DOCUMENTATIE

TEMA 3

NUME STUDENT:Cândea Claudiu
GRUPA:30227

CUPRINS

1. Obiectivul temei

Implemetarea unei aplicatii pentru gestiunea comezilor unui magazine. Stocare datelor despre clienti, poduse si comezi in tabele unei baze de date conectata la aplicatie.

Obiective secundare:

a) Analiza problemei si identificare cerintelor
b) Proiectare aplicatiei de gestiune a comenzilor
c) Implementare aplicatiei de gestiune a comenzilor
d) Testare aplicatiei de gestiune a comenzilor

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

Cerintele functionale ale proiectului:

- Aplicatia ar trebui sa permita operatiile de inserare, stergere, actualizare si selectie asupra tabelului Client din baza de date;
- Aplicatia ar trebui sa permita operatiile de inserare, stergere, actualizare si selectie asupra tabelului Product din baza de date;
- Aplicatia ar trebui sa permita operatiile de inserare si selectie asupra tabelului Orders din baza de date;

Cerintele non-functionale ale proiectului:

- Interfata grafica a plicatiei ar trebui sa fie intuitiva si usor de folosit de catre utilizator;

Use Case: inserare in tabelul Client

Actor principal: Utilizatorul;

Scenariu de success:

1. Din interfata grafica, utilizatorul selecteaza operatii pe tabelul Client
2. In noua fereastra deschisa acesta va introduce datele clientului ce se va fi inserat
3. Va alege optiunea de insert, iar nou client se va introduce in baza de date

Use Case: actualizare in tabelul Client

Actor principal: Utilizatorul;

Scenariu de success:

1. Din interfata grafica, utilizatorul selecteaza operatii pe tabelul Client
2. In noua fereastra deschisa acesta va introduce id-ul clientului ce se doreste a fi actualizat si noile date
3. Va alege optiunea de update, iar datele clientului din baza de date se vor modifica

Use Case: sterger in tabelul Client

Actor principal: Utilizatorul;

Scenariu de success:

1. Din interfata grafica, utilizatorul selecteaza operatii pe tabelul Client
2. In noua fereastra deschisa acesta va introduce id-ul clientului ce se doreste a fi sters
3. Va alege optiunea de delete, iar datele clientul se vor sterge din baza de date

Use Case: selectie in tabelul Client

Actor principal: Utilizatorul;

Scenariu de success:

1. Din interfata grafica, utilizatorul selecteaza operatii pe tabelul Client
2. In noua fereastra deschisa acesta va introduce id-ul clientului ce doreste sa il vizualizele sau va selecta optiunea findAll pentru a vizualiza toti clientii
3. Se va deschide o noua fereastra in care rezultatul selectiei va fi afisat sub forma tabelara

Use Case: inserare in tabelul Product

Actor principal: Utilizatorul;

Scenariu de success:

1. Din interfata grafica, utilizatorul selecteaza operatii pe tabelul Prodcut
2. In noua fereastra deschisa acesta va introduce datele produsului ce se va fi inserat
3. Va alege optiunea de insert, iar nou produs se va introduce in baza de date

Use Case: actualizare in tabelul Product

Actor principal: Utilizatorul;

Scenariu de success:

1. Din interfata grafica, utilizatorul selecteaza operatii pe tabelul Product
2. In noua fereastra deschisa acesta va introduce id-ul produsului ce se doreste a fi actualizat si noile date
3. Va alege optiunea de update, iar datele produsului din baza de date se vor modifica

Use Case: sterger in tabelul Product

Actor principal: Utilizatorul;

Scenariu de success:

1. Din interfata grafica, utilizatorul selecteaza operatii pe tabelul Product
2. In noua fereastra deschisa acesta va introduce id-ul produsului ce se doreste a fi sters

3. Va alege optiunea de delete, iar datele produsului se vor sterge din baza de date

Use Case: selectie in tabelul Product

Actor principal: Utilizatorul;

Scenariu de success:

1. Din interfata grafica, utilizatorul selecteaza operatii pe tabelul Product
2. In noua fereastra deschisa acesta va introduce id-ul produsului ce doreste sa il vizualizele sau va selecta optiunea findAll pentru a vizualiza toate produsele
3. Se va deschide o noua fereastra in care rezultatul selectiei va fi afisat sub forma tabelara

Use Case: inserare in tabelul Order

Actor principal: Utilizatorul;

Scenariu de success:

1. Din interfata grafica, utilizatorul selecteaza operatii pe tabelul Order
2. Acesta va selecta un client si un produs si va introduce o cantitate
3. Va alege optiunea de createOrder si se introduce in baza de date o noua comanda in functie de clientul si produsul ales

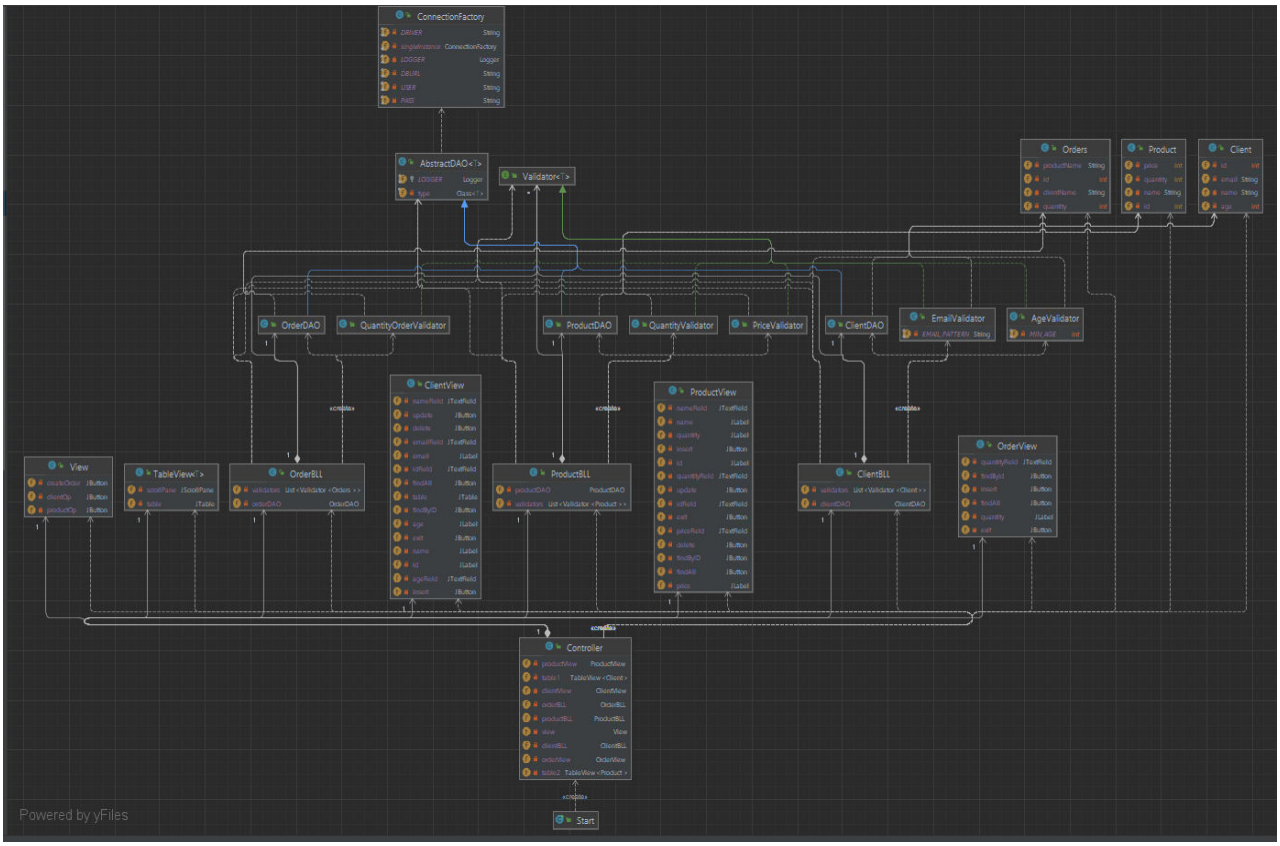
Use Case: selectie in tabelul Order

Actor principal: Utilizatorul;

Scenariu de success:

1. Din interfata grafica, utilizatorul selecteaza operatii pe tabelul Order
2. In noua fereastra deschisa acesta va selecta optiunea findAll pentru a vizualiza toate comenzile
3. Se va deschide o noua fereastra in care rezultatul selectiei va fi afisat sub forma tabelara

3. Proiectare



Clasa Controller implementeaza interfata ActionListener pentru a putea pune ascultatori asupra elementelor din Clasele de UI.

Am definit o nou interfata, Validator, ce are o singura metoda, validate. Aceasta metoda primeste un obiect si ar trebui sa spuna daca datele acestuia sunt valide sau nu. Aceasta interfata este implementata de clasele din pachetul Validators: AgeValidator, EmailValidator, PriceValidator, QuantityValidator, QuantityOrderValidator.

Ca structuri de date, in mai multe clase ale proiectului s-au folosit liste, de cele mai multe ori ArrayList-uri.

4. Implemetare

Clasa ConnectionFactory

In aceasta clasa se defines metode pentru realizarea conexiuni cu baza de date. Metodele sunt createConnection, getConnection, si 3 metode close care primesc ca parametru fie o conexiune, fie un statement, fie un resultSet pe care mai apoi le vor inchide.

Clasele din pachetul Model: Client, Product, Orders

Aceste 3 clase modeleaza tabele cu aceleasi nume din baza de date. Campurile lor sunt identice cu coloanele tabelului corespunzator. Ca metode contin doar getters si setters pentru campurile lor

Pachetul DAO: Clasa AbstractDAO

Este o clasa gennerica se foloseste de reflection pentru a creea query-uri in specifice in functie de tipul obiectului cu care lucreaza si mai apoi le executa.

Metodele createSelectQuery, createFindAllQuery si createDeleteQuery creaza query-uri pentru delete si select in fuctie de genericul T.

Metoda findAll executa un query care returneaza toate randurile dintr-un tabel. Returneaza o list cu obiecte de tip T.

Metoda findById primeste un id si executa un query care returneaza randul dintr-un tabel al carui id este egal cu parametrul metodei. Returneaza o valoare generica de tip T.

Metoda insert primeste un parametru generic T si executa un query de inserare pe unul din tabele bazei de date. Se insereaza valorile campurilor parametrului generic. Metoda returneaza id-ul randului inserat.

Metoda delete primeste un id si executa un query asurpa unuia dintre tabele pentru stergea randului cu id-ul respectiv. Returneaza id-ul randului sters.

Metoda update primeste un parametru de tip generic T si executa un query de actualizare asupra unui tabele. Randul actualizat va avea valorile coloanelor egale cu valorile campurilor parametrului generic t.

Pachet DAO: ClientDAO

Clasa mosteneste AbstractDAO.

Clasa aceasta creaza query-uri specifice pentru tabelul Client.

Metoda createInsertQuery creaza un query de inserare, iar metoda createUpdateQuery creaza un query de actualizare. Ambele nu au parametri si returneaza un string.

Metoda setInsertValues primeste ca parametru un statement si un client. Valorile campurilor clientului sunt puse pe pozitile valorilor ce trebuie inserate din statement.

Metoda setUpdateValues primeste aceasi 2 parametrii ca si metoda setInsertsValues, doar ca valorile campurilor clientului sunt adaugate intr-un statement de update.

Pachet DAO: ProductDAO

Clasa mosteneste AbstractDAO.

Clasa aceasta creaza query-uri specifice pentru tabelul Product.

Metoda createInsertQuery creaza un query de inserare, iar metoda createUpdateQuery creaza un query de actualizare. Ambele nu au parametri si returneaza un string.

Metoda setInsertValues primeste ca parametru un statement si un product. Valorile campurilor produsului sunt puse pe pozitile valorilor ce trebuie inserate din statement.

Metoda setUpdateValues primeste aceasi 2 parametrii ca si metoda setInsertsValues, doar ca valorile campurilor produsului sunt adaugate intr-un statement de update.

Pachet DAO: OrderDAO

Clasa mosteneste AbstractDAO.

Clasa aceasta creaza query-uri specifice pentru tabelul Orders.

Metoda createInsertQuery creaza un query de inserare.Nu are parametri si returneaza un string.

Metoda setInsertValues primeste ca parametru un statement si un order. Valorile campurilor comezii sunt puse pe pozitile valorilor ce trebuie inserate din statement.

Pachetul Bll: pachetul Validators

In acest pachet sunt clase care implementeaza interfata Validator si implicit metoda validate.

Clasele de validare pentru obiectele de tip Client sunt: AgeValidator, care se asigura ca clientul are varsta minima de 10 ani, si EmaiValidator, care se asigur ca email-ul clientului respecta un anumit format.

Clasele de validate pentru obiectele de tip Product sunt: PriceValidator, care se asigur ca pretul produsului nu e negativ, si QuantityValidator, care se asigura ca cantitatea produsului nu e negativa.

Clasa de validate pentru obiectele de tip Orders e QuantityOrderValidator, care se asigura ca cantitate de produse de pe comanda nu e negativa.

Toate aceste clase arunca o exceptie in cazul in care obiectul nu e valid.

Pachetul Bll: clasa ClientBLL

Aceasta clasa are ca si campuri o lista de validatori specifici pentru obiectele de tip client si un obiect instanta a clasei ClientDAO.

Sunt implementeate metode care apleaza metodele de inserare, update, delete, findAll si findById din din clasa ClientDAO. Metodele de insert si update primesc ca parametru un client pe care il si valideaza folosind obiectele din lista de validare a clasei.

Pachetul Bll: clasa ProductBLL

Aceasta clasa are ca si campuri o lista de validatori specifici pentru obiectele de tip Product si un obiect instanta a clasei ProductDAO.

Sunt implementeate metode care apleaza metodele de inserare, update, delete, findAll si findById din din clasa ProductDAO. Metodele de insert si update primesc ca parametru un produs pe care il si valideaza folosind obiectele din lista de validare a clasei.

Pachetul Bll: clasa OrderBLL

Aceasta clasa are ca si campuri o lista de validatori specifici pentru obiectele de tip Orders si un obiect instanta a clasei OrderDAO.

Sunt implementeate metode care apleaza metodele de inserare si findAll din clasa OrderDAO. Metoda de insert primeste ca parametru o comanda pe care o si valideaza folosind obiectele din lista de validare a clasei.

Metoda createPDF primeste ca paramentru o comanda si o valoare de tip int numita totalPrice. Daca comanda e valida atunci creaza un pdf care reprezinta factura pentru comanda respectiva.

Pachetul Presentation: Clasa View

Clasa mosteneste JFrame si este fereastra principala a aplicatiei.

In aceasta clasa sunt 3 butoane pentru a selecta tabelul asupra caruia vrem sa facem operatii: Client, Product sau Orders.

Clasa are ca metode doar getters pentru cele 3 butoane.

Pachetul Presentation: Clasa ClientView

Clasa mosteneste JFrame.

In aceasta clasa, care implementeaza interfata grafica pentru operatiile din tabelul Client, avem textField-uri pentru a introduce id-ul, numele, varsta si email-ul si butoane pentru operatii : insert, delete, update, findById, findAll. De asemenea avem si un buton numit exit care inchide aceasta fereastra si o deschide pe cea principala. Are metode de get pentru butoane si textField-uri.

Pachetul Presentation: Clasa ProductView

Clasa mosteneste JFrame.

In aceasta clasa, care implementeaza interfata grafica pentru operatiile din tabelul Product, avem textField-uri pentru a introduce id-ul, numele, cantitatea si pretul si butoane pentru operatii : insert, delete, update, findById, findAll. De asemenea avem si un buton numit exit care inchide aceasta fereastra si o deschide pe cea principala. Are metode de get pentru butoane si textField-uri.

Pachetul Presentatiom: Clasa OrderView

Clasa mostenteste JFrame.

In aceasta clasa, care implementeaza interfata grafica pentru operatiile din tabelul Orders, avem un textField pentru a introduce cantitate si butoane pentru operatiile de creatOrder si findAll. De asemenea avem si un buton numit exit care inchide aceasta fereastra si o deschide pe cea principala. Are metode de get pentru butoane si textField.

Pachetul Presentation: Clasa TableView

Clasa mostenteste JFrame.

Aceasta clasa contine tabelul in care se afiseaza rezultatele operatiilor de select.

Metoda createTable primeste ca parametru o lista de obiecte generice(de tip T) si folosind reflection creaza un nou tabel pe care il returneza.

Pachetul Presentation: Clasa Controller

Aceasta clasa are ca si campuri un view, un clientView, un productView, un orderView , 2 tableView-uri, unul pentru obiecte de tip Client si unul pentru obiecte de tip Product, si cate un obiect din clasele ClientBLL, ProductBLL, OrderBLL.

Implementeaza interfata ActionListener si se adauga pe sine ca si ascultator pentru obiectele din ferestrele interfetei grafice.

5. Rezultate

Ca mod de testare aplicatia a fost conectata la o baza de date cu 3 table: Client, View si Orders si pentru fiecare table s-au testat operatiile implementate ce se pot executa asupra lui. Pentru tabelul Client si Product operatiile: inset, delete, update, findById si findAll. Pentru tabelul Orders operatiile: insert (create order) si findAll. Toate aceste operatii au reusit intrucat tabelele din baza de date au suferit modificarile astepate dupa fiecare operatie.

6. Concluzii

In urma acestui proiect am invatat cum se conecteza o aplicatie java la o baza de date si cum putem extrage sau modifica informatiile din baza de date din interiorul aplicatie. De asmenea, am invatat cum se lucreaza cu reflection si cu clase generice.

Fata de celelate teme aceasta a fost putin mai complicat, insa consider ca in urma realizarii ei am dobandit niste abilitati noi si mi-am aprofundat anumite cunstinte de programare.

7. Bibliografie

https://jenkov.com/tutorials/java-reflection/index.html

https://www.baeldung.com/javadoc

https://www.baeldung.com/java-pdf-creation

https://www.baeldung.com/java-jdbc

https://mkyong.com/jdbc/how-to-connect-to-mysql-with-jdbc-driver-java/

https://dzone.com/articles/layers-standard-enterprise

https://gitlab.com/utcn_dsrl/pt-reflection-example