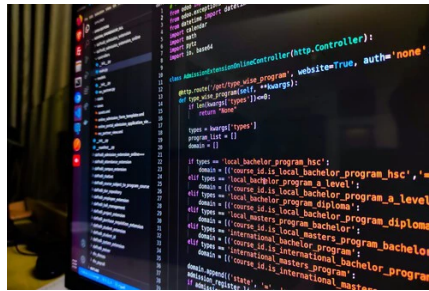




Documentación y jerarquía de código “Herencia y Polimorfismo”



Materia:

Programación orientada a objetos

Elaborado por:

Jonathan Josué Rivera Guido

Angelo Gabriel Soza Espinoza

Davis Samael Brenes Montiel

Isaac Mora Carcamo

Manuel Jirón Martínez

Docente

José Duran García

Fecha 20/09/2025

1. Documentación de código

Este proyecto de Java demuestra la aplicación de conceptos fundamentales de la programación orientada a objetos (POO), como la herencia, el polimorfismo, las clases abstractas y las interfaces. El objetivo es modelar diferentes tipos de vehículos, cada uno con un comportamiento de movimiento único, y algunos con la capacidad de recargar combustible.

La estructura del proyecto está organizada en dos paquetes para mantener un código limpio y modular:

- ❖ Paquete “vehículo”: Contiene la lógica del negocio, es decir, las clases e interfaces que definen a los vehículos.
- ❖ Paquete “run”: Contiene la clase principal (Main), que es el punto de entrada de la aplicación y donde se ejecuta la demostración.

Dentro del paquete “vehículo” se agrupa las clases e interfaces que representan a los diferentes tipos de vehículos:

- a) “Vehículo.java”: Esta es una clase abstracta que sirve como el modelo base para todos los vehículos y posee atributos como marca y modelo. Así mismo, contiene el Método abstracto: mover() que obliga a cualquier subclase a proporcionar su propia versión, imponiendo una estructura común a todas las clases de vehículos, asegurando que todos tengan una marca, un modelo y una forma de moverse.
- b) “Combustible.java”: Esta es una interfaz que define un contrato para cualquier clase que necesite la funcionalidad de "recargar". Contiene el método: recargar() que declara la existencia de la funcionalidad y permite que las clases que la implementan (Carro y Camion) tengan un comportamiento específico para recargar, sin estar ligadas a una única jerarquía de herencia.
- c) “Carro.java, Moto.java, Camion.java”: Representan tipos concretos de vehículos y heredan los atributos y el método mover() de la clase Vehiculo. Así mismo, cada una sobrescribe el método mover() con un comportamiento de movimiento único y específico. No obstante, Carro y Camion implementan la interfaz Combustible, lo que les permite tener su propio método recargar(). Moto no la implementa, lo que demuestra que solo algunos vehículos tienen esta funcionalidad.
- d) Paquete “run”: Este paquete contiene la clase que inicia y ejecuta el programa: “Main.java”, la cual es la clase principal que demuestra el polimorfismo en acción.

Finalmente, se utiliza “instanceof” para verificar si un objeto implementa la interfaz Combustible antes de llamar a su método recargar(). Esto evita errores y demuestra cómo interactuar con objetos de manera segura cuando tienen funcionalidades opcionales.

2. Diagrama de Clases UML

classDiagram
direction TB

```
class Vehiculo {  
    +marca: String  
    +modelo: String  
    +Vehiculo(marca, modelo)  
    +getMarca()  
    +getModelo()  
    <<abstract>> +mover()  
}
```

```
class Carro {  
    +Carro(marca, modelo)  
    +mover()  
    +recargar()  
}
```

```
class Moto {  
    +Moto(marca, modelo)  
    +mover()  
}
```

```
class Camion {  
    +Camion(marca, modelo)  
    +mover()  
    +recargar()  
}
```

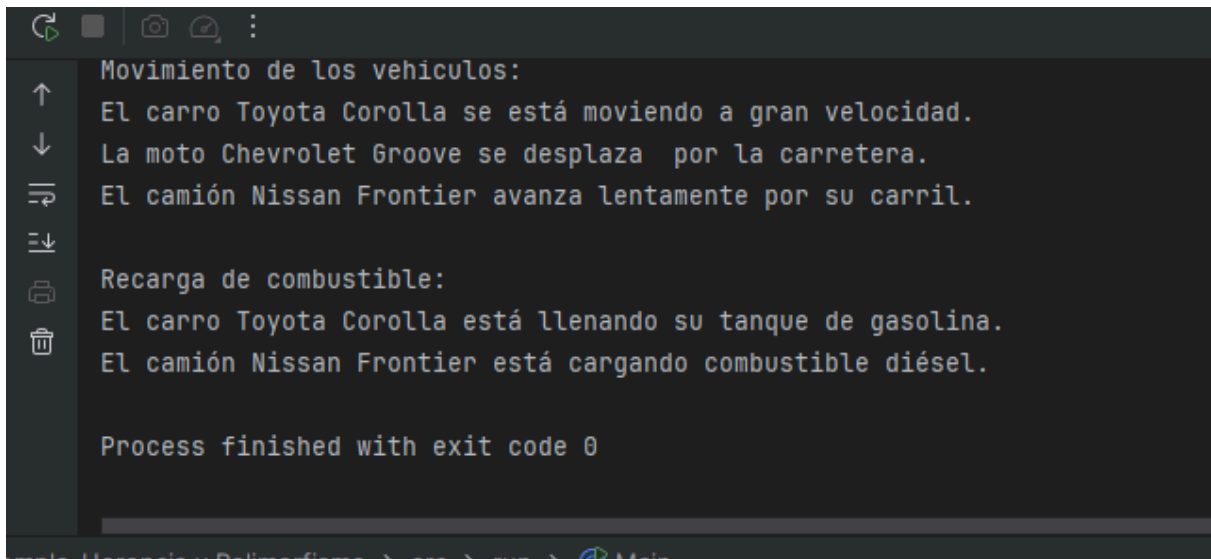
```
class Combustible {  
    <<interface>>  
    +recargar()  
}
```

```
class Main {  
    +main(args: String[])  
}
```

```
Vehiculo <|-- Carro  
Vehiculo <|-- Moto  
Vehiculo <|-- Camion
```

```
Combustible <|.. Carro  
Combustible <|.. Camion
```

3. Salida de código



A screenshot of a terminal window with a dark background. The window has a title bar with standard OS icons (back, forward, search, etc.). On the left side of the terminal, there is a vertical toolbar with icons for navigation (up, down, search, etc.) and editing (copy, paste, etc.). The main area of the terminal displays the following text in a monospaced font:

```
Movimiento de los vehiculos:  
El carro Toyota Corolla se está moviendo a gran velocidad.  
La moto Chevrolet Groove se desplaza por la carretera.  
El camión Nissan Frontier avanza lentamente por su carril.  
  
Recarga de combustible:  
El carro Toyota Corolla está llenando su tanque de gasolina.  
El camión Nissan Frontier está cargando combustible diésel.  
  
Process finished with exit code 0
```