



PRÁCTICA N° 7: Colas.

Fecha de realización: Semana del 13 al 17 de Mayo (entrega hasta una semana después).

Duración: 2.5 horas presenciales

Objetivos de la práctica

- Desarrollar una aplicación en C++ desde cero.
- Implementar una estructura tipo Cola con representación dinámica
- Usar estructuras del tipo Cola disponibles en la biblioteca STL.

Material a entregar

- Solución ejercicios previos (1, 2, 3): **pr07_previo.doc**. Se debe subir a Aula Virtual antes del **12 de Mayo a las 23:55**.
- Los ficheros de cabecera e implementación de la clase *Alumno* (**alumno.cpp**, **alumno.h**).
- Los ficheros de cabecera e implementación de la clase *Cola* (**cola.cpp**, **cola.h**).
- El programa principal que se usan las clases anteriores: **pr07.cpp**.

Introducción

En esta práctica se va a crear una clase para manejar una estructura de datos del tipo Cola con gestión dinámica y se comparará la implementación con la que proporciona la biblioteca STL.

La clase Cola maneja elementos del tipo Alumno (se incluye la definición de la clase en el fichero alumno.h). Esta es una clase muy sencilla que sólo contiene dos atributos, un código que no puede repetirse (int) y el nombre del alumno (string).

La clase Cola deberá implementar, como mínimo, los métodos descritos a continuación:

- Cola() : Constructor de la clase
- ~Cola() : Destructor de la clase
- bool Encolar(Alumno a) : Añade un alumno a la cola. Si todo va bien devuelve true, si falla devuelve false.
- bool DesEncolar(void) : Devuelve true si puede extraer un elemento false si no quedan elementos. Desencolar elimina el primer elemento de la cola.
- int Elementos(void) : Devuelve el número de elementos que hay en la cola
- void Mostrar(void) : Muestra por la consola los elementos de la cola. Si la cola está vacía muestra el mensaje "Cola vacía".
- int Vaciar(void) : Vacía la cola eliminando todos los elementos. Devuelve el número de elementos eliminados. Esta función debe ser pública.
- bool VerPrimero(Alumno &a) : Devuelve en 'a' los datos del elemento que está el primero de la cola. No extrae el elemento de la cola. Si la cola está vacía devuelve false. True en caso contrario.



- `bool VerUltimo(Alumno &a)` : Devuelve en 'a' los datos del elemento que está el último de la cola. No extrae el elemento de la cola. Si la cola está vacía devuelve false. True en caso contrario.
- `bool VerPosicion(int pos, Alumno &a)` : Devuelve en 'a' los datos del elemento que está en la posición 'pos', a partir del comienzo de la cola. El parámetro pos será 0 para el primer elemento y N-1 para el último (N=elementos en la cola). Si hay un elemento en esa posición devolverá true, y false si el valor pos es mayor que el número de elementos en la cola.
- `Operador=` : Operador asignación permitirá asignar dos colas diferentes.

Desarrollo de la práctica

Ejercicio 4: Implementar los métodos de la clase Alumno (alumno.cpp) en base al fichero alumno.h suministrado con el enunciado.

Ejercicio 5: Implementar la clase Cola con gestión dinámica (cola.cpp) que permita almacenar objetos de la clase usando la clase Alumno en base al prototipo de clase suministrado (cola.h).

Ejercicio 6: Crear un programa principal que declare objetos de la clase Cola y los use comprobando que se pueden realizar todas las operaciones definidas. El fichero pr07.cpp suministrado contiene la estructura básica e indica las operaciones mínimas que deben comprobarse. La salida por consola debe permitir determinar si el funcionamiento ha sido correcto o no.

Ejercicio 7: Dentro del programa principal anterior, repetir las mismas operaciones pero sobre objetos de tipo cola de la biblioteca STL. Hacer las mismas operaciones y comprobar que se obtienen los mismos resultados.