

PRACTICA NRO. 2

HERENCIA

1. Sea la siguiente jerarquía de clases:

- Producto(nombre,precio)
 - ProductoAlimenticio(fechaVencimiento, calorías)
 - ProductoElectrodomestico(potenciaW, marca)
 - Refrigerador(capacidadLitros, claseEnergetica) la clase energética puede ser 110 Voltios y 220 Voltios))

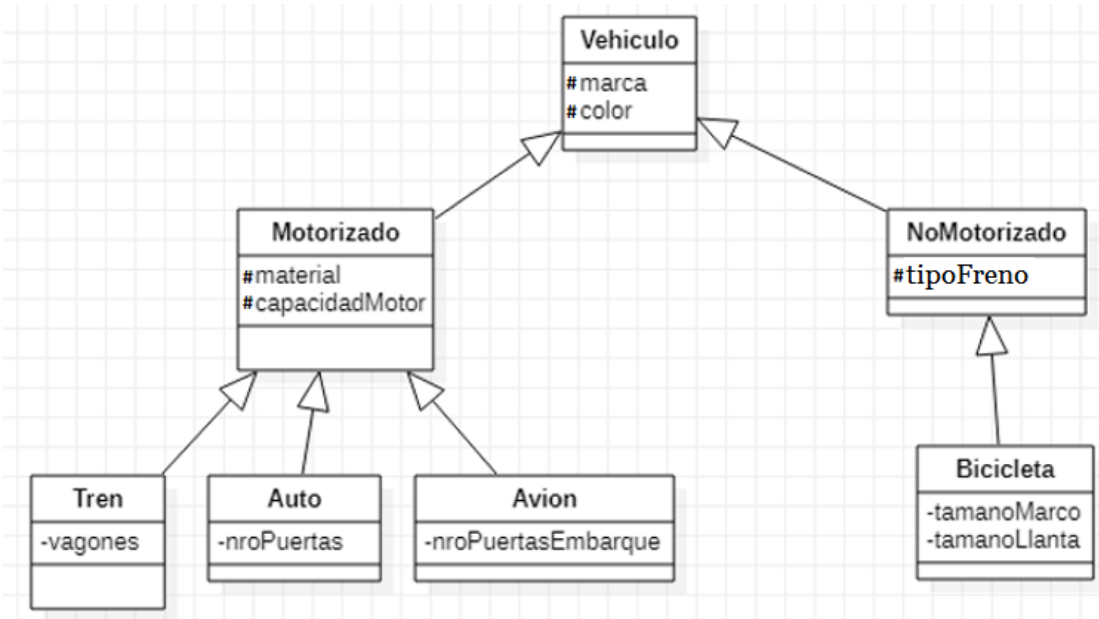
Realizar:

- a) Dada una fecha x calcula la cantidad de días restantes antes de la fecha de vencimiento de un producto alimenticio
 - b) Verifica si un producto alimenticio tiene menos de 100 calorías e emite un mensaje indicando que es saludable
 - c) Compara la capacidad de dos refrigeradores e indica cual es el mayor
2. Se desea implementar un Sistema de Gestión de Animales en el Refugio "Chuletita". Para ello se define la clase base: Animal con atributos nombre, edad, especie y métodos como: getEspecie() , hacerSonido() y las Subclases:
- Mamifero(atributos: tipoPelaje, peso)
 - Perro(atributos: raza), Método: hacerSonido()→ devuelve "Ladrar".
 - Murcielago (atributos: alcanceVueloKm, activoNoche). Método comparaAlcanceVuelo(Murcielago otro)
 - Ave(atributos: tipoDeVuelo). Método: hacerSonido()→ devuelve "Cantar".

Representa este ejercicio en un Diagrama y su solución en pseudocódigo y su implementación en Java y desarrolle las siguientes operaciones:

- a. Crear al menos 3 objetos de cada tipo de animal y almacenarlos en una lista.
- b. Implementar una nueva clase derivada Reptil con atributos adicionales como tipoEscamas y un método hacerSonido() que devuelva "Sisear".
- c. Mostrar todos los animales (nombre, especie y un sonido específico de cada tipo).
- d. Implementar un método que agrupe los animales según su especie y muestre cuántos hay de cada tipo.

3. Dado el diagrama UML, realizar los siguientes ejercicios:



Se pide:

- Implementar el diagrama de clases, con constructores por defecto, getters y setters.
- Instanciar 2 objetos de cada hijo de Motorizado y NoMotorizado.
- Compara si un auto y un tren tienen el mismo color, manejan el mismo combustible si fuera el caso, mostrar los datos de ambos objetos.
- Dadas dos bicicletas verificar si ambas tienen el mismo tipo de frenos.

AGREGACION Y COMPOSICION

- Realice 3 ejemplos de agregación, 3 de composición, 2 combinados y realice su representación en diagramas UML
- Representa en UML y soluciona en pseudocódigo e implementa en java, la siguiente jerarquía de clases:
 - Empresa: nombre, nroDeptos, Departamento depto[50]
 - Departamento: área, presupuesto, nroEmpleados, Empleados em[50]
 - Empleado: nombre, salario

Métodos:

- calcularSalarioTotalDepartamento()
- verificarEmpleado(String nombre)
- compararSalario(Empleado otro)

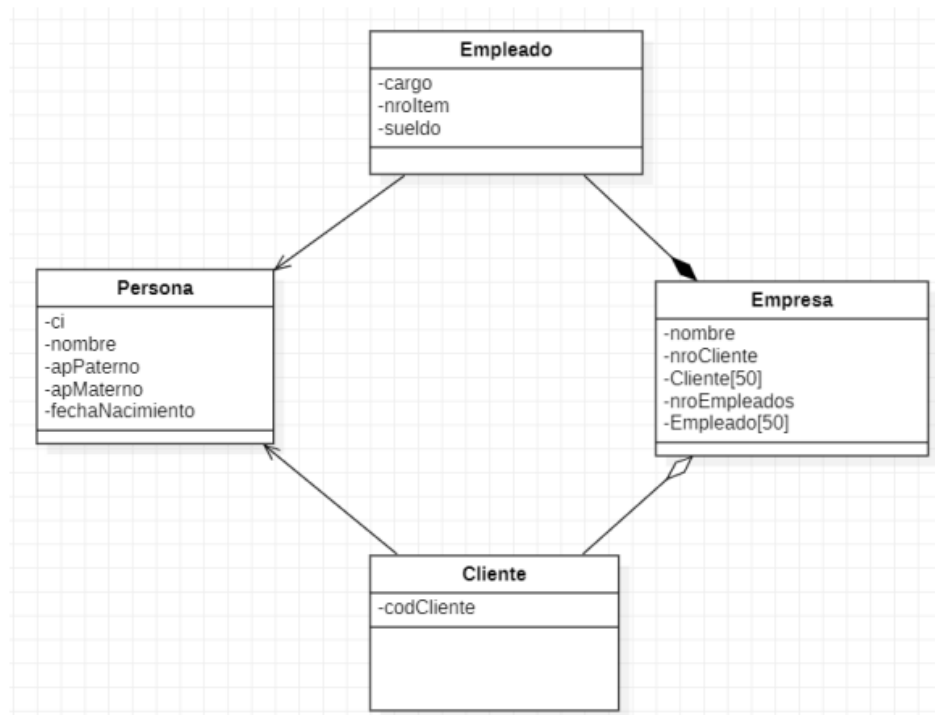
3. Representa en UML y soluciona en pseudocódigo e implementa en java, la siguiente jerarquía de clases:

- Computadora: marca, nroCpu, Cpu cpu[10]
- Cpu: modelo, nroNucleo, Nucleos nuc[10]
- Nucleo: velocidad (GHz)

Métodos:

- calcularPotenciaTotal()
- esGamer()
- compararVelocidad(Nucleo otro)

4. Dado el siguiente diagrama de clases:



Implementar todas las clases y los métodos principales

- Eliminar al(los) empleados(s) que son es (son) cliente y empleado en la empresa al mismo tiempo. Mostrar la lista actualizada de empleados
- Mostrar el cargo del empleado con CI X.

GENERICIDAD

1. Representa en UML y soluciona en pseudocodigo e implementa en java la clase genérica **Par<T1, T2>**

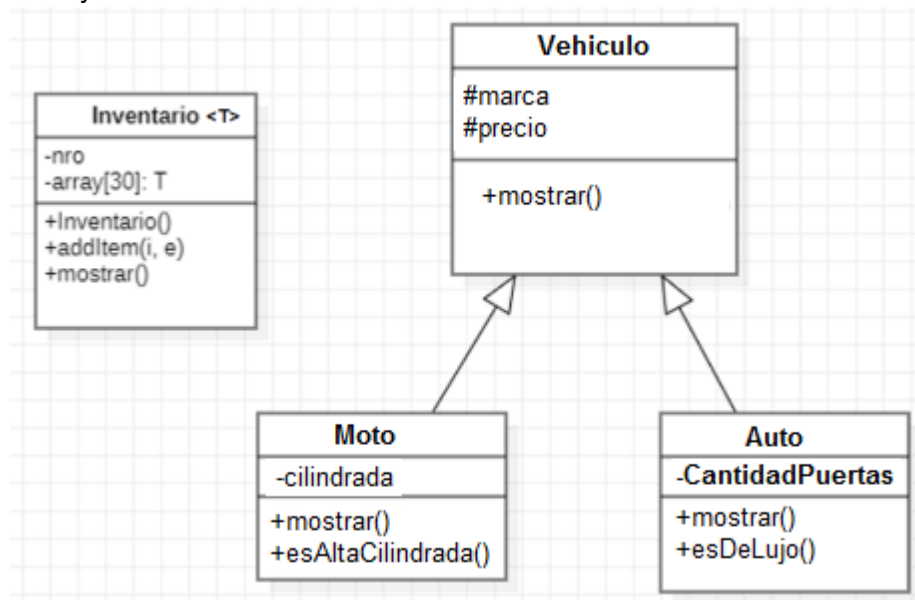
- Atributos: primero, segundo
- Métodos:
 - mostrar()
 - sonIguales()
 - tienenTiposIguales()

2. Representa en UML y soluciona en pseudocodigo e implementa en java un método genérico para hallar el máximo de 3 elementos. Por ejemplo si se comparan las frutas: manzana, pera, limón. El máximo será manzana porque tiene mayor número de caracteres.

3. Representa en UML y soluciona en pseudocodigo e implementa en java la clase genérica que permita comparar dos objetos implementando la interfaz Comparable.

- a. Determinar cuál objeto es mayor.
- b. Determinar si ambos objetos son iguales.
- c. Ordenar un array de objetos de tipo genérico.

4. Soluciona en pseudocodigo e implementa en java Implementar la clase genérica Inventario que almacena objetos Vehiculo como Moto y Auto.



Se pide:

- a) Crear una instancia de Inventario para almacenar vehiculos.
- b) Agregar al inventario dos autos y una moto.
- d) Contar cuántos vehículos superan un precio de Bs. 20,000.
- e) Mostrar solamente las motos.

5. Dada la siguiente jerarquía de clases, representa en UML y soluciona en pseudocódigo e implementa en java:

Superclase: Publicacion (herencia simple)

- Atributos:
 - titulo (String)
 - anioPublicacion (int)
- Método: mostrarInfo()

Subclases:

1. Libro → hereda de Publicacion
 - Atributo: isbn (String)
 - Atributo: editorial (asociación con Editorial)
2. Artículo → hereda de Publicacion
 - Atributo: revista (String)
 - Atributo: volumen (int)
3. Tesis → hereda de Publicacion
 - Atributo: grado (String)
 - Composición con Asesor (si la tesis se elimina, el asesor también)

Clase: Autor

- Atributos: nombre, correo
- Método: mostrarAutor()

Clase: Editorial

- Atributos: nombre, pais

Clase genérica: Repositorio<T>

- Uso de ArrayList<T>
- Métodos:
 - agregar(T item)
 - eliminar(T item)
 - mostrarTodo()

Este repositorio será utilizado para almacenar publicaciones o autores de forma genérica.

Realizar:

- Crear al menos 1 objeto de cada tipo (Libro, Artículo, Tesis).
- Asociar autores a cada publicación (**agregación**).
- Asociar una editorial al libro (**asociación**).

- Componer un asesor en la tesis (**composición**).
- Guardar las publicaciones en un repositorio genérico (Repositorio<Publicacion>).
- Mostrar toda la información usando mostrarInfo() en consola.