Tarea Final de la Unidad PROG07

Realizar un ejercicio en JAVA sobre el desarrollo de clases con Arrays.
Entrega: Deberás crear una carpeta comprimida en formato ‘zip’ con el nombre “Apellido_Nombre_Tarea_Final_PROG07”. Dentro de esta carpeta, incluye el proyecto o el paquete donde están las clases del ejercicio.
Finalmente, sube el archivo comprimido en la tarea correspondiente del aula virtual.

Enunciado del Problema

Título: Gestión de Vehículos

Descripción:

Se debe desarrollar un sistema en **Java** que permita gestionar distintos tipos de **vehículos**, aplicando los principios de **herencia, clases abstractas e interfaces**.
El programa permitirá registrar vehículos, calcular costos de mantenimiento y mostrar información detallada de cada vehículo.

Especificaciones

Clase Abstracta Vehiculo

1. **Atributos:**
    - matricula: Identificación única del vehículo.
    - marca: Marca del vehículo.
    - modelo: Modelo del vehículo.
    - anioFabricacion: Año de fabricación.
    - kilometraje: Kilómetros recorridos.
2. **Constructores:**
    - Constructor parametrizado para inicializar los atributos.
3. **Métodos Abstractos:**
    - calcularCostoMantenimiento(): Debe ser implementado por cada tipo de vehículo, calculando el costo en función de su tipo y uso.

4. **Otros Métodos:**
    - mostrarInformacion(): Devuelve una cadena con toda la información del vehículo, incluyendo el conductor si está asignado**.**

Clases Hijas de Vehículo

Cada tipo de vehículo extiende la clase Vehiculo y sobrescribe el método calcularCostoMantenimiento().

Clase Coche (extends Vehiculo)

- Atributo adicional: tipoCombustible (Gasolina, Diésel, Eléctrico).
- Implementación de calcularCostoMantenimiento():
    - **Gasolina/Diésel**: (kilometraje * 0.05) + 100
    - **Eléctrico**: (kilometraje * 0.03) + 50

Clase Moto (extends Vehiculo)

- Atributo adicional: cilindrada.
- Implementación de calcularCostoMantenimiento():
    - **Menos de 250cc**: (kilometraje * 0.03) + 50
    - **250cc o más**: (kilometraje * 0.05) + 80

Clase Camion (extends Vehiculo)

- Atributo adicional: capacidadCarga (en toneladas).
- Implementación de calcularCostoMantenimiento():
    - **Menos de 10 toneladas:** (kilometraje * 0.08) + 200
    - **10 toneladas o más:** (kilometraje * 0.10) + 300

Interfaz Mantenible

Define un método realizarMantenimiento() que imprimirá un mensaje indicando que el vehículo ha sido enviado a mantenimiento y mostrará el costo calculado.

- **La clase Vehiculo implementará esta interfaz y así las clases hijas heredarán ese método.**

Programa Principal

1. **Estructura de almacenamiento:**
    - Se utilizará un array de máximo 20 vehículos para almacenar objetos de la clase Vehiculo.
2. **Menú interactivo:**

    El programa mostrará un menú con las siguientes opciones:

    a) Registrar vehículo
    - Solicita el tipo de vehículo (Coche, Moto o Camión).
    - Solicita la información correspondiente según el tipo.
    - Añade el vehículo al array si hay espacio disponible.

    b) Realizar mantenimiento de un vehículo
    - Solicita la matrícula del vehículo.
    - Si existe, llama a realizarMantenimiento().

    c) Mostrar información de vehículos
    - Muestra la información detallada de todos los vehículos.

    d) Salir
    - Finaliza el programa.

Rúbrica de Evaluación

Criterio	Descripción	Puntos
Clase Vehiculo	Implementación correcta de atributos y constructores.	2
Método Abstracto	Implementación de calcularCostoMantenimiento() en las subclases.	2
Interfaz Mantenible	Creación de la interfaz e Implementación correcta del método en la clase Vehículo	2
Programa Principal	Menú interactivo funcional con todas las opciones.	2
Validación de datos	Se valida correctamente la entrada del usuario en todas las opciones.	1
Estructura del Código	Buen uso de herencia, clases abstractas, interfaces y estructuras de datos.	1
Total	**10 puntos**	