

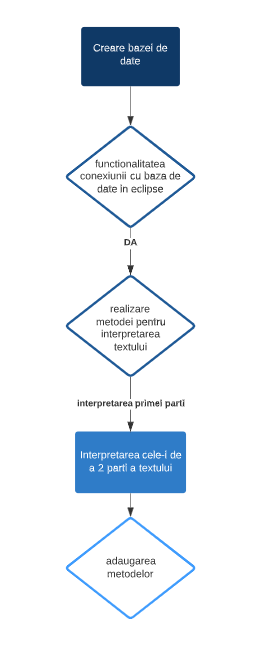
Consider an application OrderManagement for processing customer orders for a warehouse. Relational databases are used to store the products, the clients and the orders.

1. Obiectivul temei

Obiectivul principal al temei este realizarea unei aplicatii in java care sa preia comenzi dintr-un text file si sa le proceseze operatiile, sa salveze data intr-o baza de date sis a genereze rapoarte in format pdf.

Obiective secundare:

1. Crearea bazei de date – capitol 4
2. Stabilirea conexiunii cu baza de date – capitol 4
3. Creare metodelor pentru interpretare comenzilor din fisierul text -capitol 4
4. Crearea metofelor pentru procesare comenzilor – capitol 4
5. Adaugarea librariei pentru generare de fisiere pdf si crearea metofelor pentru a scrie in pdf-uri.
6. Analiza problemei, modelare, scenario, cazuri de utilizare



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

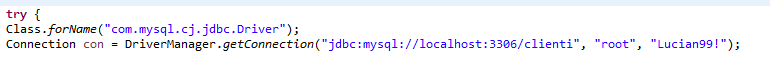
Deoarece proiectul meu este format dintr-o singura casa principala nu pot realiza o diagrama UML a acestuia.

Una din principalele decizii de proiectare a fost realizarea proiectului intr-o singura clasa.

Cu ajutorul interpretorului de text format din 2 metode reusesc sa impart fiecare linie in CERINTA si ARGUMENTE

Pe masura ce interpretez fisierele text cu ajutorul unui PARAGRAF de if then else-uri apelez metoda necesara pentru implementarea cerintei descrisa de fiecare linie.

Conectiunea cu baza de date o realizez cu ajutorul librariei mysql-connector-java

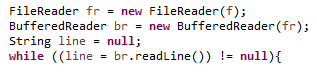


In principiu fiecare cerinta din fisierul txt este interpretata cu ajutorul unei metode diferite. In aceste metode dupa ce creez o conexiune declar pentru fiecare cerinta in parte un statement in limbaj MySql pe care il folosesc pentru updatarea valorilor din baza de date.

Rapoartele Pdf sunt create cu ajutorul librarieie itextpdf. Pentru realizarea rapoartelor folosesc o cate o metoda diferita. Ca la fiecare metoda, creez initial o conexiune dupa care imi declar un ResultSet egal cu cee ace imi returneaza executarea statementului. Dup ace obtin toate valorile de care am nevoie cu ajutorul unui while creez pentru fiecare set de valori din TABLE realizez cate un nou paragraph pe care il populez cu getString/getInt.

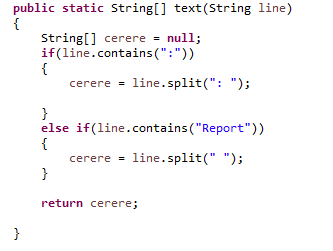
1. Implementare

Pentru implementarea metodei care citeste din fisierul text folosesc un FileReader si FileReader.



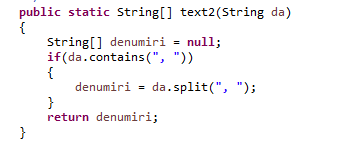
Pentrun implementarea celor 2 metode care prelucreaza textul primit de la fisier declar un array in care, daca in cazul in care cerinta contine “:” atunci stringul/cerinta va fi impartita in 2 parti cu ajutorul .split.

In caz contrar atunci stringul cu cererea se va imparti dupa primul space(in 2 cuvinte)



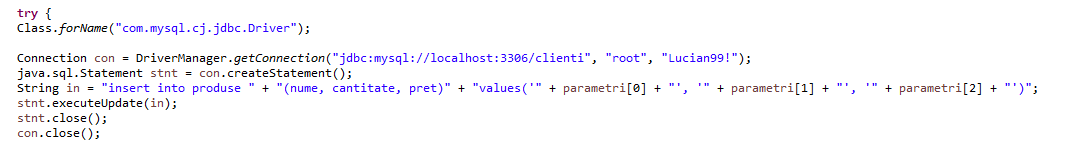
In continuare dup ace am rezolvat un caz pentru prelucrarea textului(rapoartele)/ cele carenu contin “;” acum am creat o metoda pentru a imparti cele 2/3 argumente in functie de cerinta pe care o primeste.

Am declarant un array si am oferita ca parametru functiei cea de a 2 parte a stringul returnat de prima metoda. In aceasta metoda, in cazul in care stringul contine “. “ atunci in array-ul declarant mai sus se imparte pe fiecare pozitie care un parametru.

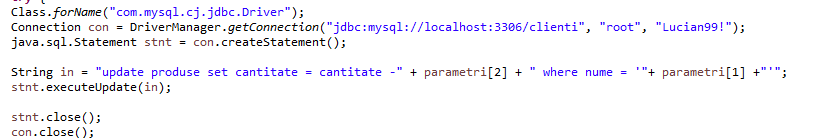


O data ce am rezolvat problema interpretarii textului am creat cate un if pentru fiecare caz posibil cerut in txt. In cazul in care if-ul este indeplinit atunci se apeleaza metoda(cu sau fara parametri in functie de caz).

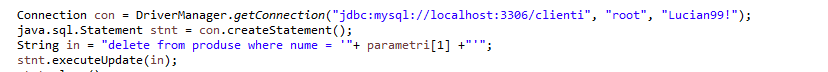
Metodele pentru inserare a clientilor a produselor si a comenzilor sunt asemanatoare. Inititl creez conexiunea cu baza de date dupa care cu ajutorul unui statement si a parametrilor(cei cititi din fisierul text) se introduce informatii in baza de date.



O alta metoda ajutatoare celei pentru inserarea comenzilor este metoda care updateaza cantitatea produselor pe baza carora s-a plasat comanda. In aceasta metoda am reusit sa fac scaderea tot in limbaj sql cu ajutorul unui query statement.



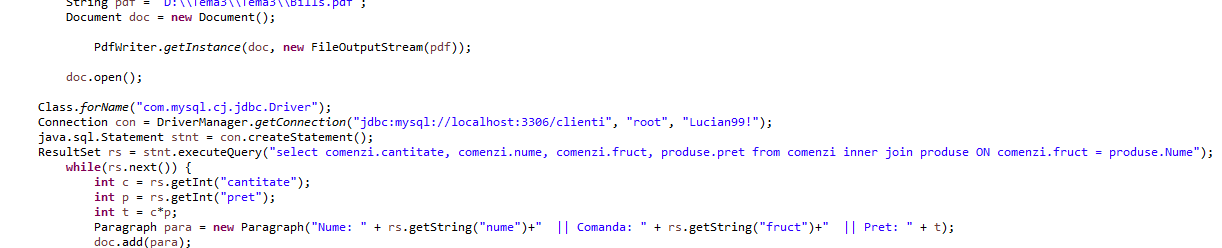
Asemanatoare cu metoda de mai sus este si metoda folosita pentru stergerea din baza de date.



Pentru metodele folosite la generarea rapoartelor pdf am avut nevoie de libraria itextpdf. Metodele sunt asemanatoare pentru fiecare dintre cele 3 rapoarte. In acestemetode initial creez o noua conexiune la baza de date. Iar cu ajutorul unui ResultSet pe care il egalez cu ce imi returneaza executarea statementului reusesc sa fac rost de datele de care am nevoie pentru a adauga in paragrafe valorile cerute.



Ultima metoda implementata este cea pentru generarea facturilor. In aceasta metoda pe langa faptul ca folosesc statements si ResultSet imi declar cate un int pentru a ma ajuta sa calculez totalul preturilor comenzilor.



1. Rezultate.

Dupa ce rulez proiectul se populeaza baza de date si mi se genereaza pdf-urile cu fiecare raport si facturile. Testarea am realizat-o ca si jar file si normal, din eclipse, cu ajutorul fisierului text.

1. Concluzii

In concluzie tema este functionala aproape in totalitate, singurul lucru care nu functioneaza fiind generarea SEPARATA a tuturor bill-urilor.

Consider ca principalul draw-back este alegere initiala de a dezvolta toata aplicatia intr-o singura clasa. Ca si imbunatatiri viitoare consider ca ar trebui inlocuite majoritatea metodelor cu o singura metoda ajutata de parametri pentru a manipula cerinta, si realizarea aplicatiei pe mai multe clase.

Din aceasta tema am invatat cum sa lucrez in java cu o baza de date, cum se genereaza pdf-urile in java si manipularea string-urilor.

1. Bibliografie

1. <https://stackoverflow.com/>

2. <https://dev.mysql.com/doc/>