

Estructura de Datos (2015-2016) Grupo: B3
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Practica de Abstracción

9 de noviembre de 2015

Índice

1. Objetivos de la Practica	1
2. Ejercicio a resolver	1
3. Resolución del Ejercicio	1
4. Conclusión	2

1. Objetivos de la Practica

Los objetivos de este guión de prácticas son los siguientes:

- Asimilar los conceptos fundamentales de abstracción, aplicado al desarrollo de programas.
- Documentar un tipo de dato abstracto (T.D.A)
- Practicar con el uso de doxygen.
- Profundizar en los conceptos relacionados especificación del T.D.A, representación del

T.D.A., función de Abstracción e Invariante de la representación. Los requisitos para poder realizar esta práctica son:

- Haber estudiado el Tema 1: Introducción a la eficiencia de los algoritmos
- Haber estudiado el Tema 2: Abstracción de datos.

2. Ejercicio a resolver

El objetivo en este ejercicio es crear una aplicación que nos genere exámenes de tipo test, con un numero determinado de preguntas, escogidas de forma aleatoria de un conjunto de preguntas con sus respectivas respuestas. Con tal fin vamos a desarrollar varios tipos de datos abstractos:

- Pregunta: Se conforma del texto de la pregunta y las respuestas posibles a dicha pregunta.
- Conjunto de Preguntas: Es una colección de objetos de tipo Pregunta.
- Test: Contiene un numero de preguntas, con sus respuestas, escogidas de forma aleatoria.

3. Resolución del Ejercicio

Después de entender cuales son los objetivos de la practica, me puse a ver la posibilidad de implementarla con un vector de la STL, pero dado que esta opción no era posible utilizarla, opte por empezar a implementar Pregunta con listas enlazadas.

Cuando ya tenia implementada la clase Pregunta, pase a implementar la clase ConjuntoPreguntas y aquí es cuando me di cuenta de que al intentar implementar el operador \ll la cosa se complicaba un poco, así que decidí simplificarlo.

Después de eso me di cuenta que las respuestas que tenían que tener las preguntas se podían almacenar en un string unas detrás de otras, así de esa manera cuando tuviera que pintar por pantalla la pregunta bastaría con incluir el carácter de escape `\n` y así las respuestas saldrían una debajo de la otra.

Aparte con las listas enlazadas también estaba el inconveniente que cuando generábamos los test aleatorios no teníamos la posibilidad de poder acceder a una posición del dato en concreto osea:

ConjuntoPreguntasCP[4]

Al intentar implementar este dato con listas enlazadas, significaba que si teníamos que acceder a una posición de memoria tal que fuera 1500 tendríamos que recorrer la colección completa hasta la posición 1500, mientras que si lo hacíamos con memoria dinámica teníamos la posibilidad de poder acceder al sitio que quisiéramos.

4. Conclusión

Después de haber estudiado las dos maneras posibles, (que se me ocurrieron) de montar la estructura para cada uno de los tipos abstractos, me decante por una cadena de strings para guardar las respuestas de las preguntas y dos vectores dinámicos para guardar la Pregunta tanto en el ConjuntoPreguntas como en Test.

Así de esta manera de lo único que tendría que preocuparme era de la gestión de memoria cuando se dejaran de usar los diferentes objetos.

Las demás variables que se pueden apreciar en la documentación de doxygen, son simplemente variables de control para saber el tamaño y otros aspectos de cada uno de los objetos, para generar la documentación basta con usar el comando *make docu*.