DOCUMENTATIE TEMA 4

FOOD DELIVERY MANAGEMENT SYSTEM

Nume student: Buzilă-Gârda Andra-Maria

Grupa: 30229

Profesor de laboratorȘ assist. Antal Marcel

Cuprins:

1. Cerinte Functionale 3
2. Obiective 4
   1. Obiectiv principal 4
   2. Obiective secundare 4
3. Analiza problemei 4
4. Proiectare 7
   1. Alegerea structurilor de date 7
   2. Diagrama de clase 7
5. Implementare 8
   1. Pachetul presentationLayer 8
   2. Pachetul businessLayer 9
   3. Pachetul dataLayer 12
6. Concluzii si dezvoltari ulterioare 13
7. Bibliografie 13
8. Cerinte functionale:

* Implementati o aplicatie care sa se ocupe de management-ul unei firme de catering. Un client poate

Sa comande un produs din meniul firmei. Sistemul trebuie sa aiba 3 tipuri de utilizatori care se pot loga in aplicatie folosind un username si o parola. Aceste 3 tipuri sunt: administrator, angajat si client. Administratorul trebuie sa execute mai multe operatii si anume: sa se logheze, sa adauge produs in meniu, sa stearga produs si sa modifice produs, de asemenea poate sa creeze un meniu propriu-zis, alcatuit din mai multe produse. Tot administratorul poate sa genereze anumite rapoarte in functie de datele introduse. Clientul poate sa se logheze, sa vada lista de produse din meniu, sa caute produse pe baza anumitor criterii cum ar fi: dupa nume, dupa rating, numar de calorii, proteine, grasimi, sodiu si pret. Clientul poate sa genereze o comanda (sau mai multe), iar nota va fi afisata intr-un fisier .txt.

* Cerinte pentru notare:
* Utilizati un limbaj de programare orientat pe obiecte;
* Implementați schema de clase din secțiunea 1. Alegeți structurile de date adecvate pentru salvarea comenzilor și a elementelor meniului.
* Definiți clasa Produs de bază cu următoarele câmpuri: titlu, rating, calorii, proteine, grăsimi, sodiu, preț. Citiți datele din fișierul products.csv folosind fluxuri și împărțiți fiecare linie în 7 părți: titlu, rating, calorii, proteine, grăsimi, sodiu, preț și creați o listă de obiecte de tipul BaseProduct. NOTĂ: fișierul conține produse duplicat, deci asigurați-vă că selectați doar unul.
* Interfață grafică:
* o fereastră de conectare
* o fereastră pentru operațiunile administratorului (vezi cerințele din secțiunea 1)
* o fereastră pentru operațiunile clientului (vezi cerințele din secțiunea 1)
* Utilizați expresii lambda și procesarea fluxului pentru a implementa funcționalitățile de căutare disponibile pentru client (a se vedea secțiunea 1).
* Documentație de bună calitate
* Utilizați modelul de design compozit pentru modelarea claselor MenuItem, BaseProduct, CompositeProduct.
* Creați factura în format .TXT.
* Proiectare prin contract: condiții prealabile și postcondiții în interfața IDeliveryServiceProcessing. Implementați-le în clasa DeliveryService folosind instrucțiunea asert. Definiți un invariant pentru clasa DeliveryService. Generați fișierele JavaDoc corespunzătoare care ar trebui să includă etichetele personalizate și descrierile asociate condițiilor de pre, post și invarianți definite.
* Fereastra pentru utilizatorul angajat: utilizați Observer Design Pattern pentru a notifica de fiecare dată când se adaugă o nouă comandă.
* Salvați informațiile din clasa DeliveryService într-un fișier (adică file.txt) folosind serializarea. Încărcați informațiile când pornește aplicația.

1. Obiective
   1. Obiectiv principal

Proiectarea si implementarea unei aplicatii care faciliteaza munca unei firme de catering prin crearea

unui sistem care inmagazineaza tot meniul, iar clientul poate foarte usor sa isi aleaga produsul dorit.

* 1. Obiective secundare

|  |  |  |
| --- | --- | --- |
| Obiectiv | Descriere | Capitol |
| Dezvoltarea de use case-uri si scenario | Dorim sa cunoastem ce se intampla in momentul inroducerii datelor si apasarii fiecarui buton | 3 |
| Alegerea structurilor de date | Modul de reprezentare in memorie | 4 |
| Impartirea problemei in clase | Cream clasele de care avem nevoie | 4 |
| Implementarea solutiilor | Descrierea modului in care am create fiecare clasa si fiecare metoda in parte | 5 |

1. Analiza problemei

Utilizatori:

* Manageri de restaurante;
* Client;
* Angajati la restaurante;

Pre-conditii:

* Utilizatorul a introdus corect username-ul;
* Utilizatorul a introdus corect parola;
* Utilizatorul a dat Register sau stie ca este deja inregistrat
* Administratorul a introdus corect titlul;
* Administratorul a introdus corect rating-ul;
* Administratorul a introdus corect caloriile;
* Administratorul a introdus corect proteinele;
* Administratorul a introdus corect grasimile;
* Administratorul a introdus corect sodiul;
* Administratorul a introdus corect pretul;
* Administratorul a selectat corect produsele din table pentru a crea produsul compus;
* Administratorul a introdus corect numele noului produs;
* Administratorul a introdus corect start hour si end hour dupa apasarea butonului Reports;
* Administratorul a introdus corect numarul de produse dupa apasarea butonului Reports;
* Administratorul a introdus corect number of clients si value of the order dupa apasarea butonului Reports;
* Administratorul a introdus corect ziua dupa apasarea butonului Reports;
* Clientul a introdus corect titlul;
* Clientul a introdus corect rating-ul;
* Clientul a introdus corect caloriile;
* Clientul a introdus corect proteinele;
* Clientul a introdus corect grasimile;
* Clientul a introdus corect sodiul;
* Clientul a introdus corect pretul;
* Clientul a selectat corect produsele din table pentru a crea comanda;

Post-conditii:

* Se afiseaza produsele;
* Se afiseaza doar anumite produse selectate in functie de anumite criterii;
* Se adauga produse in meniu;
* Se sterg produse din meniu;
* Se modifica produse din meniu;
* Se creeaza comenzi;
* Se afiseaza rapoarte.

Modul de functionare:

1. Utilizatorul este administratorul. Dupa ce s-a logat in aplicatie ii va aparea in fata interfata

“Administrator”. Daca administratorul intoduce datele si apasa pe butonul de adaugare, se va dauga un produs in meniu, daca a modificat datele si a apasat butonul modificare, datele vor fi modificate, daca a scris datele unui produs, titlul obligatoriu, si a apasat pe butonul delete, se va sterge produsul respectiv, daca a apasat pe butonul View All, va vedea toate produsele intr-un table, de aici va selecta ce produse doreste pentru a crea un produs compus, introduce titlul noului produs in spatial dedicat si va apasa pe Composed Product pentru a se crea produsul compus care se va adauga in meniu. Daca administratorul va apasa pe butonul “Reports” ii va aparea o alta interfata in care: daca va introduce start hour si end hour si va apasa pe butonul “Time interval” i se vor afisa comenzile effectuate in acel interval de timp, daca va introduce number of products si va apasa pe butonul Product Frequency i se va afisa ce produs a fost comandat mai mult decat numarul specificat, daca va introduce number of clients si value of the order si va apasa pe butonul Client Frequency i se se vor afisa clientii care au comandat mai de mai multe ori decat primul numar specificat, dar care au fost mai scumpe decat o anumita valoare (al doilea numar introdus), daca va introduce in campul intitulat “day” o valoare si va apasa pe butonul Products Ordered i se vor afisa toate comenzile effectuate intr-o anumita zi si de cate ori au fost comandate. Butoanele de Back vor trimite utilizatorul la pagina anterioara.

1. Utilizatorul este de data aceasta Clientul. Dupa ce s-a logat ii va aparea interfata intitulata “Client”.

Daca va introduce in campul titlu o valoare de forma String ii va aparea un table cu toate produsele care contin acel cuvant in denumirea lor, daca va introduce in campul rating i se vor afisa noar produsele ce au acel rating, daca va introduce in calories, i se vor afisa produsele cu respectivele calorii, daca va introduce valoare in campul proteins, va putea sa vada toate elementele care au acel numar de proteine, daca va introduce o valoare in campul fat, va putea sa vada produsele cu acel numar de grasimi, daca va introduce valoare in campul sodium, va vedea produsele cu acea catitate de sodiu in ele, iar daca va introduce in campul price va vedea produsele cu pretul respectiv. Bineinteles, toate i se vor arata dupa apasarea butonului search (dupa o introducere). Daca va dori sa vada toate produsele va apasa pe “View Menu”, iar daca va dori sa creeze o comanda va selecta din table produsele pe care le doreste si apoi va apasa pe butonul “create”.

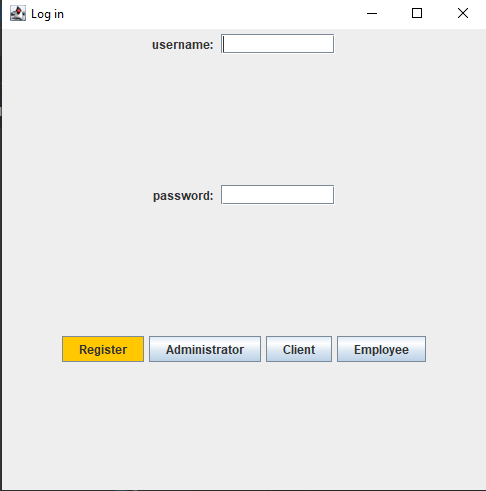


Figura 1: Interfata vazuta de toti utilizatorii

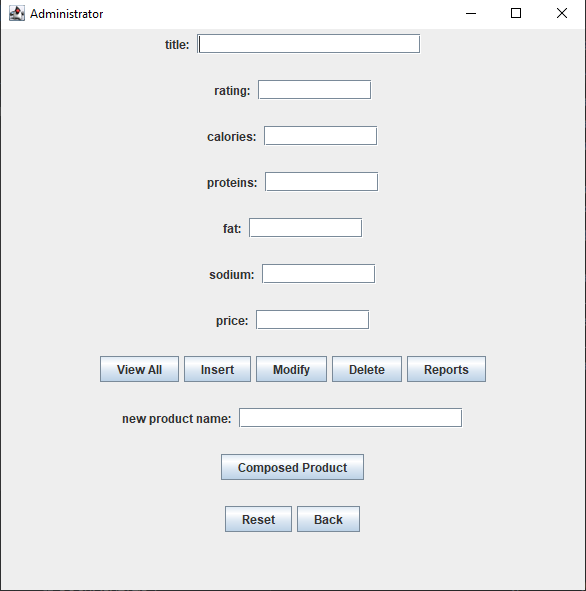


Figura 2: Interfata vazuta de administrator

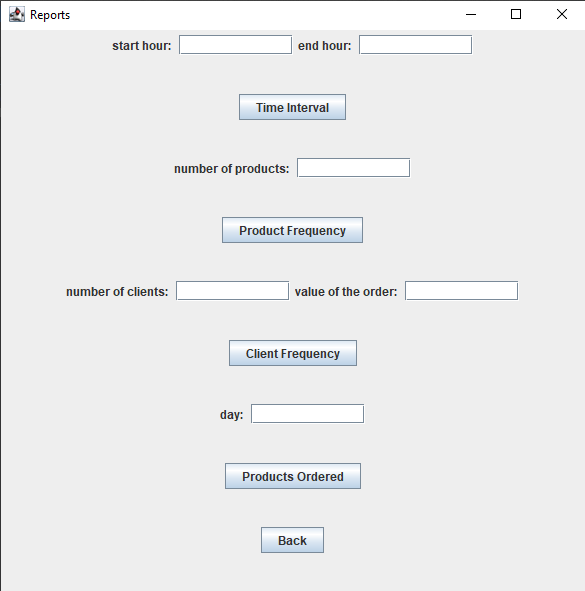


Figura 3: Interfata unde administratorul va vedea rapoartele

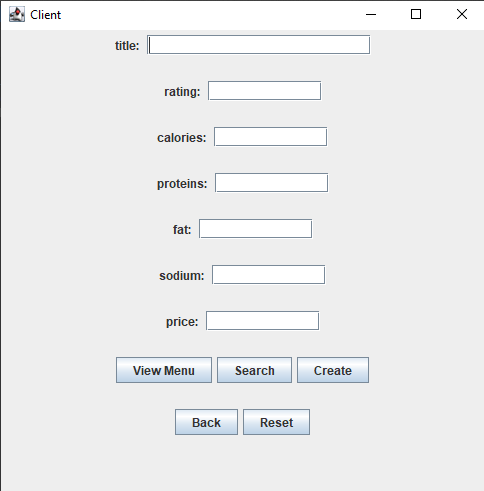


Figura 4: Interfata vazuta de client

1. Proiectare
   1. Alegerea structurilor de date

In clasa DeliveryService am o structura de HashSet<> in care voi inmagazina toate produsele din meniu

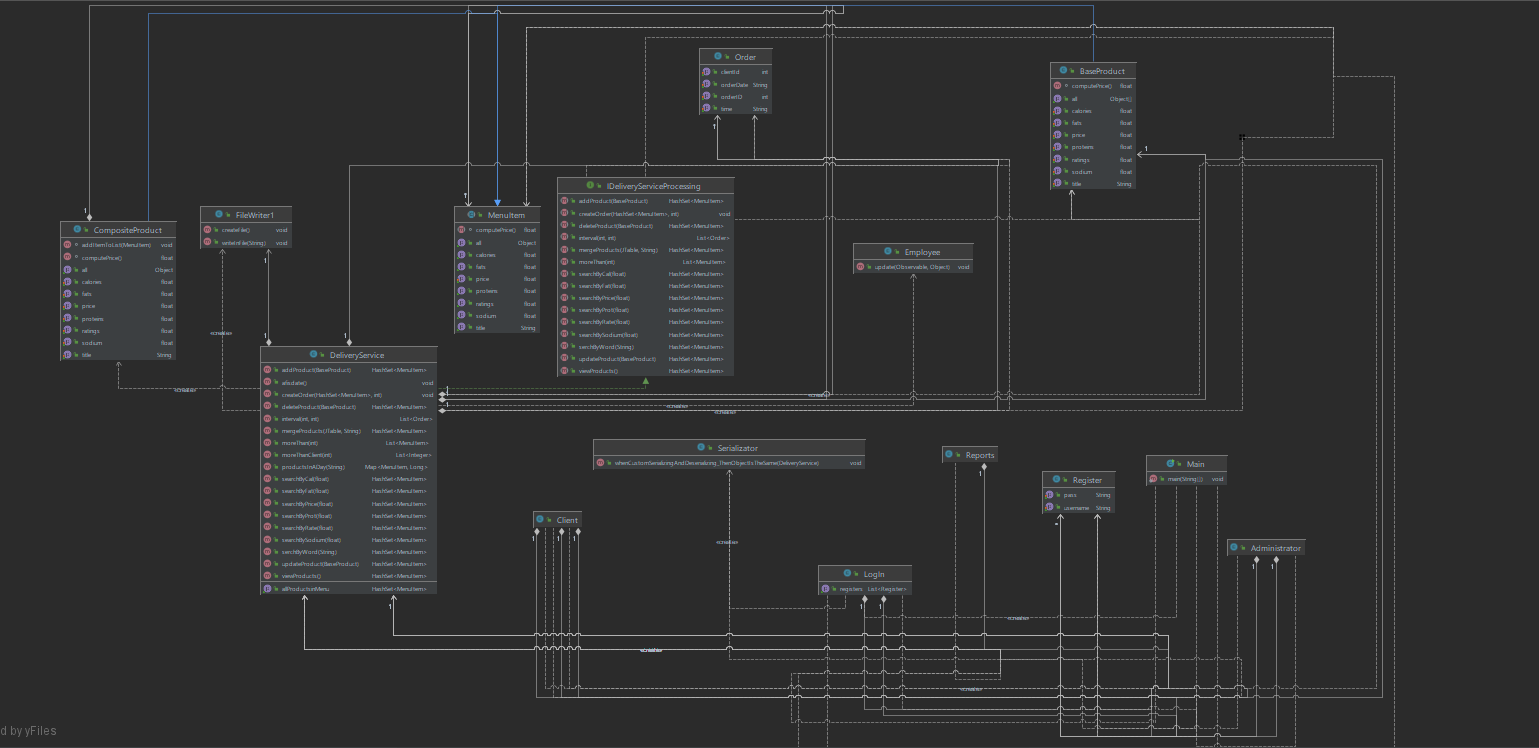
si asupra lor se vor face modificarile de catre administrator. Tot aceasta este si lista carese va afisa la apasarea butonului View All. Tot in aceasta clasa exista o structura HashMap<Order, HashSet<MenuItem>> care va inregistra pentru fiecare comanda si produsele care fac parte din comanda respectiva. Inafara de acestea 2 mai exista structuri de String[] dar si Object[][] (structura folosita pentru afisarea in tabele). De asemenea am folosit si liste.

* 1. Diagrama de clase

Unified Modeling Language (prescurtat UML) este un limbaj standard pentru descrierea de modele și specificatii pentru software. UML ofera o larga gama de diagrame pentru modelarea diferitelor situatii in cadrul unui proiect de dezvoltare software.

Diagrama de clasa este folosita pentru reprezentarea vizuala a claselor ai a interdependentelor, taxonomiei si a relatiilor de multiplicitate dintre ele. Aceste diagrame sunt folosite si pentru reprezentarea concreta a unor instante de clasa, asadar obiecte, si a legaturilor concrete dintre acestea.

Am impartit aplicatia in 3 pachete: businessLayer, dataLayer si presentationLayer. In presentation se afla clasele care se ocupa cu interfata cu utilizatorul, in businessLayer se afla clasele care contin metodele necesare pentru cee ace se intampla in momentul apasarii butoanelor, iar in dataLayer se afla clasa de scriere in fisier si cea de care avem nevie pentru serializare.



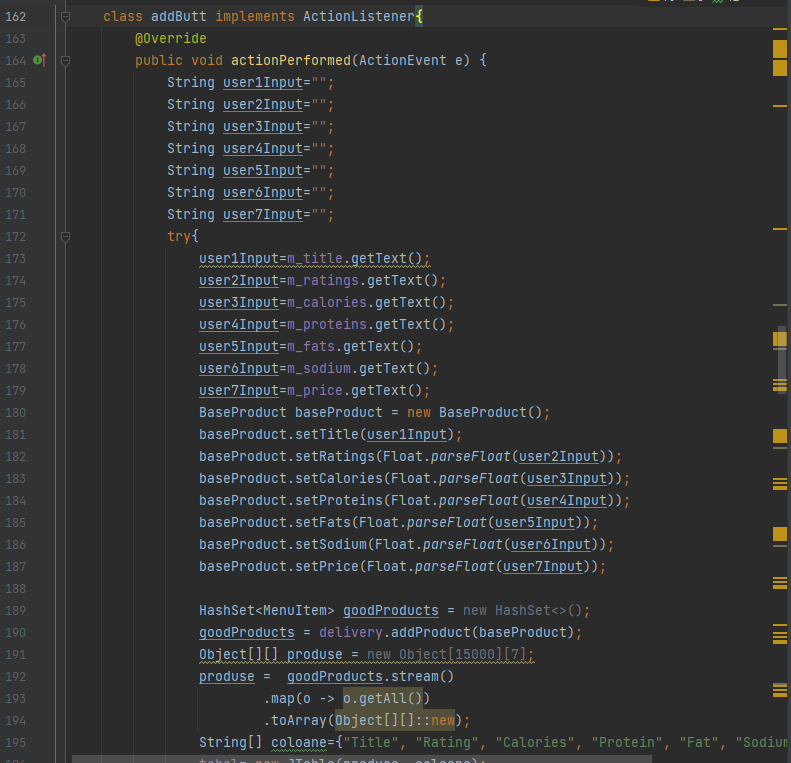
1. Implementare

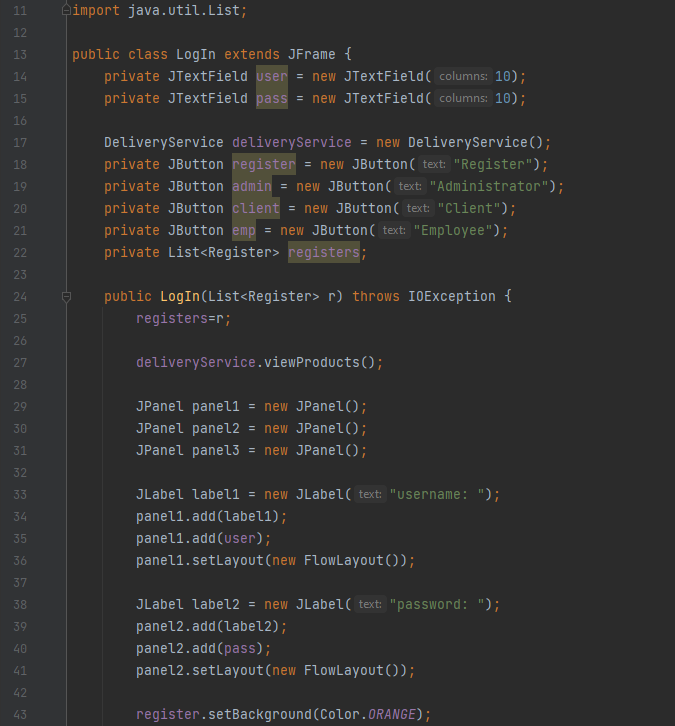
Pachetul presentationLayer

Acest pachet contine cate o clasa pentru fiecare interfata care este posibil sa apara pe ecran. In clasa

Administrator avem mai multe clase, respectiv pentru apasarea fiecarui buton, aceste clase vor apela anumite metode din clasa DeliveryService aflata in pachetul businessLayer. Ceea ce se intampla de fapt in aceste clase, se vor extrage informatiile din interfata si se va afisa in tabele rezultatele obtinute in metodele apelate. In clasa Client se intampla acelasi lucru doar ca se vor apela metode diferite pentru operatii diferite. Clasa LogIn va afisa interfata in care se vor loga utilizatorii cu un user si o parola. In clasa register se va valida logarea, iar in clasa Reports se vor afisa rapoartele in urma extragerii anumitor comenzi.



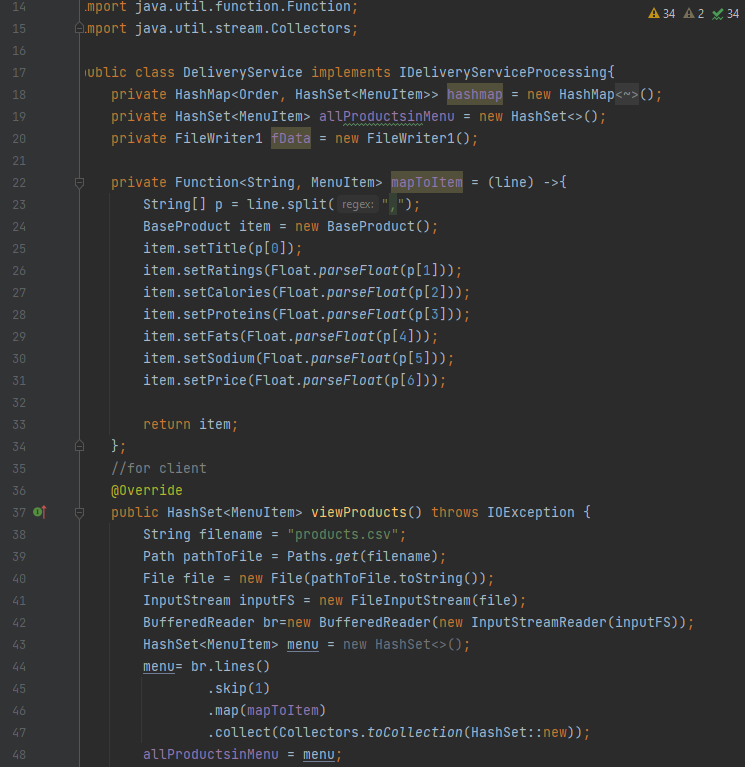






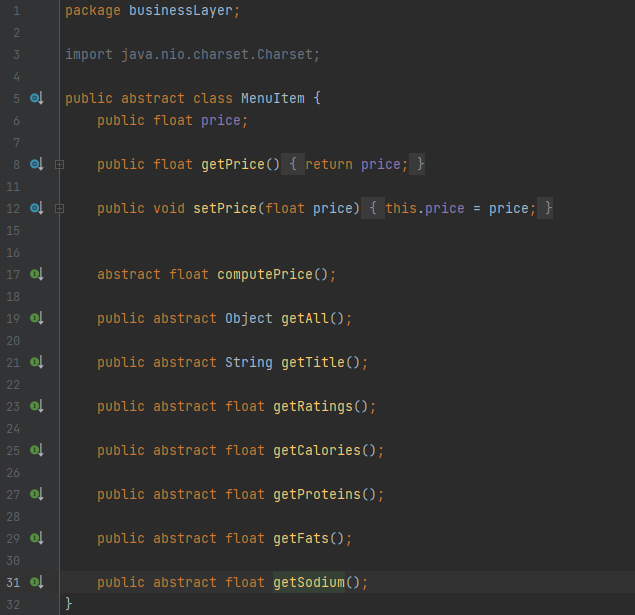
Pachetul businessLayer

Acest pachet contine clasele de baza, clasa de produs simplu, produs compus, acestea doua extind clasa abstracta MenuItem. Practic toate datele care se vor extrage din acel fisier .csv vor fi impartite si se vor crea instante de MenuItem, fiecare rand va reprezenta un produs, de aveea variabilele instanta din produs simplu si produs compus trebuie sa fie aceleasi cu titlurile coloanelor. In clasa DeliveriService care extinde interfata IdeliveryServiceProcessing se vor implementa toate metodele corespunzatoare actiunilor pe care le poate face utilizatorul acestei aplicatii. In aceasta clasa se afla o lista cu toate produsele de tipul hashSet<>, care va reprezenta de fapt meniul si va fi editata in functie de cerintele utilizatorului, dar tot aici exista o structura de HashSet<> care va inmagazina fiecare comanda si produsele aferente ei. In meniu se va adauga element, se va edita produs, se va sterge sau se va crea un produs compus, in cazul administratorului. Tot el poate si sa genereze rapoarte in functie de diferite conditii. In cazul clientului, el poate doar sa vizualizele produsele, sa caute produse dupa anumite criterii si sa creeze comenzi. In interfata IdeliveryServiceProcessing sunt afisate doar metodele neimplementate. Clasa order contine anumite variabile instanta specifice pentru fiecare comanda, cu care vom lucra la generarea rapoartelor.









Pachetul dataLayer

Acest pachet contine doar 2 clase: FileWriter1 si Serializator. Prima clasa va scrie in fisier, creand fisierul propriu zis si cu writer va scrie inauntrul sau, iar a doua va ajuta la scrierea in fisier a datelor.



1. Concluzii si dezvoltari ulterioare

In concluzie, aceasta aplicatie m-a ajutat sa imi dezvolt abilitatile in lucrul cu structuri precum

HashMap sau HashSet. Am invatat sa creez o aplicatie cu mai multe interfete care se leaga intre ele. Am invatat, de asemenea, sa lucrez cu Serialization, cu stream-uri si expresii lambda.

Ca si dezvoltari ulterioare as putea sugera sa se adauge cateva operatii realizabile si de catre un angajat comun. O alta adaugare de sediu ar fi o imbunatatire, astfel, clientului sa ii ajunga produsele in cel mai scurt timp din cel mai apropiat sediu.

1. Bibliografie

* YouTube
* https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html
* http://www.tutorialspoint.com/java/java\_serialization.htm
* http://javarevisited.blogspot.ro/2011/02/how-hashmap-works-in-java.html