



Reto 3

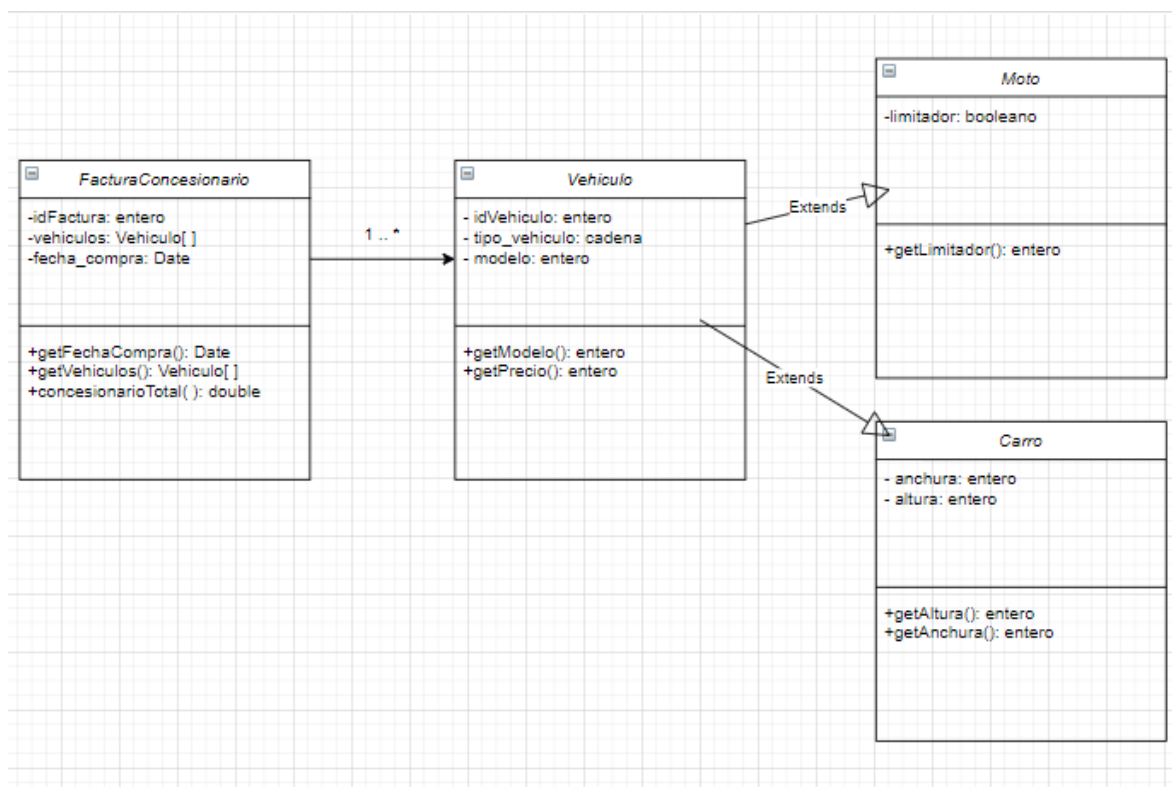
Objetivo

El objetivo de este reto es que el estudiante reconozca y aplique los elementos básicos del paradigma de la programación orientada a objetos en un escenario abstraído de la cotidianidad.

Contexto

El concesionario de carros “AutoMax” ha venido en crecimiento últimamente y cada día su sistema necesita más eficiencia para el cálculo de los totales. Por ende, se necesita un programa que retorne la cantidad total a pagar por el cliente, teniendo en cuenta que hay clientes que pueden comprar a la vez tanto un auto como una moto, teniendo ambos vehículos características únicas que cambian sus precios.

Considere el siguiente diagrama de clases para la implementación de las clases necesarias **FacturaConcesionario.java**, **Vehiculo.java**, **Carro.java** y **Moto.java**





NOTA: Las clases deben llamarse **FacturaConcesionario.java**, **Vehiculo.java**, **Carro.java** y **Moto.java**

Reto

Implemente una función llamada `concesionarioTotal()`, estando ubicada en la clase `FacturaConcesionario` y con un arreglo de tipo `Vehiculo` como miembro dato, en el cual se puede dar el caso de que el vehículo sea un carro o una moto. Dependiendo del modelo de cada vehículo se pueden presentar los siguientes casos:

- a) En caso de que el vehículo sea un carro, tendrá un precio estándar de 20'000.000
- b) Por otra parte, en caso de que el vehículo sea una moto, tendrá un precio estándar de 5'000.000
- c) Por último, en cuanto al modelo del vehículo, por cada año después del 2000 se suma 500.000 al precio del auto. En cambio, si el vehículo es mas antiguo del 2000, no se le suma ningún valor y su precio seria el estándar anteriormente mencionado.

Adicionalmente, use la siguiente imagen como referencia para la construcción de las clases con sus atributos y métodos necesarios.



```
import java.sql.Date;

public class FacturaConcesionario {
    private int id_factura;
    private Vehiculo[] vehiculos;
    private Date fecha_compra;

    public FacturaConcesionario(int id_factura, Vehiculo[] vehiculos, Date fecha_compra) {
    }

    public double concesionarioTotal() {
        double retorno = 0;

        return retorno;
    }

    public FacturaConcesionario() {
        // TODO Auto-generated constructor stub
    }

    int getId_factura() {
        return id_factura;
    }

    void setId_factura(int id_factura) {
        this.id_factura = id_factura;
    }

    Vehiculo[] getVehiculos() {
        return vehiculos;
    }

    void setVehiculos(Vehiculo[] vehiculos) {
        this.vehiculos = vehiculos;
    }

    Date getFecha_compra() {
        return fecha_compra;
    }

    void setFecha_compra(Date fecha_compra) {
        this.fecha_compra = fecha_compra;
    }
}

public class Vehiculo {

    private int id_vehiculo;
    private String tipo_vehiculo;
    private int modelo;

    public Vehiculo(int id_vehiculo, String tipo_vehiculo, int modelo) {
    }

    public Vehiculo() {
        // TODO Auto-generated constructor stub
    }

    int getId_vehiculo() {
        return id_vehiculo;
    }

    void setId_vehiculo(int id_vehiculo) {
        this.id_vehiculo = id_vehiculo;
    }

    String getTipo_vehiculo() {
        return tipo_vehiculo;
    }

    void setTipo_vehiculo(String tipo_vehiculo) {
        this.tipo_vehiculo = tipo_vehiculo;
    }

    int getModelo() {
        return modelo;
    }

    void setModelo(int modelo) {
        this.modelo = modelo;
    }
}
```



```
public class Carro extends Vehiculo {
    private int altura;
    private int anchura;

    public Carro(int altura, int anchura) {
        this.altura = altura;
        this.anchura = anchura;
    }

    public Carro(int id_vehiculo,String tipo_vehiculo,int modelo,int altura, int anchura) {
    }

    public Carro() {
        // TODO Auto-generated constructor stub
    }

    int getAltura() {
        return altura;
    }

    void setAltura(int altura) {
        this.altura = altura;
    }

    int getAnchura() {
        return anchura;
    }

    void setAnchura(int anchura) {
        this.anchura = anchura;
    }
}

public class Moto extends Vehiculo {

    private boolean delimitador;

    public Moto(boolean delimitador) {
        this.delimitador = delimitador;
    }

    public Moto(int id_vehiculo,String tipo_vehiculo,int modelo,boolean delimitador) {
        this.setId_vehiculo(id_vehiculo);
        this.setTipo_vehiculo(tipo_vehiculo);
        this.setModelo(modelo);
        this.delimitador = delimitador;
    }

    public Moto() {
        // TODO Auto-generated constructor stub
    }

    boolean getDelimitador() {
        return delimitador;
    }

    void setDelimitador(boolean delimitador) {
        this.delimitador = delimitador;
    }
}
```



Por último, para un mayor entendimiento de herencia, dejamos a continuación un ejemplo de instanciar una clase haciendo uso de herencia:

```
Empleado[] misEmpleados=new Empleado[6];
misEmpleados[0]=new Empleado("Ana",30000, 2000, 07, 07);
misEmpleados[1]=new Empleado("Carlos",50000, 1995, 06, 15);
misEmpleados[2]=new Empleado("Paco",25000, 2005, 9, 25);
misEmpleados[3]=new Empleado("Antonio",47500, 2009, 11, 9);
misEmpleados[4]=jefe_RRM; //Polimorfismo en acción. Principio de sustitución
misEmpleados[5]=new Jefatura("María",95000, 1999,5,26);
```

Casos de prueba

Finalmente, para verificar el funcionamiento del programa se sugiere considerar los siguientes casos de prueba:

# CASO DE PRUEBA	DATO DE ENTRADA	SALIDA ESPERADA						
1	<table border="1"> <thead> <tr> <th>Carro</th><th>Moto</th><th>Carro</th></tr> </thead> <tbody> <tr> <td>Id_vehiculo: 0 Tipo_vehiculo: "Carro" modelo: 1998 anchura: 75m altura: 15m</td><td>Id_vehiculo: 1 Tipo_vehiculo: "Moto" modelo: 2007 limitador: false</td><td>Id_vehiculo: 2 Tipo_vehiculo: "Carro" modelo: 2003 anchura: 95m altura: 25m</td></tr> </tbody> </table>	Carro	Moto	Carro	Id_vehiculo: 0 Tipo_vehiculo: "Carro" modelo: 1998 anchura: 75m altura: 15m	Id_vehiculo: 1 Tipo_vehiculo: "Moto" modelo: 2007 limitador: false	Id_vehiculo: 2 Tipo_vehiculo: "Carro" modelo: 2003 anchura: 95m altura: 25m	El precio a pagar por estos vehículos es: 50'000.000
Carro	Moto	Carro						
Id_vehiculo: 0 Tipo_vehiculo: "Carro" modelo: 1998 anchura: 75m altura: 15m	Id_vehiculo: 1 Tipo_vehiculo: "Moto" modelo: 2007 limitador: false	Id_vehiculo: 2 Tipo_vehiculo: "Carro" modelo: 2003 anchura: 95m altura: 25m						



2	<table><tr><th>Carro</th><th>Moto</th><th>Carro</th></tr><tr><td>Id_vehiculo: 0 Tipo_vehiculo: "Carro" modelo: 2009 anchura: 105m altura: 15m</td><td>Id_vehiculo: 1 Tipo_vehiculo: "Moto" modelo: 1999 limitador: True</td><td>Id_vehiculo: 2 Tipo_vehiculo: "Carro" modelo: 1999 anchura: 65m altura: 25m</td></tr></table>	Carro	Moto	Carro	Id_vehiculo: 0 Tipo_vehiculo: "Carro" modelo: 2009 anchura: 105m altura: 15m	Id_vehiculo: 1 Tipo_vehiculo: "Moto" modelo: 1999 limitador: True	Id_vehiculo: 2 Tipo_vehiculo: "Carro" modelo: 1999 anchura: 65m altura: 25m	El total a pagar para los asistentes a la película es: 49'500.000
Carro	Moto	Carro						
Id_vehiculo: 0 Tipo_vehiculo: "Carro" modelo: 2009 anchura: 105m altura: 15m	Id_vehiculo: 1 Tipo_vehiculo: "Moto" modelo: 1999 limitador: True	Id_vehiculo: 2 Tipo_vehiculo: "Carro" modelo: 1999 anchura: 65m altura: 25m						
3	<table><tr><th>Moto</th><th>Moto</th><th>Moto</th></tr><tr><td>Id_vehiculo: 0 Tipo_vehiculo: "Moto" modelo: 2010 limitador: True</td><td>Id_vehiculo: 1 Tipo_vehiculo: "Moto" modelo: 2007 limitador: false</td><td>Id_vehiculo: 2 Tipo_vehiculo: "Moto" modelo: 2003 limitador: True</td></tr></table>	Moto	Moto	Moto	Id_vehiculo: 0 Tipo_vehiculo: "Moto" modelo: 2010 limitador: True	Id_vehiculo: 1 Tipo_vehiculo: "Moto" modelo: 2007 limitador: false	Id_vehiculo: 2 Tipo_vehiculo: "Moto" modelo: 2003 limitador: True	El total a pagar para los asistentes a la película es: 25'000.000
Moto	Moto	Moto						
Id_vehiculo: 0 Tipo_vehiculo: "Moto" modelo: 2010 limitador: True	Id_vehiculo: 1 Tipo_vehiculo: "Moto" modelo: 2007 limitador: false	Id_vehiculo: 2 Tipo_vehiculo: "Moto" modelo: 2003 limitador: True						

Entrega:

1. Suba a la plataforma un archivo con el nombre de **FacturaConcesionario.java, Vehiculo.java, Moto.java y Carro.java** este nombre debe de respetarse, dado que, si no se nombre de dicha manera no se tendrá en cuenta para la calificación del reto.
2. **Importante:** Los métodos deben de llamarse **exactamente igual** a como se muestra en el ejemplo de la estructura del código.



El futuro digital
es de todos

MinTIC



Universidad
Pontificia
Bolivariana

Vigilada Mineducación

Mision
TIC 2022

3. **Importante:** Las salidas deben ser tal cual se muestran en los casos de pruebas. De lo contrario, el sistema no lo reconocerá.