

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

ÍNDEX

1- Introducció	pag. 3
2- Errors modificats del codi	pag. 4
3- Tests	pag. 10
4- Conclusió	pag. 11

MEMÒRIA CODI

CANVIS I MODIFICACIONS DINS DEL PROJECTE

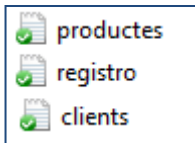
1- Introducció

El propòsit d'aquest document és documentar tots els canvis realitzats del codi en el projecte final de Enginyeria del Software del comerç DISCON 2013 per al correcte funcionament del programa.

2- Errors modificats del codi

- Es generen els següents fitxers:

- clients.txt
- productes.txt
- resgistro.txt



Cadascun dels arxius ha de contenir una dada mínima.

Com per exemple:

-1303442-Adria-Berge-47949587S-15856325874532-639183439-1303442

A més sense aquest fitxers algunes parts del codi no es poden executar.

Per poder obtenir un funcionament correcte aquests fitxers s'aniran modificant amb noves dades.

- Per poder utilitzar algunes variables dins el codi, les hem modificat de categoria. Una d'elles és la de "gestion" dintre del nostre "main" passant-la a un valor "public static". Si no fem aquesta modificació el nostre arxiu no estarà disponible en un altre lloc de la nostra aplicació.

A més el nostre objecte "gestió de compres" el modificarem amb un disseny Singleton.

```
import org.uab.etse.es.E500.GestionarCompres;  
  
public class main {  
  
    public static GestionarCompres gestion;
```

- Dins de la classe “Catàleg” hem modificat la posició de la “ID” i el “nom del producte”. També hem tingut que modificar les següents funcions:

- writeProductToTxt()

```
private void WriteProdToTxt(String nombreProd, String idProd, int unistatsProd,
// Registro de productos
FileWriter fr;
BufferedWriter ficheroRegistro;
try {
    fr = new FileWriter(new File(StorageFilename()), true);
    ficheroRegistro = new BufferedWriter(fr);
    String sCadena = nombreProd.toString() + "-" + idProd.toString() + "-" +
    ficheroRegistro.append(sCadena);
    ficheroRegistro.close();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}
```

ABANS

```
private void WriteProdToTxt(String nombreProd, String idProd, int unistatsProd,
// Registro de productos
FileWriter fr;
BufferedWriter ficheroRegistro;
try {
    fr = new FileWriter(new File(StorageFilename()), true);
    ficheroRegistro = new BufferedWriter(fr);
    String sCadena = idProd.toString() + "-" + nombreProd.toString() + "-" +
    ficheroRegistro.append(sCadena);
    ficheroRegistro.close();
} catch (IOException e) {
    e.printStackTrace();
}
}
```

DESPRÉS

- ReadProductStr()

```
private Producte ReadProdStr(String sCadena){
    Producte auxProd = new Producte();
    String[] ProdDades = sCadena.split("-");
    // Update values
    auxProd.setNombre(ProdDades[0]);
    auxProd.setIdProducte(ProdDades[1]);
    auxProd.setUnitats(Integer.parseInt(ProdDades[2]));
    auxProd.setCost(Float.valueOf(ProdDades[3]).floatValue());
    auxProd.setStockOptim(Integer.parseInt(ProdDades[4]));
    auxProd.setStockMinim(Integer.parseInt(ProdDades[5]));
    auxProd.setDescripcio(ProdDades[6]);

    return auxProd;
}
```

ABANS

```
private Producte ReadProdStr(String sCadena){
    Producte auxProd = new Producte();
    String[] ProdDades = sCadena.split("-");
    // Update values
    auxProd.setNombre(ProdDades[1]);
    auxProd.setIdProducte(ProdDades[0]);
    auxProd.setCost(Float.valueOf(ProdDades[2]).floatValue());
    auxProd.setUnitats(Integer.parseInt(ProdDades[3]));
    auxProd.setStockOptim(Integer.parseInt(ProdDades[4]));
    auxProd.setStockMinim(Integer.parseInt(ProdDades[5]));
    auxProd.setDescripcio(ProdDades[6]);

    return auxProd;
}
```

DESPRÉS

En la ultima instancia hem canviat dins de la classe “Producte” la següent funció:

- UpdateProdTxt()

```
private void UpdateProdTxt(){
    String nomRegistro = "productos.txt";
    FileReader fr;
    BufferedReader ficheroRegistro;
    FileWriter fr1;
    BufferedWriter ficheroRegistro1;

    try {
        // reading
        fr = new FileReader (new File(nomRegistro)); //for reading
        ficheroRegistro = new BufferedReader(fr);
        String sCadena;
        int max = 1000;
        String[] prodStr = new String[max];
        int pos = 0;
        ficheroRegistro = new BufferedReader(fr);
        while ((sCadena = ficheroRegistro.readLine())!=null) {
            prodStr[pos] = sCadena;
            pos++;
        }

        //writing
        fr1 = new FileWriter(new File(nomRegistro)); // for writing
        ficheroRegistro1 = new BufferedWriter(fr1);
        for (int i=0; i<pos;i++){
            if (!getIdProducto().equals(IdProdStr(prodStr[i]))){
                ficheroRegistro1.append(prodStr[i]);
                ficheroRegistro1.newLine();
            }
        }
        ficheroRegistro1.append(getNombre().toString() + "-" + getIdProducto())
        ficheroRegistro.close();
    }
```

```
private void UpdateProdTxt(){
    String nomRegistro = Constantes.RutaProductos;
    FileReader fr;
    BufferedReader ficheroRegistro;
    FileWriter fr1;
    BufferedWriter ficheroRegistro1;

    try {
        // reading
        fr = new FileReader (new File(nomRegistro)); //for reading
        ficheroRegistro = new BufferedReader(fr);
        String sCadena;
        int max = 1000;
        String[] prodStr = new String[max];
        int pos = 0;
        ficheroRegistro = new BufferedReader(fr);
        while ((sCadena = ficheroRegistro.readLine())!=null) {
            prodStr[pos] = sCadena;
            pos++;
        }
        //ArrayList<Producte> list = (ArrayList<Producte>)main.gestion.getCatleg().getProducte();

        //writing
        fr1 = new FileWriter(new File(nomRegistro)); // for writing
        ficheroRegistro1 = new BufferedWriter(fr1);
        for (int i=0; i<pos;i++){
            String IDprod = getIdProducto();
            String idstr = IdProdStr(prodStr[i]);
            if (!IDprod.equals(idstr)){
                ficheroRegistro1.append(prodStr[i]);
                ficheroRegistro1.newLine();
            }
        }
    }
```

ABANS

DESPRÉS

- Un altre canvi que hem realitzat dintre de la classe “GestionarCompres” ha sigut modificar la classe “públic static” perquè hem de poder accedir a tot el programa utilitzant més d’un patró de disseny “Singleton”

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

```
private Producte GetProd(String idProd){
    int pos = 0;
    while ((pos<producteSize())&&!(producteToArray()[pos].getIdProducto().equals(idProd))){
        pos++;
    }
    if ((pos!=producteSize()) && (producteToArray()[pos].getIdProducto().equals(idProd)))
        return producteToArray()[pos];
    else
        return null;
}
```

ABANS

```

public Producte GetProd(String idProd){
    int pos = 0;
    int size = producteToArray().length;
    Producte producte = producteToArray()[pos];
    while ((pos<size)&&(!(producte.getIdProducte().equals(idProd)))){ // Mentre
        pos++;
        producte = producteToArray()[pos];
    }
    if ((pos!=size) && (producteToArray()[pos].getIdProducte().equals(idProd)))
        return producte;
    else
        return null;
}

```

DESPRÉS

- Creem una nova classe que s'anomena "Constantes" on hi guardarem les variables públiques estàtiques i també variables públiques finals per poder accedir a elles des de qualsevol part del nostre programa per poder utilitzar-les com si fos un arxiu de configuració.

```

package org.uab.etse.es.ES00;

public class Constantes {
    private static final String Ruta = "src/org/uab/etse/es/ES00/";
    public static final String RutaClient = Ruta + "clients.txt";
    public static final String RutaProductes = Ruta + "productes.txt";
    public static final String RutaRegistros = Ruta + "registro.txt";
}

```

- Dins la classe "Lineacompra" hem implementat un mètode "GetPreu()" que retorna el preu total de la compra realitzada.

```

public float GetPreu(){
    return unitat * producte.getCost();
}

```

- Hem fet un canvi de posició dins la funció "UpdateProdTxt()" del "Getcost()" per el "GetUnitats()" dins la classe "Producte".

```
ficheroRegistro1.append(getNombre().toString() + "-" + getIdProducte().toString() + "-" + getUnitats() + "-" + getCost() + "-" + getStockOptim() +
```

ABANS

```
ficheroRegistro1.append( getIdProducte().toString()+ "-" +getNombre().toString() + "-" +getCost() + "-" + getUnitats() + "-" + getStockOptim() +
```

DESPRÉS

- Dins el "main", la opció 6 estava buida i hem cridat la opció "DesconectarClient()" per tal que funcioni correctament.

```
case 6: DesconectarClient();
      break;
```

- Modifiquem el tipus de tractat de la condició "col·lecció" per utilitzar arrays.
- 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.

```
private String IdProdStr(String sCadena){
    String[] ProdDades = sCadena.split("-");
    return ProdDades[0];
}
```


- Hem afegit al menú principal el cas 7 que permeti a l'usuari sortir de l'aplicació.

```
case 7:
    String nombre = System.getProperty("os.name");
    if (nombre.substring(0, 1).equals("L")){
        Runtime.getRuntime().exec("clear");
    }
    else{
        Runtime.getRuntime().exec("cmd.exe /cls");
    }
    System.out.println("\nGracies per utilitzar l'aplicacio!");
    System.exit(0);
    break;
default:
    System.out.println("Opcio no valida");
}
System.out.println("-----");
PrintMenu();
```

3- Test

Classe TestCatalog

- 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 "catalogFilename" 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>"

4- Conclusió

Finalitzada tota la correcció del codi podem concloure que el programa funciona correctament.