Tarea evaluable. Unidad 6. Estructuras internas. Memoria

Introducción

Hasta el momento hemos utilizado en todos los casos estructuras estáticas (arrays) para almacenar datos, llega el momento de aprovechar las bondades de las estructuras dinámicas y la funcionalidad de las colecciones.

Enunciado

Vehículos

Se trata de realizar una mejora en la aplicación ya desarrollada y subida a git en el paquete u06.vehiculos, en la que gestionamos un concesionario cuyos vehículos son insertados en un array. El objetivo es mantener los vehículos ordenados por matrícula en la estructura de datos. El objetivo de este ejercicio es:

- Hacer las modificaciones a la clase Vehiculo para que sean objetos comparables por matrícula.
- Modificar la clase Concesionario para que utilice una estructura de datos dinámica que mantenga los vehículos ordenados. Determina qué estructura es la más apropiada, justificando tu respuesta (Puedes hacerlo en la misma declaración de la propiedad como documentación Javadoc o comentarios de programador).
- Añadir la opción Eliminar Vehículo: Dada una matrícula, eliminar el vehículo cuya matrícula coincide si existe.
- Testear y solucionar los posibles errores de funcionamiento que puedan existir en el código original, dejando un comentario de la modificación que identifique el cambio.
- Añadir documentación al código de tal forma que todas las funcionalidades queden bien definidas para un futuro mantenimiento.
- Identificar dichos cambios en el código. Ejemplo:
  //CAMBIO-001 se cambia la comprobación de matrícula….

```
}
// CAMBIO-001
public int insertarVehiculo(Vehiculo v){

    if(this.numVehiculos == this.vehiculos.length){
        return -1;
    }

    if(this.buscaVehiculo(v.getMatricula()) != null){
        return -2;
    }else{
```

Diccionario

1. Crea un mini-diccionario español-inglés que contenga, al menos, 20 palabras (con su correspondiente traducción). Utiliza un objeto Map para almacenar las parejas de palabras. El programa pedirá una palabra en español y mostrará la correspondiente traducción en inglés (solo 1 palabra cada vez que se ejecuta el programa).
2. Implementa en el programa anterior dos funciones auxiliares:
    a. Un método para corregir palabras erróneas. Se le pasará la palabra a modificar y la nueva traducción
    `corregirPalabra(String palabraSpa, String nuevaTraduccionEng)`
    b. Un método para eliminar palabras del diccionario. Se le pasará una palabra y deberá eliminar la palabra que coincida o bien en español o bien en inglés.
    `eliminarPalabra(String palabra)`
3. Realiza un programa que escoja al azar 5 palabras en español del minidiccionario del ejercicio anterior. El programa irá pidiendo que el usuario teclee la traducción al inglés de cada una de las palabras y comprobará si son correctas. Al final, el programa deberá mostrar cuántas respuestas son válidas y cuántas erróneas.

Evaluación

Criterios de puntuación. Total 10 puntos.
La tarea tiene una puntuación total de 10 puntos repartidos de la siguiente forma:
Vehículos

- Se comparan correctamente objetos vehículos: 1,25 puntos.
- Se modifica el concesionario con una estructura de datos ordenada y sigue siendo funcional: 1,75 puntos.
- La opción de eliminar vehículo funciona correctamente: 1 punto.
- Documentación del código con Javadoc: 1 punto.

Mini-diccionario

- Creación de diccionario con palabras y traducciones: 0,75 puntos.
- Corregir palabra: 1 punto.
- Eliminar palabra: 1,25 puntos.
- Juevo de traducción: 2 puntos

Entrega

Entregar las clases java dentro de un archivo comprimido con extensión .zip.