Academia de Studii Economice

Facultatea de Cibernetică, Statistică şi Informatică Economică

**PROIECT**

**CALITATE SI TESTARE SOFTWARE**



Busuioc Ana-Maria

Grupa 1049

Seria A IE

**CUPRINS**

1. **Definirea design pattern-urilor folosite**
   1. Singleton
   2. Factory
   3. State
2. **Definirea metodelor testate prin Unit Testing**
3. **Definirea Test Case-urilor**
4. **Definirea Test Suite-ului**
5. **Definirea functiilor aplicatiei**
6. **Definirea design pattern-urilor folosite**
   1. **SINGLETON**

Prin intermediul clasei Casier am implementat design pattern-ul Singleton. Aplicatia este construita astfel incat per sesiune sa existe numai un Casier. Casierul primeste facturi de la clienti pe care le inregistreaza in gestiunea lui. De asemenea acesta primeste si plata aferenta fiecarei facturi .

* 1. **FACTORY**

Am creat interfata Factura ce este implementata de clasele FacturaDeGaze si FacturaDeTelefonie, astfel pot existe doua tipuri de facturi.

Clasa FactoryFabrica este clasa care in functie de un parametru primit construieste ori un tip de facture ori celalalt.

* 1. **STATE**

Acest design pattern l-am implementat in cadrul clasei Factura creand un camp stareFactura care poate lua valorile : PLATITA\_INTEGRAL, CU\_REST\_DE\_PLATA, DE\_PLATA. Acest camp este setat infunctie de suma de bani primita de la client. Casierul poate primi suma necesara pentru achitarea integrala a facturii si astfel avem o facture cu starea “PLATITA\_INTEGRAL” sau poate suma nu e suficienta si ramane un rest de plata, astfel avem facture cu starea “REST\_DE\_PLATA”.

1. **Definirea metodelor testate prin Unit Testing**
   1. **primireFactura()**

Pentru metoda primireFactura din clasa Casier s-a efectuat urmatorul test:

* testPrimireFacturaInexistenta() – este folosit pentru situatia in care facture ce se primeste este nula
  1. **incaseazaPlataFacturii()**

Pentru metoda incaseazaPlataFacturii() din clasa Casier s-au efectuat urmatoarele teste:

* testPlataFacturiiNegativa() – este folosita in situatia in care valoare inregistrata pentru suma primita este negativa
* testPlataFacturiiNula() - este folosita in situatia in care valoare inregistrata pentru suma primita este nula
* testPlataFacturiiNereceptionata() - este folosita in situatia in care factura nu a fost mai intai marcata ca si primita inaintea primirii sumei de bani aferente platii acesteia
* testPlataFacturiiInexistenta() - este folosita in situatia in care numarul facturii este un numar invalid, neexistant posibilitatea existentei unei facturi cu acel numar
  1. **validareActDeIdentitate()**

Pentru metoda validareActDeIdentitate() din clasa Casier s-a efectuat urmatorul test:

* testValidareCnp() – se verifica daca cmp-ul introdus este correct
  1. **daRest()**

Pentru metoda daRest() din clasa Casier s-a efectuat urmatorul test:

* testdaRest() – se verifica daca restul calculat de casier este echivalent cu diferenta dintre suma primita si suma de plata
  1. **verificareSoldLaSfarsitulZilei()**

Pentru metoda verificareSoldLaSfarsitulZilei() din clasa Casier s-a efectuat urmatorul test:

* testVerificareSoldLaSfarsitulZileiSoldIncorect() – se verifica daca soldul de la sfarsitul zilei este diferit de cel initial si de asemenea se verfica daca soldul este modificat dar nu s-a inregistrat nici o facture

1. **Definirea Test Case-urilor**

Am implementat 2 test case-uri: TestFlux si TestSuite. Primul test case contine variate test ce au fost definite mai sus. In cadrul celui de al doilea test case am definit un test suite.

1. **Definirea Test Suite-ului**

Test Suite este definit in clasa TestSuite\_. In cadrul acestui test suite am adaugat 3 metode din test case-ul TestFlux precum si intregul test case.

1. **Definirea functiilor aplicatiei**

Functii din cadrul clasei Casier:

* 1. **primireFactura()**

Aceasta functie este apelata in momentul in care casierul primeste o factura noua si o inregistreaza in lista facturilor din ziua respectiva.

* 1. **incaseazaPlataFacturii()**

Aceasta functie este apelata in momentul in care casierul primeste banii pentru plata facturii inregistrate la pasul anterior.

* 1. **validareActDeIdentitate()**

Aceasta functie este apelata in momentul in care casierul primeste actul de identitatea al persoanei care face plata.

* 1. **daRest()**

Aceasta functie este apelata in momentul in care casierul da restul clientului.

* 1. **verificareSoldLaSfarsitulZilei()**

Aceasta functie este apelata in momentul in care casierul isi face monetarul la sfarsitul zilei.