

Estructures de dades

Curs 2020-21

Práctica 3 - Trenes de carga

5 de maig de 2021

Esta práctica consiste en implementar una estructura de datos que nos ayude a resolver el problema presentado en este enunciado: gestión de trenes de carga.

Para ello, se os proporcionarán las implementaciones de los TAD vistos en teoría para que vosotros podáis utilizar los que os hagan falta en la implementación de vuestra estructura.

La realización de esta práctica puede ser **individual o en parejas**, en la carpeta comprimida debéis añadir un documento *participants.txt* indicando el nombre y DNIs de las personas que han realizado la práctica. Es suficiente que realice la entrega de la práctica uno de los componentes del grupo.

1 Enunciado del problema

Se tiene que diseñar una estructura de datos para gestionar trenes de carga de mercancía.

Un tren se compone de una locomotora y una serie de vagones preparados para el transporte de la carga. Tanto las locomotoras como los vagones son individuales y se montan los trenes según las necesidades. Se necesita gestionar un conjunto, n , muy grande de trenes.

Montar un tren implica seleccionar una locomotora de entre las locomotoras libres (no asignadas a ningún tren) y un conjunto x de vagones (también de entre los vagones libres).

Las locomotoras se encuentran identificadas mediante un código alfanumérico de 8 caracteres que empieza siempre por L. Las locomotoras libres se encuentran aparcadas en una vía de la estación de sentido único: de modo

que las locomotoras se aparcen al final de la fila de locomotoras aparcadas y se desaparcen por el principio de la vía (ver Figura 1).

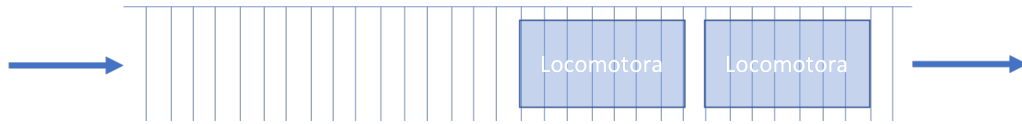


Figura 1: Aparcamiento de locomotoras.

Las vagones se encuentran identificados mediante un código alfanumérico de 8 caracteres que empieza siempre por V y, además, debe almacenarse el peso máximo de carga que puede transportar cada uno. Los vagones libres se encuentran aparcados en una vía de la estación de doble sentido que finaliza en una de las paredes de la estación: de modo que los vagones se aparcen en sentido hacia la pared y se desaparcen en sentido contrario (ver Figura 2).

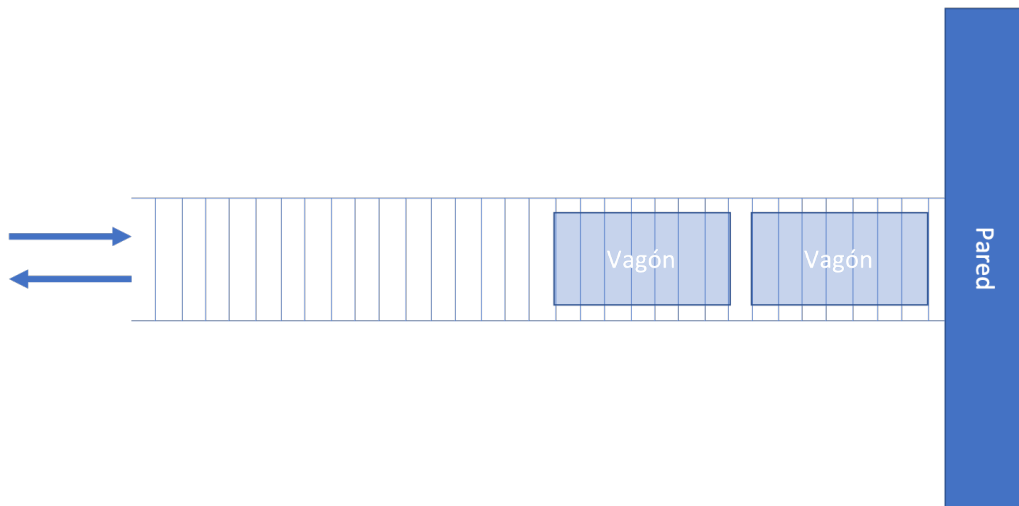


Figura 2: Aparcamiento de vagones.

Cuando se monta un tren, los vagones siempre se enganchan inmediatamente detrás de la locomotora. Para dismantelar un tren, los vagones también se desenganchan empezando por el primer vagón (el vagón que se encuentra inmediatamente detrás de la locomotora).

Las operaciones que necesitan implementarse son:

1. **procedure vacio(cia: out cTrenes);** (0.25 puntos)
 - Prepara la estructura de la compañía de trenes para trabajar con ella.
2. **procedure aparcaLocomotora(cia: in out cTrenes; k: in tcodigo);** (0.25 puntos)
 - Da de alta una nueva locomotora y la aparca en el aparcamiento de locomotoras libres.
 - Si el aparcamiento de locomotoras libres se encuentra completo, debe lanzar una excepción **aparcamiento_locomotoras_completo**.
3. **procedure aparcaVagon(cia: in out cTrenes; k: in tcodigo; pmax: in integer);** (0.25 puntos)
 - Da de alta un nuevo vagón y lo aparca en el aparcamiento de vagones libres.
 - Si el aparcamiento de vagones libres se encuentra completo, debe lanzar una excepción **aparcamiento_vagones_completo**.
4. **procedure listarTrenes(cia: in cTrenes);** (0.75 puntos)
 - Lista todos los trenes que se encuentran montados ordenados **ascendentemente por su peso de carga máxima acumulado**.
 - El peso de carga máximo acumulado de un tren se corresponde a la suma del peso máximo de carga de todos los vagones que lo forman.
 - Por cada uno de los trenes listados, además de la información del código del tren y su peso de carga máximo acumulado, debe mostrarse la información del código de la locomotora que lo forma, además del código y peso máximo de cada uno de los vagones que lo componen.
5. **procedure creaTren(cia: in out cTrenes; t: out tcodigo; num_vagones: in Integer);** (1.75 puntos)
 - Crear un nuevo tren a partir de una locomotora libre y el número indicado de vagones (de entre los vagones libres aparcados).

- El código del tren se corresponde con el código de la locomotora que lo forma pero, en lugar de empezar por L, debe empezar por T.
- Si no quedan locomotoras libres, debe lanzar la excepción **locomotoras_agotadas**.
- Si no quedan suficientes vagones libres, debe lanzar la excepción **vagones_agotadas**.
- Si no se dispone de espacio suficiente para almacenar el tren montado, debe lanzar la excepción **inventario_trenes_completo**

6. **procedure consultaTren(cia: in cTrenes; t: in tcodigo);** (0.75 puntos)

- Debe mostrar la información del tren con el código identificativo *t*.
- La información del tren que debe mostrarse es: código del tren, código de su locomotora y, por cada uno de los vagones que lo componen, el código y el peso máximo de cada vagón.
- Si no existe en el inventario de trenes montados un tren con el código identificativo *t*, debe lanzar la excepción **tren_no_existe**.

7. **procedure desmantelarTren(cia: in out cTrenes);** (2 puntos)

- Desmantela el tren con el menor peso de carga máxima acumulada.
- Desmantelar un tren significa:
 - Desenganchar todos sus vagones (empezando por el vagón que se encuentra inmediatamente detrás de la locomotora) y aparcarlos en el aparcamiento de vagones libres.
 - Aparcar su locomotora en el aparcamiento de locomotoras libres.
 - Eliminar el tren del inventario de trenes montados.
- Si no hay ningún tren montado, debe lanzar la excepción **tren_no_existe**.
- Si el aparcamiento de vagones libres se encuentra completo, debe lanzar una excepción **aparcamiento_vagones_completo**.
- Si el aparcamiento de locomotoras libres se encuentra completo, debe lanzar una excepción **aparcamiento_locomotoras_completo**.

2 Entrega de la práctica

La entrega de la práctica constará de dos entregas individuales:

1. Entrega del diseño de la estructura (3 puntos): 26/05/2021; 23:59h.
 - En la primera entrega debe entregarse una documentación (concreta y breve) donde se describan las diferentes estructuras utilizadas y la justificación de su utilización.
 - Proponer una estructura de datos para gestionar el problema. Explicad y justificad la estructura propuesta. Dibujad un diagrama de la solución. Toda suposición que se haga y no esté especificada en el enunciado, tiene que describirse.
2. Entrega de la implementación de la estructura (7 puntos): 14/06/2021; 23:59h.
 - Especificación del tipo abstracto de datos (parte genérica, parte pública y parte privada) para gestionar el problema (.ads): 1 punto.
 - Implementar las operaciones listadas (*package body*, .adb): 6 puntos.
 - La segunda entrega constará de una carpeta comprimida que contendrá el proyecto GPS con el código fuente correspondiente a la implementación de la estructura, además de un programa principal donde se testeen las diferentes operaciones implementadas. En caso de no disponer de proyecto GPS, debe entregarse una carpeta comprimida con los ficheros .ads y .adb.