

Ingeniería de Software I

La TDA Lista

Objetivo

Utilizar las operaciones de la TDA Lista para eliminar elementos repetidos en una lista (la purga de elementos repetidos)

Descripción

La actividad trata sobre completar el programa en la carpeta “Programas a completar\02 TDA Lista”.

Requerimientos Generales

1. Entregar y utilizar todo el código fuente de la carpeta citada en la descripción.
2. Del citado código fuente modificar solamente el archivo “main.cpp” en la operación `purgaLista()` y sin modificar la firma de dicha operación.
3. La subrutina `purgaLista()` declare tantas variables locales de tipo “posicion” como sean necesarias y usar dentro de dicha subrutina variables sólo de tipo “posicion”.
4. Desde el interior de la subrutina `purgaLista()` sean llamadas solo subrutinas (métodos) alojadas en el objeto de tipo `Lista` recibido como parámetro.
5. Utilizar a la función `Lista::mismo()` para el efecto de comparar dos elementos de la lista.
6. La subrutina `purgaLista()` elimine solo los elementos repetidos en la lista.
7. El algoritmo en `purgaLista()` contemple:
 - a) que los elementos repetidos podrían estar tanto en posiciones consecutivas como no consecutivas;
 - b) la posibilidad de recibir una lista de cualquier cantidad de elementos.
 - c) evitar aplicar (o que se asuma haber aplicado) un algoritmo de ordenamiento a la lista.
8. Al interior de la subrutina `purgaLista()` no se debe hacer alusión alguna a que las posiciones de la lista son secuenciales en el arreglo, esto es, se prohíbe el uso de los operadores: “+”, “++”, “-“, “--“, “+=”, “-=“, “>”, “>=”, “<” y “<=” al interior de la subrutina.
9. INDISPENSABLE PARA DERECHO A CALIFICACIÓN MAYOR A 5/100: cumplir con los puntos #1, #2, #3, #4 y 8.
10. Habiendo ya cumplido con todos los requisitos anteriores cumplir con lo siguiente:

- a) Incorporar en el programa una opción de menú y una subrutina adicional que permitan escribir todos los registros en memoria RAM hacia un archivo de registros, en el que se aplique la técnica de registros de longitud variable con delimitadores.
- b) Al iniciar el programa, antes de mostrar el menú principal, mediante la implementación adicional de una subrutina de inicialización, se cargue el listado de registros, si es que existe un archivo.
- c) Incluir en la entrega un archivo “.txt” con un listado de registros que se cargue al iniciar el programa.

Requerimientos Optativos para Puntos Extras

11. Entregar en el mismo comprimido carpetas diferentes para cada lenguaje:

- a) Una nombrada “C” con los códigos fuente en C (ANSI);
- b) Una nombrada “C++” con los códigos fuente en C++;
- c) Una nombrada “Java” con los códigos fuente en Java;
- d) Una nombrada “C#” con los códigos fuente en C#;
- e) Una nombrada “Python” con los códigos fuente en Python;
- f) Una nombrada “English” con los códigos fuente escritos en su totalidad en inglés (solo para el lenguaje de su preferencia).

...las primeras opciones se redacten absolutamente en español y la última absolutamente en inglés. Si incluye 2 carpetas con lenguajes OO, uno de ellos puede redactarse en inglés y sin necesidad de entregar la versión en español para ese lenguaje.

Criterios de Evaluación

- Los establecidos en las “Reglas de Operación y Evaluación” del curso.
- Cumplir con la fecha límite de entrega citada en el Excel de Actividades.
- Calificación en base a cobertura de requerimientos y fecha de entrega.
- Cumplir con Requerimientos de Valor Agregado en Código Fuente (hasta el req. “PP”).
- Entrega en un lenguaje: C++ (ANSI).
- Es indispensable la entrega de un programa cuyo código fuente sea completamente en idioma Español (a excepción de lo correspondiente a la API del lenguaje).