



Laboratorio de Estructuras de Datos

Práctica 5. Arreglos unidimensionales ordenados

Unidad Temática: 1. Introducción a las estructuras de datos y estructuras fundamentales

📄 👤 Profesor: Dr. Aldonso Becerra Sánchez

Índice

1	Objetivo de la tarea	1
2	Tiempo aproximado de realización	1
3	Fecha de entrega	1
4	Fecha de entrega con extensión y penalización	1
5	Introducción	1
6	Actividades a realizar	1
6.1	Actividad Inicial	1
6.2	Actividad 1	1
6.3	Actividad 2	1
6.4	Actividad 3	2
6.5	Actividad 4	2
6.6	Actividad 5	2
6.7	Actividad 6	2
6.8	Actividad 7	2
6.9	Actividad 8	2
7	Contáctame	2
	References	2

más eficiente. El único detalle con esta cuestión es que es poco flexible el número de elementos que podemos manipular, ya que se requiere conocer a priori la cantidad de elementos a guardar [1], [2], [3].

6. Actividades a realizar

6.1. Actividad Inicial

Lea primero toda la práctica ⚠️. No inicie a programar sin leer todo cuidadosamente primero. Recuerde que debe generar el reporte en formato IDC con todos sus componentes.

6.2. Actividad 1

Primero genere la Introducción 📄.

6.3. Actividad 2

Información importante

Esta actividad debe entrar en la parte de *Desarrollo* 📄.

Defina el TDA llamado “ArregloListaInfoEstaticaOrden”, el cual debe tener atributos y métodos correspondientes. Recuerde que el orden en números no es lo mismo que el orden en cadenas y otros tipos de objetos; ya que debe personalizar este mecanismo según sea el tipo de contenido. Nota: no utilice ningún método de ordenamiento formal. Se pide que realice los siguientes métodos dentro de esta clase (deben estar enfocados para datos estrictamente ordenados, lo cual no es igual para datos desordenados):

- **public ArregloListaInfoEstaticaOrden(int tamaño, TipoDeOrden orden).** Es el constructor del arreglo con orden. Este arreglo además de indicar el tamaño, debe crear un enumerado en donde se indique el orden/acomodo de las inserciones, ya sea ASCENDENTE (0) o DESCENDENTE (1). Ese orden se respetará en todo el conjunto de métodos.
- **public Integer nuevo(Object valor).** Insertar elementos mediante mecanismos ordenados en un orden definido en el constructor.
- **public Object buscar(Object valor).** Buscar elementos mediante mecanismos ordenados.
- **(declarado en interface ArregloListaDatos) public boolean modificar(Object valorViejo, Object valorNuevo, int numApariciones).** Este método debe reorganizar los datos después de ser modificado el valor indicado. Note que numApariciones siempre tenderá a ser uno, dado que en este tipo de arreglos no se permiten valores duplicados.

1. Objetivo de la tarea

Crear un TDA arreglo unidimensional ordenado y desordenado para su posterior uso en aplicaciones comunes.

2. Tiempo aproximado de realización

🕒 5 horas.

3. Fecha de entrega

📅 28 agosto de 2024.

4. Fecha de entrega con extensión y penalización

📅 29 agosto de 2024.

5. Introducción

La facilidad que los arreglos tienen para permitir guardar más de un dato con una sola variable lo hace pertinentes para la resolución de muchos problemas donde se requiere esta situación. En muchos casos cuando el arreglo está ordenado, facilita búsquedas

51 ■ **public boolean modificar(int posicion, Object valor).** Este
52 método deberá sobrescribirse para que el contenido del arreglo
53 se reorganice.
54 ■ **public Object quitar(Object valor).** Eliminar un elemento del
55 arreglo.
56 ■ **(declarado en interface ArregloListaDatos) public boolean**
57 **agregarArreglo(ArregloListaDatos arreglo2).** Debe permitir
58 agregar los elementos de arreglo2 (que debe validar que sea un
59 arreglo [ordenado o desordenado] en este caso) en el arreglo
60 actual. Debe reorganizar todos los elementos insertados, de tal
61 manera que el arreglo siga ordenado. Recuerde que el arreglo
62 ordenado no permite valores duplicados. arreglo2 debe ser de
63 tipo ArregloListaInfoEstaticaOrden.
64 ■ **(declarado en interface ArregloListaDatos) public void**
65 **voltrear().** Debe invertir el orden de los elementos del arreglo. Al
66 mismo tiempo que se cambia el orden, debe cambiarse el valor
67 de la variable TipoDeOrden, si no mostrará inconsistencias.
68 ■ **(declarado en interface ArregloDatos) public void lle-**
69 **nar(Object valor).** “valor” indica el límite superior en los valo-
70 res a rellenar, siempre y cuando sea numérico. Por ejemplo, si
71 valor es 6, se insertaron elementos del 1 al 6 (en las posiciones
72 de 0, 1, 2, 3, 4, 5) [orden ASCENDENTE]. Si valor es 6 y orden
73 es DESCENDENTE, se insertaron elementos del 6 al 1 (en las
74 posiciones de 0, 1, 2, 3, 4, 5). Si el valor es negativo, por ejemplo
75 -6, se debe insertar elementos del -1 al -6 (en las posiciones 0, 1,
76 2, 3, 4, 5) [orden DESCENDENTE]. Si el valor es negativo y el
77 orden es ASCENDENTE, se debe insertar elementos del -6 al -1
78 (en las posiciones 0, 1, 2, 3, 4, 5). Si es una letra, por ejemplo H,
79 se deberán insertar elementos como A, B, C, D, E, F, G, H (en
80 las posiciones 0, 1, 2, 3, 4, 5...) [orden ASCENDENTE]. Si es
81 una letra, por ejemplo H, se deberán insertar elementos como
82 H, G, F, E, D, B, A (en las posiciones 0, 1, 2, 3, 4, 5...) [orden
83 DESCENDENTE]. Si es cadena, por ejemplo “hola”, debe insertar
84 solo este elemento [orden ASCENDENTE]. Si es otro tipo de
85 objeto, también debe insertar solamente ese valor de manera
86 única, ya que no se permiten valores duplicados. Recuerde que
87 este método está en función del tamaño del arreglo, no debe
88 sobrepasarse.
89 ■ **public ArregloDatos arregloDesordenado().** Debe regresar
90 un arreglo desordenado, de tal manera que los elementos alma-
91 cenados deben reburujarse a tal grado que ya no estén ordenados,
92 no solo regresar un objeto tipo ArregloListaEstatico.
93 ■ **(declarado en interface ArregloListaDatos) public boolean**
94 **esSubArreglo(ArregloListaDatos arreglo2).** Debe indicar si
95 la arreglo2 (que es otro arreglo ordenado) es una sublista o sub-
96 conjunto de la lista actual. Por ejemplo, la lista actual es: 1, 2, 3,
97 4, 5 y la lista2 es: 3, 4, 5; el valor de retorno debe ser true.
98 ■ **(declarado en interface ArregloListaDatos) public boolean**
99 **cambiarArreglo(ArregloListaDatos arreglo2, ArregloLista-**
100 **Datos arreglo2Nuevos).** Debe cambiar los elementos de arre-
101 glo2 que se encuentren en el arreglo actual con los elementos de
102 la arreglo2Nuevos. Cada elemento de arreglo2 coincide en posi-
103 ción con su nuevo valor a cambiar en arreglo2Nuevos. Ejemplo,
104 arreglo2= 2, 3, 4, arreglo2Nuevos=50, 40, 80. Quiere decir que
105 si encuentra un 2 en lista actual debe sustituirlo por un 50, si
106 encuentra un 3 en lista actual debe sustituirlo por 40, etc.
107 ■ **(declarado en interface ArregloListaDatos) public boolean**
108 **retenerArreglo(ArregloListaDatos arreglo2).** Debe dejar en
109 la lista actual solo los elementos que se encuentran en arreglo2.

el método para que sea sobre-escrito en la clase ArregloListaIn-
foEstaticaOrden de tal manera que se valide que siga ordenado.
■ **(declarado en interface ArregloListaDatos) public boolean**
copiarArreglo(ArregloListaDatos arreglo2). Copiar el conte-
nido de arreglo2 a la lista actual. Debe validarse el tamaño que
sea del mismo tipo. El contenido de la lista actual se perderá al
ser substituido por la arreglo2. También modifique el método
para sea sobreescrito en la clase ArregloListaInfoEstaticaOrden
tal manera que se valide que siga ordenado. Es decir, si tiene un
orden definido en arreglo2, se debe poder copiar si se respeta el
orden ASCENDENTE O DESCEDENTE, en caso contrario no
debe poderse hacer.


6.5. Actividad 4

Pruebe el funcionamiento del programa de la actividad 2 con todo y
sus capturas de pantalla.

6.6. Actividad 5

Realice la sección de Código agregado  (diagrama de clases UML).





6.7. Actividad 6

Realice la sección de Pre-evaluación  (use los lineamientos esta-
blecidos).

6.8. Actividad 7

Finalmente haga las Conclusiones .


6.9. Actividad 8


Subir los entregables (pdf  y zip  con código )  a Moodle.


7. Contáctame

Puedes contactarme a través de los siguientes medios.

 <https://moodle.ingsoftware.uaz.edu.mx/>

 a7donso@gmail.com

 Cubículo

 Salón CC2-IS

Referencias

- [1] O. Cairo y S. Guardati, *Estructura de datos*. McGraw-Hill.
- [2] L. Joyanes Aguilar, *Fundamentos de programación, algoritmos u estructura de datos*. McGraw-Hill.
- [3] M. A. Weiss, *Estructura de datos en Java*. Addison Wesley.

6.4. Actividad 3

Complete el TDA llamado “ArregloListaInfoEstatica”. Para esto debe-
rá codificar el método:

- **(declarado en interface ArregloListaDatos) public boolean**
nuevo(int posicion, Object valor). Este método insertará en
la posición “posicion” el contenido de valor. También modifique