

# Tema POO 2017 – 2018

## Sistem de facturi fiscale

**Punctaj:** 2p din nota finală (2.5p cu bonus)

**Termen predare:** 13 ianuarie 2018, ora 23:50

**Atenție!** Tema se va uploada pe cs.curs doar până la această dată, urmând ca la ultimul laborator (săptămâna 14), fiecare student să-și prezinte personal tema la grupa lui, preferabil pe laptopul personal pentru a evita eventualele probleme de rulare!!

Se va rula aplicația din arhiva trimisă - pentru aceasta e bine să vă păstrați această arhivă!

### Arhitectura:

#### Interfete:

**IMagazin** – interfata cu metodele de calcul a diverselor tipuri de sume pentru un magazin

#### Clase abstracte:

**Magazin** (implementează IMagazin) – clasa abstractă corespunzătoare interfetei IMagazin

#### Clase concrete (instantiabile):

1. **Produs** – definește un produs ce poate fi regăsit pe o factură
2. **MiniMarket** (extinde Magazin) – conține facturile asociate unui magazin de dimensiuni mici
3. **MediumMarket** (extinde Magazin) – conține facturile asociate unui magazin de dimensiuni medii
4. **HyperMarket** (extinde Magazin) – conține facturile asociate unui magazin de dimensiuni mari
5. **Factura** – conține produsele facturate de un magazin la un moment dat
6. **ProdusComandat** – definește linia unei facturi
7. **Gestiune** – conține datele aplicației: produse, facturi, magazine, taxe etc. și metodele acestora.

### Descriere clase și interfețe – schelet minimal

#### Produs

##### Date private:

- denumire
- categorie
- taraOrigine
- pret

##### Metode:

- setDenumire(String)
- getDenumire()
- setCategorie(String)
- getCategorie()
- setTaraOrigine(String)
- getTaraOrigine()
- setPret(double)
- getPret()

Obiectele de tipul **Produs** vor fi create în urma citirii și parsării fișierului **produse.txt**.

Fisierul va conține pe prima linie denumirea țărilor separate prin spațiu (primele 2 cuvinte vor fi ignorate fiind de ordin descriptiv), pe prima coloană denumirile produselor existente și pe coloana a doua categoria asociată fiecărui produs. Restul informațiilor vor fi numere de tip double separate prin spațiu reprezentând pretul neimpozitat al produselor caracterizate de denumirea liniei pe care se află și țara coloanei asociate.

#### ProdusComandat

##### Date private:

- produs - obiect de tip **Produs**
- taxa
- cantitate

*Metode:*

- setProdus(Produs)
- getProdus()
- setTaxa(Double)
- getTaxa ()
- setCantitate(int)
- getCantitate()

Obiectele de tipul **ProdusComandat** vor fi create in urma citirii si parsarii unei linii din fisierul **facturi.txt**.

Fisierul va contine facturile emise de o serie de magazine.

Daca o linie incepe cu cuvantul cheie *Magazin* inseamna ca facturile ce urmeaza in continuare sunt asociate unui nou magazin de tipul MiniMarket in cazul primului magazin din fisier avand numele *MegaImage1*. Antetul recunoasterii unui magazin nou este: *Magazin:TipMagazin:NumeMagazin*, unde *TipMagazin* este MiniMarket, MediumMarket sau HyperMarket iar nume magazin este denumirea magazinului. Se garanteaza ca nu vor exista 2 magazine care sa aiba aceeasi denumire.

O factura noua va fi semnalizata cu cuvantul cheie *Factura*(unde i este indexul facturii unui magazin). Facturile vor fi despartite de un rand liber intre ele. Fiecare factura va avea pe prima linie antetul acesteia: *DenumireProdus Tara Cantitate*. In continuare fiecare linie va contine denumirea produsului facturat, tara de provenienta si cantitatea facturata, toate separate prin spatiu. Pentru fiecare astfel de linie se va crea un obiect de tip *ProdusComandat*.

## Factura

*Date:*

- denumire - obtinuta din fisier(prima linie a facturii). In cazul de mai sus: Factura1, Factura2 pentru MegaImage1 si Factura1 pentru Cora5
- Vector<ProdusComandat>

*Metode:*

- getTotalFaraTaxe() – va calcula totalul facturii fara taxe
- getTotalCuTaxe() - va calcula totalul facturii cu taxe
- getTaxe() - va calcula totalul taxelor facturii
- getTotalTaraFaraTaxe(String) - va calcula totalul facturii fara taxe pentru o anumita tara
- getTotalTaraCuTaxe(String) - va calcula totalul facturii cu taxe pentru o anumita tara
- getTaxeTara(String) - va calcula totalul taxelor facturii pentru o anumita tara

Obiecte de tipul *Factura* vor fi create in urma citirii si parsarii unei facturi din fisierul **facturi.txt**. Pentru fiecare factura din fisier se va crea un obiect de tip *Factura*.

## IMagazin

*Metodele interfetei:*

- getTotalFaraTaxe() – calcul total facturi magazin fara taxe
- getTotalCuTaxe() - calcul total facturi magazin cu taxe
- getTotalCuTaxeScutite() - calcul total facturi magazin cu scutirea de taxe corespunzatoare

Obs. Daca nu exista scutiri de taxe atunci metoda intoarce aceeasi valoare cu getTotalCuTaxe()

- getTotalTaraFaraTaxe(String) - calcul total facturi magazin fara taxe, pentru o anumita tara
- getTotalTaraCuTaxe(String) - calcul total facturi magazin cu taxe, pentru o anumita tara
- getTotalTaraCuTaxeScutite(String) - calcul total facturi magazin cu scutire de taxe, pentru o anumita tara(procentajul scutirii este cel calculat de metoda calculScutiriTaxe() si se aplica sumei returnate de metoda getTotalTaraCuTaxe(String))
- calculScutiriTaxe() – calculeaza procentajului taxelor de care sunt scutite magazinele

Magazin – clasa abstracta, implementeaza *IMagazin*

*Date:*

- nume– obtinut din fisier(MegaImage1 si Cora5)
- Vector<Factura>

*Metode:* Implementeaza toate metodele din IMagazin, mai putin metoda *calculScutiriTaxe()* care ramane abstracta!

MiniMarket, MediumMarket, HyperMarket – clase concrete, mostenesc *Magazin*

Date:

- Tip - **MiniMarket, MediumMarket, HyperMarket**

Metode:

- calculScutiriTaxe() - calculeaza procentajul reducerii aplicate sumei totale a magazinului (suma returnata de metoda getTotalCuTaxe()) in functie de indeplinirea urmatoarelor criterii:
  - **MiniMarket** –10% daca exista o tara penru care suma produselor comandate (inclusiv taxe) depaseste 50% din totalul vanzarilor realizate de catre magazin (inclusiv taxe)
  - **MediumMarket** – 5% daca exista o categorie de produse pentru care suma produselor comandate(inclusiv taxe) depaseste 50% din totalul vanzarilor realizate de catre magazin (inclusiv taxe)
  - **HyperMarket** –1% daca exista cel putin o factura a carei suma totala(inclusiv taxe) depaseste 10% din totalul vanzarilor realizate de catre magazin (inclusiv taxe)

Obiecte de tipul *MiniMarket, MediumMarket sau HyperMarket* vor fi create in urma citirii si parsarii fisierului **Facturi.txt** prezentat mai jos.

**Atentie!** Pentru fiecare magazin se va crea un obiect corespunzator de tip *MiniMarket, MediumMarket sau HyperMarket*, in functie de tipul magazinului, folosind **Factory pattern** . [1]

Gestiune

Date:

- produse - lista obiecte **Produs**
- magazine – lista de obiecte *Magazin – modificare a enuntului!!*
- taxe - dictionar de perechi (**tara, dictionar de perechi (categorie, procent)**)

Legatura dintre o tara, o categorie de produs si taxa asociata combinatiei in cauza va fi retinuta sub forma unui dictionar in urma procesarii fisierului **taxe.txt** exemplificat mai jos:

Fisierul contine pe prima linie denumirea tarilor separate prin spatiu(primul cuvant va fi ignorat fiind de ordin descriptiv) si pe prima coloana categoriile produselor existente. Restul informatiilor vor fi numere intregi separate prin spatiu reprezentand procentul cu care se vor impozita produsele caracterizate de categoria liniei pe care se afla si tara coloanei asociate.

Fisierul in cauza specifica ca produsele ce fac parte din categoria *Patiserie* si sunt importate din *Italia* vor fi impozitate cu 15% din pretul specificat in fisierul **produse.txt**. Astfel o factura ce va contine produsul *Paine* importat din Italia va avea pretul final cu care urmeaza a fi achizitionat(dupa impozitare) de 2.415( $2.1 * 115\%$ ).

**Atentie!** Clasa Gestiune se va implementa folosind **Singleton pattern** . [0]

## Detalii suplimentare

1. Pe lângă aceste date si metode puteti adăuga orice altceva considerati util
2. Se va utiliza **tipizarea** colectiilor folosite (utilizati `ArrayList<Produs>` nu `ArrayList`!)
3. Pentru realizarea interfeței grafice a aplicației se vor folosi cât mai multe din facilitățile și componentele Swing - panel-uri (ex. `JTabbedPane`), componente MVC - `JComboBox`, `JList`, `JTable` etc, precum și tot ce poate să facă interfața cât mai atractivă!
4. Toate clasele trebuie să aibă metoda **toString** implementată!!
5. Exemple pentru toate fisierele utilizate (**facturi.txt, taxe.txt, produse.txt, out.txt**) gasiti in arhiva atasata temei!

## Cerinte

**Task 1 (12p):** In fisierul **out.txt** completati urmatoarele informatii ca in exemplul din arhiva:

1. Prima linie va contine tipul magazinului
2. In continuare va fi detaliat fiecare magazin avand tipul mentionat. Astfel linia a doua va contine numele primului magazin considerat in ordinea sumelor totale fara taxe aplicate urmat de un rand liber.
3. Linia urmatoare va incepe cu cuvantul „Total” si va fi urmata de 3 valori numerice si de un rand liber:
  - Total magazin fara taxe
  - Total magazin cu taxe
  - Total magazin cu taxe scutite

4. In continuare urmeaza o linie ce va contine cuvantul cheie „Tara” urmat de atatea randuri cate tari sunt identificate in fisierul **taxe.txt**. Fiecare rand va incepe cu denumirea tarii si va fi urmata de 3 valori numerice si de un rand liber:
  - Total tara fara taxe
  - Total tara cu taxe
  - Total tara cu taxe scutite
 In cazul in care nu exista produse comandate dintr-o anumita tara se va afisa valoarea 0.
5. In continuare va fi detaliata fiecare factura a magazinului in cauza. Astfel linia urmatoare va contine numele primei facturi considerata in ordinea sumelor totale cu taxe urmata de un rand liber. Pentru fiecare factura se vor afisa aceleasi informatii ca cele afisate pentru magazin (pasii 3, 4 si 5 de mai sus, cu mentiunea ca nu vor fi specificate sumele totale cu taxe scutite, avand in vedere ca pentru clasa *Factura* nu avem definite metodele asociate)
6. In urma afisarii tuturor informatiilor necesare legate de toate magazinele cu toate facturile din categoria MiniMarket se va trece la afisarea acelorasi informatii pentru categoriile MediumMarket si HyperMarket.

**Punctaj 12p** din care: **2p** - implementare cu *Singleton pattern*, **2p** - implementare cu *Factory pattern*

## Task 2: (8p + 5p Bonus ) Interfata grafică - Java Swing

Pentru gestiunea aplicatiei, veti crea o interfata grafica folosind Java Swing care trebuie să cuprindă următoarele elemente:

1. pagina de start a aplicatiei ce va contine link-uri catre urmatoarele pagini [**Bonus** – logare]
2. pagina de incarcare a fisierelor **taxe.txt**, **produse.txt** si **facturi.txt** - din ierarhia de fisiere intr-un obiect **Gestiune** - si de creare a fisierului **out.txt**
3. pagina afisare si administrare a produselor din fisierul **produse.txt** cu urmatoarele operatii obligatorii:
  - afisarea produselor existente cu posibilitate de ordonare:
    - alfabetica dupa denumirea produsului
    - dupa tara
  - adaugarea unui nou produs la lista(daca produsul exista deja utilizatorul va fi attentionat)
  - stergerea unui produs
  - [**Bonus**]: editarea unui produs
  - [**Bonus**]: posibilitatea de cautare a unui produs

Observatie! Produsele adaugate/editate vor fi actualizate in fisierul **produse.txt**!

4. pagina de afisare a statisticilor:
  - magazinul cu cele mai mari vanzari (total cu taxe) si a datelor acestuia - denumire, total fara taxe, total cu taxe si total cu taxe scutite
  - magazinul cu cele mai mari vanzari pentru fiecare tara in parte si a datelor acestuia (ca mai sus)
  - magazinul cu cele mai mari vanzari pentru fiecare categorie in parte si a datelor acestuia
  - factura cu suma totala (fara taxe) cea mai mare

**BONUS:** Pentru fiecare din cele 3 bonusuri primiti cate 1p iar pentru o interfata grafica intuitivă și complexă, frumos realizată se va mai acorda un punctaj suplimentar de maxim 2p!!

## Important!

- Pentru testare veti trimite aplicatia si cu fisierele **taxe.txt**, **produse.txt** si **facturi.txt**. Datele aplicatiei (de la o rulare la alta) vor fi păstrate în fisiere pentru a asigura consistenta datelor.
- Temele sunt **INDIVIDUALE**. Copierea de la alți colegi se va sancționa cu **punctaj 0** atât pentru cel care a copiat cât și pentru cel de la care s-a copiat. Copierea de pe Internet atrage de asemenea punctaj 0 pe temă pentru cel care a copiat.
- Tema o veți trimite într-o arhivă .zip de forma **grupa\_NUME\_Prenume.zip** care va conține:
  - un folder SURSE ce conține doar sursele Java si fisierele de test
  - un folder PROIECT ce conține proiectul NetBeans sau Eclipse
  - un fișier README în care veți specifica numele, grupa, gradul de dificultate al temei, timpul alocat, modul de implementare și alte observații dacă există. Lipsa fișierului README duce la o **depunere de 10p**.

[0] Singleton pattern

[http://www.tutorialspoint.com/java/java\\_using\\_singleton.htm](http://www.tutorialspoint.com/java/java_using_singleton.htm)

<http://howtodoinjava.com/2012/10/22/singleton-design-pattern-in-java/>

[http://en.wikipedia.org/wiki/Singleton\\_pattern](http://en.wikipedia.org/wiki/Singleton_pattern)

[1] Factory pattern

<http://www.oodeesign.com/factory-pattern.html>

<http://howtodoinjava.com/2012/10/23/implementing-factory-design-pattern-in-java/>

[http://en.wikipedia.org/wiki/Factory\\_method\\_pattern](http://en.wikipedia.org/wiki/Factory_method_pattern)