

Simulacro

Una administradora ferroviaria necesita una aplicación que le ayude a administrar las formaciones que tiene disponibles. Una formación es lo que habitualmente llamamos “un tren”, tiene una locomotora y uno o varios vagones. Los vagones pueden ser de carga o de pasajeros.

Implementar la clase **Vagon** y sus especializaciones, la clase **Locomotora** y la clase **Formacion**.

En la clase **Vagon**:

1. Se registra la siguiente información: año de instalación, largo, ancho, peso del vagón vacío. Si se trata de un vagón de pasajeros también se almacena la cantidad máxima de pasajeros que puede transportar, la cantidad de pasajeros que está transportando y el peso promedio de los pasajeros. Es importante tener en cuenta que la variable de instancia que representa el peso del vagón se calcula de acuerdo a la cantidad de pasajeros que se está transportando y considerando un peso promedio por pasajero de 50 Kilos.. Si se trata de un vagón de carga se almacena el peso máximo que puede transportar y el peso de su carga transportada. El peso del vagón va a depender del peso de su carga más un índice que coincide con un 20 % del peso de la carga, dicho índice se guarda en cada vagón de carga.

2. Implementar el método **incorporarPasajeroVagon** que recibe por parámetro la cantidad de pasajeros que ingresan al vagón y tiene la responsabilidad de actualizar las variables instancias que representan el peso y la cantidad actual de pasajeros. El método debe devolver verdadero o falso según si se pudo o no agregar los pasajeros al vagón.
3. Implementar el método **incorporarCargaVagon** que recibe por parámetro la cantidad de carga que va a transportar el vagón y tiene la responsabilidad de actualizar las variables instancias que representan el peso y la carga actual. El método debe devolver verdadero o falso según si se pudo o no agregar la carga al vagón.
4. Implementar el método **calcularPesoVagon** y redefinirlo según sea necesario. No olvidar agregar el peso que tiene el vagón vacío.

En la clase **Locomotora**:

1. Se registra la información de su peso y velocidad máxima.

En la clase **Formacion**:

1. Se almacena la referencia a un objeto de la clase Locomotora, la colección de objetos de la clase Vagón y el máximo de vagones que puede contener. **Se tiene una única colección de vagones**
2. Implementar el método **incorporarPasajeroFormacion** que recibe la cantidad de pasajeros que se desea incorporar a la formación y busca dentro de la colección de vagones aquel vagón capaz de incorporar esa cantidad de pasajeros. Si no hay ningún vagón en la formación que pueda ingresar la cantidad de pasajeros, el método debe retornar un valor falso y verdadero en caso contrario. Puede utilizar la función `is_a` para saber si se trata de un vagón de carga o de pasajeros.
3. Implementar el método **incorporarVagonFormacion** que recibe por parámetro un objeto vagón y lo incorpora a la formación. El método retorna verdadero si la incorporación se realizó con éxito y falso en caso contrario.
4. Implementar el método **promedioPasajeroFormacion** el cual recorre la colección de vagones y retorna un valor que represente el promedio de pasajeros por vagón que se encuentran en la formación. Puede utilizar la función `is_a` para saber si se trata de un vagón de carga o de pasajeros.
5. Implementar el método **pesoFormacion** el cual retorna el peso de la formación completa.
6. Implementar el método **retornarVagonSinCompletar** el cual retorna el primer vagón de la formación que aún no se encuentra completo

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

1. Se crea un objeto locomotora con un peso de 188 toneladas y una velocidad de 140 km/h.
2. Se crea 3 objetos con la información visualizada en la tabla:

Vagón	pesoVacio	PesoCarga	cantidad_maxima_pasajeros	cantidad_actual_pasajeros
vagon1	15000	–	30	25
vagonCarga	15000	55000 (peso Carga)	--	--
vagon	15000	–	50	50



FACULTAD DE INFORMÁTICA
CÁTEDRA INTRODUCCIÓN POO



3. Se crea un objeto formación que tiene la locomotora y los vagones creados en (1) y (2) usando el método **incorporarVagonFormacion**.
4. Invocar al método **incorporarPasajeroFormacion** con el parámetros cantidad de pasajero = 6 y visualizar el resultado.
5. Realizar un print de los 3 objetos vagones creados.
6. Invocar al método **incorporarPasajeroFormacion** con el parámetros cantidad de pasajero = 14 y visualizar el resultado.
7. Realizar un print del objeto locomotora
8. Invocar al método **promedioPasajeroFormacion** y visualizar el resultado obtenido.
9. Invocar al método **pesoFormacion** y visualizar el resultado obtenido.
10. Realizar un print de los 3 objetos vagones creados.