



PRÁCTICA 9: BOLSAS QUE CREAN ÁRBOLES EN LUGAR DE DESTRUIRLOS

1. Introducción

Un Tipo Abstracto de Datos (TAD) es un tipo que se define, es decir, se *especifica*, de forma independiente a su representación, es decir, a su *implementación*. La especificación algebraica de un TAD sirve para indicar de forma precisa y no ambigua:

- (1) cómo se construye el conjunto de individuos o *términos* que pertenecen a un tipo;
- (2) qué términos son iguales entre sí;
- (3) qué operaciones se le pueden aplicar a dichos términos
- (4) bajo qué condiciones es aplicable cada operación;
- (5) qué hace exactamente cada operación.

Existen infinitas maneras de implementar una especificación, cada una con sus propias ventajas e inconvenientes, siendo éstas dependientes del contexto. Es por tanto tarea de un buen ingeniero software saber elegir la implementación más adecuada para cada contexto.

En general, para implementar un TAD se suelen usar vectores (*arrays*), listas enlazadas, cursores¹, tablas de dispersión o árboles autobalanceados, entre otras estructuras. En el caso concreto de esta práctica, implementaremos un TAD *Bolsa* usando para ello una implementación hecha por terceros de un árbol roji-negro (un tipo concreto de árbol binario de búsqueda autobalanceado).

Por tanto, cada bolsa que creamos en nuestra aplicación generará un nuevo árbol, a diferencia de lo que ocurre en los centros comerciales, donde por cada bolsa que usamos, destruimos parte de un árbol.

2. Objetivos

Los objetivos de esta práctica son:

- (1) Aprender a implementar TADs a partir de sus especificaciones algebraicas.
- (2) Aprender a diferenciar el concepto de interfaz e implementación.
- (3) Aprender a implementar bolsas sobre árboles binarios de búsqueda autobalanceados.
- (4) Aprender a usar implementaciones realizadas por terceros de estructuras de datos comunes.

¹ Los cursores no han sido vistos en clase.



3. Actividades

Usando la especificación algebraica del TAD *Bolsa* disponible en el Tema 5 de la plataforma *moodle*:

- (1) Crear una interfaz *IBag* que corresponda a la interfaz de dicho TAD.
- (2) Usando la clase *RBTree* proporcionada a través de *moodle*, implementar la interfaz *IBag*. Tener en cuenta que la clase *RBTree* no admite elementos repetidos, por tanto, hemos de añadir un contador a cada elemento almacenando en el *RBTree* para mantener un control de las instancias que hemos de almacenar.
- (3) Crear un programa de prueba que sirva para testear las operaciones del TAD *Bolsa* en un amplio abanico de casos.

4. Criterios de Autoevaluación y Aclaraciones

En ningún momento se calificará la práctica. Los criterios expuestos en esta sección están destinados a permitir que el alumno pueda autoanalizar cuando la solución dada a la práctica cumple con los criterios de calidad necesarios para superar la asignatura.

En general, para una correcta resolución de la práctica, se deberán satisfacer los siguientes criterios:

- (1) Se cumplen las normas expuestas en la *Guía de Buenas Prácticas en Programación*.
- (2) La definición de la interfaz *IBag* está libre de detalles de la representación.
- (3) La implementación de la interfaz *IBag* usando un árbol roji-negro es correcta.
- (4) Las pruebas realizadas sobre los métodos implementados son exhaustivas.
- (5) El diseño de la clase que implementa los métodos es eficiente y lo más simple posible.
- (6) Los métodos es lo más genérico posible, y no se imponen restricciones innecesarias que dificulten la reutilización de los elementos creados.

No es necesario entregar la práctica, aunque en caso de no realizarla o realizar de forma insatisfactoria, no existe garantía alguna de superar la asignatura.

Pablo Sánchez Barreiro.