Restaurant Management

Profesor: Student:

Ana-Maria Nanes Cristiana Tintesan

Grupa:30227

Disciplina: Tehici de Programare

2019-2020

# Cuprins

[Cuprins 2](#_Toc40711555)

[1.Analiza problemei 3](#_Toc40711556)

[2.Obiectivul lucrarii 3](#_Toc40711557)

[2.1 Obiectivul principal 3](#_Toc40711558)

[2.2 Obiectivul secundar 4](#_Toc40711559)

[3.Proiectare 5](#_Toc40711560)

[3.1 Decizii de proiectare 5](#_Toc40711561)

[3.4 Implementarea pachetelor 6](#_Toc40711562)

[3.5 Diagrama de clase 7](#_Toc40711563)

[4.Implementare 7](#_Toc40711564)

[4.1 Pachetul PresentationLayer 7](#_Toc40711565)

[4.2 Pachetul BusinessLayer 9](#_Toc40711566)

[4.3 Pachetul DataLayer 10](#_Toc40711567)

[5.Rezultate 10](#_Toc40711568)

[6.Concluzii 14](#_Toc40711569)

[7.Bibliografie 14](#_Toc40711570)

# 1.Analiza problemei

Cerinta acestui assignment: Implementati un sistem pentru managementul unui restaurant.Sistemul ar trebui sa aiba trei tipuri de utilizatori: administrator, chelner si bucatar. Administratorul poate adauga, sterge sau modifica pretul unui produs deja existent in meniul restaurantului. Chelnerul poate crea o noua comanda pentru o masa din incinta localului si poate genera o factura pentru o masa din restaurant. Bucatarul este anuntat de fiecare data cand trebuie sa prepare o comanda generata de chelner. Pentru a simplifica aplicatia, consideram ca este doar un administrator, doar un chelner si doar un bucatar care se ocupa de gestiunea restaurantului.

Pentru a indeplini cerintele mentionate, primul lucru pe care trebuie sa il facem este sa intelegem atributiile fiecarei persoane care se ocupa de restaurant. Administratorul are rolul principal.El trebuie, in primul rand, sa aduge produse in meniu.Acestea pot fi BaseProduct sau CompisiteProduct, dar cu conditia ca un CompositeProduct are la baza mai mult BaseProduct.Mai mult, administratorul poate sterge un produs din meniu, sau poate modifica pretul unui produs existent in lista preparatelor din restaurant. Chelnerul aplicatiei poate genera comenzi pentru produsele dorite de client. De asemenea, chelnerul poate genera facturi care reprezinta nota totala de plata pentru o masa specificata din restaurant.

# 2.Obiectivul lucrarii

## 2.1 Obiectivul principal

Obiecticul principal al proiectului este de a implementa o aplicatie care sa gestioneze sistemul unui restaurant. Operatiile de baza, pentru a asigura o buna functionare sunt:

* Adaugarea unui nou produs in meniu. Aceasta sarcina revine administratorului;
* Stergerea unui produs din meniu. Aceasta sarcina revine administatorulului;
* Modificarea pretului unui produs existent in meniu. Aceasta sarcina revine tot administratorului;
* Generarea unui comenzi.Clientii de la o anumita masa pot comanda produsele din meniu, iar chelnerul va prelua comanda aceastora.
* Generarea unui facturi pentru o anumita masa din restaurant.Chelnerul va genera factura in care este specificat constul total al produselor comandate de clinetii de la o anumita masa.

## 2.2 Obiectivul secundar

|  |  |  |
| --- | --- | --- |
| Obiectiv secundar | Descriere | Capitol |
| Alegerea structurilor de date | Structurile de date folosite pentru a duce la capat obiectivul principal | 3 |
| Structurarea in pachete si apoi in clase a proiectului | Sunt folosite trei pachete. Primul este pachetul BusinessLayer in care sunt implementate clasele: BaseProduct, CompositeProduct, MenuItem, Order, Restaurant si interfata IRestaurantProcessing. Al doilea pachet este PresentationLayer in care se regasesc clasele: AdministratorGraphicalUserInterface, WaiterGraphicalUserInterface si clasa Main care ruleaza aplicatia. Ultimul pachet este DataLayer in care este descrisa o singura clasa: FileWriter care se ocupa cu generarea facturilor in format txt. | 3 |
| Implementarea solutiei | Vor fi descriese, pentru fiecare clasa in parte, campurile si metodele importante | 4 |
| Testare | Vor fi descries cateva scenarii de testare.In functie de un anumit set de date de intrare, se vor prezenta rezultatele rularii programului. | 5 |
| Dezvoltarea de use case-uri si scenarii care pot sa apara | Intr-un sistem de software un “use-case” este o lista de actiuni sau pasi de evenimente care definesc in mod obisnuit interactiunile dintre un rol si un sistem in atingerea unui obiectiv. | 3 |

# 3.Proiectare

# 3.1 Decizii de proiectare

Pentru realizarea proiectului, am luat urmatoarele decizii de implementare:

* Am impartit proiectul in trei pachete care vor fi detaliate ulterior.Proiectul contine in total zece clase;
* Am incercat sa folosesc cat mai multe metode simple, care sa nu contina multe linii de cod cu scopul de a oferi o intelegere rapida si usoara a codului;
* Folosirea unor nume sugestive pentru variabile;
* Pastrarea codului cat mai lizibil si usor de urmarit.

3.2 Structuri de date folosite

* In clasa CompositeProduct este declarata o colectie de tipul List in care sunt memorate elemente de tipul MenuItem pentru a stoca informatiile necesare despre produsele din meniu:

Private List<MenuItem> items;

* In clasa Restaurant se regaseste o colectie de tipul Map cu scopul de a stoca informatiile de tipul (cheie, valoare) dintre comenzi si lista produselor din meniu si o coloectie de tipul List pentru meniul restaurantului.

private Map<Order, List<MenuItem>> orders;

private List<MenuItem> menu;

3.3 Use case-uri/Scenarii

Utilizarea programului presupune citirea datelor prin intermediul unei interfete grafice.In aplicatie avem o interfata grafica pentru administratorul rastaurantului.Aici avem trei butoane si patru JtextArea in care putem introduce date de la tastatura. Cele trei butoane sunt: Add Menu Item, Delete Menu Item si Update Price Item. In cele patru JTextArea putem introduce datele despre un produs: in primul introducem id-ul produsului care trebuie sa fie de tipul int, in al doliea introducem numele produsului care trebuie sa fie un String, in al treilea pretul produsului, numar de tipul double, iar in al patruluea cantitatea produslui, care este, de seamenea, un numar intreg. Trebuie sa introducem corect datele pentru a asigura o buna functioneare a aplicatiei. In interfata grafica pentru chelner aveam doua butoane Place order si Generate Bill si un JTextArea in care trebuie sa introducem un numar intreg care indica numarul mesei din restaurant pentru care se va genera comanda sau factura. Pentru a genera cu succes o comanda trebuie selectat din tabelul cu produse unul sau mai multe preparate si trebuie introdus numarul mesei. Pentru a genera o factura trebuie selectate produsele din tabelul din stanga si comanda din tabelul din dreapta. Daca acestea nu sunt selectate, aplicatia nu va putea genera factura sau va genera o factura cu un pret gresit.

## 3.4 Implementarea pachetelor

Proiectul este format din trei pachete esentiale:

* Pachetul BusinessLayer – Contine clasele: BaseProduct, CompositeProduct, MenuItem, Order, Restaurant si interfata IRestaurantProcessing;
* Pachetul PresentationLayer – Contine clasele AdministratotGraphicalUserInterface, WaiterGraphicalUserInterface si clasa Main care ruleaza aplicatia;
* Pachetul DataLayer in care este descrisa clasa FileWriter, care are rolul principal de a crea fisierelel txt in care sunt generate facturile solocitate de chelner.

## 3.5 Diagrama de clase

A screenshot of a cell phone

Description automatically generated

# 4.Implementare

## 

## 4.1 Pachetul PresentationLayer

Aceast pachet contine trei clase:

* Clasa AdministratorGraphicalUserInterface. Aceasta clasa este folosita pentru a crea interfata grafica a administratorului.Pentru inferfata sunt create trei butoane: Add Menu Item, Delete Menu Item si Update Price Item, 4 JTextArea pentru a putea introduce datele despre produse, 4 JLabel-uri pentru a afisa inaintea fiecarui JTextArea un scurt text prin care specificam ce date trebuie introduse si un JCheckBox prin care determinam daca un produs este BaseProduct sau CompositeProduct.Daca JCheckBox este bifat, atunci produsul nou pe care dorim sa il adaugam in meniu face parte din categoria CompositeProduct. Pentru a adauga un produs de tipul CompositeProduct trebuie selectate din tabelul in care este afisat meniul, produsele de baza din care este compus. Pretul si gramajul unui CompositeProduct vor fi egale cu suma preturilor, respectiv a gramajelor a produselor de tipul BaseProduct care il compun.Daca nu este selectat niciun BaseProduct, atunci un mesaj de eroare va fi afisat pe ecran, iar produsul CompositeProduct nu va fi adaugat in meniu. Atunci cand unul dintre cele trei butoane este apasat, datele introduse sunt prelucrate prin apelarea metodelor din pachetul BusinessLayer.

deleteMenuItemBtn.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int[] selectedRows = menuTable.getSelectedRows();

if (selectedRows.length > 0) {

for (int row : selectedRows) {

Object x = menuTable.getValueAt(row, 1);

String a = x.toString();

restaurantProcessing.deleteMenuItem(a);

}

} else {

System.out.println("Selectati produsul pe care doriti sa il stergeti!!!");

}

view.updateMenuTable(restaurantProcessing.getMenu());

}

});

* Clasa WaiterGraphicalUserInterface. Aceasta clasa este folosita pentru a crea interfata grafica a chelnerului. Aici avem doar doua butoane: Place Order si Generate Bill si doar un JTextArea in care trebuie sa introduce un numar integ care indica numarul mesei din restaurant pentru care dorim sa generam comanda sau factura. Pentru a putea genera cu success o comanda, este necesar sa selectam produsele din meniu care alcatuiesc comanda, iar pentru a genera cu success o factura, este necesar sa selectam produsele care alcatuiesc factura si comanda din tabelul comenzilor( va fi prezentat un exemplu la punctul 5).
* Clasa Main este clasa de baza a programului deoare aceasta are rolul de a rula aplicatia si de a lega impreuna cele doua interfete grafice. Tot in aceasta clasa sunt realizate si formatele tabelelor in care sunt afisate produsele din meniu sau comenzile din restaurant.

## 4.2 Pachetul BusinessLayer

* Clasa BaseProduct mosteneste clasa MenuItem. Este o clasa simpla, in care sunt metodele getName(), setName(), getWeight(), setWeight(), getPrice(), computePrice() si update – metoda pe care o apelam pentru a modifica pretul unui produs din meniu. Constructorul acestei clase are patru parametrii: name, weight, id, price. Este folosit cuvantul cheie super: super(name, weight, id, price) pentru a face referinta la obiectele din clasa parinte: clasa MenuItem;
* Clasa CompositeProduct mosteneste clasa MenuItem. Constructorul acestei clase are patru parametrii: name weight, id, price. Este folosit cuvantul cheie super: super(name, weight, id, price) pentru a face referinta la obiectele din clasa parinte: clasa MenuItem. Metodele esentiale din aceasta clasa sunt getWeight si computePrice deoarece aceste doua metode calculeaza suma preturilor si a gramajelor dintre produsele de tipul BaseProduct care compun un produs de tipul CompositeProduct.
* Clasa MenuItem este clasa parinte pentru clasele BaseProduct si CompositeProduct.Sunt descrise patru variabile pentru aceasta clasa: id, name, weight si price.
* Clasa Order este clasa unde stocam informatiile despre o comanda. Avem doua variabile de tipul int: idOrder si table si o variabile de tipul Date – indica data in care comanda a fost procesata. Constructorul clasei primeste ca si parametrii doua variabile de tipul int si una de tipul Date si creeaza un nou obiect de tipul Order folosind datele primate ca si parametru. In aceasta clasa am creat metodele folosite de functia de mapare numite hashCode si equals.
* IRestaurantProcessing reprezinta interfata creata pentru a fi implementata de clasa Restaurant. Aici am declarant toate metodele care trebuie implementate de clasa Restaurant pentru a putea descrie toate capabilitatiile administratorului si alea chelnerului: crearea unui nou produs, stergerea unui produs din meniu, actualizarea pretului unui produs, crearea unei noi comenzi, generarea unui facturi in format txt.
* Clasa Restaurant este clasa care implementeza interfata IRestaurantProcessing. Clasa Restaurant are o lista de tipul MenuItem, care poate fi de doua tipuri: base sau composite product.Colectia Map este folosita pentru a stoca informatii de tipul (cheie, valoare) dintre comenzi si lista produselor din meniu. In aceasta clasa sunt suprescrise metodele delcarate in interfata IRestaurantProcessing si este descrisa functionalitatea lor.De exemplu, metoda deleteMenuItem primeste ca si parametru un String care indica numele produsului pe care dorim sa il stergem.Daca lista produselor din meniu este goala, atunci se va afisa un mesaj de eroare. O alta metoda importanta din clasa Restaurant este createNewOrder, care primeste ca si parametrii campurile necesare pentru a crea o noua comanda si o lista cu produsele din meniu.

## 4.3 Pachetul DataLayer

* Contine o singura clasa: FileWriter. Aceasta contine o singura metoda statica createBill pentru a crea fisierele txt in care vor fi afisate facturile generate. Metoda primeste ca si parametrii: int order – indica id-ul facturii, String date – data in care s-a realizat factura, int tableNb – numarul mesei din rasturant pentru care s-a generat factura, double price – indica pretul total.

# 5.Rezultate

In urma testarii programumlui, s-a constatat:

* Pentru administrator: adaugarea unui nou produs in meniu functioneaza corect, stergerea unui produs din meniu functioneaza corect, actualizarea pretului unui produs existent in meniu functioneaza corect;
* Pentru chelner: generarea unui comenzi care contine produsele selectate din meniu functioneaza corect. Crearea unui facturi in format txt functioneaza corect, daca din interfata grafica a chelnerului sunt selectate produsele din comanda corespunzatoare pentru generarea facturii.

In continuearea, vom prezenta si ilustra cum fuctioneaza operatiile descrise mai sus. Pentru ca aplicatia sa functioneze corect, este necesar ca datele introduse de la tastatura sa fie corect si respecte tipul celor cerute. De exemplu, pentru pretul unui produs trebuie sa introducem un numar real. In cazul in care vom introduce alte caractere, decat cele care constituie un numr, aplicatia nu va functiona.

Exemple:

* Dorim sa inseram produsele: 1 cartofi prajiti 18 200 ; 2 friptura 30 350. Aceste produse vor fi de tipul BaseProduct. Apoi dorim sa formam produsul CompositeProduct: 2 friptura cu cartofi prajiti 48 550.

A screenshot of a computer

Description automatically generated

Prima data am inserat cele doua produse:1 si 2, iar apoi dorim sa avem produsul cu id-ul 3.Pentru acest produs este necesar sa introducem id-ul si denumirea corect, iar la pret si gramaj putem pune orice numere pentru ca aceste vor fi oricum calculate in functie de cele doua produse deja existente in tabel, dar nu putem lasa ultimele doua JTextArea necompletate.De asemnea daca bifam Composite trebuie sa ne asiguram ca a slectam din meniu produsele care il compun.

Daca apasam acum butonul Add Menu Item vom obtine:

A screenshot of a social media post

Description automatically generated

* Pentru generarea unui comenzi:

A screenshot of a social media post

Description automatically generated

Am selectat produsele un id-ul 3 si 4 si am introdus numarul mesei: 2. Apasam butonul Place Order, iar comanda va fi generate si afisat in tabelul alaturat.

* Pentru generarea unui facturi: Selectam produsele si comanda, iar apoi apasam butonul Generate Bill. Daca se doreste sa se genereze factura pentru un produs de tipul CompisteProduct, atunci din tabelul din stanga trebuie selectate toate produsele BaseProduct care il compun, fara a mai selecta produsul compus.

A screenshot of a social media post

Description automatically generated

Factura a fost generata.

A screenshot of a social media post

Description automatically generated

# 

# 6.Concluzii

In urma realizarii acestu assignment, am reusit sa invat destul de multe lucruri despre GUI in Java. Cea mai grea parte la aceasta tema a fost interfata grafica, in mod special legatura din administrator si chelner.

Posibilitati de dezvoltare ulterioare: crealizarea interfetei grafice si pentru bucatarul restaurantului, salvarea informatiilor despre restaurant.

# 7.Bibliografie

<https://javarevisited.blogspot.com/2011/02/how-hashmap-works-in-java.html>

<https://docs.oracle.com/javase/8/docs/technotes/guides/language/assert.html>