Dirección General de Educación Secundaria, Formación Profesional y Enseñanzas de Régimen Especial
CONSEJERÍA DE EDUCACIÓN, JUVENTUD Y DEPORTE
Comunidad de Madrid

Convocatoria: Ordinaria Pendientes **Ciclo:** Desarrollo de Aplicaciones Web

Módulo: Programación

DATOS DEL ASPIRANTE		CALIFICACIÓN
APELLIDOS:		
NOMBRE:	Fecha	

INSTRUCCIONES GENERALES PARA LA REALIZACIÓN DE LA PRUEBA

La prueba consiste en un único proyecto en el que se pide implementar un programa principal y las clases especificadas en el enunciado.
No se podrá hacer uso de internet durante el examen.
Para entregar el examen deberás comprimir todas las carpetas del proyecto y guardar el archivo comprimido con el nombre APELLIDOS_NOMBRE.ZIP. Deberás subir dicho archivo a la plataforma virtual por lo que tendrás que conocer tu usuario y contraseña.

CRITERIOS DE CALIFICACIÓN Y VALORACIÓN

Para aprobar el examen es imprescindible que éste no tenga errores de compilación. Un examen con errores de compilación tendrá una nota máxima de 2 puntos.
Los criterios de calificación serán los siguientes:

- Los datos se guardan y se recuperan correctamente en el fichero binario (1,5 puntos)
- Se finalizan correctamente los trabajos y se exportan correctamente al fichero de (Finalizar trabajo y generar factura) (1 punto)
- Jerarquía e implementación de las clases. Los atributos y métodos llevan los modificadores adecuados y están en las clases que les corresponde. (1,5 punto)
- Los métodos funcionan correctamente según se especifica en el enunciado. Se crea correctamente el id de trabajo, se modifican las horas y precios de material correctamente, etc. (1,5 punto)
- Manejo de la lista de trabajos (añadir trabajo, modificar trabajo existente,… (1,5 puntos)
- Se muestra correctamente el listado de trabajos (0,5 punto)
- Funcionamiento del programa *main*. (1 punto)
- Control de las excepciones (1,5 puntos)

ENUNCIADO

Se desea realizar una aplicación que permita a los mecánicos de un garaje registrar, consultar y actualizar los trabajos (reparaciones y revisiones) que han sido realizados o que están en proceso de realización en el garaje.

Cada trabajo se identifica unívocamente por su "identificador de trabajo". El "identificador de trabajo" es un código de 8 caracteres alfanuméricos cuyo primer carácter es T seguido 7 dígitos numéricos y que se asocia con el trabajo en el momento que se registra. El primer trabajo registrado tendrá el identificador "T0000000", el segundo "T0000001" y así sucesivamente, y será asignado por el programa.

Los trabajos incluyen una pequeña descripción de la reparación o revisión a realizar.

Todos los trabajos incluyen el número de horas que van siendo necesarias para su realización. Al crear un trabajo el número de horas es 0. El número de horas irá aumentando a medida que los mecánicos van dedicando tiempo a realizar la reparación o la revisión. Cuando el trabajo se ha finalizado se marca como "finalizado" y el número de horas no puede volver a cambiarse.

Todos los trabajos deben incluir un método *incrementarHoras(int horas)* que incrementará el número de horas dedicadas a dicho trabajo, siempre y cuando el trabajo no esté en estado finalizado.

Las reparaciones incluyen el precio del material utilizado (piezas o pintura). Al registrar una reparación el precio del material es 0 y va aumentando a medida que los mecánicos van utilizando material en la reparación. Las reparaciones tendrán un método llamado *usarMaterial(double precio)* que aumentará el importe total del material utilizado en la reparación. Una vez que la reparación se marca como "finalizada" no se puede cambiar el precio del material utilizado.

El precio a cobrar para cada trabajo se compone de una parte fija que resulta de multiplicar el número de horas empleadas por 30€. Además, dependiendo del tipo de trabajo el coste varía de la siguiente manera:

- Reparación mecánica: su precio se calcula como fijo más el coste material multiplicado por 1.1
- Reparación de chapa y pintura: su precio se calcula como fijo más el coste material multiplicado por 1.3
- Revisión: su precio se calcula como fijo más extra independiente del número de horas de 20€.

Para calcular el precio todos los trabajos incluirán un método llamado *calcularPrecio()*.

Para poder mostrar la información de un trabajo todas las clases implementarán un método toString que generará un String con el tipo de trabajo, código, descripción, número de horas, coste del material en el caso de las reparaciones y precio final.

El programa principal deberá gestionar una única lista de trabajos, para lo que mostrará un menú con las siguientes opciones:

1. Nuevo trabajo
2. Añadir horas
3. Añadir material
4. Finalizar trabajo y generar factura
5. Listar trabajos
6. Salir

- **Nuevo trabajo**: registrará un nuevo trabajo. Para ello le preguntará al usuario el tipo de trabajo (reparación mecánica, reparación de chapa y pintura o revisión) y la descripción del mismo, añadiendo el nuevo trabajo creado a la lista de trabajos.
- **Añadir horas:** preguntará al usuario el id del trabajo al que le va a incrementar las horas y el número de horas a incrementar. Si el id del trabajo existe y no está finalizado, incrementará sus horas.
- **Añadir material**: preguntará al usuario el id del trabajo al que le va a añadir el coste de material y el importe de dicho material. Si el id del trabajo existe, no está finalizado y además se trata de una reparación, incrementará el importe del material.
- **Finalizar trabajo y generar factura**: preguntará por id de un trabajo y, si éste existe, Marcará el trabajo como finalizado y generará un fichero de texto llamado TXXXXXXX.FAC, siendo TXXXXXXX el id del trabajo, con la información del trabajo tal y como se muestra por pantalla.
- **Listar trabajos**: muestra por pantalla la información de todos los trabajos de la lista
- **Salir**: finaliza el programa. Antes de salir deberá guardar en el fichero binario trabajos.dat la lista de trabajos.

Cada vez que se ejecute el programa se comprobará si existe un fichero binario llamado trabajos.dat donde se almacenarán todos los trabajos y, si existe, lo carga en la lista de trabajos del programa.
Antes de finalizar el programa se guardará la lista de trabajos actual en el fichero trabajos.dat.
Se implementará como mínimo las clases *Trabajo*, *Reparacion*, *Revision* y *Taller* (clase que incluye el método *main*), aunque se podrá implementar más clases si se considera oportuno.