

Taller de C++ 3: Árboles Binarios de Búsqueda Implementación de Conjunto mediante ABB

Normativa

- **Límite de entrega:** Miércoles 28 de septiembre de 2016 a las 22:00 hs.
- **Límite de re-entrega:** Miércoles 5 de octubre de 2016 a las 12:00 hs.
- Este taller se realiza de forma **individual** y con entrega digital.
- **Normativa completa:** Para más detalles véase ‘Información sobre la cursada’ en el sitio Web de la materia: <http://www.dc.uba.ar/materias/aed2/2016/2c/cursada>

Enunciado

El objetivo de los talleres es familiarizarse con el lenguaje C++ y las características del mismo que se usarán en esta materia (clases, templates, memoria dinámica, etc.), de manera de llegar mejor preparados a afrontar un desarrollo más grande y complicado como el TP3.

Se pide:

1. Implementar C++ la clase `Conjunto` para elementos de tipo genérico, cuya interfaz será entregada por la cátedra en el archivo `Conjunto.hpp` adjunto. Dicha interfaz está basada en el TAD `Conjunto` provisto con anterioridad en el apunte de TADs Básicos.
2. Pueden agregar funciones y variables auxiliares en la parte privada, pero nada en la parte pública de la clase `Conjunto`.
3. Para la entrega por mail, adjuntar únicamente el archivo `Conjunto.hpp`.

Normas de entrega adicionales

- Al finalizar la implementación se deberá enviar el archivo con la solución a la siguiente dirección de correo electrónico: `algo2.dc+taller3@gmail.com`. El asunto deberá indicar su libreta universitaria en formato: `LU XXX/YY`.
- El trabajo se evaluará automáticamente utilizando para ello una batería de tests que no serán provistos a los alumnos. Si el trabajo resulta aprobado se reenviará, también de forma automática, un correo respuesta informando esta situación. En caso negativo la respuesta indicará que tipo de problema no pudo resolver el código en cuestión. El sistema debería responder en un tiempo razonable si el TP fue aprobado o no. Si no recibe confirmación de la recepción de su TP luego de 30 min., por favor informe su situación a la lista de docentes. Se pueden hacer tantas entregas como se necesiten, hasta la fecha límite de entrega.
- La implementación dada no debe perder memoria en ningún caso. Se recomienda la verificación mediante el uso de *valgrind*.
- Está prohibido utilizar la biblioteca estándar de C++ (STL) con la excepción de las siguientes bibliotecas: **`iostream`, `string` y `cassert`**.
- Se admite el estándar de C++ hasta el año 2003. Está prohibido usar las funcionalidades que agrega el estándar C++11 o posteriores.

Recomendaciones

- Respecto de los archivos provistos, si intentan compilar `tests.cpp` podrán hacerlo, pero no podrán linkearlo y generar un binario porque, por supuesto, va a faltar la implementación de todos los métodos de la clase testada.
- Una sugerencia para empezar es dejar la implementación de todos los métodos necesarios escrita, pero vacía, de manera de poder compilar. Pueden comentar todos los tests que requieran métodos aún no implementados de manera de poder usar la aplicación para el testing a medida que van implementando. Tengan en cuenta que algunos métodos pueden ser necesarios para muchas de las funciones de test, por lo tanto, es aconsejable empezar por esos test.
- Además de los casos de test provistos por la cátedra les recomendamos fuertemente que realicen sus propios tests.

- Dado que su implementación no debe perder memoria, es una buena práctica utilizar la herramienta *valgrind* durante el desarrollo, como mínimo en la parte de testing final.
- Se sugiere chequear las precondiciones con **assert** para facilitar la depuración de errores.
- Se sugiere repasar el capítulo 10 del Cormen (ver bibliografía en la página de la materia).