

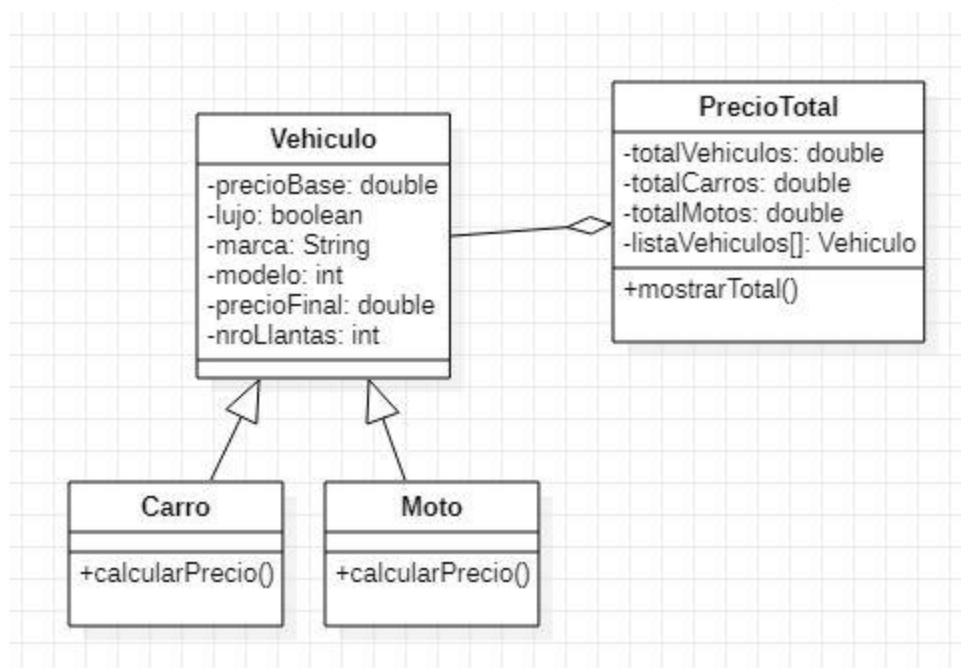


Ciclo 2 Fundamentos de programación

Reto 2

Descripción del problema:

Se le ha contratado a usted para desarrollar un sistema a una empresa que alquila vehículos (carros o motos). Se ha determinado que el modelo de clases mediante el cual se resolverá el problema es el siguiente:



Todos los Vehículos comparten los atributos `precioBase`, `lujo`, `marca` y `modelo` los cuales son valores que pueden modificar su `precioFinal`, se cuenta también con el atributo `precioBase`, el cual representa el precio del Vehículo antes de sumar los respectivos valores según la evaluación de los atributos `lujo`, `marca` y `modelo`. El método `calcularPrecio()` permite obtener el precio final para un Vehículo de la siguiente forma:

Los Vehículos del tipo **Carro** cuentan con los atributos `marca` y `modelo` los cuales agregan un valor adicional al precio de alquiler del Vehículo de la siguiente forma:

1. Si es marca Mazda:
 - Su `precioBase` es de \$10.000 por hora.
 - Si el modelo del Vehículo es mayor a 2012:
 - ✓ Se añade un extra de \$15.000 al precio por hora del vehículo.
 - ✓ Si el vehículo es de lujo se añaden \$10.000 extra al precio por hora.
 - Si el modelo del Vehículo es menor a 2012:
 - ✓ Se añade un extra de \$10.000 al precio por hora del vehículo.
 - ✓ Si el vehículo es de lujo se añaden \$8.000 extra al precio por hora.
2. Si es marca Renault:
 - Su `precioBase` es de \$8.000 por hora.
 - Si el modelo del Vehículo es mayor a 2012:
 - ✓ Se añade un extra de \$12.000 al precio por hora del vehículo.
 - ✓ Si el vehículo es de lujo se añaden \$8.000 extra al precio por hora.
 - Si el modelo del Vehículo es menor a 2012:
 - ✓ Se añade un extra de \$8.000 al precio por hora del vehículo.



- ✓ Si el vehículo es de lujo se añaden \$5.000 extra al precio por hora.

Los Vehículos del tipo Moto cuentan con los atributos marca y modelo los cuales agregan un valor adicional al precio de alquiler del Vehículo de la siguiente forma:

- Si es marca Yamaha:
 - Su precioBase es de \$5.000 por hora.
 - Si el modelo del Vehículo es mayor a 2012:
 - ✓ Se añade un extra de \$2.000 al precio por hora del vehículo.
 - ✓ Si el vehículo es de lujo se añaden \$5.000 extra al precio por hora.
 - Si el modelo del Vehículo es menor a 2012:
 - ✓ Se añade un extra de \$1.000 al precio por hora del vehículo.
 - ✓ Si el vehículo es de lujo se añaden \$2.000 extra al precio por hora.
- Si es marca Suzuki:
 - Su precioBase es de \$3.000 por hora.
 - Si el modelo del Vehículo es mayor a 2012:
 - ✓ Se añade un extra de \$1.000 al precio por hora del vehículo.
 - ✓ Si el vehículo es de lujo se añaden \$3.000 extra al precio por hora.
 - Si el modelo del Vehículo es menor a 2012:
 - ✓ Se añade un extra de \$700 al precio por hora del vehículo.
 - ✓ Si el vehículo es de lujo se añaden \$1.000 extra al precio por hora.

Los atributos de la clase PrecioTotal son: totalVehiculos, totalCarros, totalMotos y listaVehiculos. Este último atributo contiene todos los Vehículo, los cuales son almacenados en un array (tipo Vehículo) y son entregados al constructor de la clase PrecioFinal en el método main(), desde donde se llama al método mostrarTotales(), el cual debe imprimir en consola:

El precio total de los Vehiculos es de { totalVehiculos }

La suma del precio de los Carros es de { totalCarros }

La suma del precio de las Motos es de { totalMotos }

Ejemplo:

Pruebas	Salida
<pre>Vehiculo vehiculos[] = new Vehiculo[4]; vehiculos[0] = new Carro(10000, true, "Mazda", 2015); vehiculos[1] = new Carro(8000, true, "Renault", 2010); vehiculos[2] = new Moto(5000, true, "Yamaha", 2015); vehiculos[3] = new Moto(3000, true, "Suzuki", 2012); PrecioTotal solucion = new PrecioTotal(vehiculos); solucion.mostrarTotales();</pre>	<pre>El precio total de los Vehiculos es de 72700.0 La suma del precio de los Carros es de 56000.0 La suma del precio de las Motos es de 16700.0</pre>

Pruebas	Salida
<pre>Vehiculo vehiculos[] = new Vehiculo[4]; vehiculos[0] = new Carro(10000, false, "Mazda", 2015); System.out.println(vehiculos[0].getPrecioFinal()); vehiculos[1] = new Carro(8000, false, "Renault", 2010);</pre>	<pre>El precio total de los Vehiculos es de 51700.0 La suma del precio de los Carros es de 41000.0 La suma del precio de las Motos es de 10700.0</pre>



```
System.out.println(vehiculos[1].getPrecioFinal());
vehiculos[2] = new Moto(5000, false, "Yamaha",
2015);
System.out.println(vehiculos[2].getPrecioFinal());
vehiculos[3] = new Moto(3000, false, "Suzuki",
2012);
System.out.println(vehiculos[3].getPrecioFinal());
PrecioTotal solucion = new PrecioTotal(vehiculos);
solucion.mostrarTotales();
```

NOTA: Las pruebas son ejecutadas en la clase main App. Esta clase no se debe subir a la plataforma como parte de la solución.

Esqueleto:

```
// Inicio de la solución
public class Vehiculo {
    private final double precioBase;
    private final boolean lujo;
    private String marca;
    private int modelo;
    private double precioFinal;
    private int nroLlantas;

    public Vehiculo(double precioBase, boolean lujo, String marca, int modelo,
int nroLlantas) {
        this.precioBase = precioBase;
        this.lujo = lujo;
        this.marca = marca;
        this.modelo = modelo;
        this.nroLlantas = nroLlantas;
    }
    // Los getters y setters de ser necesario
}

public class Carro extends Vehiculo {

    public Carro(double precioBase, boolean lujo, String marca, int modelo)
{
        super(precioBase, lujo, marca, modelo, 4);
        calcularPrecio();
    }

    public void calcularPrecio() {
        double precio = 0.0;
        // Calculos
        super.setPrecioFinal(precio);
    }
}
```



```
}  
}  
  
public class Moto extends Vehiculo {  
  
    public Moto(double precioBase, boolean lujo, String marca, int modelo) {  
        super(precioBase, lujo, marca, modelo, 2);  
        calcularPrecio();  
    }  
  
    public void calcularPrecio() {  
        double precio = 0.0;  
        // Calculos  
        super.setPrecioFinal(precio);  
    }  
}  
  
public class PrecioTotal {  
    private double totalVehiculos;  
    private double totalCarros;  
    private double totalMotos;  
    private Vehiculo[] vehiculos;  
  
    public PrecioTotal(Vehiculo[] vehiculos) {  
        this.vehiculos = vehiculos;  
    }  
  
    public void mostrarTotales() {  
        // Calculos  
        System.out.println("El precio total de los Vehiculos es de " + this.  
totalVehiculos);  
        System.out.println("La suma del precio de los Carros es de " + this  
.totalCarros);  
        System.out.println("La suma del precio de las Motos es de " + this.t  
otalMotos);  
    }  
}  
  
// Fin de la solución  
  
// Esta clase es para las pruebas, no se debe subir como parte de la solución  
public class App {  
    public static void main(String[] args) {  
        // Casos de Pruebas  
        Vehiculo vehiculos[] = new Vehiculo[4];  
        vehiculos[0] = new Carro(10000, true, "Mazda", 2015);  
        vehiculos[1] = new Carro(8000, true, "Renault", 2010);  
        vehiculos[2] = new Moto(5000, true, "Yamaha", 2015);
```



```
vehiculos[3] = new Moto(3000, true, "Suzuki", 2012);
PrecioTotal solucion = new PrecioTotal(vehiculos);
solucion.mostrarTotales();

Vehiculo vehiculos2[] = new Vehiculo[4];
vehiculos[0] = new Carro(10000, false, "Mazda", 2015);
vehiculos[1] = new Carro(8000, false, "Renault", 2010);
vehiculos[2] = new Moto(5000, false, "Yamaha", 2015);
vehiculos[3] = new Moto(3000, false, "Suzuki", 2012);
PrecioTotal solucion2 = new PrecioTotal(vehiculos);
solucion.mostrarTotales();
}
```

Nota: Recuerde que cada una de las clases debe ser codificada en una clase (archivo independiente), pero se deben de cargar juntas en iMaster.