



PRÁCTICA Nº 4: 1 sesión

(del 22 al 26 de Abril de 2002)

Utilización del tipo abstracto de dato Cola. Simulación de colas de despegue de aviones de un aeropuerto.

0. OBJETIVO

El objetivo de esta práctica es la implementación del tipo abstracto de datos cola y su utilización en la simulación de colas de despegue de aviones de un aeropuerto.

1. TIPO ABSTRACTO DE DATOS COLA

Las colas son estructuras de datos lineales, denominadas así, porque consisten en una secuencia de elementos $a_0, a_1, \dots, a_{n-1}$ dispuestos en una dimensión. Dependiendo del tipo de elementos que queramos almacenar, podremos tener distintos tipos de colas. Así, tendremos colas de enteros, de caracteres, etc, pudiendo tener también colas de otros TAD's. En el caso de la práctica actual se almacenará en la cola una identificación del avión o vuelo.

Dentro de las estructuras lineales, las colas también se conocen como estructuras FIFO (First In, First Out). El nombre FIFO hace referencia al modo en que se introducen y se accede a los elementos, ya que todas las inserciones se realizan por un lugar (fin de cola) y supresiones tienen lugar en el otro extremo (principio de cola).

2. IMPLEMENTACIÓN DE LA CLASE COLA

Deseamos crear una clase **Cola** que represente al TAD Cola de Aviones. El alumno deberá hacer 2 implementaciones distintas tal y como se ha visto en clase: una implementación estática y otra dinámica.

En este caso hemos supuesto que el identificador del vuelo será un string, por lo que solamente almacenaremos en la cola ese identificador (id). El alumno podrá considerar modificar o ampliar la cantidad de datos almacenados en cada elemento de la cola. En el caso de representación estática, se preverá un máximo de 100 vuelos (elementos guardados en la cola).

En cualquier caso, las operaciones de uso de la cola serán las definidas por el TAD y estarán presentes en la parte pública de la clase:

Cola (void);

Constructor de la clase. El inicio y final de cola están en el mismo sitio y la cola está vacía.

bool ColaVacía (void);

Comprobación de si hay o no elementos en la cola.

bool PrimeroCola (string & id);

Proporciona el primer elemento de la cola y devuelve false. Si la cola está vacía devuelve true.

bool Encolar (string id);

Inserta en la cola el vuelo id y devolverá false (no se ha producido ningún error.). Si no cabe en la cola, devolverá true (se ha producido un error.)

bool Desencolar (string & id);

Si hay elementos en la cola, pondrá el valor de inicio de cola en id, lo sacará de la cola y devolverá false. Si no hay elementos en la cola devolverá true.

En el caso de que el alumno lo considere necesario se podrían también implementar otros métodos como:

LongitudCola
ColaLlena



3. SIMULACIÓN DE COLAS DE DESPEGUE DE UN AEROPUERTO

Vamos a tratar de simular la colas de despegue de un aeropuerto. Para ello se deberá implementar una clase cola donde los datos guardados dentro de la cola será como mínimo un identificador del vuelo.

Los identificadores válidos están formados por 2 caracteres iniciales con letras mayúsculas, que corresponden a la compañía y 4 caracteres numéricos que corresponden al número de vuelo. Ejemplos: IB0012, LH8892, AF1251.

El alumno desarrollará un programa que simulará las colas de despegue del aeropuerto usando la clase cola previamente definida.

En este programa se va a simular el flujo de salida de aviones en un aeropuerto. El funcionamiento de las salidas es el siguiente:

- Entre un despegue y el siguiente se debe esperar un tiempo determinado: 10 segundos
- Cuando un avión esté en la cola de despegue y sea el que tiene el turno para despegar, se recurrirá a una función que identifique si el avión tiene problemas para despegar o no. Para ello esta función de “problema de despegue” deberá considerar aleatoriamente que el 10% de los aviones que van a despegar sufren retrasos. Si la función “problema de despegue” indica que el avión concreto que se encuentra en el inicio de la cola está entre los del 10% que tienen problemas, no despegará. Se sacará de la cola y se pondrá en el final de la cola para poder volver a intentar despegar cuando le toque otra vez.
- Si al introducir un avión en la cola, hace más de diez segundos que despegó el último, éste despegará directamente sin hacerle esperar, sin esperar el tiempo restante hasta los diez segundos que deben transcurrir entre un vuelo y el siguiente.

Monitorización de vuelos por pantalla:

El programa podrá estar en cualquiera de los 3 siguientes estados:

- 1.- Introducir vuelos nuevos en la cola de despegue
- 2.- Ver estado de vuelos
- 3.- Pasar un segundo

Que se podrán seleccionar desde un menú.

Para introducir un nuevo vuelo en la cola de salida, se pasará al modo 1 de trabajo en el que se permitirán introducir vuelos pedirá el identificador de vuelo por teclado, y si éste es válido se introducirá en la cola de salida. Cuando sea introducido en la cola de salida y se pase al modo 2 se mostrará por pantalla el siguiente mensaje:

“Vuelo *iden_vuelo* ->salida prevista en *min:secs*.”

Ejemplo: Vuelo LH0889 -> Salida prevista en 03:15

Cuando un vuelo consiga despegar se mostrará el siguiente mensaje: “*ident_vuelo* despegando”

Para simular el 10% de vuelos que tiene problemas de despegue y no consiguen despegar cuando les toca, utilizaremos una función que de forma aleatoria nos diga si el vuelo despegará o no. Así cuando a un vuelo le toque despegar, se llamará a esta función, que devuelve “true” en caso de que tenga problema de despegue y “false” en caso de que pueda despegar:

- Si devuelve false. El vuelo despegará, con lo cual, el vuelo será extraído de la cola de despegue”, y se mostrará el mensaje indicado anteriormente en pantalla.



- Si devuelve true. El vuelo no despegará, con lo cual, se extraerá del inicio de la cola y se insertará al final, dando el mensaje:

“vuelo *ident_vuelo* con problemas, nueva previsión de salida en *min.:secs.*”

Además se insertará el vuelo en otra cola en la se guardarán los vuelos que han tenido problemas en el despegue.

El paso del tiempo lo supondremos por selección de menú. Cada vez que el usuario seleccione una opción del menú, supondremos que pasa un segundo.

El programa finaliza cuando el usuario pulse una tecla de fin, pero antes de finalizar mostrará por pantalla la siguiente información:

- Número de vuelos que consiguieron despegar,
- Volcará por pantalla la cola de vuelos que queden por despegar, (si quedan)
- Volcará por pantalla la cola de vuelos que tuvieron problemas en el despegue, (si los hubo)

4. TAREAS QUE REALIZAR

En esta práctica tenemos que realizar **un solo programa** de simulación de las colas del aeropuerto, además de las clase cola correspondiente **ColaE.h** y **ColaE.cpp** (y **ColaD.h** y **ColaD.cpp**)

5. REQUISITOS

Recordar que para poder acceder a la sesión de prácticas correspondiente es necesario entregar un pequeño esquema (documentación del programa) de lo que se va a realizar en la práctica. Si no se presenta la documentación ANTES de acceder al laboratorio, no se permitirá la entrada a la sesión de prácticas.

6. ENTREGA

La entrega de esta tercera práctica debe incluir los siguientes archivos:

- 1.- **simula##.cpp**: con la función principal y todas las funciones que éste requiera.
- 2.- **ColaE##.cpp**, **ColaE##.h**: Implementación estática de la clase ‘Cola’
- 3.- **ColaD##.cpp**, **ColaD##.h**: Implementación dinámica de la clase ‘Cola’

La entrega puede realizarse por e-mail a la dirección del profesor encargado del grupo o en disquette al iniciar la siguiente sesión de prácticas, pero siempre ANTES de empezar la siguiente sesión de prácticas.

Cualquier práctica entregada fuera de plazo no será admitida para su corrección.

Fecha límite de entrega de prácticas: Sesiones del 29 de abril al 3 de mayo