Deusto
Facultad de Ingeniería
Ingeniaritza Fakultatea

EXAMEN DE PROGRAMACIÓN II

25-MARZO-2021

Nombre: _____

DNI: _____ *Aula:* _____ *Fila:* _____ *Columna:* _____

1. Diseño

Debido a la liberalización del sector, desde Deusto hemos decidido crear DeustoTrain, una empresa ferroviaria que gestione el transporte de mercancías y pasajeros. De cada tren nos interesa saber la potencia en cv (int), el modelo (String) y un código único generado automáticamente a partir de un contador común. DeustoTrain gestionará diferentes rutas para las que almacenaremos el tren encargado de hacer la ruta y una lista de los trayectos que componen la ruta. Además, de las rutas de mercancías queremos guardar la carga máxima en toneladas (double) y de las rutas de pasajeros el número de vagones de primera (int), segunda (int) y coche-cama (int), además de la lista de pasajeros. De cada trayecto nos interesa saber la estación de origen, la estación de destino y el número de km que se recoren. De cada estación almacenamos su nombre, latitud (de -90.0 a 90.0) y longitud (de -180.0 a 180.0) y si disponen de cafetería (boolean). De los pasajeros nos interesa almacenar su nombre, dni y teléfono.

Queremos saber cuánto beneficio podría darnos cada ruta. Para ello, necesitamos calcular el total de km de cada ruta y multiplicarlo por el beneficio por km de cada tipo de ruta. Todas las rutas tendrán un método getBeneficioPorKm() que se calculará de manera diferente por cada tipo de ruta: 1) las rutas de mercancías lo calcularán dividiendo su carga en toneladas por la potencia en cv de su tren; 2) las rutas de pasajeros lo calcularán multiplicando por 50 el número total de vagones.

Las rutas de pasajeros serán reservables, lo que implica implementar dos métodos: 1) `reservar`(), que recibe un Pasajero y devuelve un boolean indicando si se ha podido añadir a la lista de pasajeros de la ruta y 2) `anular`() que recibe un Pasajero devuelve un boolean indicando si se ha podido eliminar de la lista de pasajeros de la ruta. Las estaciones también serán reservables porque un pasajero puede reservar una mesa en su cafetería (es necesario que guardemos una lista de pasajeros en cada estación y gestionemos las reservas y anulaciones con los mismos métodos, teniendo en cuenta que si una estación no tiene cafetería no podremos reservar ni anular).

Además, en la clase Ruta necesitamos que haya un método `getRutaMasBeneficiosa`() que reciba un ArrayList de Rutas y devuelva la Ruta que dé mayor beneficio del ArrayList recibido.

Dibuja un diagrama de clases para contemplar este caso.

Deusto
Facultad de Ingeniería
Ingeniaritza Fakultatea

EXAMEN DE
PROGRAMACIÓN II

25-MARZO-2021

2. Implementación

1. Programa la jerarquía de clases diseñada en el apartado 1 en Java. Cada clase deberá tener **constructores** con argumentos y sin argumentos, **getters y setters** y **toString**.

2. Programa una clase DeustoTrain con un método main en el que crees al menos **2 rutas de mercancías y 2 rutas de pasajeros** que tengan al menos **3 trayectos cada una** entre diferentes estaciones. Para el caso de las rutas de pasajeros, crea al menos **2 pasajeros** en cada una.

3. Muestra desde el main de DeustoTrain cuál es la ruta más beneficiosa usando el método **`getRutaMasBeneficiosa`**() de la clase Ruta.

Extra: crea en el main de DeustoTrain una VentanaGrafica de 800x600 píxeles de resolución y muestra cada ruta dibujando una flecha por cada trayecto, desde el origen al destino (traduce la latitud a la altura disponible, donde -90.0 será 0 y 90.0 será la altura de la ventana, y lo mismo para la longitud: -180.0 será 0 y 180.0 será la anchura de la ventana). Las rutas de mercancías serán rojas (Color.RED) y las de pasajeros serán verdes (Color.GREEN). Dibuja un círculo naranja (Color.ORANGE) para representar cada Estación que tenga cafetería.

Deusto
Facultad de Ingeniería
Ingeniaritza Fakultatea

EXAMEN DE
PROGRAMACIÓN II

25-MARZO-2021