

Algoritmos y Estructura de Datos – TI

Guía de Problemas - Semana 12

Herencia y polimorfismo: conceptos básicos.

Problema_12_1

Implemente en el paquete **padre**, la clase **Vehiculo** con los atributos protegidos de tipo cadena: marca, modelo y placa, un constructor que inicialice los atributos, el método **datosDelVehiculo()** que retorne en una cadena los datos del vehículo más un código generado que se obtiene de concatenar marca, placa y modelo. ✓

Implemente en el paquete **hijos**, la clase **Auto** que herede (usando *extends*) la clase padre **Vehiculo** y agregue el atributo privado precio. A través del constructor reciba los cuatro atributos y derive (usando *super*) los tres primeros a la clase padre **Vehiculo**. Implemente el método público **datosDelAuto()** que retorne en una cadena los datos completos del auto incluyendo el importe a pagar a través del método público **aPagar()** y considere que por campaña cuenta con 10% de descuento. ✓

Implemente en el paquete **hijos**, la clase **Camioneta** que herede (usando *extends*) la clase padre **Vehiculo** y agregue el atributo privado precio. A través del constructor reciba los cuatro atributos y derive (usando *super*) los tres primeros a la clase padre **Vehiculo**. Implemente el método público **datosDeLaCamioneta()** que retorne en una cadena los datos completos de la camioneta incluyendo el importe a pagar a través del método público **aPagar()** y considere que por campaña cuenta con 20% de descuento. ✓

A la pulsación del botón Procesar en la clase principal cree un objeto de tipo **Vehiculo**, **Auto** y **Camioneta** respectivamente. Finalmente, visualice a través de tres métodos listado la información completa de cada objeto. ✓

Cambie la sintaxis de los métodos que retornan los datos completos de cada clase por el de **datosCompleto()** y haga uso de *super* en las clases **hijos** para diferenciarlos. ✓

Utilice un solo método listado que reciba como parámetro la referencia de la clase padre **Vehiculo**, haga uso del operador *instanceof* para distinguir el tipo de objeto recibido y visualice la información completa de los objetos. ✓

Problema_12_2

Implemente en el paquete **padre**, la clase **Licencia** con los atributos protegidos de tipo cadena: clase, categoría, fechaEmision (dd/mm/aa) y fechaRenovacion (dd/mm/aa), un constructor que inicialice los atributos, el método **datosDelBrevete()** que retorne en una cadena los datos de la licencia más un código generado que se obtiene de concatenar clase y fechaEmision. ✓

Implemente en el paquete **hijos**, la clase **Particular** que herede (usando *extends*) la clase padre **Licencia** y agregue el atributo privado dni de tipo entero. A través del constructor reciba los cinco atributos y derive (usando *super*) los cuatro primeros a la clase padre **Licencia**. Implemente el método público **datosLicenciaParticular()** que retorne en una cadena los datos completos de la licencia incluyendo un segundo código generado a través del método público **segundoCodigo()** que se obtiene de concatenar categoría y dni. ✓

Implemente en el paquete **hijos**, la clase **Profesional** que herede (usando *extends*) la clase padre **Licencia** y agregue el atributo privado ruc de tipo cadena. A través del constructor reciba los cinco atributos y derive (usando *super*) los cuatro primeros a la clase padre **Licencia**. Implemente el método público **datosLicenciaProfesional()** que retorne en una cadena los datos completos de la licencia incluyendo un segundo código generado a través del método público **segundoCodigo()** que se obtiene de concatenar categoría y ruc. ✓

A la pulsación del botón Procesar en la clase principal cree un objeto de tipo **Licencia**, **Particular** y **Profesional** respectivamente. Finalmente, visualice a través de tres métodos listado la información completa de cada objeto.

Cambie la sintaxis de los métodos que retornan los datos completos de cada clase por el de **datosCompleto()** y haga uso de *super* en las clases **hijos** para diferenciarlos.

Utilice un solo método listado que reciba como parámetro la referencia de la clase padre **Licencia**, haga uso del operador *instanceof* para distinguir el tipo de objeto recibido y visualice la información completa de los objetos.

"Quien no da un paso adelante vive sólo de la esperanza y la oportunidad."

MP