



PRACTICA 1

“HERENCIA”

Facultad de ciencias de la computación
Ing. En ciencias de la computación

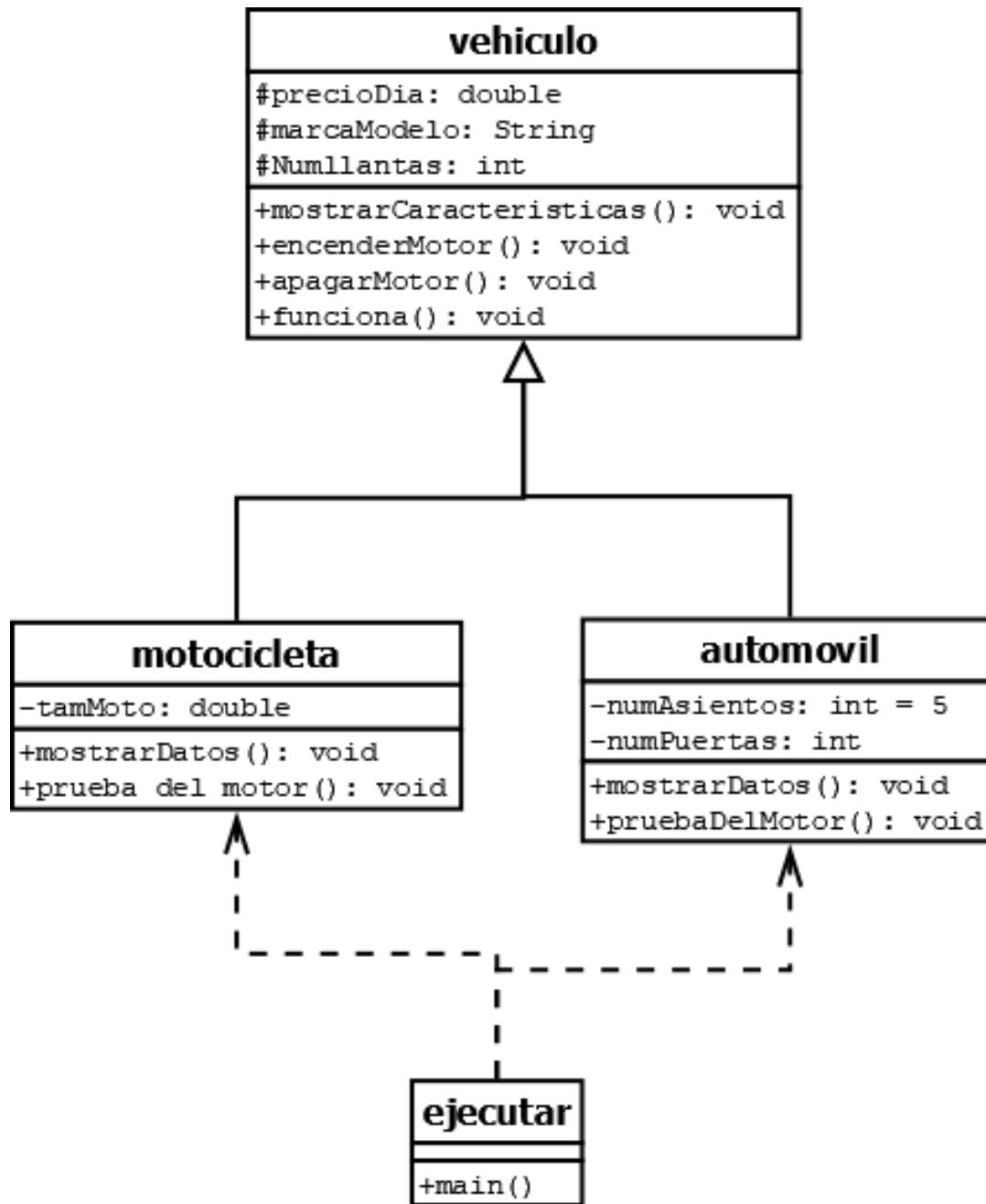


IRVYN XICALE CABRERA
201963582

Introducción

Esta práctica nos ayuda a comprender el funcionamiento de las herencias.

Diagramas ULM



Código

Vehiculo.java

```
public class vehiculo{
    protected int precioDia;
    protected String marcaModelo;
    protected int numllantas;

    // constructor
    public vehiculo(String marcaModelo1, int precioDia1, int numllantas1) {
        marcaModelo = marcaModelo1;
        precioDia = precioDia1;
        numllantas = numllantas1;
        System.out.println("construyo el vehiculo");
    }

    // metodos
    public void mostrarCaracteristicas(){
        System.out.println("Marca: " + marcaModelo);
        System.out.println("Precio: " + precioDia);
        System.out.println("Numero de llantas: " + numllantas);
    }
    public void encenderMotor(){
        System.out.println("el motor se encendio");
    }
    public void apagarMotor(){
        System.out.println("el motor se apago");
    }
}
```

```
public void funciona(){  
    System.out.println("el motor de " + marcaModelo + " funciona correctamente");  
}  
}
```

Motocicleta.java

```
public class motocicleta extends vehiculo{  
    // variables  
    private double tamMoto;  
  
    public motocicleta(String marcaModelo,int precioDia,int numllantas, double tam){  
        super(marcaModelo,precioDia,numllantas);  
        tamMoto = tam;  
        System.out.println("ha creado un auto");  
    }  
    public void mostrarDatos(){  
        super.mostrarCaracteristicas();  
        System.out.println( "Tamaño de la moto: " + tamMoto + "cm");  
    }  
    public void pruebaDelMotor(){  
        super.encenderMotor();  
        super.apagarMotor();  
        super.funciona();  
    }  
}
```

automovil.java

```
public class automovil extends vehiculo{
    // variables
    private int numAsientos = 5;
    private int numPuertas;

    public automovil(String marcaModelo,int precioDia,int numllantas, int asientos, int puertas){
        super(marcaModelo,precioDia,numllantas);
        numAsientos = asientos;
        numPuertas = puertas;
        System.out.println("ha creado un auto");
    }
    public void mostrarDatos(){
        super.mostrarCaracteristicas();
        System.out.println( "Asientos: " + numAsientos);
        System.out.println( "Puertas: " + numPuertas);
    }
    public void pruebaDelMotor(){
        super.encenderMotor();
        super.apagarMotor();
        super.funciona();
    }
}
```

ejecutar.java

```
public class ejecutar {
    public static void main(String[] args){
        // automovil
        automovil v1 = new automovil("volvo 550",120,4,5,4);
    }
}
```

```
v1.mostrarDatos();  
v1.pruebaDelMotor();  
  
// moto  
motocicleta m1 = new motocicleta("Italika",50,2,120);  
m1.mostrarDatos();  
m1.pruebaDelMotor();  
}  
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\irvyn\OneDrive\Pictures\paginaweb\practica1> javac vehiculo.java
PS C:\Users\irvyn\OneDrive\Pictures\paginaweb\practica1> javac motocicleta.java
PS C:\Users\irvyn\OneDrive\Pictures\paginaweb\practica1> javac automovil.java
PS C:\Users\irvyn\OneDrive\Pictures\paginaweb\practica1> javac ejecutar.java
PS C:\Users\irvyn\OneDrive\Pictures\paginaweb\practica1> java ejecutar
construyo el vehiculo
ha creado un auto
Marca: volvo 550
Precio: 120
Numero de llantas: 4
Asientos: 5
Puertas: 4
el motor se encendio
el motor se apago
el motor de volvo 550 funciona correctamente
construyo el vehiculo
ha creado un auto
Marca: Italika
Precio: 50
Numero de llantas: 2
Tamaño de la moto: 120.0cm
el motor se encendio
el motor se apago
el motor de Italika funciona correctamente
PS C:\Users\irvyn\OneDrive\Pictures\paginaweb\practica1> |
```

Conclusión

Esta practica refuerza lo visto en clases sobre la herencia y las relaciones en diagramas UML.

PROGRAMACIÓN II

RÚBRICA HOJA DE EVALUACIÓN DE PRÁCTICAS

MATRICULA: 201963582

FECHA: 24/01/2021

NOMBRE: Xicale Cabrera Irvyn

No. PRACTICA: 1

INDIVIDUAL (x) COLABORATIVA ()

CRITERIOS	Deficiente	Suficiente	Bueno	Excelente	Calificación Obtenida	
	1-5.9	6-7.9	8-9	9.1-10	%	Puntos
CONOCIMIENTO TEÓRICO 20%	Conocimiento deficiente de los fundamentos teóricos de POO y no puede aplicarlos en el diseño de clases.	Conocimiento confuso de los fundamentos teóricos de POO y el diseño de las clases y relaciones es incompleto.	Conocimiento claro de los fundamentos teóricos POO pero requiere mejorar el modelado de las clases, sus métodos y sus relaciones entre clases.	Dominio del Conocimiento de los fundamentos teóricos POO y puede aplicarlos de forma completa en el modelado de las clases, métodos y todas las relaciones entre clase.		
EJECUCIÓN DE LA PRÁCTICA 30%	No puede realizar la práctica ya que desconoce el entorno de trabajo y desarrollo de la práctica en lenguaje UML y Java.	Realiza la práctica de forma incompleta ya que desconoce el entorno de trabajo del lenguaje UML y Java	Realiza la práctica pero requiere mejorar en el manejo del entorno de trabajo del lenguaje de programación(sintaxis y semántica)	Realiza la práctica de forma correcta y completa, demuestra dominio del entorno de trabajo del lenguaje de programación (sintaxis y semántica).		

SOLUCIÓN DE LA PRÁCTICA 40%	No puede generar las soluciones o programas a los problemas planteados ya que no posee el dominio teórico y práctico del modelado y el lenguaje de programación.	Propone soluciones confusas o programas incompletos a los problemas planteados, ya que carece del dominio del modelado y lenguaje de programación	Genera soluciones con poca profundidad y los programas no están orientados de acuerdo a los problemas solicitados, por lo cual no tiene un dominio profundo de la temática y del Lenguaje.	Genera soluciones con profundidad y los programas son correctos de acuerdo a los problemas planteados, por lo cual demuestra un dominio de la temática y del Lenguaje de Programación.		
ACTITUD DE APRENDER Y COLABORAR EN EQUIPO DE TRABAJO 10%	No posee una actitud proactiva para un aprendizaje autónomo y no le gusta participar y trabajar en equipo.	Posee una actitud propositiva para un aprendizaje autónomo, participa pero no le gusta trabajar en equipo.	Posee una actitud propositiva logrando un aprendizaje autónomo, colaborativo, le gusta trabajar en equipo pero requiere mejorar su participación y portaciones de forma profunda.	Posee una actitud proactiva logrando un aprendizaje autónomo, participa con propuestas concretas y profundas, le gusta trabajar en equipo y asume su responsabilidad dentro para lograr éxito del equipo		
Total						

Firma del Alumn@

Vo. Bo. Docente

