

```

/*
 * 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.taller;

/**
 *
 * @author brc-9
 */
public class Reparacion extends Trabajo {

    protected double precioMaterial;

    public double getPrecioMaterial() {
        return precioMaterial;
    }

    public void setPrecioMaterial(double precioMaterial) {
        this.precioMaterial = precioMaterial;
    }

    public Reparacion(String descripcion) {
        super(descripcion);
        this.precioMaterial = 0;
    }

    public void usarMaterial(double precio) {
        if (!estaFinalizado) {
            precioMaterial = precioMaterial + precio;
            System.out.println("Precio material actual: " + getNumeroHoras());
        } else {
            System.err.println("El trabajo esta finalizado");
        }
    }
}

```

```

/*
 * 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.taller;

/**
 *
 * @author brc-9
 */
public class RepChapaPintura extends Reparacion {

    public RepChapaPintura(String descripcion) {
        super(descripcion);
    }

    @Override
    protected double calcularPrecio() {
        return (super.calcularPrecio()) + (precioMaterial * 1.3); // To change
        // body of
        // generated
        // methods,
        // choose
        // Tools |
        // Templates.
    }

    @Override
    public String toString() {
        return "\nReparacion Chapa y Pintura\n"
            + "-----\n" + "Identificador Trabajo:"
            + identificadorTrabajo + "\n Descripcion: " + descripcion
            + "\n Numero Horas: " + numeroHoras + "\n Trabajo finalizado: "
            + estaFinalizado + "\n Precio Material: " + precioMaterial
            + "\n Precio del servicio: " + calcularPrecio()
            + "\n-----";
    }
}

```

```

/*
 * 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.taller;

/**
 *
 * @author brc-9
 */
public class RepMecanica extends Reparacion {

    public RepMecanica(String descripcion) {
        super(descripcion);
    }

    @Override
    protected double calcularPrecio() {
        return (super.calcularPrecio()) + (precioMaterial * 1.1); // To change
        // body of
        // generated
        // methods,
        // choose
        // Tools |
        // Templates.
    }

    @Override
    public String toString() {
        return "\nReparacion Mecanica\n" + "-----\n"
            + "Identificador Trabajo:" + identificadorTrabajo
            + "\n Descripcion: " + descripcion + "\n Numero Horas: "
            + numeroHoras + "\n Trabajo finalizado: " + estaFinalizado
            + "\n Precio Material: " + precioMaterial
            + "\n Precio del servicio: " + calcularPrecio()
            + "\n-----";
    }
}

```

```
/*
 * 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.taller;
```

```
/**
```

```
 *
```

```
 * @author brc-9
```

```
 */
```

```
public class Revision extends Trabajo {
```

```
    public Revision(String descripcion) {
        super(descripcion);
    }
```

```
    @Override
    protected double calcularPrecio() {
        return (super.calcularPrecio()) + 20;
    }
```

```
    @Override
    public String toString() {
        return "\nRevision\n" + "-----\n"
            + "Identificador Trabajo:" + identificadorTrabajo
            + "\n Descripcion: " + descripcion + "\n Numero Horas: "
            + numeroHoras + "\n Trabajo finalizado: " + estaFinalizado
            + "\n Precio del servicio: " + calcularPrecio()
            + "\n-----";
    }
```

```
}
```

```
/*
 * 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.taller;
```

```
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;
```

```
/**
 *
 * @author brc-9
 */
```

```
public class Taller {
```

```
    static String rutaFichero = "trabajos.dat";
    static ArrayList<Trabajo> trabajos = null;
```

```
    public static void main(String[] args) {
        leerArchivoBinario();
        recuperarUltimoIdentificador();
```

```
        int opcionMenu = menu();
```

```
        while (opcionMenu != 6) {
            switch (opcionMenu) {
                case 1 :
                    nuevoTrabajo();
                    break;
                case 2 :
                    anadirHoras();
                    break;
                case 3 :
                    anadirMaterial();
                    break;
                case 4 :
                    finalizarTrabajo();
                    break;
                case 5 :
                    listarTrabajos();
                    break;
```

```

        }
        opcionMenu = menu();
    }

    escribirArchivoBinario();
}

private static void leerArchivoBinario() {
    File fichero = new File(rutaFichero);
    if (fichero.exists()) {
        ObjectInputStream objInputStream = null;
        try {
            objInputStream = new ObjectInputStream(
                new FileInputStream(fichero));
            trabajos = (ArrayList<Trabajo>) objInputStream.readObject();
        } catch (IOException e) {
            System.err.println("Error abriendo el fichero");
        } catch (ClassNotFoundException e) {
            System.err.println("Formato incorrecto de fichero");
        } finally {
            try {
                if (objInputStream != null) {
                    objInputStream.close();
                }
            } catch (IOException e) {
                System.err.println("Error cerrando el fichero");
            }
        }
    } else {
        trabajos = new ArrayList<Trabajo>();
    }
}

private static void escribirArchivoBinario() {
    File fichero = new File(rutaFichero);
    ObjectOutputStream objOutputStream = null;
    try {
        objOutputStream = new ObjectOutputStream(
            new FileOutputStream(fichero));
        objOutputStream.writeObject(trabajos);
    } catch (IOException e) {
        System.err.println("Error abriendo el fichero");
    } finally {
        try {
            if (objOutputStream != null) {
                objOutputStream.close();
            }
        } catch (IOException e) {
            System.err.println("Error cerrando el fichero");
        }
    }
}

```

```

    }
}

private static int menu() {
    String mensajeMenu = "1. Nuevo trabajo\n" + "2. Añadir horas \n"
        + "3. Añadir material \n"
        + "4. Finalizar trabajo y generar factura\n"
        + "5. Listar trabajos\n" + "6. Salir\n";
    int opcionMenu;
    opcionMenu = Teclado.leerEntero(mensajeMenu, 1, 6);
    return opcionMenu;
}

private static void nuevoTrabajo() {
    boolean respuestaValida = false;
    String descripcion = "";
    int trabajoElegido = tipoTrabajo();
    Trabajo trabajoNuevo = null;
    while (!respuestaValida) {
        switch (trabajoElegido) {
            case 1 :
                descripcion = Teclado.leerTexto(
                    "Indica una descripcion para el trabajo.");
                trabajoNuevo = new RepMecanica(descripcion);
                trabajos.add(trabajoNuevo);
                respuestaValida = true;
                break;
            case 2 :
                descripcion = Teclado.leerTexto(
                    "Indica una descripcion para el trabajo.");
                trabajoNuevo = new RepChapaPintura(descripcion);
                trabajos.add(trabajoNuevo);
                respuestaValida = true;
                break;
            case 3 :
                descripcion = Teclado.leerTexto(
                    "Indica una descripcion para el trabajo.");
                trabajoNuevo = new Revision(descripcion);
                trabajos.add(trabajoNuevo);
                respuestaValida = true;
                break;
        }
    }
}

private static void anadirHoras() {
    String id = preguntarID();
    int indiceEncontrado = comprobarID(id);
}

```

```

        int numeroHoras;
        if (indiceEncontrado != -1) {
            numeroHoras = Teclado
                .leerEntero("Indica el numero de horas a añadir");
            trabajos.get(indiceEncontrado).incrementarHoras(numeroHoras);
        }
    }

    private static void anadirMaterial() {
        String id = preguntarID();
        int indiceEncontrado = comprobarID(id);
        double precioMaterial;
        if (indiceEncontrado != -1) {
            precioMaterial = Teclado
                .leerDouble("Indica el importe del material a añadir");
            if (trabajos.get(indiceEncontrado) instanceof RepMecanica) {
                ((RepMecanica) trabajos.get(indiceEncontrado))
                    .usarMaterial(precioMaterial);
            } else if (trabajos
                .get(indiceEncontrado) instanceof RepChapaPintura) {
                ((RepChapaPintura) trabajos.get(indiceEncontrado))
                    .usarMaterial(precioMaterial);
            }
            trabajos.get(indiceEncontrado);
        }
    }

    private static void finalizarTrabajo() {
        String id = preguntarID();
        int indiceEncontrado = comprobarID(id);
        if (indiceEncontrado != -1) {
            trabajos.get(indiceEncontrado).setEstaFinalizado(true);
            generarFicheroTexto(
                trabajos.get(indiceEncontrado).getIdentificadorTrabajo(),
                trabajos.get(indiceEncontrado).toString());
        }
    }

    private static void listarTrabajos() {
        for (Trabajo value : trabajos) {
            System.out.println(value.toString());
        }
    }

    private static int tipoTrabajo() {
        // 1 = Reparacion Mecanica, 2 = Reparacion Chapa Pintura 3 = Revision
        int trabajoElegido;
        String mensajeTipoTrabajo = "1 = Reparacion Mecanica\n"
            + "2 = Reparación Chapa Pintura\n" + "3 = Revision\n";
    }

```



```

        trabajoElegido = Teclado.leerEntero(mensajeTipoTrabajo, 1, 3);
        return trabajoElegido;
    }

    private static String preguntarID() {
        String id;
        id = Teclado.leerTexto("Indica el ID del trabajo");
        return id;
    }

    private static int comprobarID(String id) {
        boolean noEstaEnListaOEstaFinalizado = false;
        for (int i = 0; i < trabajos.size(); i++) {
            if (trabajos.get(i).getIdentificadorTrabajo().equals(id)
                && !trabajos.get(i).isEstaFinalizado()) {
                return i;
            } else {
                noEstaEnListaOEstaFinalizado = true;
            }
        }
        if (noEstaEnListaOEstaFinalizado) {
            System.out.println(
                "El identificador no esta en la lista o el trabajo esta
finalizado.");
        }
        return -1;
    }

    private static void generarFicheroTexto(String rutaFichero,
        String informacionAEscribir) {
        File ficheroTexto = new File(rutaFichero);
        BufferedWriter buffWriter = null;
        try {
            buffWriter = new BufferedWriter(new FileWriter(ficheroTexto));
            buffWriter.write(informacionAEscribir);
        } catch (IOException e) {
            System.err.println("Error escribiendo el fichero " + rutaFichero);
        } finally {
            try {
                if (buffWriter != null) {
                    buffWriter.close();
                }
            } catch (IOException e) {
                System.err.println("Error cerrando el fichero.");
            }
        }
    }
}

```

```
private static void recuperarUltimoidentificador() {  
    if (!trabajos.isEmpty()) {  
        String ultimoidentificador = trabajos.get(trabajos.size() - 1)  
            .getIdentificadorTrabajo();  
        String parteNumericaString = ultimoidentificador.replace("T", "0");  
        int parteNumerica = Integer.valueOf(parteNumericaString);  
        Trabajo.setContadorIdentificador(parteNumerica);  
    }  
}
```

```

/*
 * 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.taller;
```

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

```
/**
```

```
*
```

```
* @author brc-9
```

```
*/
```

```
public class Teclado {
```

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

```
    public static int leerEntero(String mensaje, int min, int max) {
        int numero = 0;
        boolean respuestValida = false;
        System.out.println(mensaje);
        while (!respuestValida) {
            try {
                numero = sc.nextInt();
                if (numero >= min && numero <= max) {
                    respuestValida = true;
                } else {
                    System.err.println(
                        "Introduce un numero entre " + min + " y " + max);
                }
            } catch (InputMismatchException e) {
                sc.nextLine();
                System.err.println("Introduce un numero entero");
            }
        }
        return numero;
    }
}
```

```
    public static int leerEntero(String mensaje) {
        int numero = 0;
        boolean respuestValida = false;
        System.out.println(mensaje);
        while (!respuestValida) {
            try {
                numero = sc.nextInt();
                respuestValida = true;
            } catch (InputMismatchException e) {

```

```

        sc.nextLine();
        System.err.println("Introduce un numero entero");
    }
}
return numero;
}

public static double leerDouble(String mensaje, int min, int max) {
    double numero = 0;
    boolean respuestValida = false;
    System.out.println(mensaje);
    while (!respuestValida) {
        try {
            numero = sc.nextDouble();
            if (numero >= min && numero <= max) {
                respuestValida = true;
            } else {
                System.err.println(
                    "Introduce un numero entre " + min + " y " + max);
            }
        } catch (InputMismatchException e) {
            sc.nextLine();
            System.err.println("Introduce un numero decimal");
        }
    }
    return numero;
}

```

```

public static double leerDouble(String mensaje) {
    double numero = 0;
    boolean respuestValida = false;
    System.out.println(mensaje);
    while (!respuestValida) {
        try {
            numero = sc.nextDouble();
            respuestValida = true;
        } catch (InputMismatchException e) {
            sc.nextLine();
            System.err.println("Introduce un numero decimal");
        }
    }
    return numero;
}

```

```

public static String leerTexto(String mensaje) {
    String texto;
    System.out.println(mensaje);
    texto = scString.nextLine();
    return texto;
}

```

}

}

```
/*
 * 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.taller;
```

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

```
/**
```

```
 *
```

```
 * @author brc-9
```

```
 */
```

```
public abstract class Trabajo implements Serializable {
```

```
    protected String identificadorTrabajo;
```

```
    protected String descripcion;
```

```
    protected int numeroHoras;
```

```
    protected boolean estaFinalizado;
```

```
    static int contadorIdentificador;
```

```
    public String getIdentificadorTrabajo() {
```

```
        return identificadorTrabajo;
```

```
    }
```

```
    public void setIdentificadorTrabajo(String identificadorTrabajo) {
```

```
        this.identificadorTrabajo = identificadorTrabajo;
```

```
    }
```

```
    public String getDescripcion() {
```

```
        return descripcion;
```

```
    }
```

```
    public void setDescripcion(String descripcion) {
```

```
        this.descripcion = descripcion;
```

```
    }
```

```
    public int getNumeroHoras() {
```

```
        return numeroHoras;
```

```
    }
```

```
    public void setNumeroHoras(int numeroHoras) {
```

```
        this.numeroHoras = numeroHoras;
```

```
    }
```

```
    public boolean isEstaFinalizado() {
```

```
        return estaFinalizado;
```

```
    }
```

```

public void setEstaFinalizado(boolean estaFinalizado) {
    this.estaFinalizado = estaFinalizado;
}

public static int getContadorIdentificador() {
    return contadorIdentificador;
}

public static void setContadorIdentificador(int contadorIdentificador) {
    Trabajo.contadorIdentificador = contadorIdentificador;
}

public Trabajo(String descripcion) {
    contadorIdentificador++;
    this.identificadorTrabajo = generarIdentificador();
    this.descripcion = descripcion;
    this.numeroHoras = 0;
    this.estaFinalizado = false;
}

private String generarIdentificador() {
    String identificador = "T";
    int numeroDeCeros;
    numeroDeCeros = 7 - String.valueOf(contadorIdentificador).length();
    for (int i = 0; i < numeroDeCeros; i++) {
        identificador += "0";
    }
    identificador += String.valueOf(contadorIdentificador);
    return identificador;
}

protected void incrementarHoras(int horas) {
    if (!estaFinalizado) {
        numeroHoras = numeroHoras + horas;
        System.out.println("Numero de horas actuales: " + getNumeroHoras());
    } else {
        System.err.println("El trabajo esta finalizado");
    }
}

protected double calcularPrecio() {
    double precio;
    precio = 30 * numeroHoras;
    return precio;
}

@Override
public String toString() {
    return "";
}

```

}

}