Nuestro amigo Pepe quiere que le implementemos una aplicación para gestionar su pequeña ferretería. El la ferretería se venden productos de los cuales ha de guardar la referencia (código numérico), la descripción y el precio. Los productos que venden pueden ser de dos tipos: aquellos que se venden por unidades, en cuyo caso el precio que guardan es el de cada unidad y los que se venden al peso, que almacenan el peso por kilo. 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 el código, 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.

```
package extra21;

import java.io.*;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Scanner;

/**
 *
 * @author Profesor
 */
public class Extra21 {

    public static ArrayList<Producto> listaProductos = new ArrayList<Producto>();
    public static String fichero = "productos.bin";

    public static void guardarProductos()
    {
        ObjectOutputStream oos = null;
        try
        {
            oos = new ObjectOutputStream(new FileOutputStream(fichero));
            oos.writeObject(listaProductos);
        }
        catch (IOException e)
        {
            System.out.println("Error guardando el fichero");
        }
        finally
        {
            try
```

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

public static void cargarProductos()
{
    ObjectInputStream ois = null;
    try
    {
        ois = new ObjectInputStream(new FileInputStream(fichero));
        listaProductos = (ArrayList<Producto>)ois.readObject();
    }
    catch (ClassNotFoundException e)
    {
        System.out.println("Error en los datos del fichero");
    }
    catch (FileNotFoundException e)
    {
        System.out.println("No hay fichero de productos");
    }
    catch (IOException e)
    {
        System.out.println("Error leyendo el fichero");
    }
```

```
    catch (Exception e)
    {
        System.out.println("Error en los datos");
    }
    finally
    {
        try
        {
            if (ois!=null)
                ois.close();
        }
        catch (IOException e)
        {
            System.out.println("Error cerrando el fichero");
        }
    }
}



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 = Util.leerInt();
    }
    return op;
```

```
}

public static void registrarProducto()
{
   System.out.print("CÃ³digo: ");
   int c = Util.leerInt();
   System.out.print("DescripciÃ³n: ");
   String d = Util.leerLinea();
   int tipo = 0;
   while (tipo!=1 && tipo!=2)
   {
      System.out.println("Â¿Se vende por peso (1) o por unidades (2)?");
      tipo = Util.leerInt();
   }
   System.out.print("Precio por "+((tipo==1)?"kilo":"unidad"));
   double p = Util.leerDouble();
   if (tipo==1)
      listaProductos.add(new PPeso(c, d, p));
   else
      listaProductos.add(new PUnidad(c, d, p));
}

public static void mostrarProductos()
{
   Iterator <Producto> it = listaProductos.iterator();
   while (it.hasNext())
      System.out.println(it.next());
}

public static Producto buscar(int cod)
{
```

```
    Iterator<Producto> it = listaProductos.iterator();
    while (it.hasNext())
    {
        Producto p = it.next();
        if (p.getCodigo()==cod)
            return p;
    }
    return null;
}

public static String generarFactura(ArrayList<Producto> l,
                                    ArrayList<Integer> pesos, double pTotal)
{
    String texto ="";
    for (int i = 0; i < l.size(); i++)
    {
        Producto p = l.get(i);
        texto += p.getCodigo()+"\t"+p.getDesc()+"\t";

        if (p instanceof PPeso)
        {
        texto+=p.getPrecio()+"â,¬/kilo"+"\t"+((PPeso) p).calcularPrecio(pesos.get(i))+"â,¬\n";
        }
        else
        {
            texto+=p.getPrecio()+"â,¬/u"+"\t"+p.getPrecio()+"â,¬\n";
        }
    }
    texto+="TOTAL: "+pTotal;
    return texto;
}
```

```
public static void guardarFactura(String factura, String fichero)
{
   BufferedWriter bw = null;
   try
   {
      bw = new BufferedWriter(new FileWriter(fichero));
            bw.write(factura);
   }
   catch (IOException e)
   {
      System.out.println("Error escribiendo factura");
   }
   finally
   {
      try
      {
         if (bw!=null)
            bw.close();
      }
      catch (IOException e)
      {
         System.out.println("Error cerrando fichero de factura");
      }
   }
}

public static void realizarCompra()
{
   ArrayList<Producto> listaCompra = new ArrayList<Producto>();
   ArrayList<Integer> pesos = new ArrayList<Integer>();
```

```
double precioTotal = 0;
mostrarProductos();
System.out.println("Indica los cÃ³digos de los productos a comprar. Termina con 0");
int c = Util.leerInt();
while (c!=0)
{
   Producto p = buscar(c);
   if (p==null)
      System.out.println("No existe ningÃºn producto con cÃ³digo "+c);
   else if (p instanceof PPeso)
   {
      System.out.print("Peso en gramos: ");
      int gramos = Util.leerInt();
      listaCompra.add(p);
      pesos.add(gramos);
      precioTotal+=((PPeso) p).calcularPrecio(gramos);
   }
   else
   {
      listaCompra.add(p);
      pesos.add(0);
      precioTotal+=p.getPrecio();
   }
   mostrarProductos();
   c = Util.leerInt();
}
String factura = generarFactura(listaCompra, pesos, precioTotal);
System.out.println(factura);
int op = 0;
while (op!=1 && op!=2)
{
```

```
            System.out.println("Â¿Desea guardar la factura en un fichero?\n1. SÃ-\n2.no");
            op = Util.leerInt();
        }
        if (op==1)
        {
            System.out.print("Nombre del fichero: ");
            guardarFactura(factura, Util.leerLinea());
        }

    }

    public static void main(String[] args) {
        cargarProductos();
        int op = menu();
        while (op!=3)
        {
            switch (op)
            {
                case 1: registrarProducto(); break;
                case 2: realizarCompra(); break;
            }
            op = menu();
        }
        guardarProductos();
    }

}
```

PRODUCTO

```java
package extra21;

import java.io.Serializable;

/**
 *
 * @author Profesor
 */
public abstract class  Producto implements Serializable{

    protected int codigo;
    protected String desc;
    protected double precio;

    public Producto(int c, String d, double p)
    {
        codigo = c;
        desc = d;
        precio = p;
    }

    public int getCodigo() {
        return codigo;
    }

    public double getPrecio() {
        return precio;
    }

    public String getDesc() {
        return desc;
    }


}
```

PUNIDAD

```java
package extra21;

/**
 *
 * @author Profesor
 */
public class PUnidad extends Producto{

    public PUnidad(int c, String d, double p)
    {
        super(c,d,p);
    }


    @Override
    public String toString()
    {
        return codigo+"\t"+desc+"\t"+precio+"â,¬/unidad";
    }

}
```

PPESO

```java
package extra21;

/**
 *
 * @author Profesor
 */
public class PPeso extends Producto implements AlPeso{

    public PPeso(int c, String d, double p)
    {
        super(c,d,p);
    }

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

    @Override
    public String toString()
    {
        return codigo+"\t"+desc+"\t"+precio+"â,¬/kilo";
    }
```

```
}
```

ALPESO

```
package extra21;

/**
 *
 * @author Profesor
 */
public interface AlPeso {

    public double calcularPrecio(int peso);

}
```