

ENUNCIADO: La tienda de golosinas de Jaime Nuestro amigo Jaime quiere que le implementemos una aplicación para gestionar su pequeña tienda de golosinas. En la tienda se venden productos de los cuales ha de guardar la referencia, la descripción y el precio. Las golosinas que venden pueden ser de dos tipos: aquellas que se venden por unidades, en cuyo caso el precio que guardan es el de cada unidad y las que se venden al peso, que almacenan el precio por kilo. La referencia es un objeto de la clase Referencia cuyo atributo es un código alfanumérico formado por 3 letras y 4 dígitos. Las letras serán PAP para los productos que se venden al peso y PPU para los productos que se venden por unidades. Los números serán correlativos empezando por 0001 y será asignado por el programa de forma que no habrá dos productos cuya parte numérica coincida. Los que se venden al peso deberán implementar una interfaz llamada AlPeso que definirá un único método llamado calcularPrecio. Dicho método recibirá un peso expresado en gramos y nos devolverá el precio de venta. Tendremos una clase principal con un atributo que será una lista de productos donde se irán guardando todos los productos que se registren en el sistema. Se mostrará una lista de opciones, que se describen a continuación: 1. Registrar producto. Se pedirá la descripción y precio del producto y se añadirá a la lista de productos. A la hora de pedir los datos del producto se deberá distinguir entre los que se venden por unidades o al peso. 2. Realizar compra. Se mostrará la lista de todos los productos y se irá pidiendo al usuario el código de los productos que se añaden a la compra. Cuando se añade un producto al peso se pedirá también el peso en gramos. Cuando el usuario no desee comprar más productos lo indicará escribiendo un 0 como código del producto y entonces el programa mostrará una factura con la compra donde se mostrará una serie de líneas cada una de las cuales contendrá un producto comprado mostrando el código, la descripción, el peso (en el caso de los productos al peso) y el peso total del producto. La última línea contendrá el precio total de la compra. A continuación se preguntará si se desea guardar factura y, en caso de responder que sí, se guardará en un fichero de texto cuyo nombre se preguntará al usuario la misma información que se ha mostrado por pantalla. 3. Salir del programa. Mientras no se pulse esta opción se volverá a mostrar el menú, el programa sólo terminará al indicar esta opción. Antes de cerrar el programa se deberá guardar toda la información de los productos en un fichero binario que deberá recuperarse al arrancar nuevamente el programa. Se deberá controlar todas las posibles excepciones.

MAIN.JAVA

```
package golosinasjaime;
```

```
import java.io.*;
```

```
import java.util.ArrayList;
```

```
import java.util.Iterator;
```

```
public class GolosinasJaime {
```

```
    public static ArrayList<Producto> productos = new ArrayList<Producto>();
```

```
    public static String fProductos = "productos.bin";
```

```
    public static int menu()
```

```
    {
```

```
        int op = 0;
```

```
        while (op<1 || op >3)
```

```
        {
```

```
            System.out.println("1. Registrar producto");
```

```
            System.out.println("2. Realizar compra");
```

```
            System.out.println("3. Salir");
```

```
            op = Lector.leerInt();
```

```
        }
```

```
        return op;
```

```
    }
```

```
    public static int menuProducto()
```

```
    {
```

```
        int op = 0;
```

```
        while (op<1 || op >2)
```

```

{
    System.out.println("1. Producto al peso");
    System.out.println("2. RProducto por unidad");
    op = Lector.leerInt();
}
return op;
}

```

```

public static void registrar()
{
    String desc = Lector.leerLinea("Descripción: ");
    double precio = Lector.leerDouble("Precio: ");
    int tipo = menuProducto();
    if (tipo==1)
        productos.add(new PPeso(desc,precio));
    else
        productos.add(new PUnidad(desc,precio));
}

```

```

public static Producto buscar(String cod)
{
    Iterator<Producto> it = productos.iterator();
    Producto p = null;
    while (it.hasNext())
    {
        p = it.next();
        if (p.getRef().equals(cod))
            return p;
    }
    return null;
}

```

```

public static void comprar()
{

    String compra = "FACTURA\n-----\n\n";
    Iterator<Producto> it = productos.iterator();
    System.out.println("LISTA DE PRODUCTOS");
    System.out.println("-----");
    while (it.hasNext())
        System.out.println(it.next());
    System.out.println("-----\n");

    System.out.println("Indique los productos que desea comprar. Terminar con código 0");
    String cod = Lector.leerLinea("Código: ");
    double precioTotal = 0;
    while (!cod.equals("0"))
    {
        Producto p = buscar(cod);
        if (p!=null)
        {
            if (p instanceof PPeso)
            {
                int peso = Lector.leerInt("Indique el peso en gramos: ");
                compra+=((PPeso) p).mostrar(peso)+"\n";
                precioTotal+=((PPeso) p).calcularPrecio(peso);
            }
            else
            {
                compra+=p.toString()+"\n";
                precioTotal+=p.getPrecio();
            }
        }
    }
}

```

```

    }

    cod = Lector.leerLinea("Código: ");
}

compra+= "-----\n";
compra+= "TOTAL_____"+precioTotal+"â,-\n";
compra+= "-----\n";

System.out.println(compra);

String resp = Lector.leerLinea("¿Desea guardar la factura en un fichero? (sS/nN)");
while (!resp.equalsIgnoreCase("s") && !resp.equalsIgnoreCase("n"))
{
    resp = Lector.leerLinea("¿Desea guardar la factura en un fichero? (sS/nN)");
}

if (resp.equalsIgnoreCase("S"))
    guardarFactura(compra);
}

public static void guardarFactura(String c)
{
    String f = Lector.leerLinea("Indique el nombre del fichero: ");

    BufferedWriter bw = null;

    try
    {
        bw = new BufferedWriter(new FileWriter(f));

        bw.write(c);
    }

    catch (IOException e)
    {

```

```

        System.out.println("Error generando factura");
    }
    finally
    {
        try
        {
            if (bw!=null)
                bw.close();
        }
        catch (IOException e)
        {
            System.out.println("Error cerrando el fichero");
        }
    }
}

public static void guardarProductos()
{

    ObjectOutputStream oos = null;
    try
    {
        oos = new ObjectOutputStream(new FileOutputStream(fProductos));
        oos.writeObject(productos);
    }
    catch (IOException e)
    {
        System.out.println("Error guardando los productos");
    }
    finally

```

```

{
    try
    {
        if (oos!=null)
            oos.close();
    }
    catch (IOException e)
    {
        System.out.println("Error cerrando el fichero");
    }
}
}

```

```

public static void recuperarProductos()
{

    ObjectInputStream ois = null;
    try
    {
        ois = new ObjectInputStream(new FileInputStream(fProductos));
        productos = (ArrayList<Producto>)ois.readObject();
    }
    catch (FileNotFoundException e)
    {

    }
    catch (ClassNotFoundException e)
    {
        System.out.println("Contenido erroneo el fichero");
    }
}

```

```

        catch (IOException e)
        {
            System.out.println("Error leyendo los productos");
        }
        catch (Exception e)
        {
            System.out.println("Contenido erroneo el fichero");
        }
        finally
        {
            try
            {
                if (ois!=null)
                    ois.close();
            }
            catch (IOException e)
            {
                System.out.println("Error cerrando el fichero");
            }
        }
    }
}

```

```

public static void main(String[] args) {

```

```

    recuperarProductos();

```

```

    int op = menu();

```

```

    while (op!=3)
    {

```

```

        switch (op)

```



```
{  
    case 1: registrar(); break;  
    case 2: comprar(); break;  
}  
  
op = menu();  
  
}  
  
guardarProductos();  
  
}  
  
}
```

LECTOR.JAVA

```
package golosinasjaime;
```

```
import java.util.Scanner;
```

```
public class Lector {
```

```
    public static int leerInt()
```

```
    {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        while (true)
```

```
        {
```

```
            try
```

```
            {
```

```
                return sc.nextInt();
```

```
            }
```

```
            catch (Exception e)
```

```
            {
```

```
                System.out.println("Número incorrecto");
```

```
                sc.nextLine();
```

```
            }
```

```
        }
```

```
    }
```

```
    public static int leerInt(String texto)
```

```
    {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        while (true)
```

```
        {
```

```
            try
```

```
            {
```

```
                System.out.print(texto);
```

```
                return sc.nextInt();
```

```
            }
```

```
            catch (Exception e)
```

```
            {
```

```
                System.out.println("Número incorrecto");
```

```
                sc.nextLine();
```

```
            }
```

```
        }
```

```
    }
```

```
    public static double leerDouble()
```

```
    {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        while (true)
```

```
        {
```

```
            try
```

```
            {
```

```
                return sc.nextDouble();
```

```
            }
```

```
            catch (Exception e)
```

```

        {
            System.out.println("Número incorrecto");
            sc.nextLine();
        }
    }

}

public static double leerDouble(String texto)
{
    Scanner sc = new Scanner(System.in);

    while (true)
    {
        try
        {
            System.out.print(texto);
            return sc.nextDouble();
        }
        catch (Exception e)
        {
            System.out.println("Número incorrecto");
            sc.nextLine();
        }
    }
}

public static String leerLinea()
{
    Scanner sc = new Scanner(System.in);

    return sc.nextLine();
}

public static String leerLinea(String texto)
{
    Scanner sc = new Scanner(System.in);

    System.out.print(texto);
    return sc.nextLine();
}
}

```

ALPESO

```
package golosinasjaime;

public interface AlPeso {

    public double calcularPrecio(int gramos);

}
```

PPESO

```
package golosinasjaime;

public class PPeso extends Producto implements AlPeso{

    private static final String cod = "PAP";

    public PPeso(String desc, double precio)
    {
        super(cod,desc,precio);
    }

    @Override
    public double calcularPrecio(int gramos)
    {
        return gramos*precio/1000;
    }

    @Override
    public String toString()
    {
        return super.toString()+precio+"â,¬/Kg";
    }

    public String mostrar(int gramos)
    {
        return
super.toString()+gramos+"gr"+"\\t"+calcularPrecio(gramos)+"â,¬";
    }

}
```

REFERENCIA.JAVA

```
package golosinasjaime;

import java.io.Serializable;

public class Referencia implements Serializable{

    private String ref;
    private static int cont=1;

    public Referencia(String letras)
    {
        ref = cont+"";
        while (ref.length()<4)
            ref = "0"+ref;
        ref = letras+ref;
        cont++;
    }

    public String getRef()
    {
        return ref;
    }

    @Override
    public String toString()
    {
        return ref;
    }
}
```

```
package golosinasjaime;
```

```
public class PUnidad extends Producto{

    private static final String cod = "PPU";

    public PUnidad(String desc, double precio)
    {
        super(cod,desc,precio);
    }

    @Override
    public String toString()
    {
        return super.toString()+precio+"â,¬/u";
    }

}
```

```
import java.io.Serializable;
```

```
public abstract class Producto implements Serializable{

    protected Referencia ref;
    protected String desc;
    protected double precio;

    public Producto (String letras, String desc, double precio)
    {
        ref = new Referencia(letras);
        this.desc = desc;
        this.precio = precio;
    }

    public String getRef() {
        return ref.getRef();
    }

    public double getPrecio() {
        return precio;
    }

    @Override
    public String toString()
    {
        return ref+"\t"+desc+"\t";
    }

}
```