

Tecnologías de la Información



Universidad de Huelva

Estructuras de Datos I

Práctica 1

Introducción

Con esta práctica se pretende afianzar los conceptos relativos al uso de **ficheros**, así como hacer uso de **TADs lineales**, concretamente la clase *Cola*.

Para ello, se empleará un **fichero binario** que contendrá información relativa a los **pacientes** de una **clínica**. Dicho fichero tendrá la siguiente estructura:

Nº de pacientes	Paciente 1	Paciente 2	...	Paciente n
-----------------	------------	------------	-----	--------------

Al inicio del fichero se guarda el número de pacientes almacenados en ese momento en el fichero, siendo por tanto un dato de tipo entero. A continuación van los datos de los n pacientes, cada uno con la siguiente estructura:

```
struct Paciente {  
    int id;  
    cadena nombre;  
    cadena apellidos;  
    int edad;  
};
```

Clase Clínica

La clase *Clínica* será usada para gestionar dicho fichero, y es la siguiente:

```
class Clinica {  
    private:  
        FILE *fichero;  
        int numPacientes;  
        void posicionar(int p);  
    public:  
        Clinica(char nombreFichero[]);  
        int getNumPacientes()  
            { return numPacientes; };  
        void consultar(char inicial);  
        void insertar(Paciente p);  
        void modificar(int id);  
        void eliminar(int id);  
        void simularOperaciones();  
};
```

Un objeto *Clinica* está vinculado a un fichero, siendo el constructor el método en el que se debe abrir tal fichero, según la cadena pasada como parámetro (con el nombre del fichero físico). Si tal fichero ya existiera se leería el valor de **numPacientes**, y si no, se procedería a crearlo (asignando y guardando el valor de 0 para el número de pacientes).

El método privado **posicionar** se encarga de situarse en el fichero en el inicio del paciente *p* pasado como parámetro.

El método **consultar** se encarga de mostrar por pantalla los datos de todos los pacientes cuyos apellidos comiencen por el carácter pasado por parámetro.

El método **insertar** realiza la inserción de los datos de un nuevo paciente, a continuación del último de los ya existentes.

El método **modificar** se encarga de actualizar los datos del paciente cuyo identificador se pasa por parámetro, pidiendo dichos datos previamente. Si el identificador no estuviera en el fichero de pacientes, se mostraría un mensaje al respecto.

El método **eliminar** realiza la eliminación del paciente cuyo identificador se pasa por parámetro. Si el identificador no estuviera en el fichero de pacientes, se mostraría un mensaje al respecto. Para eliminar un paciente del fichero, se desplazan una posición a la izquierda todos los pacientes existentes a continuación del que se elimina (para no dejar huecos).

El método **simularOperaciones** se encarga de realizar una simulación del paso de los pacientes por los quirófanos de la clínica. Su detalle se explica más adelante en este documento.

Programa principal

El programa a desarrollar deberá comenzar creando un objeto *Clinica* a partir del fichero físico "*clinica*".

Luego mostrará el siguiente menú:

```
CLINICA  'Los Milagros'
-----
Pacientes: 5

1. Consultar pacientes
2. Introducir un nuevo paciente
3. Modificar un paciente
4. Eliminar un paciente
5. Simular operaciones
6. Salir
```

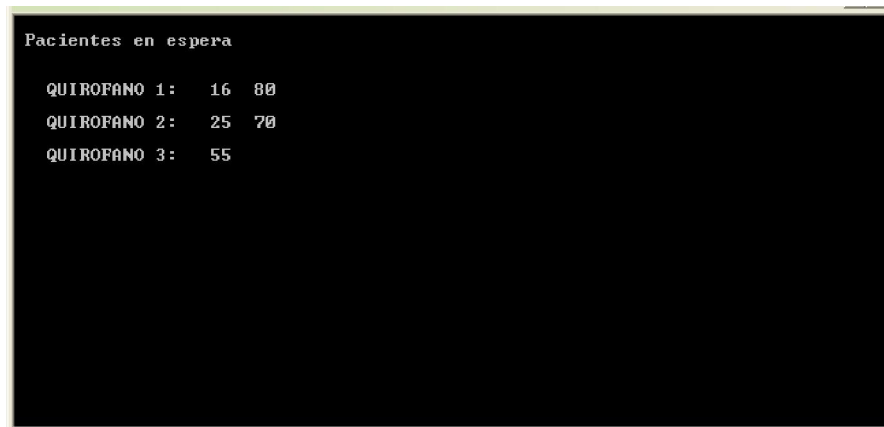
Las opciones del menú se corresponden con las llamadas a los métodos de la clase *Clinica*.

Simulación de las operaciones

Para realizar el método ***simularOperaciones*** se supondrá que la clínica dispone de 3 quirófanos, procediéndose a mostrar por pantalla las colas de espera para cada uno de ellos, mientras haya algún paciente en espera.

Es decir, se mostrarán los identificadores de los pacientes en espera para entrar a cada quirófano, insertándose al principio aleatoriamente cada uno de los pacientes existentes en la clínica.

Como ejemplo, en un instante dado la salida por pantalla podría ser así:



```
Pacientes en espera
QUIROFANO 1: 16 80
QUIROFANO 2: 25 70
QUIROFANO 3: 55
```

Se proporciona una función ***azar*** que devuelve un número aleatorio en el intervalo $[1, n]$, así como una función ***pacienteOperado*** que devuelve si ha terminado la operación de un paciente cualquiera.

En base a esto, el método ***simularOperaciones*** deberá inicialmente utilizar un array de 3 objetos *cola*, para asignar aleatoriamente a una cola cualquiera cada uno de los identificadores de los pacientes del fichero. Y luego, hasta que todos los pacientes hayan entrado en su quirófano, se mostrarían los identificadores en cada cola, actualizando cada una de ellas si el primer paciente ha entrado en quirófano (usando la función ***pacienteOperado***).

Ficheros proporcionados

Se proporcionan los ficheros *clinica*, *Pl.cpp*, *Clinica.h*, *Clinica.cpp*, *TADCola.h* y *TADCola.cpp*.

El fichero *clinica* contiene 5 pacientes de ejemplo, con los siguientes datos:

Identificador:	25
Nombre:	MARISA
Apellidos:	DOMINGUEZ
Edad:	33
Identificador:	70
Nombre:	FRANCISCO
Apellidos:	DOMINGUEZ
Edad:	41
Identificador:	80
Nombre:	MARIA
Apellidos:	PEREZ
Edad:	71
Identificador:	16
Nombre:	JUAN
Apellidos:	FERNANDEZ
Edad:	28
Identificador:	55
Nombre:	IRENE
Apellidos:	GUTIERREZ
Edad:	62