

Práctica 6b.

Mapas

En esta práctica introducimos los Mapas, estructuras Java que nos permiten recuperar información de colecciones de una forma más simple y eficiente.

Cuestiones Previas.

Copia todo tu código generado del paquete P6a de la práctica anterior, a un nuevo paquete P6b y realiza en este todas las modificaciones necesarias para la práctica.

1 Actividad 1. Introduciendo Mapas en la clase CarDB.

En esta actividad incorporamos mapas para manejar de forma más eficiente la información del conjunto de coches que entran y salen del parking, que hasta este momento hemos recogido en la colección `cityCars`.

Para ello, en la clase `CarDB`, modifica la declaración de `cityCars` como colección `ArrayList` a un mapa `LinkedHashMap` de matrícula¹ como clave, y `Car` como valor.

Incorpora los cambios necesarios en el código de aquellos métodos de la clase `CarDB` que hagan referencia a la variable `cityCars`, ahora definida como mapa, según se describe a continuación:

- En general, en aquellos métodos que necesiten recorrer la colección de coches del mapa (`saveCarsToFile`, `computeTotalPower`, ...) usa la colección de valores del mapa `cityCars.values()`.
- Método `getCarFromPlate`. Modifica el código para recuperar el coche del mapa a partir de su matrícula.
- Método `readCityCarsFile`. Modifica las inserciones al `ArrayList` por inserciones (`puts`) al mapa, usando como valor el objeto de clase `Car` (que hayas creado a partir de los datos de la línea del fichero) y como clave la matrícula correspondiente a ese objeto.
- Métodos de ordenación `sortByPlate` y `sortByBatteryChargeAndPlate`. Para ordenar por cualquiera de los criterios sigue los pasos siguientes:
 - Declara y crea una colección `ArrayList` temporal de coches.
 - Pueba el `ArrayList` a partir de la colección de valores del mapa.
 - Ordena el `ArrayList` por el criterio correspondiente.
 - Borra todo el contenido del mapa y vuelve a poblarlo a partir del `ArrayList` ordenado.

¹ Puesto que no hemos creado una clase específica para la matrícula, usa la clase `String` para modelar la clave del mapa.

2 Actividad 2. Clase principal.

Crea la clase P6b, que contendrá el método main y será la clase principal de la práctica. En el método main:

- Recibe y lee seis argumentos, los nombres de cinco ficheros:
 - file1: un fichero existente con la estructura y contenido de un parking.
 - file2: un fichero existente de entradas y salidas.
 - file3: el nombre del fichero en el que guardar el resultado de la actualización del parking.
 - file4: un fichero existente con los coches de ciudad.
 - file5: el nombre del fichero donde guardar la colección de coches cityCars.
 - file6: el nombre del fichero donde guardar el dibujo del parking.
- Crea un objeto cdb de la clase CarDB e invoca su método readCityCarsFile para leer el fichero file4 con los coches y que estos se guarden en el array cityCars del objeto creado cdb.
- Crea un objeto de la clase Parking (asígnalo a una variable global miParking) a partir del fichero file1.
- Actualiza miParking a partir del fichero de entradas/salidas file2 (llamando a processIO(file2)).
- Guarda miParking invocando el método saveParking con file3.
- Ordena cityCars por matrícula invocando el método sortByPlate desarrollado en la actividad 1.²
- Guarda cityCars en el fichero file5, invocando el método saveCarsToFile.
- Crea el dibujo del parking invocando el metodo toMap() de la clase Parking y salva el resultado en el fichero file6.

² Observa que este punto difiere ligeramente del mismo punto de la practica 6a. Ahora debes ordenar por matricula y no por bateria+matricula.