**DOCUMENTATIE TEMA 4**

**RESTAURANT MANAGEMENT SYSTEM**

**Csillag Szabolcs Andras**

**Grupa 30227**

**Profesor: Dan Mitrea**

Cuprins:

1. Cerintele temei

2. Analiza problemei

3.Proiectare

4.Implementare

5.Testare

6.Dezvoltari ulterioare si Concluzii

7.Bibliografie

# Cerintele Temei

( cerintele problemei )

Implementati un restaurant management system. Sistemul ar trebui sa aiba 3 tipuri de users: administrator, waiter si chef.

* Administratorul poate adauga , sterge sau modifica produse existente din meniu.
* Waiterul poate crea un order pentru o masa , adauga elemente din meniu, si a forma nota de plata
* Cheful e notificat de fiecara data cand trebuie sa gateasca mancare , care e ordonat d eun waiter .

**Obiective:**

## a)Obiectiv Principal:

(Obiectivul principal pe care il urmarim in acest proiect)

Gestionarea unui restaurant in care administratorul poate adauga , sterge , modifica produsele , cheful primeste doar order-urile efectuate si waiter poate construi un order si de asemenea sa ii faca nota. (factura de plata).Restaurantul sa aiba un management logic prin care utilizatorul sa comunice usor cu aplicatia ,sa ofere comenzile indicate sis a pastreze cu rigoare diagram data de cerinta,care va fi prezentata mai jos.

## b)Obective Secundare:

( comenzille pe care le poate oferi utilizatorul)

|  |  |  |  |
| --- | --- | --- | --- |
| **Numele Comenzii** | **Sintaxa comenzii** | | **Descrierea** |
| Adaugarea unui produs | Scrierea numelui si pretul produsului | | Se va insera in meniul restaurantului |
| Stergerea unui produs | Scrierea numelui produsului | | Sterge din restaurant produsul cu numele inserat |
| Modificarea unui produs | Modificarea unui produsului indeicat de nume ,cu pretul inserat | | Se va modifica pretul produsului daca acesta exista in stoc |
| Comanda de ospatar | Waiterul poate oferi un anumit order introducand masa ,si selectand cateva produse din meniu | | Comanda se va memora intrun HashMapcu cheia order si cu valuarea o lista de produse |
| Crearea notei de plata | Apasarea butonului computeBill | | Se va genera un fisier de tipul txt : bill.txt , in care se va specifica numele mesei si produsele comandate si suma totala care trebuie platita |
| La chef se afiseaza doar order-urile. | |

# Analiza Problemei

( use case-uri / scenarii care au o probabilitate mare sa apara )

**Use case-uri / Scenarii**

Utilizarea acestei aplicatii se face cu rularea programului care ofera o casuta in care trebuie sa alegem ce suntem, administrator, waiter sau chef bucatar.

* Administratorul , care adauga ,sterge , modifica sau numele unui produs si pretul
* Waiter : Poate creao camanda sau sa ii faca factura
* Chef : se afiseaza order-urile

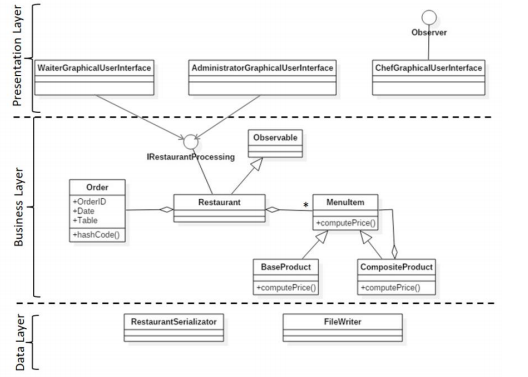
Factura(bill) se va genera si sub forma de txt.file la apasarea butonului computeBill.

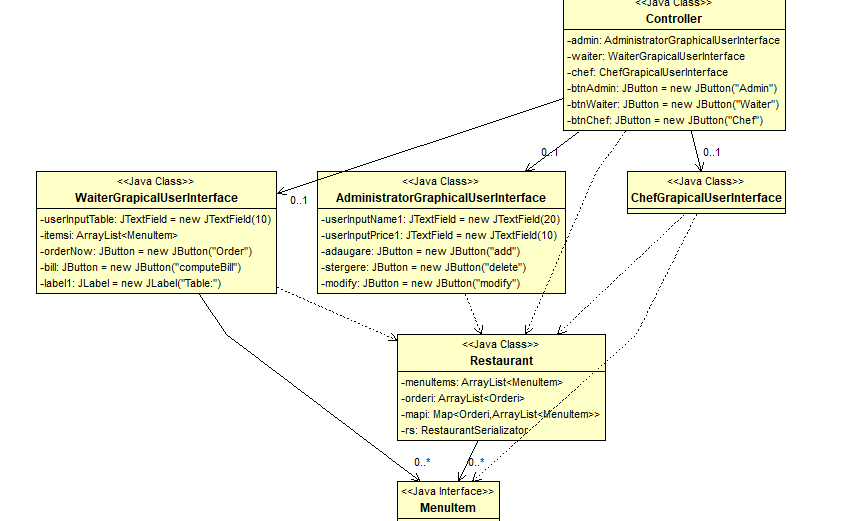
# Proiectare

Aici voi descrie etapele de proiectare folosite pentru realizarea acestui proiect ,dar mai ales gandirea constructiei ,si metodele alese.

(Ce structuri de date am folosit ?) Pentru implementarea meniului ,se foloseste implementarea design pattern cu BaseProduct fiind o frunza , CompositeProduct poate fi compus din mai multe produse , astea fiind un MenuItem. Am folosit si Observer pattern in care observer este chef si oservable este restaurantul , voi descrie in mod special cand voi vorbi despre fiecare clasa in parte . Am folosit si JCF collection pentru a face hashmap , cheia fiind un anumit order si valoarea fiind o lista de MenuItems si desigur suprascriind metodele equals si hashcode in clasa Orderi

## b)Diagrama de clase





Prima diagrama este luat exact din cerinta ,iar a doua este cea generata . Diagrama de clase,adica schema UML a proiectului am generat-o in ecplise si apareau prea multe detalii. Dar ca sa incapa frumos pe pagina ,am decupat un pic de jos. Am lasat doar ceea ce este esential, pentru ca metodele in sine le voi discuta mai jos in detaliu pe fiecare.Am adaugat cateva clase ajutatoare pentru aceasta aplicatie ,dar am pastrat cu exactitate cele prezentate in diagram cerintei si mai ales relatiile pe care le au intre ele;

# Implementare

Sunt implementate 3 pachete mari ,in care sunt multe clase, si fiecare cu rolul ei,voi descrie in mare pe fiecare.

**A Pachetul Business Layer**

**A.1 Clasa BaseProduct :**

-implementeaza interfata MenuItem

-Are un nume de tip string si un pret de tip double . are implementate gettere si settere si metoda toString.

**A.2 Clasa Composite Product**

-implementeaza interfata MenuItem

-Seamana cu un BaseProduct doar ca acesta contine un ArrayList de tipul MenuItem , deci acesta este un produs compus ,care este format din mai multe MenuItemi ,adica ori din cateva BaseProduct sau chiar mai multi CompositeProduct

**A.3 Interfata MenuItem**

-fiind o interfata descrie doar antetul a patru functii : getName , toString , computePrice , setPret

**A.4 Interfata IRestaurantProcessingOspatar**

-sunt descrise cele doua metode care pot fi aplicate de un waiter: addOrderAndElements si computeBillForAnOrder , care primesc ca parametrii un order si o lista de MenuItem.

**A.5 Interfata IRestaurantProcessingAdministrator**

-sunt descrise cele trei metode care pot fi aplicate de un administrator: addProduct , deleteProduct si modifyProduct ; delete primind ca parametru doar un nume, si ceilalti primind ca parametru numele produsului si pretul .

**A.6 Clasa MainClass**

-este clasa care contine metoda main , initial este implementata diferit un pic cu ce este lasat in comentariu pentru a adauga si manual cu cod anumite iteme, pe urma aici construiesc un restaurant pe care il primesc prin urmarea unei deserializari din clasa RestaurantSerializator construiesc un Controller si desigur la final serializez restaurantul posibil schimbat prin interfatele grafice.

**A.7 Clasa Abstracta Observable**

-este o clasa implementata de mine care are o lista de observatori ,are doua metode prin care poate adauga un observator sau sa ii dea remove , si desigur un seter ,prin care le transmite tuturor observatorilor pe care le contine sa faca un update.

**A.8 Clasa Orderi**

-aicieste descrisi cum arata un simplu order avand un ordered de tip int , o data de tip Date si un table de tip int,care sa precizeze masa la care este facuta comanda respectiva.

-are un constructor prin care primeste orderId si table , pentru ca data o va construi direct cu new Date() ,care va reprezenta momentul actual in care se face comanda respective

-am generat gettere si settere pentru variabilele instanta si am rescris metoda hashcode si metoda equals.

**A.9 Clasa Restaurant**

-este cea mai complexa clasa pe care defapt o si serializam si o deserializam, contine un ArrayList de tipul MenuItem , contine tot asa un ArrayList de tipul Orderi , areun RestaurantSerializator si are HashMap in care cheia este un Orderi si valoarea este o lista de MenuItem

-extinde clasa Observable , si implementeaza IRestaurantProcessingAdministrator si Ospatar

-aici sunt implementate metodele addProduct ,prin care efectiv adaugam la lista mare de MenuItems , produsul cu numele si pretul primit de parametru; deleteProduct : se va sterge produsul cu numele primit ca parametru din lista mare; modifyProduct : va modifica elemental,care are numele primit ,cu valoarea care ii este data tot din parametru

-aici sunt implementate si metodele prin care adaugam un order ,ei bine adaugam orderul in lista de orderi si punem in hashmap combinatia dintre order si ArrayListul de MenuItem primit ca parametru,dupa care notificam cheful ca avem un nou order; computeBill : in care doar cream un nou FileWriter si ii zicem sa ne faca nota de plata

-sunt generate si gettere si settere pt ce e relevant

-aplicam in mai multe functii serializable ,pentru ca atunci cand cream un produs,il stergem,il modificam,adaugam o comanda

**B Pachetul Data Layer**

**B.1 Clasa FileWriter :**

-are un constructor gol , si metoda computeBill care genereaza fisierul cu bill.txt . Fac un new File si daca acest fisier nu exista deja atunci il creez . Metoda primeste in parametrii un Order si o lista de produse , ei bine nota de plata va avea nr mesei acelei order si pe urma fiecare produs cu pretul sau si la final suma totala calculate prin adunarea preturilor produselor.

**B.2 RestaurantSerializator :**

-implementeaza Serializable ,acest Serializable este implementat de clasele care for contine si ele la randul lor date in fisierul serializat adica, : Restaurant , Orderi, MenuItem ( chiar daca e inerfata ,atunci va extinde) , BaseProduct , CompositeProduct .

-are doua metode esentiale : prima ar fi pentru deserializare cand incercam sa deserializam dintrun fisier un restaurant sis a il returnam ; a doua ar fi metoda de serializare in care primim ca argument Restaurantul si il punem in fiserul nostrum “fisieri.txt” .

**C Pachetul Presentation Layer**

**C.1 Interfata Observer:**

-are rol de a scrie un antet metodei update care va fi suprascrisa in chef ;

**C.2 Clasa Controller:**

- este o clasa adaugata de mine pentru a oferi un management mai bun restaurantului, prin care se deschide o interfata grafica si trebuie sa alegem ce anume vrem sa fim : admin , waiter , chef ;

-pentru asta acest Controller are toate celelalte 3 interfete grafice disponibile , adica sunt create aici ,si prin apasarea butonului dorit de utlizator aceel frame din clasa respective va devein vizibila ,iar controllerul nu va mai fi vizibil, pentru a incerca altceva , trebuie rulat din nou programul .

**C.3 Clasa ChefGrapicalUserInterface:**

- contine un tabel cu orderi si cu produsele orderului respective , aici intampin o problema pentru ca mi se afiseaza doar ultimul order si nu inteleg de ce.

-El are si o metadata in care pune in tabel orderul primit ca parametru si desigur lista de produse pentru acel order . initial aceasta metoda este chemat cu un foreach pentru a afisa toate order-urile existente

**C.4 Clasa AdministratorGrapicalUserInterface:**

- contine un label in care se zice ”Nume” si un textField unde trebuie introdus numele produsului,

- un alt label cu “Pret” unde e alt TextField ,unde trebe introdus pretul

-avem trei butoane care folosesc ActionListener pentru a adauga acel produs in restaurant, pentru a il sterge ( pentru stergere e destul doar numele produsului) si eventual pentru a modifica pretul acestuia

**C.5 ClasaWaiterGrapicalUserInterface:**

- contine un tabel cu meniul restaurantului in care sunt toate produsele oferite de acesta ,putem selecta cateva doar oferi un click pe ele , chiar daca pe urma ce dam click pe altul iluzia optica ar fi ca cea initiala nu mai e selectata, dar stati linistiti e deja memorata in orderul care va fi sa urmeze.

-contine un label cu “table” si un textField in care o sa scriem masa unde dorim sa facem rezervarea , langa el este butonul order, care va procesa Orderul respective in restaurant

-butonul computeBill este pentru a face nota de plata pentru ultimul order efectuat, se face automat pentru acela nu inteleg de ce, am stat ore in sir sa inteleg cum as putea face sa afisez toti orderii sis a fie selectati ,dar primeam eroare la incercari, asa ca am lasat asa.

# Testare

Pentru verificarea programului am construit initial un produs ,apa:3lei,lamaie:5lei si limonada care contine cele doua produse deci va fi 8 lei , le-am adaugat in restaurant si de acum programul se va rula in asa fel incat din fisierul fisieri.txt se va deserializa restaurantul.

# Concluzii si Dezvoltari Ulterioare

Acest Restaurant system management nu mi-a iesit super bine , o sa vedeti si in cod ,poate ca nu este exact ce sa cerut ,pentru ca este dubios acolo de ce se face doar pentru ultimul order, daca cititi asta, va rog sa imi semnalati cum as putea face aia ,in caz ca nu primesc nota mai mare decat 5 , m-am chinuit mult pana am inteles cerinta ,si cum sa adaptez , tema asta din punctual meu de vedere a fost foarte grea de implementat ,mi-au luat 4 zile non-stop , poate si din cauza ca nu am inteles exact cerintele ,sic and rulam uneori primeam niste erori de stateam cateva ore pe el .

Observatie Personala , nu neaparat despre continutul temei:

Stiu ca aceasta tema nu este facuta super la asteptarile profesorului , dar sper sa nu existe doar metoda de evaluare picat sau facut foarte bine , daca ar fi sa nu trec , va rog sa imi spuneti ce anume sa corectez mai exact sis a imi specificati timpul necesar pe care il voi avea. Desigur daca voi primi un raspuns tarziu , sper sa se calculeze ca am avut timp de atunci sa refac.

# Bibliografie

<http://www.coned.utcluj.ro/~salomie/PT_Lic/4_Lab/Assignment_4/Assignment_4.pdf>

<http://www.coned.utcluj.ro/~salomie/PT_Lic/4_Lab/Assignment_4/Assignment_4_Indications.pdf>

<https://www.baeldung.com/java-serialization>

<https://www.youtube.com/watch?v=fgjIk7qQong>

<https://www.baeldung.com/java-composite-pattern>

<https://www.baeldung.com/java-observer-pattern>