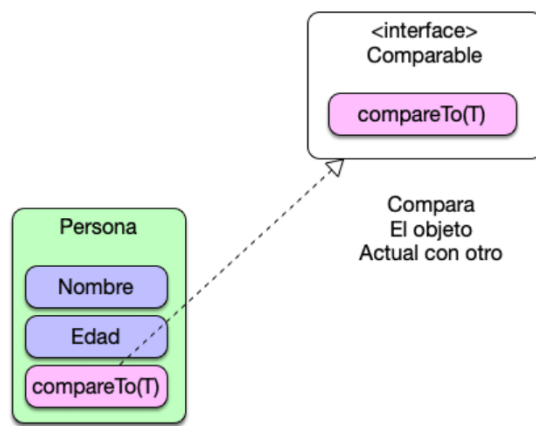


## Guía Nro. 3 - Laboratorio Programación Ingeniería en Computación

### Interface Comparable y método compareTo (API Java) Comparar objetos

Ejercicio acerca de cómo una clase debe hacer uso de la interfaz Comparable. Esto nos va a permitir que en dicha clase se pueda realizar la comparación de objetos, permitiendo hacer ordenaciones de los mismos.

```
public class Persona implements Comparable<Persona>{  
    public int edad;  
    public int rut;  
    public Persona( int r, int e){  
        this.rut = r;  
        this.edad = e;  
    }  
}
```



#### INSTRUCCIONES:

- Implementar Método de Interface Comparable **public int compareTo(Persona o)**
- En nuestro ejemplo nos vamos a inventar esta ordenación y vamos a decir que una persona es más grande que otra si tiene la edad mayor, en cambio será menor si tiene una edad menor y serán iguales si y solo si su edad es igual y el rut también es igual. En este último caso si dos personas tienen la misma edad, se considerará mayor el que tenga el rut más grande. Fíjate que el criterio de ordenación lo hemos decidido nosotros. Una vez decidido, lo implementaremos en forma de código.
- Utilizar librería Scanner para el ingreso de las variables por teclado, para comparar dos personas, necesariamente se deben ingresar los datos de ambas.
- Utiliza clase Tester que contenga en **main** principal para testear el programa.

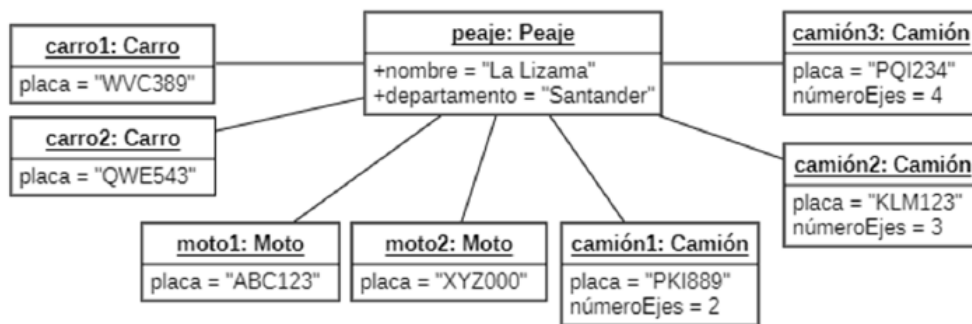
### Enunciado: clase Peaje y jerarquía de herencia de vehículos

Se requiere desarrollar un programa que modele un sistema de peaje. Las estaciones de peaje tienen un nombre, departamento en que están ubicadas y un valor numérico que representa el valor total de peaje recolectado.

Los vehículos que llegan a un peaje tienen una placa (tipo *String*). El peaje cobra diferentes valores de peaje según el tipo de vehículo. Si es un carro, el valor del peaje es de \$10000. Si es una moto, \$5000. Si es un camión, el valor del peaje depende del número de ejes, se cobra \$5000 por cada eje.

Se requiere que la estación de peaje calcule el valor del peaje de cada vehículo que llegue y el total de peajes recolectados. Así, al finalizar, el sistema debe *imprimir* en pantalla un listado con los vehículos que llegaron al peaje y el total acumulado.

Diagrama de objetos



- Se requiere codificar el programa en Java
- Se requiere modelar el diagrama de Clases UML
- La clase TestPeaje debe ser independiente
- El programa debe tener interacción con el usuario.