**Tehnici de programare**

**TEMA 4**

**Restaurant**

**Cucui Mihai-Viorel**

**Grupa: 30228**

**I.Obiectiv:**

Propunerea si implementarea unei aplicatii pentru gestiunea unui local de tipul restaurant, aplicatie care poate furniza o solutie eleganta pentru o gestiune a unui meniu de produs al unui restaurant.

**II. Analiza problemei, modelare, scenarii, cazuri de utilizare**

Analiza problemei constituie un prim punct de plecare in determinarea si in alegerea solutiei. Analiza problemei presupune analiza obiectivului, analiza resurselor si alegerea unei idei de implementare, idee care sufera diferite modificari de-a lungul procesului de implementare.

Alegerea solutiei constituie baza unei aplicatii solide. Inceperea muncii pentru o idee neimplementabila presupune o munca fara un rezultat finit, o munca grea, deci o munca degeaba.

Problema, in cazul nostru, gestiunea produselor si comenzilor unui restaurant, presupune o implementare a unui restaurant real, care are un patron, un chelner si un bucatar. Patronul decide preturile, denumirile produselor, id-urile produselor si meniul afisat clientilor.

Implementarea in viata reala presupune mult mai multe, dar tema aceasta are o aplicabilitate in viata reala. Meniurile digitale sunt un exemplu foarte bun de aplicabilitate a acestei aplicatii Java. Meniurile virtuale dispun de mai multe pagini, sunt implementate deobicei pe device-uri care permit selectarea si comandarea produselor dorite.

In aplicatia realizata, s-a impus creearea unui chelner, care duce la bucatar comenzile, chelner care nu este necesar, implementarea crearii de comenzi putand fi realizata( dupa cum era normal) de un client venit in restaurant.

In cerintele aplicatiei, este necesara creearea unui “Chef”( bucatar). In cazul nostur bucatarul nu trebuie sa pregateasca niciun meniu, el doar afiseaza comenzile care ar trebui pregatite. Chef-ul ar putea sa pregateasca mancarea, sa semnalizeze chelnerului, care mai apoi ar trebui sa duca mancarea ceruta de clienti la masa cu numarul respectiv, dar aceste lucruri ar presupune o munca in plus si deci un timp de rezolvare mult mai mare fata de cel cerut initial.

Datorita interventiei chelnerului, datele din restaurant nu sunt preluate direct, ci sunt preluate dintr-o variabila instanta de catre chelner, date care sunt trimise mai apoi spre chef.

**III. Proiectare**

Proiectarea presupune mult efort de gandire plus efortul fizic de tastare, fiind realizata in eclipse, un compilator care sufera la capitolul predictibilitate si auto filling. Realizarea proiectelor in compilatoare ca IntelliJ este mult mai usoara de scris, fiind oferite predictii aproape la fiecare cuvant, codul fiind scris folosind aproape mereu “Enter” si “Left Arrow”, “Right Arrow”, “Up Arrow”, “Down Arrow”.

Proiectarea este realizata alegand clasele specifice obiectelor reale. Alegerea corecta a claselor are ca efect rezultant o usurare a muncii programatorului. Clasele Alese au fost AdminGUI, WaiterGUI, ChefGUI, MainGUI, MainClass, Restaurant, Order, DataN, BaseProduct, CompositeProduct,Restaurant, MainGUI.

Clasele reprezinta obiectele palpabile din lumea reala. Pe langa alegerea claselor, a fost si alegerea pachetelor, care au fost alese specific, folosind cate un nume reprezentativ, astfel caracterizand fiecare clasa si astfel realizand o impartire obiectiva a fiecarei .

Pachetele sunt reprezentative aplicatiilor obiectuale, continand nume de pachete precum “Business Logic”, “Data logic”, “Main package”, “Controller”, “Presentation Layer”.

Se mai poate vedea structurat pachetul Main, pachet care presupune realizarea fizica a obiectelor creeate, obiecte care realizeaza fiecare functie in parte si apoi isi implementeaza rolul.

In functie de clasa aleasa, se pot regasi metode specifice, metode care au o anumita functionalitate, metode care realizeaza functionarea propriu zisa a programului. Programul nu poate functiona fara o clasa main, clasa care realizeaza unirea tuturor claselor si interconectarea obiectelro “reale”. Clasele realizate se observa ca sunt interconectate dupa cum spuneam, in clasa Main.

Interfata a fost realizata prin diferentierea in trei clase, fiecare clasa implementand un frame, clase care realizeaza afisarea fiecarei functionalitati a aplicatiei. Sunt afisate clase de interfata grafica cu utilizatorul pentru Admin, Chef, MainGUI, si Chelner.

Daca utilizatorul va alege interfata waiter acesta va avea posibilitatea sa introduca o noua comanda dupa care sa o plaseze, sa afiseze datele comenzii si sa obtina o facturea.Daca se alege interfata chef v-a trebui deschisa si interfata waiter pentru a adauga comenzi,astfel chef-ul v-a fi notificat ca a primit o noua comnda.Interfta cu chef-ul v-a afisa si datele referitoare la comanda.

**III. Clase**

Proiectarea este realizata alegand clasele specifice obiectelor reale. Alegerea corecta a claselor are ca efect rezultant o usurare a muncii programatorului. Clasele Alese au fost AdminGUI, WaiterGUI, ChefGUI, MainGUI, MainClass, Restaurant, Order, DataN, BaseProduct, CompositeProduct,Restaurant.

Clasele au fiecare o functionalitate specifica, si realizeaza un anumit rol in aplicatie.

AdminGUI reprezinta afisarea grafica pentru un admin, interfata din care se realizeaza obiectele din meniu, obiecte care sunt atat obiecte compuse care pot fi realizate si din alte obiecte compuse si din obiecte simple, la alegerea adminului. Pot fi realizate si obiecte de tipul BaseProduct, obiecte care sunt obiecte simple, de baza, fiind reprezentate de materia prima din care sunt realizate anumite mancaruri.

Clasa MenuItem reprezinta o clasa care realizeaza un item dintr-un meniu, clasa care reprezinta o intrare in meniu, mai exact reprezinta un fel de mancare sau o mancare de baza(carne, lapte, sunca, oua, branza, etc si multe altele).

MenuItem contine atributele id, nume, pret care vor fi mostenite de clasele BaseProduct si CompositeProduct.Aici se afla metodele de set si get care se mostenesc de cele 2 clase.

Clasa menuItem contine atributele id, pret, nume si in plus de asta, clasa care o extinde, Composite Item contine si o lista cu obiectele din care sunt formate.

Clasa Order este reprezentarea virtuala a unei comenzi din viata reala( o comanda data unui chelner la restaurant; nu comenzi de tipul “Stai”, “Mergi”, comenzi de tipul “Cina”, “Pranz”, “Mic dejun”.

Order-clasa contine atributele id, data da tip date care ia data curenta a sistemului si numarul mesei la care se realizeaza comanda.Functia de hashCode() este suprascrisa asfel incat fiecare comanda se va pune la id-ul introdus de waiter. In aceasta clasa avem si metoda toString() care returneaza detalile comenzii.

Clasele de tipul GUI sunt realizate utilizand Swing. Swing este un subset JFC (Java Foundation Classes) şi constă dintr-o serie de componente vizuale care extind (îmbunătăţesc) componentele AWT, şi furnizează noi facilităţi precum tabele şi arbori. Structura de clase din Swing este asemănătoare cu cea din AWT, în sensul că toate componentele interfeţei grafice sunt derivate dintr-un singur părinte numit JComponent (care este derivat din clasa AWT Container). Pachetul de clase Swing reprezintă soluţia furnizată de Sun pentru crearea unor interfeţe utilizator grafice complet portabile pe orice platformă. În Swing, toate numele claselor încep cu litera J, şi atunci când este posibil, numele este acelaşi cu cel al clasei AWT pe care o înlocuieşte. La fel ca la AWT, punctul de plecare pentru un program bazat pe Swing, este clasa JFrame sau clasa JApplet.

Clasele GUI realizate in cadrul proiectului sunt AdminGUI, ChefGUI, WaiterGUI si MainGUI. In clasa MainGUI sunt reunite toate celelalte clase realizate anterior. Clasa AdminGUI, dupa cum am amintit, se ocupa de realizarea obiectelor din cadrul meniului, de editarea acestora si nu in ultimul rand de stergerea lor.

Clasa ChefGUI este reprezentata de un obiect de tipul JTextArea, la prima vedere. Aceasta clasa are de altfel in spate si un JPanel si in plus de asta un JFrame. JFrame-urile reprezinta ferestrele “Windows” care sunt vizibile in interiorul aplicatiei. JFrame este o versiune extinsă a clasei Frame care adaugă suport pentru un comportament de desenare special. Adiţional, JFrame permite componentelor Swing MenuBars să fie ataşate nu numai în partea de sus a ferestrei dar oriunde în fereastră. Toate obiectele asociate unui JFrame sunt manipulate de o instanţă a clasei JRootPane, care este singura componentă-fiu a unei instanţe JFrame.

Clasa WaiterGUI dispune, la fel, de un JFrame, de mai multe JPaneluri pentru structuarea in cadrul JFrame-ului a obiectelor de tipul vizual. Clasa WaiterGUI, impreuna cu clasa AdminGUI au functionalitatea de baza, realizata cu ajutorul unor JComboBox, care ofera libertate de exprimare atat grafica, cat si fizica, acestea retinand chiar referinta obiectelor adaugate in interiorul lor. Aceasta mobilitate ofera programatorului oportunitatea de a accesa diferite obiecte chiar din cadrul interfetei grafice.

De altfel, alt obiect intalnit in interfetele grafice este JButton, fiind reprezentarea virtuala a unui buton real. Cu aceste butoane se realizeaza, de obicei operatii. Fiecare buton necesita implementarea unui Action Listener. Cu ajutorul butoanelor se poate prelua textul din TextField-uri, de exemplu. Pentru preluarea caracterelor introduse într-un câmp text trebuie tratat evenimentul ActionEvent, generat de un obiect JTextField la apãsarea tastei "Enter". Pânã la apãsarea tastei “Enter” este posibilã si editarea (corectarea) textului introdus.

Unele aplicatii preferã sã adauge un buton care sã declanseze actiunea de utilizare a textului introdus în câmpul text, în locul tastei “Enter” sau ca o alternativã posibila.

Clasele Serializable au rolul de a oferi mobilitate in timp programului, retinandu-se astfel istoricul programului, nefiind nevoie inceperea de la 0 la fiecare pornire. Aceasta functionalitate ofera posibilitatea realizarii unui management in timp a programului, nefiind nevoie, la fiecare rulare de efectuarea acelorasi operatii. Aceasta performanta este notabila, fiind chiar esentiala in executarea unor task-uri.

Utilizand fluxurile putem scrie aplicatii care salveaza si incarca datele in fisiere. Java permite si un mecanism mai avansat si anume serializarea obiectelor. In forma cea mai simpla serializarea obiectelor inseamna salvarea si restaurarea starii obiectelor. Obiectele oricarei clase care implementeaza interfata *Serializable*, pot fi salvate intr-un stream(fluxde date) si restaurate din acesta. Pachetul java.io continedoua clase speciale*ObjectInputStream r*espectiv *ObjectOutputStream* pentru serializarea tipurilor primitive. Subclasele claselor serializabile vor fi si ele serializabile. Mecanismul de serializare salveaza variabilele membri nestatice si netransiente. Daca un obiect este serializat, atunciorice alt obiect care contine o referinta la obiectul serializat va sisi el insusi serializat. Se poate serializa orice multime de obiecte interconectateintr-o structura de graf.

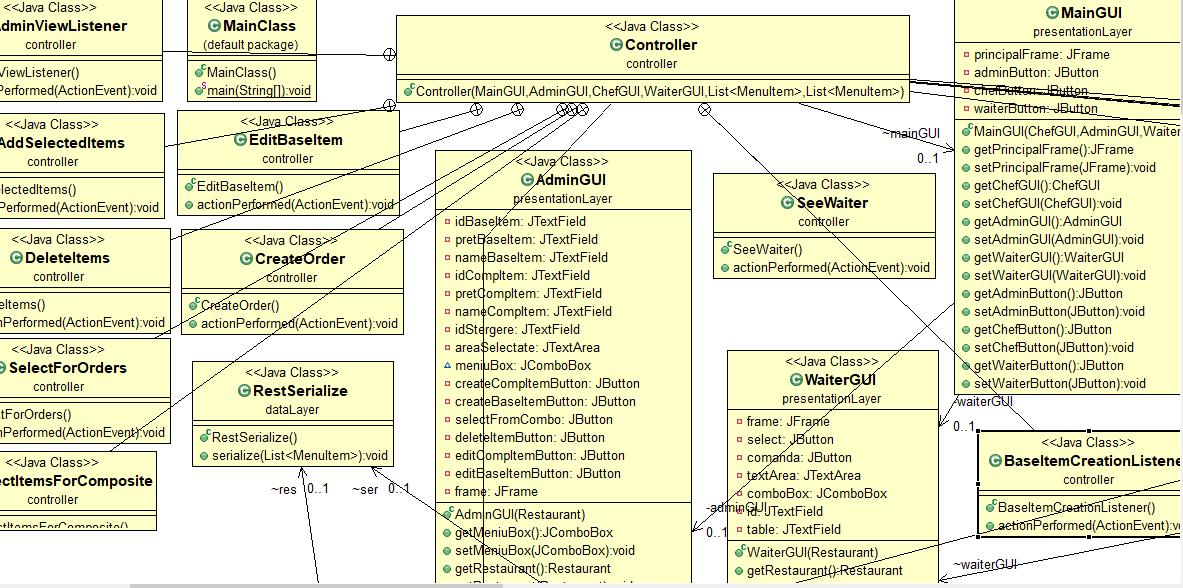
Aceasta interfata nu declara nici metode, nici atribute, serveste doar pentru identificarea obiectelor serializabile. Clasele care nu implementeaza aceasta interfata nu vor putea fi serializate sau deserializate. Operatia de deserializare presupune ca clasa obievtului poseda si un constructor cu vizibilitate publica si fara argumente. Atributele obiectului vor fi initializate dintr-un flux de date. In cazul serializarii unui obiect care apartine unui graf de obiecte daca se ajunge la un obiect care nu implementeaza interfata Serializable, atunci se genereaza exceptia *NoSerializableException* . Clasele care necesita o tratare speciala pe parcursul serializarii si a deserializarii trebuie sa implementeze urmatoarele doua metode cu urmatoarele signaturi:

private void writeObject(java.io.ObjectOutputStream out) throws IOException

private void readObject(java.io.ObjectInputStream in) throws IOException, ClassNotFoundException;

**Diagrama de clase UML:**

**Unified Modeling Language** (prescurtat **UML**) este un limbaj standard pentru descrierea de [modele](https://ro.wikipedia.org/w/index.php?title=Modelare_orientat%C4%83_pe_obiect&action=edit&redlink=1) și [specificații](https://ro.wikipedia.org/w/index.php?title=Specifica%C8%9Bie&action=edit&redlink=1) pentru [software](https://ro.wikipedia.org/wiki/Software). Limbajul a fost creat de către consorțiul [Object Management Group](https://ro.wikipedia.org/w/index.php?title=Object_Management_Group&action=edit&redlink=1) (OMG) care a mai produs printre altele și standardul de schimb de mesaje intre sisteme [CORBA](https://ro.wikipedia.org/w/index.php?title=CORBA&action=edit&redlink=1). UML a fost la bază dezvoltat pentru reprezentarea complexității programelor orientate pe obiect, al căror fundament este structurarea programelor pe clase, și instanțele acestora (numite și obiecte). Cu toate acestea, datorită eficienței și clarității în reprezentarea unor elemente abstracte, UML este utilizat dincolo de domeniul IT. Așa se face că există aplicații ale UML-ului pentru management de proiecte, pentru *business Process Design* etc.



**V. Concluzii si dezvoltari ulterioare**

Realizarea unor dezvoltari ulterioare presupune regandirea conceptului programului, si astfel si realizeaza o imbunatatire a sa. Din programul realizat se poate dezvolta un sistem de management pentru o firma de cattering sau un restaurant mic, doar ca aplicabilitatea tehnologiei la firmele mici nu este atat de in voga, fapt care ar putea sa fie datorat si de costurile mari pe care le prezinta aceasta tehnologie.

Datorita unor costuri mari multe firme aleg optiuni care reduc costurile momentan, dar nu definitiv. In cazul unei firme care presupune o dezvoltare in timp, un sistem informatic poate aduce o reducere substantiala a cheltuielilor cu angajatii si reduce mult costurile, creand astfel un management eficient.

O posibila dezvoltare ulterioara poate fi realizarea unui meniu placut ochiului, structurat in pagini, avand un font elegant, oferind astfel cititorului impresia unui restaurant luxos, chiar daca se foloseste tehnologia digitala. In acest fel se poate imbina utilul cu placutul si bineinteles cu viteza. Astfel clientii vor fi multumiti de timpul de servire si nu vor mai fi nevoiti sa astepte dupa preluarea comenzilor de catre un chelner, comenzile fiind trimise direct la bucatarie si apoi doar servite de chelner.