Universitatea Tehnică din Cluj-Napoca

Facultatea de Automatică și Calculatoare

Gestionarea comenzilor

Matei Anechitei

Grupa 30221

1.Obiectivul temei

Proiectati o aplicatie de gestionare a comenzilor folosita in procesarea comenzilor unor clienti pentru un depozit.

Detalii:

* Se foloseste o baza de date relationala pentru a stoca produsele, clientii si comenzile.
* Aplicatia va fi structurata in pachete folosind o arhitectura pe nivele si folosind urmatoarele tipuri de clase:
  + Model - modelele de date ale aplicatiei;
  + Business Logic – implementarea logicii aplicatiei;
  + Presentation – implementarea I/E
  + Data access – implementarea accesului la baza de date.
* Aplicatia proceseaza comenzi primite dintr-un fisier text dat ca argument, apoi executa opratiile cerute, salveaza date in baza de date si genereaza rapoarte in format pdf.

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

Pentru a putea realiza aplicatie, avem nevoie de o baza de date relationala (in acest caz folosim MySQL), si pachete speciale pentru a realiza conexiunea intre aplicatia Java si baze de data si generat rapoarte pdf.

3.Proiectare

Pentru o structurare cat mai clara a aplicatiei, folosesc 6 pachete diferite, fiecare cu 1 sau mai multe clase specializare:

1)start – Pachetul start contine 2 clase, Start si Reflection. Clasa Start e folosita ca main-ul programului, avand rolul doar de a prelua numele fisierului text de intrare prin argument si executarea procesarii acestuia. Clasa Reflection contine o metoda ce inlocuieste toString(), putand fi folosita pentru orice obiect cu scopul de a-l afisa cat mai clar.

2)connection – Pachetul connection contine doar clasa ConnectionFactory. Rolul ei este sa realizeze si sa inchida conexiunea la baza de date a aplicatiei.

3)model – Pachetul model contine 3 clase, specifice intrarilor in baza de date: Client, Product, Order.

4)dataaccess – Pachetul dataaccess contine 4 clase care au rolul de crea si executa query-urile SQL necesare.

5)businesslogic – Pachetul businesslogic verifica corectitudinea comenzilor primite si genereaza rapoartele cerute unde este cazul.

6)presentation – Pachetul presentation proceseaza fisierul text dat in comenzi ce pot fi recunoscute de metodele din pachetul businesslogic.

Rularea aplicatiei se poate face atat dintr-un IDE, setand argumentele de intrare, dar exista si un fisier .jar care poate fi rulat din linia de comanda conform sintaxei:



4.Implementare

In cele ce urmeaza sunt prezentate clasele din pachete care contin metode importante.

presentation.CommandControl

Clasa presentation.CommandControl cotine 3 metode folosita la parsarea fisierului text in comenzi, apoi executarea acestora prin intermediului metodelor din pachetul businesslogic. Cele 3 metode sunt: fileProcessing, commandSplit, commandProcessing.

Metoda fileProcessing preia fisierul text dat ca intrare si il parseaza, introducand intr-o lista fiecare linie a acestuia.

Metoda commandSplit preia o linie a fiserului text din lista si o separa in mai multe stringuri referitoare la numele comenzii si campurile necesare executiei acesteia cau ajutorul unui RegEx.

Metoda commandProcessing preia stringurile returnate de metoda precedenta si trimite nivelului buisnesslogic comanda pentru a fi verificata.

model.Client

Clasa model.Client este folosita pentru stocarea unui client cu toate campurile regasite si in baza de date.

In afara de constructori, setter-e si getter-e, clasa mai contine si un @Override al metodei toString() folosit special pentru crearea unui string valid pentru query-uri de tip SQL INSERT.

model.Product

Clasa model.Product este folosita pentru stocarea unui produs cu toate campurile regasite si in baza de date.

In afara de constructori, setter-e si getter-e, clasa mai contine si un @Override al metodei toString() folosit special pentru crearea unui string valid pentru query-uri de tip SQL INSERT.

model.Order

Clasa model.order este folosita pentru stocarea unui comanda cu toate campurile regasite si in baza de date.

In afara de constructori, setter-e si getter-e, clasa mai contine si un @Override al metodei toString() folosit special pentru crearea unui string valid pentru query-uri de tip SQL INSERT.

dataaccess.AbstractDAO

Clasa dataaccess.AbstractDAO cotine mai multe metode facute pentru crearea query-urilor de tip SQL SELECT, SQL INSERT si SQL DELETE si executarea acestora in baza de date. Datorita abstractizarii acesteia, query-urile prezentate sunt valide pentru toate cele 3 tipuri de obiecte specifice tabelelor din baza de date:Client, Product, Order.

Metodele createSelectQuery, createInsertQuery, createDeleteQuery sunt responsabile pentru construirea query-urilor ce trebuie sa fie executate.

Metodele insert, delete realizeaza insertia/stergerea unui obiect in baza de date.

Metoda createObjects creaza un obiect Java primind un ResultSet rezultat in urma unui query in baza de date.

Metoda findByName se foloseste de SQL SELECT si createObjects pentru a returna un obiect a carui nume se regaseste in baza de date.

dataaccess.ClientDAO/

dataaccess.ProductDAO/

dataaccess.OrderDAO

Clasele dataccess.ClientDAO/dataaccess.ProductDAO/dataaccess.OrderDAO extind AbstractDAO astfel ca metodele sa poata fi folosite individual pentru fiecare tip de obiect in parte. Pe langa aceste metode, clasa ProductDAO mai contine 2 metode: createProductUpdateQuery, update. Prima metoda creeaza un query de tip SQL UPDATE prin care cantitatea produsului este modificata, iar a doua metoda ruleaza query-ul creat.

businesslogic.ClientBLL

Clasa businesslogic.ClientBLL contine verificarea operatiilor si apelarea executiilor acestora pentru comenzile specifice clientilor. Contine 3 metode: insertClient, deleteClient, reportClient. Metoda reportClient genereaza un pdf care contine in forma tabulara tot table-ul din baza de date al clientilor.

businesslogic.ProductBLL

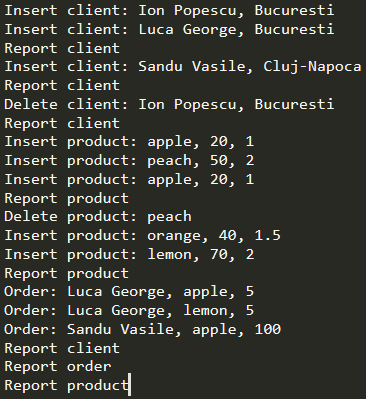
Clasa businesslogic.ProductBLL contine verificarea operatiilor si apelarea executiilor acestora pentru comenzile specifice produselor. Contine 3 metode: insertProduct, deleteProduct, reportProduct. Metoda reportProduct genereaza un pdf care contine in forma tabulara tot table-ul din baza de date al produselor.

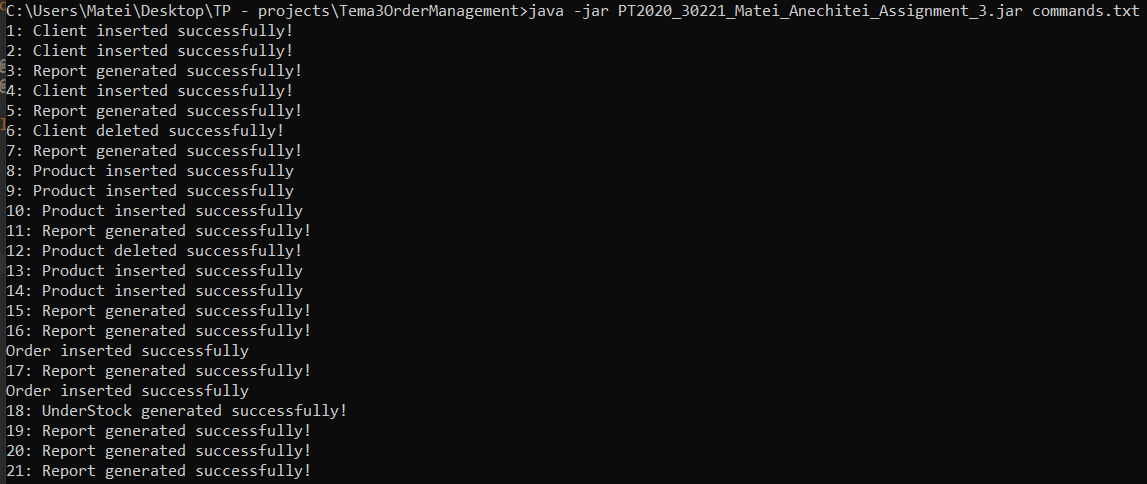
businesslogic.OrderBLL

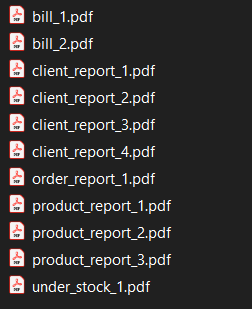
Clasa businesslogic.OrderBLL contine verificarea operatiilor si apelarea executiilor acestora pentru comenzile specifice comenzilor. Contine 4 metode: insertOrder, generateBill, generateUnderStock, reportOrder. Metoda reportOrder genereaza un pdf care contine in forma tabulara tot table-ul din baza de date al comenzilor. Metoda generateBill genereaza un pdf care contine ordinul de plata al oricarei comenzi. Metoda generateUnderStock genereaza un pdf in cazul in care o comanda de tip Order pentru care cantitatea ceruta nu este disponibila.

5.Rezultate

In fisierul cu proiectul se regaseste un foler numit Reports cu toate pdf-urile generate in urma executiei fisierului commands.txt. Continutul fisierului poate fi vazut mai jos, impreuna cu afisarile din linia de comanda si continutul folder-ului Reports.







6.Concluzii

In urma implementarii acestui program am invatat sa folosesc baze de date intr-un program Java, si anume conexiunea intre cele 2, crearea de string-uri folosite pentru diferitele SQL Queries, rularea acestora. Totodata, am revenit asupra RegEx-urilor si a organizarii in pachete, astfel avand o arhitectura pe nivele, bazata si pe concepte de abstractizare si reflexie, folositoare la generalizarea opreatiilor.

In fisierul programului, pe langa aplicatia in sine se regasesc si ambele .jar-uri necesare executarii acestuia, folerul cu toate rapoarte generate pe baza fisierului commands.txt, fisierul .jar din care se poate rula aplicatia, fisierele JavaDoc, un dump al bazei de date si aceasta documentatie.

Posibile imbunatatiri:

* Imbunatatirea securitatii bazei de date;
* Adaugarea mai multor tipuri de comenzi posibile;
* Asigurarea functionarii programului si in cazul micilor iregularitati la nivel de sintaxa a comenzilor, programul actual fiind destul de rigid din punct de vedere al format-ului acceptat.

7.Bibliografie

* <http://coned.utcluj.ro/~salomie/PT_Lic/4_Lab/Assignment_3/Assignment_3.pdf>
* <http://coned.utcluj.ro/~salomie/PT_Lic/4_Lab/Assignment_3/Assignment_3_Indications.pdf>
* <https://bitbucket.org/utcn_dsrl/pt-reflection-example/src/master/>
* <https://www.baeldung.com/javadoc>
* <https://dev.mysql.com/downloads/connector/j/5.1.html>
* <https://jar-download.com/artifacts/com.itextpdf/itextpdf/5.5.9/source-code>