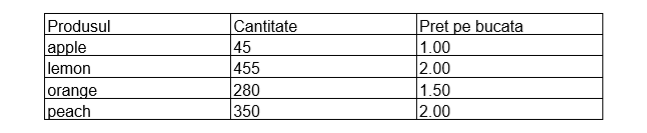
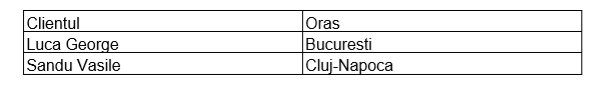
Order: Sandu Vasile, apple, 100

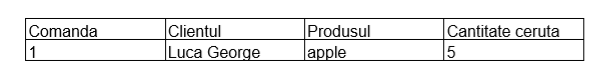
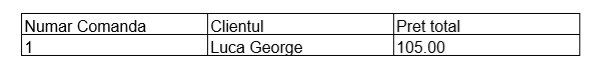
Report client

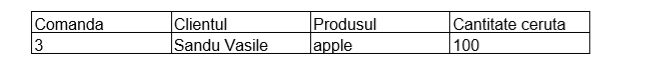
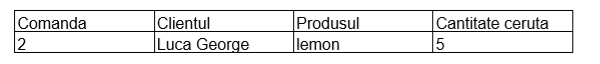
Report order

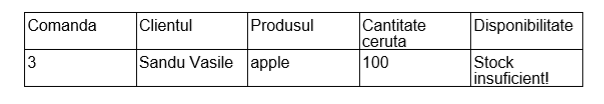
Report product

Programul a fost generat cu success, generand-use toate pdf-urile de mai jos:









7.Concluzie

In concluzie programul a fost finalizat cu success, in timpul dezvoltarii acestui proiect am invatat sa lucrez cu baza de date, sa creez pdf-uri si JAVA-Doc-uri. Ca dezvoltari ulterioare ar putea fi optimizarea programului, evitarea unor exceptii care apar din cauza duplicarii primary key(sa il puna in tabel doar daca primary key-ul nu e folosit).

7.Bibliografie

Am lucrat pe scheletul primit ca support, linkul este mai jos. <http://coned.utcluj.ro/~salomie/PT_Lic/4_Lab/Assignment_3/Assignment_3_Indications.pdf>

De asemenea m-am documentat pentru creearea pdf-urilor de pe site-ul: <http://thinktibits.blogspot.com/2012/12/iText-JDBC-SQL-Table-PDF-Report-Java-Example-Program.html>