



25-5-2024

Introducción a la programación



Cristopher Larios Cristopher Larios

UNIVERSIDAD NACIONAL DE INGENIERÍA

Área de Conocimiento de Tecnología de la Información y Comunicación

Introducción a la programación

Docente: Ing. Cristopher Larios

Estudiante: Ronald Ramon Oporta Sequeira

Numero de carnet: 2024-1623U

Guía didáctica #5: Guía Clase Practica Herencia y Encapsulamiento

- I. Introducción El objetivo de esta clase práctica es familiarizarse con los conceptos de herencia y encapsulamiento en el lenguaje de programación Java, a través de la creación de una jerarquía de clases que modelen un sistema de gestión de vehículos. Al final de esta práctica, deberías ser capaz de:
- II.
 - Crear una clase base y clases derivadas utilizando herencia.
- III.
 - Utilizar el encapsulamiento para proteger y manipular los datos de las clases.
- IV.
 - Sobrescribir métodos para personalizar el comportamiento en las clases derivadas.
- V.
 - Implementar un programa principal que interactúe con estas clases y demuestre sus funcionalidades.

Independiente Agrega un método realizarMantenimiento en la clase Vehículo que simule un mantenimiento básico del vehículo. Este método será sobrescrito en cada clase derivada para realizar tareas de mantenimiento específicas. En cada clase derivada (Auto, Camion), implementa el método realizarMantenimiento con un mensaje que describa las acciones específicas de mantenimiento para ese tipo de vehículo. En la clase Auto, agrega un método llamado calcularCostoMantenimiento que utilice programación estructurada para calcular el costo de mantenimiento del auto. Detalles del Método:

- El método debe aceptar como parámetros el costo por kilómetro y el número de kilómetros recorridos.
- Debe calcular el costo total del mantenimiento y devolverlo.
- $\text{costoTotal} = \text{costoPorKm} * \text{kilometrosRecorridos}$;

Nuevo archivo...		
Nueva carpeta...		
Mostrar en el Explorador de archivos	Mayús+Alt+R	
Abrir en terminal integrado		
New Java File	>	Class...
New Java Project...		Interface...
Maven	>	Enum...
Buscar en carpeta...	Mayús+Alt+F	Record...
Cortar	Ctrl+X	Annotation...
Copiar	Ctrl+C	Abstract Class...
Pegar	Ctrl+V	
Copiar ruta de acceso	Mayús+Alt+C	
Copiar ruta de acceso relativa	Ctrl+K Ctrl+Mayús+C	
CodeMaker AI	>	
Cambiar nombre...	F2	
Eliminar	Delete	

EXPLORADOR	...	J Vehiculo.java X	J VehiculoPasajeros.java	J VehiculoCarga.java	J Auto.java	J Camion.java	J App.java
> EDITORES ABIERTOS		src > J Vehiculo.java					
▼ HIRENCIA Y CAPSULAMINE...		1 public class Vehiculo {					
> .vscode		2					
> lib		3 private String marca;					
▼ src		4 private String modelo;					
J App.java		5 private int año;					
J Auto.java		6 private int kilometraje;					
J Camion.java		7 //constructor					
J Vehiculo.java		8 public Vehiculo(String marca,String modelo,int año, int kilometraje){					
J VehiculoCarga.java		9 this.marca = marca;					
J VehiculoPasajeros....		10 this.modelo = modelo;					
① README.md		11 this.año = año;					
		12 this.kilometraje = kilometraje;					
		13 }					
		14 //Getters y Setters					
		15 > public String getMarca() { ...					
		17 }					
		18 > public void setMarca(String marca) { ...					
		20 }					
		21					
		22 public String getModelo() {					
		23 return modelo;					
		24 }					
		25 > public void SetModelo(String modelo) { ...					
		27 }					
		28 public int getAño(int año) {					
		29 return año;					
		30 }					
		31 public void setAño(int año) {					
		32 this.año=año;					
		33 }					
		34 > public int getkilometraje() { ...					
		36 }					
		37 > public void setkilometraje(int kilometraje) { ...					
		39 }					
> ESQUEMA		40					
> LÍNEA DE TIEMPO							

EXPLORADOR	...	J Vehiculo.java X	J VehiculoPasajeros.java	J VehiculoCarga.java	J Auto.java	J Camion.java	J App.java
> EDITORES ABIERTOS		src > J Vehiculo.java					
▼ HIRENCIA Y CAPSULAMINE...		1 public class Vehiculo {					
> .vscode		40					
> lib		41					
▼ src		42 //metodo para mostrar la info del vehiculo					
J App.java		43 public void mostrarInfo() {					
J Auto.java		44 System.out.println("Marca:" +marca);					
J Camion.java		45 System.out.println("Modelo:" +modelo);					
J Vehiculo.java		46 System.out.println("Año:" +año);					
J VehiculoCarga.java		47 System.out.println("kilometraje:" +kilometraje);					
J VehiculoPasajeros....		48 }					
① README.md		49 }					
		50 //Mectodo para realizar mantenimiento basico					
		51 public void realizarMantenimiento(double costoPorkm, double kilometrosRecorridos) {					
		52 double costoTotal = costoPorkm * kilometrosRecorridos;					
		53 System.out.println("Costo total del mantenimiento: \$" +costoTotal);					
		54 }					
		55 public static void main(String[] args) {					
		56 Auto elAuto = new Auto();					
		57 Camion elcamion = new Camion();					
		58 }					
		59 }					

EXPLORADOR	...	J Vehiculo.java	J VehiculoPasajeros.java X	J VehiculoCarga.java	J Auto.java	J Camion.java	J App.java
EDITORES ABIERTOS							
src > J VehiculoPasajeros.java							
<pre>1 public class vehiculosPasajeros extends Vehiculo{ 2 3 private int numPasajeros; 4 5 //constructor 6 public vehiculosPasajeros(String marca ,String modelo, int año, int kilometraje, int numPasajeros) { 7 super(marca, modelo, año, kilometraje); 8 this.numPasajeros = numPasajeros; 9 } 10 11 //getter y Setter para numPasajeros 12 public int getNumPasajeros() { 13 return numPasajeros; 14 } 15 16 public void setNumPasajeros(int numPasajeros) { 17 this.numPasajeros = numPasajeros; 18 } 19 //Sobreescribir el metodo mostrar Info() 20 @Override 21 public void mostrarInfo() { 22 super.mostrarInfo(); 23 System.out.println("Numero de Pasajeros: " + numPasajeros); 24 } 25 } 26 27 }</pre>							

EXPLORADOR	...	J Vehiculo.java	J VehiculoPasajeros.java	J VehiculoCarga.java X	J Auto.java	J Camion.java	J App.java
EDITORES ABIERTOS							
src > J VehiculoCarga.java							
<pre>1 public class Vehiculocarga extends Vehiculo { 2 3 private int capacidadcarga; 4 5 //constructor 6 public Vehiculocarga(String marca, String modelo, int año, int kilometraje, int capacidadcarga) { 7 super(marca, modelo, año, kilometraje); 8 this.capacidadcarga = capacidadcarga; 9 } 10 //Getter y Setter para capacidad de carga 11 public int getCapacidadcarga() { 12 return capacidadcarga; 13 } 14 15 public void setCapacidadCarga(int capacidadcarga) { 16 this.capacidadcarga = capacidadcarga; 17 } 18 19 //sobreescribir el metodo MostrarInfo 20 @Override 21 public void mostrarInfo(){ 22 super.mostrarInfo(); 23 System.out.println("capacidad de carga:" +capacidadcarga + " kg"); 24 } 25 26 }</pre>							

EXPLORADOR	...	J Vehiculo.java	J VehiculoPasajeros.java	J VehiculoCarga.java	J Auto.java X	J Camion.java	J App.java
> EDITORES ABIERTOS		src > J Auto.java					
▼ HIRENCIA Y CAPSULAMINE...		1 public class Auto extends vehiculosPasajeros {					
> .vscode		2 private String tipoCombustible;					
> lib		3					
▼ src		4 public Auto(String marca, String modelo, int año, int kilometraje, int numPasajeros, String tipoCombustible) {					
J App.java		5 super(marca, modelo, año, kilometraje, numPasajeros);					
J Auto.java		6 this.tipoCombustible = tipoCombustible;					
J Camion.java		7 }					
J Vehiculo.java		8 //getter and setter					
J VehiculoCarga.java		9 public String getTipoCombustible() {					
J VehiculoPasajeros...		10 return tipoCombustible;					
① README.md		11 }					
		12					
		13 public void setTipoCombustible(String tipoCombustible) {					
		14 this.tipoCombustible = tipoCombustible;					
		15 }					
		16 //sobreescribir el metodo mostrarInfo()					
		17 @Override					
		18 public void mostrarInfo() {					
		19 super.mostrarInfo();					
		20 System.out.println("Tipo de combustible:" + tipoCombustible);					
		21 }					
		22 //Sobreescribo el mostrar mantenimiento					
		23 @Override					
		24 public void realizarMantenimiento(double costoPorkm, double kilometrosRecorridos) {					
		25 super.realizarMantenimiento(costoPorkm, kilometrosRecorridos);					
		26 System.out.println("Verificar la alineacion de las ruedas.");					
		27 System.out.println("Inspeccionar la direccion asistida.");					
		28 System.out.println("Lubricar componentes del sistema de direccion.");					
		29 }					
		30 }					

EXPLORADOR	...	J Vehiculo.java	J VehiculoPasajeros.java	J VehiculoCarga.java	J Auto.java	J Camion.java X	J App.java
> EDITORES ABIERTOS		src > J Camion.java					
▼ HIRENCIA Y CAPSULAMINE...		1 public class Camion extends VehiculoCarga{					
> .vscode		2 private int numEjes;					
> lib		3 public Camion(String marca, String modelo, int año, int kilometraje, int capacidadCarga, int numEjes) {					
▼ src		4 super(marca, modelo, año, kilometraje, capacidadCarga);					
J App.java		5 this.numEjes = numEjes;					
J Auto.java		6 }					
J Camion.java		7 //getter y setter para Numejes					
J Vehiculo.java		8 public int getnumEjes() {					
J VehiculoCarga.java		9 return numEjes;					
J VehiculoPasajeros...		10 }					
① README.md		11					
		12					
		13 public void setnumEjes(int numEjes) {					
		14 this.numEjes = numEjes;					
		15 }					
		16 //sobreescribir el metodo mostrarInfo()					
		17 @Override					
		18 public void mostrarInfo() {					
		19 super.mostrarInfo();					
		20 System.out.println("Numero de Ejes:" + numEjes);					
		21 }					
		22 //sobreescribo el realizar mantenimiento					
		23 @Override					
		24 public void realizarMantenimiento(double costoPorkm, double kilometrosRecorridos) {					
		25 System.out.println("El mantenimiento del frenado en el camion.");					
		26 super.realizarMantenimiento(costoPorkm, kilometrosRecorridos);					
		27 }					
> ESQUEMA		28					
> LÍNEA DE TIEMPO							

EXPLORADOR

> EDITORES ABIERTOS

HRENCI...

> .vscode

> lib

> src

App.java

Auto.java

Camion.java

Vehiculo.java

VehiculoCarga.java

VehiculoPasajeros....

README.md

src > J App.java

```
1 public class App {
2     public static void main(String[] args) throws Exception {
3         System.out.println("Acerca del carro");
4
5         Auto auto= new Auto("Toyota", "Corrola", 2020, 15000, 5, "Super");
6
7         Camion camion = new Camion("Volvo", "FH", 2019, 75000, 20000, 4);
8
9         //Mostrar la informacion de cada Vehiculo
10        System.out.println("Informacion del Auto:");
11        auto.mostrarInfo();
12        System.out.println("Mantenimiento del sistema de direccion en el Auto.");
13        auto.realizarMantenimiento(4, 14);
14
15        System.out.println("informacion del camion:");
16        camion.mostrarInfo();
17        camion.realizarMantenimiento(12, 30);
18    }
19 }
```