

EJERCICIOS

1. ¿En qué se diferencian el concepto de objeto del modelo orientado a objetos y el concepto de entidad del modelo entidad-relación?

Un objeto en programación orientada a objetos es como una cosa que no solo tiene características (como color, tamaño, etc.), sino que también puede hacer cosas (como moverse, cambiar de color, etc.). Por ejemplo, un coche como objeto no solo tiene atributos como el color o la marca, sino que también puede acelerar o frenar.

En cambio, una entidad en bases de datos es más simple, es como un registro o un grupo de datos que solo describe algo, pero no puede "hacer" nada. Por ejemplo, en una base de datos, un coche sería una entidad que solo tiene datos como el color, la marca y el modelo, pero no puede "hacer" nada como acelerar.

2. Una compañía de alquiler de coches tiene una base de datos de los vehículos de su flota actual. Para todos los vehículos incluye el número de identificación de cada uno, el número de la matrícula, el fabricante, el modelo, la fecha de adquisición y el color. Se incluyen datos específicos para algunos tipos de vehículos:

- **Camiones:** capacidad de carga.
- **Coches deportivos:** potencia, requisitos de edad del conductor.
- **Camionetas:** número de plazas.
- **Vehículos todo terreno:** altura de los bajos, eje motor (tracción a dos o a las cuatro ruedas).

Constrúyase la definición del esquema de una base de datos orientada a objetos para esta base de datos. Utilícese la herencia donde resulte conveniente.

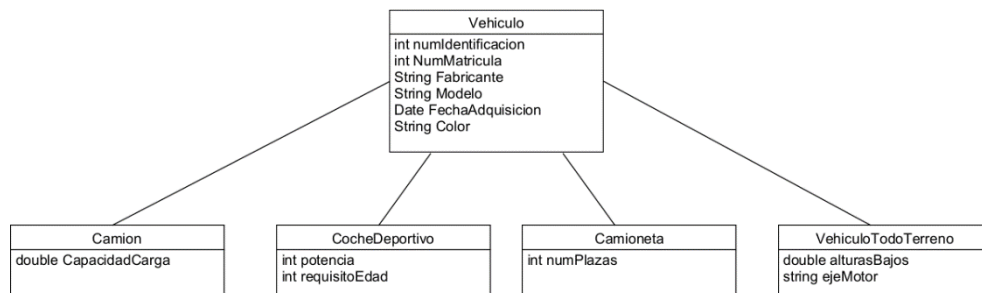
```
// Clase base para todos los vehículos
class Vehiculo {
    int numeroidentificacion;
    int numeroMatricula;
    String fabricante;
    String modelo;
    Date fechaAdquisicion;
    String color;
}

// Clase para camiones, hereda de Vehiculo
class Camion extends Vehiculo {
    double capacidadCarga; // en toneladas
}

// Clase para coches deportivos, hereda de Vehiculo
class CocheDeportivo extends Vehiculo {
    int potencia; // en caballos de fuerza (HP)
    int requisitosEdadConductor; // edad mínima del conductor
}
```

```
// Clase para camionetas, heredade Vehiculo
class Camioneta extends Vehiculo {
    int numeroPlazas; // número de asientos
}

// Clase para vehículos todo terreno, hereda de Vehiculo
class VehiculoTodoTerreno extends Vehiculo {
    double alturaBajos; // en centímetros
    String ejeMotor; // tracción a dos o cuatro ruedas
}
```



3. ¿Qué entiende por la herencia múltiple?

Donde una clase hija hereda atributos o métodos de 2 o mas clases de tal forma que dicha clase pueda acceder a los atributos de todas las clases de la cual hereda

4. Explíquese el motivo de que pueda haber ambigüedad en caso de herencia múltiple. Ilústrese la explicación con un ejemplo.

La ambigüedad puede existir por ejemplo cuando las clases padres tienen un método con el mismo nombre, entonces al momento de la clase hija heredar los métodos de la clase padre, existirá ambigüedad al llamar a dicho método, pues no se sabrá con exactitud a cual de los dos métodos se refiere.

Por ejemplo:

```
class A {
    void saludar() {
        System.out.println("Hola desde A");
    }
}

class B {
    void saludar() {
        System.out.println("Hola desde B");
    }
}

class C extends A, B {
    // Aquí hay ambigüedad: ¿Qué método "saludar" se debería usar?
}
```