

Examen de la Tercera Evaluación del módulo Programación

Se propone realizar un programa/software que tendrá que ser desarrollado siguiendo el paradigma de programación orientada a objetos POO.

El software a desarrollar consistirá en un programa que deberá tratar los datos de un concesionario de coches, que dispone de su propio taller de reparación de vehículos. Este concesionario deberá almacenar información de tres tipos de vehículos, los recientemente comprados, los que están en venta y los coches que han sido dejado por los clientes en el taller del concesionario. Para la realización de este software, se deberán seguir las siguientes indicaciones:

- Existirá una clase llamada Coches que deberá contener:
 - Los siguientes atributos:
 - **matricula (tipo String):** Contendrá el número de matrícula del coche y seguirá el formato XXXX-YYY (ej. 5678-BFG).
 - **marca (tipo String):** Contendrá el nombre de la marca del coche.
 - **modelo (tipo String):** Contendrá el nombre del modelo del coche.
 - **Tipo_ope (tipo enumerado "Operacion"):** Contendrá los distintos tipos de operaciones que se realizan en el taller. Estas operaciones serán: **Compra**, **Venta**, **Reparacion**
 - **fmatric (tipo LocalDate):** Contendrá la fecha de matriculación del coche, con el siguiente formato YYYY-MM-DD (formato por defecto).
 - **antiguedad (tipo int):** Contendrá los años que tiene el coche desde su matriculación. Será calculado por medio del método **calAntiguedad**.
 - **valor (tipo int):** Contendrá el valor del coche en euros.
 - Tendrá un **constructor** con **6 parámetros**: la **matricula (tipo String)**, la **marca (tipo String)**, el **modelo (tipo String)**, el **tipo de operación (tipo String)**, la **fecha de matriculación (tipo LocalDate)** y el **valor del vehículo (tipo entero)**.

Nota: En el caso de tipo de operación, como al constructor le pasamos un String, y el atributo es de tipo enumerado "Operación", deberemos realizar una conversión de String a tipo Operacion (tipo enumerado), para ello, podéis usar el siguiente método:

```
Operacion.valueOf(valor); -> valor un String
```

Dentro del constructor se llamará al método:

void calAntiguedad: Que no devolverá nada, pero nos servirá para inicializar el valor del atributo antiguedad, se le pasará como parámetro únicamente la fecha de matriculación del vehículo.

- Habrá un método `get` y `set` por cada uno de los atributos de esta clase, a excepción del atributo `antigüedad` que solo tendrá un método `get`.
- Además se implementará el método abstracto de la interfaz `Comparable`, que permitirá ordenar los vehículos por su número de matrícula. Teniendo en cuenta que para comparar dos matrículas siempre se tendrá que comparar primero las letras de las mismas, y en caso de que sean iguales, compararemos los números.
- También habrá que implementar un método `String toString()` que nos permita mostrar los datos de la siguiente forma:

Coche{Nº de Matrícula: 3467-LKJ, Marca: Fiat, Modelo: Ducato, Tipo de Operación: Compra, Fecha de Matriculación: 2010-12-04, Antigüedad: 12, Valor (€): 12000}

- La clase principal del programa deberá hacer lo siguiente:
 - Inicialmente creará una colección de objetos de tipo `Coches`, llamado **regCoches**.
 - Cargará los datos almacenados en el fichero binario, **concesionario.dat**, en la colección de objetos de tipo `Coches`, **regCoches**.
 - A continuación deberá mostrar el siguiente menú al usuario, del que solo saldremos si se elige la opción 5:

Elija alguna de las siguientes opciones:

1. Añadir un nuevo coche al concesionario.

2. Borrar un coche del concesionario.

3. Crear informe del concesionario en un fichero de texto

4. Mostrar todos los datos de los coches del concesionario

5. Salir

- **Opción 1:** Añade un nuevo Coche a la colección de objetos, **regCoches**. Se deberá pedir al usuario que vaya introduciendo los valores de ese vehículo.
- **Opción 2:** Borra un determinado Coche de la colección de objetos, **regCoches**. Se pedirá al usuario que introduzca el número de matrícula del coche que se desea borrar, ese será el criterio para buscar ese objeto Coche y eliminarlo, reduciendo el tamaño de la colección de objetos, **regCoches**.
- **Opción 3:** Permitirá crear un informe en un fichero de texto, llamado **Informe Concesionario.txt**, en este informe se deberá mostrar la siguiente información:

- 1. Número de Coches Comprados.**
- 2. Número de Coches en Venta.**

3. Número de Coches en Reparación.
4. Número total de Coches en el concesionario.
5. Valor total de los coches comprados.
6. Valor total de los coches en venta.
7. Saldo (Valor de euros obtenidos en las ventas menos el valor de euros pagados en las compras).

Un ejemplo del formato a seguir se muestra a continuación:

```
Informe Concesionario: Bloc de notas
Archivo Edición Formato Ver Ayuda
Número de Coches Comprados: 2
Número de Coches en Venta: 2
Número de Coches en Reparación: 1
Número total de Coches en el Concesionario: 5
Valor total de los coches comprados: 23230
Valor total de los coches en venta: 30434
Saldo (Venta - Comprados): 7204
```

- **Opción 4:** Permitirá mostrar por consola los datos de la colección de objetos, `regCoches`, siguiendo el formato del método `toString()` de la clase `Coches`.
- **Opción 5:** Permitirá salir del programa, pero antes deberá guardar los datos actuales de la colección de objetos, `regCoches`, en el fichero binario, `concesionario.dat` para que al iniciar el programa en la siguiente ocasión podamos recuperar los datos de los Coches. Además **se deberá guardar en una base de datos llamada Concesionario** (de MySQL), **únicamente** los datos de los coches cuyo tipo de operación sea **Compra**, **antes de guardar esos datos se deberán borrar todos los datos que contenga la tabla.**

NOTA: A lo largo del programa habrá que realizar el correspondiente manejo y captura de las excepciones que se puedan dar.

CRITERIOS DE EVALUACIÓN

Para la calificación de este examen, se van a seguir los siguientes criterios de calificación:

EXAMEN 3ª EVA.	Creación de las distintas clases y sus correspondientes miembros.	0,5 puntos
	Cargar correctamente los datos desde el fichero binario al array.	1 puntos
	Guardar correctamente los datos del array en el fichero binario.	1 puntos
	Crear el informe en el fichero de texto según las especificaciones.	2 puntos
	Mostrar todos los datos de los coches del concesionario tal y como se especifica en el enunciado.	1 puntos

Ordenación de los datos mostrados según el criterio indicado en de Comparable.	0,5 puntos
Insertar los nuevos datos de un coche en la colección de objetos correctamente.	1 punto
Borrar los datos de un coche en la colección de objetos de forma correcta y siguiendo las especificaciones.	1,5 puntos
Guardar en la BD, la información de los coches de la colección de objetos cuando se cumplan los criterios marcados.	1,5 puntos

En los criterios “Cargar correctamente los datos desde el fichero binario al array” y “Guardar correctamente los datos del array en el fichero binario”, si existe algún error serán calificados con 0 puntos.

En el resto de los criterios se actuará de la siguiente forma:

- **1 fallo supone perder el 50% del valor** correspondiente de ese criterio.
- **2 fallos suponen obtener 0 puntos en ese criterio.**

FORMA DE ENTREGA

El examen se entregará en un único archivo comprimido que contendrá la carpeta del proyecto realizado por el alumno. Este archivo comprimido será el que se deberá subir a la tarea, **“Examen 3ª Evaluación”**, en el aula virtual del módulo de Programación.

Cualquier entrega fuera de tiempo o sin seguir el formato de entrega, hará que dicho examen sea calificado con 0 puntos.

¡¡IMPORTANTE!!: En el caso de encontrar que el trabajo (programa/software) de un equipo no es original, sino que es una copia parcial o total de alguno de los otros equipos, ambos equipos estarán calificados con 0 pto.