

Segundo Parcial: Terminal de Trenes

Una Terminal de Trenes desea guardar y gestionar los viajes que llegan y parten de las diferentes empresas. Para ello la terminal guarda la colección de empresas de trenes, las cuales administran los diferentes viajes que se ofrecen, a diferentes destinos. Cada viaje tiene asignada una fecha, una hora de llegada, una hora de partida y el conductor responsable del viaje. Para ello implementar las clases: **Terminal, Empresa, Viaje y Responsable**.

Los viajes pueden ser Nacionales o Internacionales. Si el viaje es internacional se debe almacenar si requiere o no documentación adicional y el porcentaje correspondiente a impuestos que deben ser aplicados al costo del viaje (por defecto el valor aplicado es del 45%). Si el viaje es Nacional se almacena porcentaje de descuento que puede ser aplicado al monto del viaje (por defecto el descuento aplicado es del 10%).

En la clase **Responsable**:

1. Se registra la siguiente información: nombre, apellido, Nro de Documento, dirección, mail y teléfono.
2. Método constructor que recibe como parámetros los valores iniciales para los atributos definidos en la clase.
3. Los métodos de acceso de cada uno de los atributos de la clase.
4. Redefinir el método **toString** para que retorne la información de los atributos de la clase.

En la clase **Viaje**:

1. Se registra la siguiente información: destino, hora de partida, hora de llegada, número, importe, fecha, cantidad de asientos totales, cantidad de asientos disponibles, y una referencia a la persona responsable del viaje.
2. Método constructor que recibe como parámetros los valores iniciales para los atributos definidos en la clase.
3. Los métodos de acceso de cada uno de los atributos de la clase.
4. Redefinir el método **toString** para que retorne la información de los atributos de la clase.
5. Implementar la jerarquía de herencia que corresponda para implementar los viajes Nacionales e Internacionales.
6. Implementar el método **calcularImporteViaje()** que se calcula en base al monto base del viaje, la cantidad de asientos disponibles y la cantidad total de asientos. El cálculo que se realiza es el siguiente:
$$\text{importe} = \text{monto base} + (\text{monto base} * \text{asientosVendidos} / \text{asientos totales})$$
7. Redefinir el método que permite calcular el importe de un viaje según corresponda.

En la clase **Empresa**:

1. Se registra la siguiente información: identificación, nombre y la colección de Viajes que realiza.
2. Método constructor que recibe como parámetros los valores iniciales para los atributos.
3. Los métodos de acceso de cada uno de los atributos de la clase.
4. Redefinir el método **toString** para que retorne la información de los atributos de la clase.
5. Implementar el método **buscarViaje(codViaje)** que dado un código de viaje que se recibe por parámetro, retorna el objeto viaje correspondiente a ese código.
6. Implementar el método **darCostoViaje(codViaje)** que dado un código de viaje retorna el importe correspondiente a ese viaje.

En la clase **Terminal**:

1. Se registra la siguiente información: denominación, dirección y la colección empresas registradas en la terminal.
2. Método constructor que recibe como parámetros los valores iniciales para los atributos de la clase.
3. Los métodos de acceso para cada una de las variables instancias de la clase.
4. Redefinir el método **toString** para que retorne la información de los atributos de la clase.



5. Implementar el método **darViajeMenorValor()** recorre cada una de las empresas vinculadas a la terminal y retorna una colección de objetos viaje. Cada viaje es el de menor valor dentro de la colección de viajes de esa empresa.

Implementar un script **TestTerminal** en el cual:

1. Se crea una colección con un mínimo de 2 empresas, ejemplo Metrovías y Tren Patagónico.
2. A cada empresa se le incorporan 2 instancias de la clase viaje Nacionales y 3 instancias de la clase Viaje Internacionales.
3. Se crea un objeto Terminal con la colección de empresas creadas en el punto 1.
4. Invocar y visualizar el resultado obtenido de invocar al método darViajeMenorValor() a partir de la instancia Aeropuerto creada en el punto 3.

Invoque a los métodos implementados cuando crea necesario.