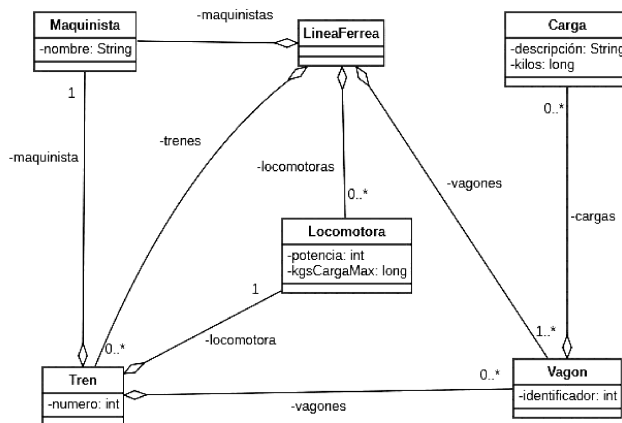


Se pretende realizar un programa en Java para gestionar los trenes y el transporte de mercancías en una línea férrea. Para ello, se han diseñado las siguientes clases, en las que se muestran sólo sus atributos:

<pre>public class LineaFerreia { Tren[] trenes; Locomotora[] locomotoras; Vagon[] vagones; Maquinista[] maquinistas; }</pre>	<pre>public class Tren { private int numero; private Locomotora locomotora; private Maquinista maquinista; private Vagon vagones[]; }</pre>	<pre>public class Locomotora { private int potencia; private long kgsCargaMax; }</pre>
<pre>public class Vagon { int identificador; Carga cargas[]; }</pre>	<pre>public class Carga { private String descripcion; private long kilos; }</pre>	<pre>public class Maquinista { private String nombre; }</pre>

Por lo que el diagrama de clases UML es el siguiente:



Se pide realizar los siguientes métodos, todos en la clase **LineaFerreia.java**, que es la única que el alumno debe modificar y entregar (las otras se van a usar tal como están):

- (2 Puntos) Realizar el método **public long totalCargaDeProducto(int numTren, String descripcion)** que devuelve un **long** con el número de kilos del producto cuyo nombre se pasa mediante el parámetro **descripcion**, que lleva el tren cuyo número es el pasado por el parámetro **numTren**.
- (1 Puntos) Realizar el método **public Tren[] trenesConducidosPor(String nombreMaquinista)** que devuelve una tabla de objetos de la clase **Tren** con todos los trenes conducidos por el maquinista cuyo nombre se pasa en el parámetro **nombreMaquinista**. El tamaño de la tabla debe ser el del número de objetos no nulos que contiene.
- (2 Puntos) Realizar el método **public int numMaquinistasConducenAlMenos(int numTrenes)** que devuelve un número entero con el número de maquinistas que conducen al menos el número de trenes pasado por el parámetro **numTrenes**.
- (1,5 Puntos) Realizar el método **public boolean hayProblemas()** que devuelve un booleano indicando si algún tren lleva más kilos de carga de los que puede arrastrar su locomotora según el atributo **kgsCargaMax** de la locomotora.
- (2 Puntos) Realizar el método **public void eliminaVagonesConCargaDelTren(String descripcionCarga, int numTren)** que del tren cuyo número sea el pasado por el parámetro **numTren**, elimina todos los vagones que tengan alguna carga cuya descripción sea la pasada por el parámetro **descripcionCarga**. Es indiferente que en los vagones a eliminar haya otras cargas distintas o no.

- f) (1,5 Puntos) Realizar el método `public Locomotora[] locomotorasQuePuedenLlevar (Vagon[] vagones)` que devuelve una tabla (con el tamaño ajustado a los elementos que contiene) de objetos tipo `Locomotora` que son capaces de arrastrar el conjunto de vagones pasado en la tabla `vagones` (según el atributo `kgsCargaMax`).

Se deben realizar todas las funciones auxiliares (si son necesarias) que se usen en la clase `LineaFerreia.java`. Los métodos no deben pedir datos al usuario, ni mostrar ningún resultado. Los resultados se muestran con las instrucciones que ya hay en `Principal.java`, y que hacen uso de los métodos que el alumno debe hacer en `LineaFerreia.java`.

Valoraciones:

- Las clases (los ficheros fuente .java a entregar) deben compilar sin errores.
- No se deben producir excepciones.
- Sólo se puede presuponer alguna condición si no contradice el enunciado.
- Se valorará el código correcto, indentado y comentado; la reutilización de código, la descomposición en funciones en los casos adecuados, la eficiencia y claridad de los algoritmos y la inexistencia de código o variables superfluas.

Ejemplo de salida de datos con los datos de trenes que se proporcionan en `Principal.java`, que son los siguientes:

```
Tren num: 443  Maquinista: Mari
Locomotora: Potencia: 30000 HP  Carga Máxima: 450000 Kgs.
Vagones:
  Num vagon: 0  Cargas: (Albaricoque: 1623K, Calabaza: 7575K, Berenjena: 1978K, Limon: 5547K, Cebolla: 8106K)

Tren num: 552  Maquinista: Juan
Locomotora: Potencia: 45000 HP  Carga Máxima: 230000 Kgs.
Vagones:
  Num vagon: 1  Cargas: (Oliva: 3396K, Albaricoque: 5373K, Patata: 8408K, Limon: 5508K)
  Num vagon: 2  Cargas: (Manzana: 7723K, Berenjena: 1585K, Lechuga: 7668K)
  Num vagon: 3  Cargas: (Calabaza: 9383K)

Tren num: 126  Maquinista: Pepe
Locomotora: Potencia: 20000 HP  Carga Máxima: 110000 Kgs.
Vagones:
  Num vagon: 4  Cargas: (Coliflor: 7593K, Lechuga: 5195K)
  Num vagon: 5  Cargas: (Cebada: 5713K, Albaricoque: 6825K, Chalota: 7388K, Patata: 9963K, Calabaza: 8225K,
                      Cebolla: 4318K)

Tren num: 975  Maquinista: Lola
Locomotora: Potencia: 20000 HP  Carga Máxima: 110000 Kgs.
Vagones:
  Num vagon: 10 Cargas: (Berenjena: 2863K, Coliflor: 707K, Albaricoque: 6828K, Manzana: 443K, Oliva: 8355K)

Tren num: 486  Maquinista: Jose
Locomotora: Potencia: 30000 HP  Carga Máxima: 450000 Kgs.
Vagones:
  Num vagon: 11 Cargas: (Garbanzo: 866K, Chalota: 155K, Limon: 570K, Cafe: 8498K)
  Num vagon: 12 Cargas: (Garbanzo: 20943K, Cebolla: 5211K, Patata: 8258K)
  Num vagon: 13 Cargas: (Berenjena: 3314K, Esparrago: 780K)

Tren num: 266  Maquinista: Mari
Locomotora: Potencia: 45000 HP  Carga Máxima: 230000 Kgs.
Vagones:
  Num vagon: 14 Cargas: (Lechuga: 292K, Cebolla: 4025K, Garbanzo: 409K, Manzana: 2735K, Albaricoque: 841K)
  Num vagon: 15 Cargas: (Calabaza: 3663K, Coliflor: 54049K, Berenjena: 564K)
```

La salida debería ser:

```
Apartado a) Carga de garbanzos en tren 486: 21809
Apartado b) Número de los trenes conducidos por Mari: 443 266
Apartado c) Maquinistas que conducen al menos 2 trenes: 1
Apartado d) Problemas antes de cargar plomo: false
Apartado d) Problemas después de cargar plomo: true
Apartado e) Tren 486 después de eliminar vagones con garbanzo:
Tren num: 486  Maquinista: Jose
  Locomotora: Potencia: 30000 HP  Carga Máxima: 450000 Kgs.
  Vagones:
    Num vagon: 13 Cargas: (Berenjena: 3314K, Esparrago: 780K)
Apartado f) Locomotoras que pueden llevar conjunto de vagones definido:
  Potencia: 30000 HP  Carga Máxima: 450000 Kgs.
  Potencia: 45000 HP  Carga Máxima: 230000 Kgs.
```