Primer Parcial: Terminal de Trenes

Una Terminal de Trenes desea guardar y gestionar los viajes que llegan y parten de las diferentes empresas de la región. 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*** .

En la clase ***Responsable:***

1. Se registra la siguiente información: nombre, apellido, Nro de Documento, direccion, mail y telefono.
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 el método ***asignarLugaresDisponibles($cantAsientos)*** que recibe por parámetros la cantidad de asientos que desean asignarse. El método retorna verdadero en caso que la asignación pueda concretarse y falso en caso contrario.

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 ***darViajeADestino($elDestino)*** que recibe por parámetro un destino junto a una cantidad de asientos y retorna una colección con todos los viajes disponibles a ese destino.
6. Implementar el método ***incorporarViaje*** que recibe como parámetro un viaje, verifica que no se encuentre registrado ningún otro viaje al mismo destino, en la misma fecha y con el mismo horario de partida. El método retorna verdadero si la incorporación del viaje se realizó correctamente y falso en caso contrario.
7. Implementar el método ***venderViajeADestino($cantAsientos, $destino)*** método que recibe por parámetro la cantidad de asientos y el destino y se registra la asignación del viaje en caso de ser posible. (invocar al método ***asignarAsientosDisponibles***). El método retorna la instancia del viaje asignado o null en caso contrario.
8. Implementar el método ***montoRecaudado*** que retorna el monto recaudado por la Empresa. (tener en cuenta los asientos vendidos y el importe del 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 ***ventaAutomatica($cantAsientos, $fecha, $destino, $empresa)*** que recibe por parámetro la cantidad de asientos que se requieren, una fecha, un destino y la empresa con la que se desea viajar. Automáticamente se registra la venta del viaje. *(Para la implementación de este método debe utilizarse uno de los métodos implementados en la clase Viaje).*
6. Implementar el método **empresaMayorRecaudacion** retorna un objeto de la clase empresa que se corresponde con la de mayor recaudación.
7. Implementar el método ***responsableViaje($numeroViaje)*** que recibe por parámetro un número de viaje y retorna el responsable del viajle.

Implementar un script **TestTermial** 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.
3. Se crea un objeto Terminal con la colección de empresas creadas en el pnto1.
4. Invocar y visualizar el resultado del método ***ventaAutomatica*** con cantidad de asientos 3 y como destino alguno de los destinos de viaje creados en 2.
5. Invocar y visualizar el resultado del método **empresaMayorRecaudacion.**
6. Invocar y visualizar el resultado del método ***responsableViaje*** *correspondiente a uno de los números de viajes del punto 2.*