

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package Examen.carniceria;
```

```
/**
 *
 * @author anusk
 */
public class Administrador extends Usuario {

    public Administrador(String u, String p) {
        super(u, p);
    }

}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package Examen.carniceria;
```

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Scanner;
```

```
/**
 *
 * @author anusk
 */
```

```
public class CarniceriaManolo {
```

```
    public static ArrayList<Producto> productos = new ArrayList<Producto>();
    public static ArrayList<Usuario> usuarios = new ArrayList<Usuario>();
    public static double ganancias = 0;
    public static Usuario logueado = null;
    public static String fUsuarios = "usuarios.bin";
    public static String fProductos = "productos.bin";
    public static String fGanancia = "ganancia.txt";
```

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

```
    public static void mostrarProducto() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Búsqueda: ");
        String busq = sc.nextLine().toUpperCase();
        Iterator<Producto> it = productos.iterator();
        while (it.hasNext()) {
            Producto p = it.next();
```

```

        if (String.valueOf(p.getCodigo()).contains(busq)
            || p.getDesc().toUpperCase().contains(busq))
            System.out.println(p);
    }
}

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

public static void nuevoProducto() {
    Scanner sc = new Scanner(System.in);
    System.out.print("Código: ");
    int cod = Util.leerInt();

    Producto p = buscar(cod);
    if (p == null) {
        System.out.print("Descripción: ");
        String desc = sc.nextLine();
        System.out.print("Stock: ");
        int stock = sc.nextInt();
        System.out.print("Precio: ");
        double precio = Util.leerDouble();
        productos.add(new Producto(cod, desc, stock, precio));
        System.out.println("Producto añadido");
    } else {
        System.out.println("Stock actual: " + p.getStock());
        System.out.print("Nuevo stock: ");
        int stock = sc.nextInt();
        p.setStock(stock);
        System.out.println("El stock ha sido modificado");
    }
}

public static void mostrarGanancia() {
    System.out.println("Ganancia acumulada: " + ganancias);
}

public static Usuario existe(String us) {
    Iterator<Usuario> it = usuarios.iterator();
    while (it.hasNext()) {
        Usuario u = it.next();
        if (u.getUser().equals(us))

```

```

        return u;
    }
    return null;
}

public static void altaUsuario() {
    Scanner sc = new Scanner(System.in);
    int op = 0;
    while (op != 1 && op != 2) {
        System.out.println(
            "Qué tipo de usuario desea dar de alta?\n\t1.
Administrador\n\t2. Cliente");
        op = Util.leerInt();
    }
    System.out.print("Usuario: ");
    String user = sc.nextLine();
    while (existe(user) != null) {
        System.out.print("Usuario: ");
        user = sc.nextLine();
        if (existe(user) != null)
            System.out.println("Usuario ya existente en el sistema");
    }
    System.out.print("Password: ");
    String pass = sc.nextLine();
    if (op == 1)
        usuarios.add(new Administrador(user, pass));
    else
        usuarios.add(new Cliente(user, pass));
}

public static void desconectar() {
    logueado = null;
}

public static void comprar() {
    Scanner sc = new Scanner(System.in);

    System.out.print("Código: ");
    int cod = Util.leerInt();

    Producto p = buscar(cod);

    if (p == null) {
        System.out.println("No existe el producto solicitado");
    } else {
        System.out.print("Unidades: ");
        int cantidad = sc.nextInt();
        if (p.getStock() >= cantidad) {

```

```

        p.setStock(p.getStock() - cantidad);
        ganancias += p.getPrecio() * cantidad;
    } else {
        System.out.println("No existe stock suficiente");
    }
}
}

public static void registrar() {
    Scanner sc = new Scanner(System.in);

    System.out.print("Usuario: ");
    String user = sc.nextLine();
    while (existe(user) != null) {
        System.out.print("Usuario: ");
        user = sc.nextLine();
        if (existe(user) != null)
            System.out.println("Usuario ya existente en el sistema");
    }
    System.out.print("Password: ");
    String pass = sc.nextLine();
    Cliente c = new Cliente(user, pass);
    usuarios.add(c);
    logueado = c;
}

public static void registrarAdmin() {
    Scanner sc = new Scanner(System.in);

    String user = null;

    System.out.print("Usuario: ");
    user = sc.nextLine();
    System.out.print("Password: ");
    String pass = sc.nextLine();
    Administrador ad = new Administrador(user, pass);
    usuarios.add(ad);
    logueado = ad;
}

public static void login() {
    Scanner sc = new Scanner(System.in);
    System.out.print("Usuario: ");
    String user = sc.nextLine();
    System.out.print("Password: ");
    String pass = sc.nextLine();
    Usuario u = existe(user);
    if (u == null || !u.getPass().equals(pass))
        System.out.println("Usuario o contraseña inválidos");
}

```

```

        else
            logueado = u;
    }

    public static int menuAdministrador() {
        int op = 0;
        while (op < 1 || op > 7) {
            System.out.println("1. Mostrar todos los productos");
            System.out.println("2. Mostrar un producto por código/descripción");
            System.out.println("3. Añadir producto / modificar stock");
            System.out.println("4. Mostrar Ganancia");
            System.out.println("5. Dar de alta a un usuario");
            System.out.println("6. Desconectar");
            System.out.println("7. Salir");
            op = Util.leerInt();
        }

        return op;
    }

    public static int menuCliente() {
        int op = 0;
        while (op < 1 || op > 5) {
            System.out.println("1. Mostrar todos los productos");
            System.out.println("2. Mostrar un producto por código/descripción");
            System.out.println("3. Comprar");
            System.out.println("4. Desconectar");
            System.out.println("5. Salir");
            op = Util.leerInt();
        }

        return op;
    }

    public static int menuInvitado() {
        int op = 0;
        while (op < 1 || op > 5) {
            System.out.println("1. Mostrar todos los productos");
            System.out.println("2. Mostrar un producto por código/descripción");
            System.out.println("3. Registrar");
            System.out.println("4. Login");
            System.out.println("5. Salir");
            op = Util.leerInt();
        }
    }

```

```

        return op;
    }

    public static void guardarDatos() {

        ObjectOutputStream oos = null;
        try {
            oos = new ObjectOutputStream(new FileOutputStream(fUsuarios));
            oos.writeObject(usuarios);
        } catch (IOException e) {
            System.out.println("Error guardando datos");
        } finally {
            try {
                if (oos != null)
                    oos.close();
            } catch (IOException e) {
                System.out.println("Error cerrando el fichero");
            }
        }

        oos = null;
        try {
            oos = new ObjectOutputStream(new FileOutputStream(fProductos));
            oos.writeObject(productos);
        } catch (IOException e) {
            System.out.println("Error guardando datos");
        } finally {
            try {
                if (oos != null)
                    oos.close();
            } catch (IOException e) {
                System.out.println("Error cerrando el fichero");
            }
        }

        BufferedWriter bw = null;
        try {
            bw = new BufferedWriter(new FileWriter(fGanancia));
            bw.write(String.valueOf(ganancias));
        } catch (IOException e) {
            System.out.println("Error guardando datos");
        } finally {
            try {
                if (bw != null)
                    bw.close();
            } catch (IOException e) {
                System.out.println("Error cerrando el fichero");
            }
        }
    }

```

```
}
```

```
public static void recuperarDatos() {
```

```
    boolean inicio = true;
    ObjectInputStream ois = null;
    try {
        ois = new ObjectInputStream(new FileInputStream(fUsuarios));
        usuarios = (ArrayList<Usuario>) ois.readObject();
        inicio = false;
    } catch (FileNotFoundException e) {
        registrarAdmin();
    } catch (ClassNotFoundException e) {
        System.out.println("Error leyendo datos");
    } catch (IOException e) {
        System.out.println("Error leyendo datos");
    } catch (Exception e) {
        System.out.println("Error leyendo datos");
    } finally {
        try {
            if (ois != null)
                ois.close();
        } catch (IOException e) {
            System.out.println("Error cerrando el fichero");
        }
    }
    if (!inicio) {
        ois = null;
        try {
            ois = new ObjectInputStream(new FileInputStream(fProductos));
            productos = (ArrayList<Producto>) ois.readObject();
        } catch (ClassNotFoundException e) {
            System.out.println("Error leyendo datos");
        } catch (IOException e) {
            System.out.println("Error leyendo datos");
        } catch (Exception e) {
            System.out.println("Error leyendo datos");
        } finally {
            try {
                if (ois != null)
                    ois.close();
            } catch (IOException e) {
                System.out.println("Error cerrando el fichero");
            }
        }
    }
}
```

```
    BufferedReader br = null;
```

```
    try {
        br = new BufferedReader(new FileReader(fGanancia));
```



```

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

```

```

public static void main(String[] args) {

    recuperarDatos();
    int op = 0;

    boolean salir = false;

    while (!salir) {
        if (logueado == null) {
            op = menuInvitado();
            switch (op) {
                case 1 :
                    mostrarTodosProductos();
                    break;
                case 2 :
                    mostrarProducto();
                    break;
                case 3 :
                    registrar();
                    break;
                case 4 :
                    login();
                    break;
                case 5 :
                    salir = true;
            }
        } else if (logueado instanceof Administrador) {
            op = menuAdministrador();
            switch (op) {
                case 1 :
                    mostrarTodosProductos();
                    break;
                case 2 :

```

```

        mostrarProducto();
        break;
    case 3 :
        nuevoProducto();
        break;
    case 4 :
        mostrarGanancia();
        break;
    case 5 :
        altaUsuario();
        break;
    case 6 :
        desconectar();
        break;
    case 7 :
        salir = true;
    }
} else if (logueado instanceof Cliente) {
    op = menuCliente();
    switch (op) {
        case 1 :
            mostrarTodosProductos();
            break;
        case 2 :
            mostrarProducto();
            break;
        case 3 :
            comprar();
            break;
        case 4 :
            desconectar();
            break;
        case 5 :
            salir = true;
    }
}
}
guardarDatos();
}
}

```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package Examen.carniceria;
```

```
/**
 *
 * @author anusk
 */
public class Cliente extends Usuario {

    public Cliente(String u, String p) {
        super(u, p);
    }

}
```

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

```
package Examen.carniceria;
```

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

```
/**
```

```
 *
```

```
 * @author anusk
```

```
 */
```

```
public class Producto implements Serializable {
```

```
    private int codigo;  
    private String desc;  
    private int stock;  
    private double precio;
```

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

```
    public int getCodigo() {  
        return codigo;  
    }
```

```
    public String getDesc() {  
        return desc;  
    }
```

```
    public int getStock() {  
        return stock;  
    }
```

```
    public double getPrecio() {  
        return precio;  
    }
```

```
    public void setStock(int stock) {  
        this.stock = stock;  
    }
```

```
    @Override  
    public String toString() {
```

```
        return codigo + "\t" + desc + "\t" + stock + " unidades\t" + precio  
            + " €/u";  
    }  
}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package Examen.carniceria;
```

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

```
/**
```

```
 *
```

```
 * @author anusk
```

```
 */
```

```
public abstract class Usuario implements Serializable {
```

```
    protected String user;
```

```
    protected String pass;
```

```
    public Usuario(String u, String p) {
```

```
        user = u;
```

```
        pass = p;
```

```
    }
```

```
    public String getUser() {
```

```
        return user;
```

```
    }
```

```
    public String getPass() {
```

```
        return pass;
```

```
    }
```

```
}
```

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Examen.carniceria;

import java.util.Scanner;

/**
 *
 * @author anusk
 */
public class Util {

    public static int leerInt() {
        Scanner sc = new Scanner(System.in);
        while (true) {
            try {
                return sc.nextInt();
            } catch (Exception e) {
                System.out.println("Número no válido");
                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 no válido");
                sc.nextLine();
            }
        }
    }
}

```