

Tarea Semana 2

Fecha de entrega: viernes 22 de mayo, 14:00 Hrs.

Artefactos entregables:

- Problema 1: ED_Tarea_S3_P1.zip
- Problema 2:
 - Código: ED_Tarea_S3_P2.zip
 - Documento: ED_Tarea_S3_P2.pdf
 - Diagramas:
 - diagrama_v1.png
 - diagrama_v2.png

Problema 1:

- Para resolver este problema considere el proyecto IntelliJ disponible en el archivo ED_Tarea_S3_P1.zip. Aquí encontrará la clase ListaEnlazadaUtils.
- En la clase ListaEnlazadaUtils deberá implementar los siguientes métodos:
 - insertarOrdenado(), removerValoresMaximos(); y contieneSubsecuencia();

Especificación de cada método:

insertarOrdenado: este método asume que el parámetro de entrada lista enlazada está ordenado en orden no descendente (es decir, cada elemento es mayor o igual que el anterior), e inserta el valor int de entrada en la posición correcta de la lista. Note que el método no retorna un valor, pero modifica la lista enlazada. Considere además que, si la lista enlazada de entrada es null, este método debería hacerse cargo de esta excepción.

removerValoresMaximos: este método remueve todas las instancias de los N valores más grandes en la lista enlazada. Si la entrada lista enlazada es null o si N es no positivo, este método debería retornar que todas las instancias se han removido.

contieneSubsecuencia: este método determina si alguna parte de la primera lista enlazada contiene todo los elementos de la segunda en el mismo orden sin otros elementos en la secuencia. Es decir, el método debería retornar "true" si la segunda lista enlazada es una subsecuencia de la primera, y "false" si no lo es. El método debe devolver falso si cualquiera de las entradas es nula o está vacía.

Documentación listas enlazadas: <https://docs.oracle.com/javase/8/docs/api/java/util/LinkedList.html>

En el Problema 1, considere las siguientes instrucciones:

Tarea Semana 2

1. Implemente cada uno de los métodos antes mencionados, considerando todos los casos de excepción. Recuerde que en clases se revisó sólo algunos excepciones y, por lo tanto, debe investigar otros casos que puedan hacer que su programa no funcione correctamente.
2. Observación 1: si bien se recomienda utilizar pruebas unitarias para evaluar el correcto funcionamiento de los métodos que implemente, no debe incluirlas en su entrega.
3. Observación 2: considerando el punto 1, cabe señalar que su código será evaluado automáticamente. Por lo tanto, sólo existirán métodos correctos o incorrectos.
4. Observación 3: considerando los puntos 1 y 2, es necesario recalcar que no debe modificar la estructura del código de la clase ListaEnlazada, salvo en el cuerpo de los métodos antes mencionados. Considere que si realiza algún cambio fuera de ellos, su código no podrá ser evaluado y, por lo tanto, recibirá puntuación mínima en este problema.

Problema 2:

Contexto del problema:

- Para el desarrollo de este problema considere todas las actividades a completar en el Sistema de gestión de Guitarras.

Entregables:

- Entrega de los problemas 1 y 2, en el siguiente formato:
 - Problema 1: ED_Tarea_S3_P1.zip
 - Problema 2:
 - Código: ED_Tarea_S3_P2.zip
 - Documento: ED_Tarea_S3_P2.pdf (por favor no entregar en formato word).
 - Diagramas:
 - diagrama_v1.png
 - diagrama_v2.png

Nota:

- No olvide las consideraciones de diseño de software revisadas en clase.