

Scutaru Lucian

Gr 30225

1. Obiectivul temei:

Obiectivul temei este implementarea unei aplicatii destinate managerierii unui restaurant. Implementarea unei interfete destinate pentru 3 tipuri de useri(administrator, chef, bucatar). Fiecare user are la dispozitie diferite operatii.

Obiective secundare:

1. Implementarea GUI-ului
2. Adaugarea frame-urilor pentru fiecare din cei 3 useri
3. Interschimbarea frame-urilor pentru ca fiecare user sa poata efectua operatiile destinate
4. Alegerea structurilor de date pentru a salva obiectele necesare efectuarii operatiilor
5. Adaugarea si implementarea componentelor si operatiilor pentru fiecare user
6. Generarea bill-urilor in format txt.

1 – Capitolul 4

2 – Capitolul 2

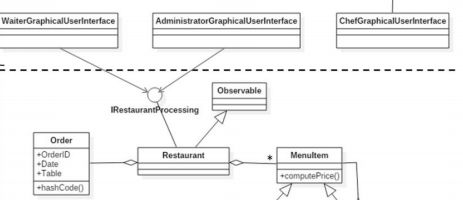
3 – Capitolul 2

4 – Capitolul 3

5 – Capitolul 4

6 – Capitolul 5

1. Analiza problemei, modelare, scenarii, cazuri de utilizare:

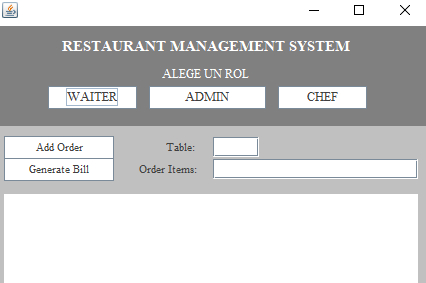


Scenarii:

1. Implementarea, meniului pentru un restaurant.
2. Adaugarea si modificare elementelor din meniu si apreturilor
3. Implementarea unei baze de date care tine evidenta comenzilor
4. Implementarea unui sistem de alertare pentru user-ul chef
5. Generarea de facturi

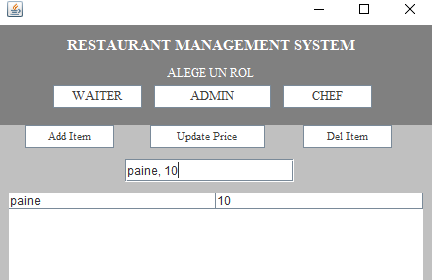
Modelare:

Am ales sa realizez toate cele 3 interfete diferite pentru fiecare user intr-un singur frame cu ajutorul unui layered-panel interschimbabil cu ajutorul butoanelor(care deserves ca si meniu pentru alegerea tipului de user). Pe fiecare panel am adaugat componentele necesare realizarii operatiilor destinate user-ului respectiv.



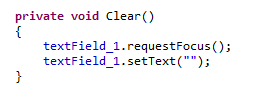
1. Proiectare (decizii de proiectare, diagrame UML, structuri de date, proiectare clase, interfete, relatii, packages, algoritmi, interfata utilizator)

Principala Structura de date folosita pentru implementarea aplicatiei este ArrayList-ul. Am ales sa salvez fiecare item din meniu intr-un arraylist. Odata cu adaugarea unui nou item in meniu acesta este adaugat si in lista. Cand butonul ”Adauga” este apasat, se creeaza un nou item cu ajutorul user-input-ului, editat cu ajutorul unei metode ce separa string-ul inserat de utilizator intr-un array cu fiecare cuvant separate. In tabela din zona destinate administratorului este adaugat item-ul si pretul acestuia.

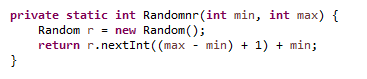


Asemanatoare este si metoda prin care se creaza si adauga in tabela comenzile, acestea fiind salvate tot intr-un arraylist.

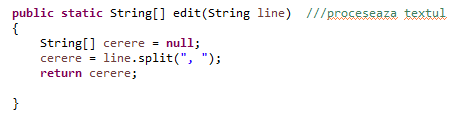
Metode si algoritmi:



Aceasta metoda vine in ajutorul utilizatorului, textul introdus de acesta disparand odata cu apasarea unui buton, pentru a face loc urmatoarelor date de introdus.



Este metoda care ma ajuta sa atribui un id unic fiecarei comenzi in parte.

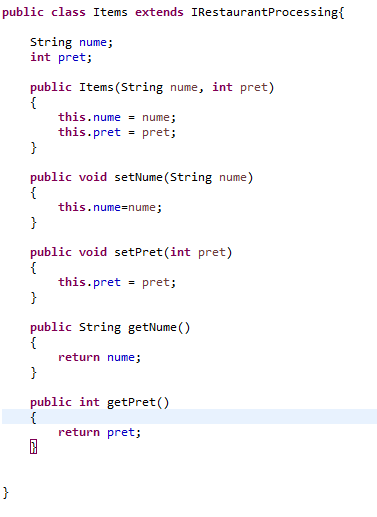


Cu ajutorul acestei metode processes informatiile primate de la utilizator, Stringul adaugat de acesta este impartit cuvant cu cuvant.

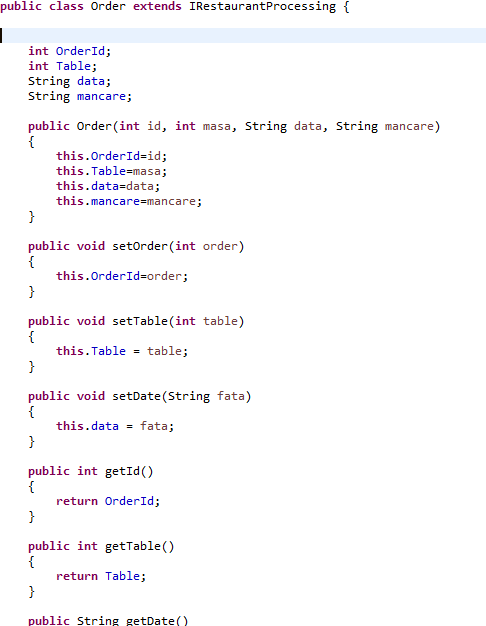
1. Implementare

Se va descrie fiecare clasa cu campurile si cu metodele importante. Se va descrie implemantarea interfetei utilizator.

Clasa Items contine cele 2 variabile necesare descrierii Itemelor ce urmeaza sa fie adaugate in meniu. Un constructor setters si getters.



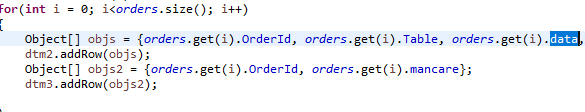
Clasa Order este asemanatoare celei mai sus prezentate



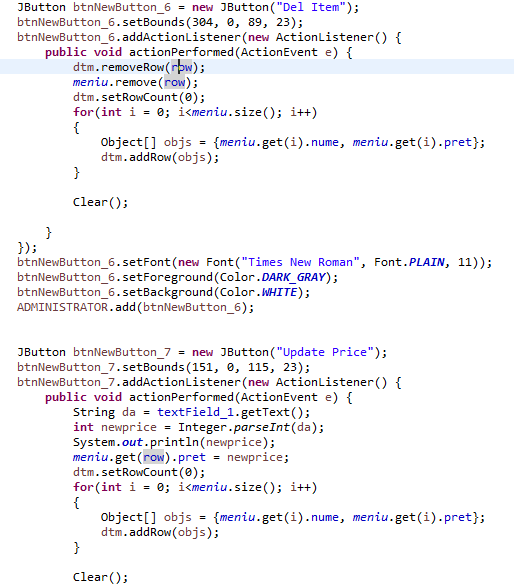
Data este reprezentata aici de un string insa atunci cand creez obiectele cu ajutorul clasei Date generez ziua si ora actuala, astfel ca fiecare comanda are o data unica.

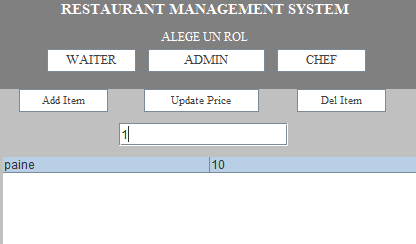


De fiecare deta cand obiectele sunt create, creez de asemenea si un array cu obiecte, pe care il populez cu datele obiectului creat si il adaug in tabela

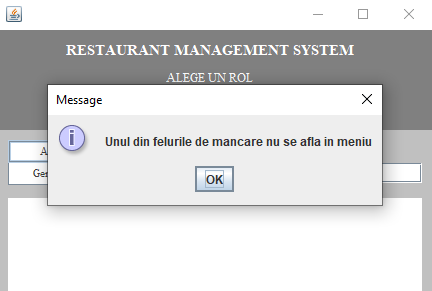


Pentru metodele care sterg si modifica iteme din meniu sau comenzile am ales sa selectez din tabel obiectul pe care doresc s ail modific, cu ajutorul event-click(dup ace apas pe un obiect din tabel, un int declarant anterior se actualizeaza si ma ajuta mai apoi sa efectuez operatiile necesare)

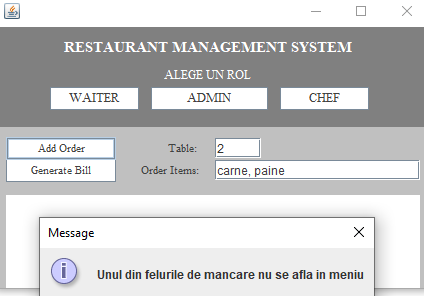




Una din metodele cele mai importante este cea care se asigura ca, inainte de a genera o noua comanda, produsele pe care waiter-ul doreste sa le adauge exista in meniu. In caz contrar o eroare va aparea pe ecran.



In aceasta metoda string-ul introdus de utilizator este despartit cuvant cu cuvant iar mai apoi numele produselor pe care acesta le-a introdus este comparat cu meniul. Daca unul din produse nu este prezent in meniu atunci generarea comenzii este imposibila.

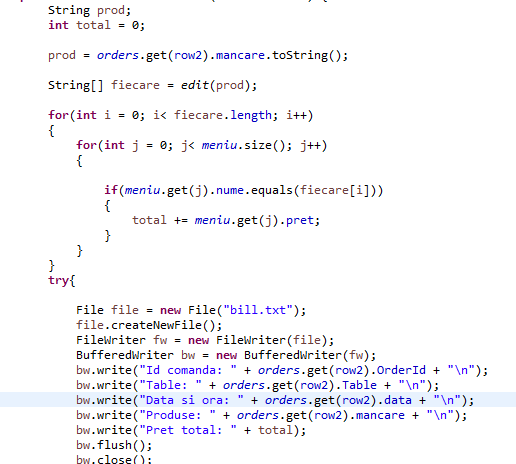


Notificarea utilizatorului CHEF am implementat-o simplu cu ajutorului unui buton. Daca o comanda este generate cu success atunci pe interfata destinate Chef-ului id-ul acesteia si produsele vor apare, impreuna cu un buton “Comanda noua!”, care odata apasat(confirmare primirii comenzii) dispare.



Am adaugat tot-odata la interfata Waiter-ului un buton pentru generarea facturilor. Cu ajutorul librariilor FileWriter si BufferedWriter creez fisierul text si scriu in el. Id-ul, Masa, Data si ora si Produsele comenzilor le adaug cu ajutorul.get(row) dupa ce, asemanator cu ce am facut la delete si modify items, selectez din tabel comanda pentru care vreasu sa generez bill-ul.

Pretul total al comenzii il calculez avand initial un int = 0, daca produsele din order se regasesc in meniu, atunci pretul acestora este adaugat la suma totala.



1. Concluzii

In concluzie, Interfata create este utilizabila, singura problema este faptul ca pentru fiecare table nu am implementat un scroll-bar, astfel ca meniul si lista cu comenzi este limitata.

Consider ca cel mai important lucru invatat din aceasta tema a fost lucrul cu Listele si tabel-urile.

Dezvoltarile ulterioare sun adaugarea unui scroll-bar vertical pentru toate tabelele, imbunatatirea sistemului da alertare a bucatarului si adaugarea mai multor campuri text pentru a usura munca utilizatorilor.

1. Resurse
2. <https://stackoverflow.com/>
3. <https://ro.wikipedia.org/wiki/Wiki>
4. <https://www.tutorialspoint.com/index.htm>
5. <https://www.geeksforgeeks.org/>