**Especificació Memòria**

**Codi i Test Errors**

Adrià Bergé Aguilar – 1303442

Martí Miquel Mas Susi – 1304576

Elies Martos Robles – 1333775

Jesus Orduña Heredia - 1271279

Flavius Stefan Nicu - 1304371

Contingut

[1-Historial de Revisions 3](#__RefHeading__463_1255263808)

[2-Introducció 4](#__RefHeading__465_1255263808)

[3-Errors modificats del codi 6](#__RefHeading__467_1255263808)

[4-Test 9](#__RefHeading__469_1255263808)

[5-Conclusió 10](#__RefHeading__471_1255263808)

1. Historial de Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versió** | **Descripció** | **Autor** |
| 27/02/2013 | 0.0 | Esquelet Projecte | TOTS |
| 10/04/2013 | 1.0 | Primera revisió d’errors | TOTS |
| 12/04/2013 | 1.1 | Revisió i correcció dels errors | TOTS |
| 15/04/2013 | 1.2 | Creació dels Test | TOTS |
| 20/05/2013 | 1.3 | Modificació del text | TOTS |
| 26/05/2013 | 1.4 | Modificació projecte | TOTS |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Introducció

La companyia comercial Discon Market treballa des de fa 25 anys en la venta de productes a particulars i disposa d'un establiment de venda al públic. La filosofia de l'empresa es: "Tots els productes que tenim estan a disposició del client", es a dir, que tot el local comercial ha de ser dedicat a la venda, fins i tot el magatzem.

La venda es realitza en el mateix local els dies laborables de 10 : 00 a 14 : 00 i de 17:00 a 22 : 00.

La companyia vol disposar d'un sistema informàtic per millorar la gestió de l'empresa i continuar creixent.

La companyia esta dividida des del punt de vista organitzatiu en quatre departaments:

1. Gestió de vendes.

2. Gestió d'entregues.

3. Gestió logística.

4. Gestió contable.

1. Errors modificats del codi

**Classe Catàleg**

* El primer canvi realitzat ha sigut modificar les posicions de els variables “idProd” i el “nombreProd”

A més hem modificat les següents funcions:

* writeProductToTxt()

Aquesta funció modificarà la forma en la que inserim el nostre producte dintre del txt i també la forma en la que el tractem

* readProductStr()

Llegeix el producte i el passa a String perquè el puguem tractar.

Modifiquem la funció “GetProd()” de la classe “Catàleg” a un mètode “públic” per tal de poder accedir al objecte producte del nostre programa.

**Classe Client**

Hem creat variables noves i la primera funció que hem modificat es:

* Inicialització de variables.
* Sets i Gets.

Aquestes modificacions es fan a partir de les noves variables que hem creat.

* Identificar-se

Sense aquest mètode no podem comprovar que el nostre client es el correcte

**Classe Compra**

Hem modificat algunes funcions com:

* EfectuarCompra()
* RegistrarCompra()
* AfegirProducteCarro()

Perquè ens modifiqui el stock que hi ha dintre del nostre txt i a més les compres realitzades pel usuari.

**Classe Constantes**

En aquesta classe hi guardarem les variables publiques estàtiques i també variables publiques finals per poder accedir a elles des de qualsevol part del nostre programa per poder utilitzar-les com si fos un arxiu de configuració.

**Classe GestionarCompres**

* El primer canvi que vam realitzar es un canvi dintre del “main” a una variable de valor “públic static” perquè la poguéssim utilitzar en qualsevol lloc de la nostre aplicació.
* Canviem el patró de disseny a un model “Singleton”

**Classe LiniaCompra**

Hem implementat un mètode “GetPreu()” que retorna el preu total de la compra realitzada.

* Creació de Sets i Gets.

**Classe Main**

DesconectarClient() ho hem inserit dintre d’aquesta classe a la opció 6 perquè no ens doni errors

* Modifiquem el tipus de tractat de la condició “col·leccions” per utilitzar arrays.
* Afegim dintre del nostre menú principal el cas 7 perquè ens permeti sortir de l’aplicació.

**Classe Productes**

Hem modificat la funció:

* UpdateProdTxt()

Canviem la posició dins de la funció “UpdateProdTxt()” del “Getcost()” per el “GetUnitats()”

Hem modificat la funció “IdProdStr” de la classe ”Producte” a la següent forma:

* ProdDades[1] -> ProdDades[0]

Així en retornarà l’ID del nostre producte en comptes del nom del nostre producte.

A més utilitzem el str perquè ens doni tota la línia del nostre producte i poder modificar-la.

**Altres**

Per poder generar tot el nostre sistema hem creat 3 fitxers anomenats:

* Clients.txt
* Productes.txt
* Registro.txt

Aquests fitxers s’aniran actualitzant amb les dades noves que poden ser d’un usuari, d’un producte o un registre de compra.

Aquest es un dels primers errors que vam trobar al codi, sense aquests fitxers, el codi no executava res.

1. Test

**Classe ConstantesTests**

Aquest test lo que ens farà serà comprovar que hi ha una ruta o un objecte que existeix.

**Classe TestCataleg**

- Hem inicialitzar els vectors "names" i "description" com "Vector<String> i, el vector "prizes" com "Vector<Float>

- Hem influït els .txt necessaris per comprovar els test i hem creat la classe "ConstantesTests.java" on declarem les rutes dels arxius.

- Hem inicialitzat la variable "catalegFilename" amb la ruta que tenim guardada a la classe "ConstantesTests.java"

- Hem fet una crida a la funció "testAfegirProducte" a la funció principal del test.

- A la funció "testContainsProducte" hem iniciat l’iterador com "Iterator<Producte>".

- Implementem la funció “TestSetProducte”, que afegeix a la llista un nou producte amb el nom, la descripció i el preu que l’introduim.

- Modifiquem la funció “testEliminarProductes”, a la que afegim el paràmetre “String numero” per poder passar-li a la funció “eliminarProducto” dins de la classe “Cataleg”

**Classe TestClient**

-El primer canvi que hem tingut que realitzar ha sigut el ”SetUp” on hem afegit tots els test de la classe ( “testDefaultClient” , “testFull”, “testModifiers”, “testIdentificar”, “testCompra”).

**Classe TestProducte**

Dins d’aquesta classe utilitzarem diferents mètodes per a comprovar que podem inserir i veure si els productes estan correctament inserits o son iguals.

1. Conclusió

Una vegada realitzat la feina de implementació del codi de l anterior equip de treball, podem afirma que s'han modificat una gran quantitat de línies de codi i que s'han realitzat tests per confirma que s'ha fet una bona implementació i que tot el codi es sòlid de manera que passa els tests descrits per el client per acceptar el treball realitzat.