



Universidad de Guanajuato

DIVISIÓN DE CIENCIAS E INGENIERIAS

COMPOSICIÓN ENTRE CLASES - CASO VEHÍCULO

Práctica 6

Vicente Yael Baca Montero 392206
Brian Gabriel Barajas Guerrero 314225
Israel Aaron Segoviano Caudillo 314477

Abril 2023

1 Introduction

En esta práctica se llevo a cabo la implementación de composición entre clases para el caso particular de un vehículo y sus componentes. Partimos del código "Composicion3.java" como referencia y a partir de ahí agregamos componentes e hicimos uso de la composición entre clases. Para esta práctica en particular el uso de la composición fue de gran utilidad puesto que algunos atributos de la clase automovil tenian a su vez atributos propios por lo que era necesario definirlos como clases.

2 Descripcion

1. Tómese como referencia las clases del sistema "Composicion3.java" (Clase Vehículo y clases relacionadas).
2. Agregar más atributos simples, al menos dos más, a cada clase: Vehículo, Motor, Transmisión.
3. Crear al menos UNA clase más que sea un COMPONENTE de vehículo, esto es, que sea una PARTE de vehículo tal como lo es Motor y Transmision. Tips: Chasis, cajuela, etc.
4. Hacer la captura de los atributos simples tanto de vehículo como de motor, transmisión y la NUEVA PARTE. Al menos capturar DOS atributos simples de cada parte.
5. Desplegar los valores de los atributos capturados.

En el código lo primero que vemos es la creación de las clases que serán atributos de la clase automovil, en particular vemos la creación de la clase motor donde definimos sus atributos y creamos el set motor donde se crean objetos para cada atributo. Después creamos la clase motor que solo tiene un atributo (pero podríamos ponerle más de ser necesario) y creamos su setTransmisión. Finalmente llegamos a la clase Chasis a la que le asignamos 4 atributos (numero de serie, color, numero de accesorios, numero de golpes) y creamos su respectivo setChasis. Después de ahí llegaremos a la clase automovil donde haremos uso de la composición al asignar atributos de tipo Motor, Transmicion y Chasis que son las clases creadas con anterioridad con sus respectivos atributos, después, crearemos los set de forma normal para los atributos convencionales de tipos String o int y para los atributos de tipo "Clase" los crearemos por medio de otro set respectivamente.

Por último llegamos a la clase main donde asignamos el valor de cada atributo y ahí mismo imprimimos los datos por medio de println.

3 Código

```
class Motor {  
int cilindros;
```

```

int potencia;
String combustible;

void setMotor(int cilindros, int potencia, String combustible){
    this.cilindros = cilindros;
    this.potencia = potencia;
    this.combustible = combustible;
    System.out.println("setMotor() de Motor");
}

void setCilindros(int cilindros) {
    this.cilindros = cilindros;
}

} //fin de clase Motor

class Transmision{
    String tipo;

    void setTransmision(String tipo){
        this.tipo = tipo;
        System.out.println("setTransmision() de Transmision");
    }

} //Fin clase Transmision

class Chasis{
    String num_serie;
    String color;
    int num_accesorios;
    int num_golpes;

    void setChasis(String num_serie, String color, int num_accesorios, int num_golpes){
        this.num_serie = num_serie;
        this.color = color;
        this.num_accesorios = num_accesorios;
        this.num_golpes = num_golpes;
    }
}

class Automovil {
    String marca;
    int modelo;
    Motor motor = new Motor();
    Transmision transm = new Transmision();
    Chasis chas = new Chasis();
}

```

```

void setMarca (String marca) {
    this.marca = marca;
}
void setModelo(int modelo){
    this.modelo = modelo;
}

void setMotor(int cilindros, int potencia, String combustible) {
    this.motor.setCilindros(cilindros);
    System.out.println("setMotor() de Automovil");
}

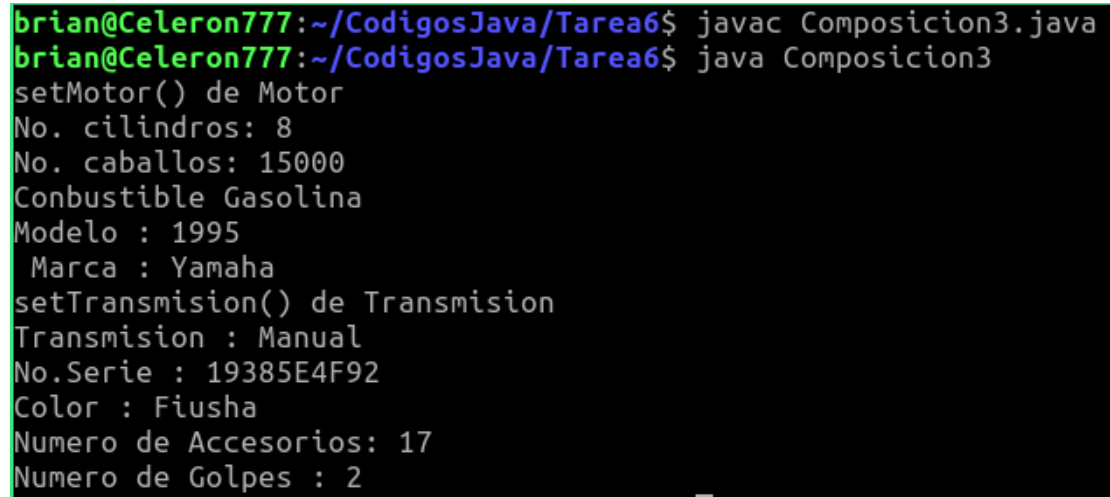
void setTransmision(String tipo){
    this.transm.setTransmision(tipo);
    System.out.println("setTransmision() de Automovil");
}

} //fin clase Automovil

public class Composicion3{
    public static void main(String args[]){
        Automovil auto = new Automovil();
        auto.motor.setMotor(8, 15000, "Gasolina");
        System.out.println("No. cilindros: " + auto.motor.cilindros);
        System.out.println("No. caballos: " + auto.motor.potencia);
        System.out.println("Combustible " + auto.motor.combustible);
        auto.setMarca("Yamaha");
        auto.setModelo(1995);
        System.out.println("Modelo : " + auto.modelo + "\n Marca : " + auto.marca);
        auto.transm.setTransmision("Manual");
        System.out.println("Transmision : " + auto.transm.tipo);
        auto.chas.setChasis("19385E4F92","Fiusha",17,2);
        System.out.println("No. Serie : " + auto.chas.num_serie);
        System.out.println("Color : "+ auto.chas.color);
        System.out.println("Numero de Accesorios: "+ auto.chas.num_accesorios);
        System.out.println("Numero de Golpes : " + auto.chas.num_golpes);
    }
} //fin clase Composicion3

```

4 Pantallas de Ejecución



```
brian@Celeron777:~/CodigosJava/Tarea6$ javac Composicion3.java
brian@Celeron777:~/CodigosJava/Tarea6$ java Composicion3
setMotor() de Motor
No. cilindros: 8
No. caballos: 15000
Combustible Gasolina
Modelo : 1995
  Marca : Yamaha
setTransmision() de Transmision
Transmision : Manual
No.Serie : 19385E4F92
Color : Fiusha
Numero de Accesorios: 17
Numero de Golpes : 2
```

Figure 1: Captura de la ejecución del código

5 Conclusiones

5.1 Vicente Yael Baca Montero

Trabajamos con los distintos tipos de herencia, en particular la composicion, asi como aprender a manipular estos diferentes tipos de herencia, construir los setters, getters y metodos constructores. Además seguimos practicando como acceder a las variables y metodos publicos dependiendo de como declaramos la clase y en que directorio se encuentra el archivo class.

5.2 Brian Gabriel Barajas Guerrero

En esta práctica pudimos practicar la composición de clases en un ejemplo bastante didáctico puesto que un automovil esta compuesto de diferentes piezas que a su vez estan compuestas o caracterizadas por distintos atributos, en esta práctica la composición de clases nos ayudó sobre todo en la parte de abstracción pues al abstraer un objeto complejo es más fácil si imaginamos primero sus piezas más grandes y después pensamos en cómo están constituidas estas piezas por piezas más pequeñas, además de que ayuda mucho a la organización.

5.3 Isael Aaron Segoviano Caudillo

Se practicó la forma en la que se puede abstraer una idea y plasmarla con clases de forma que se tenga composición de clases, siendo una parte de la otra, además de el uso de las variables de cada clase, la forma de utilizarlas correcta y adecuadamente dependiendo de la parte de la sección del código en la que se esté trabajando. En este caso se vio la diferencia de hacer setMotor desde automovil como tambien el setMotor desde motor.