Tecnologías de la Información

Universidad de Huelva

PRÁCTICA 2

TAD Lineales Dinámicos

VERSIÓN 1

Enunciado

Ante la situación excepcional provocada por el COVID-19 se plantea el desarrollo de una aplicación para la gestión de la cola de urgencias de una Clínica que permita la atención a pacientes positivos en dicho virus, atención cuanto más temprana mejor, teniendo en cuenta los factores de riesgo de los pacientes afectados, entre los que destaca la edad.

Para ello aplicaremos, en esta primera versión de la práctica, los conceptos estudiados de TAD lineales y su implementación usando tablas dinámicas, dejando para una versión posterior la aplicación de las estructuras dinámicas mediante nodos enlazados .

Se empleará un fichero binario para almacenar la información de los pacientes que llegan a las urgencias de la Clínica que 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;
};
```

Siendo el tipo cadena definido como, **typedef char** cadena[TAM_CADENA]; y TAM_CADENA una constante de valor 30, **const** int TAM_CADENA = 30;

Clase *ColaPri*

La clase *Co*laPri será usada para gestionar la información de los pacientes en urgencias que se almacenará en una Cola con Prioridades, implementada como una Lista de N colas, siguiendo la implementación vista en el último ejercicio de la relación de problemas del tema 3, adaptándola a los datos que vamos a utilizar y modificando la implementación vista con tablas estáticas a una implementación con tablas dinámicas.

Se elige la implementación vista en teoría mediante una **lista de n colas**:

- Se utiliza una cola separada para cada nivel de prioridad, que habrá que implementar de manera dinámica según lo visto en el tema 4.

- Para agrupar todas las colas, se utiliza un array.
- Cada celda del array representa un nivel de prioridad y contiene a la cola correspondiente.

Para la asignación de prioridades tendremos en cuenta la información de los expertos, los datos respecto al riesgo estadístico que supone el virus. El estudio epidemiológico más extenso hasta la fecha ha determinado las tasas de morbilidad y mortalidad en cada grupo de población, **por edades:**

Para la población en general, **el 80,9% de los casos son leves, mientras que el 13,8% son graves y solo el 4,7% son críticos.** La peor parte se la llevan los mayores de 80 años, en los que el virus alcanza un 14,8% de mortalidad. Estas son las tasas de mortalidad observadas para cada grupo de edad, por lo que la prioridad irá en aumento al aumentar la edad, y tendremos en cuenta los siguientes 9 rangos, para establecer los 9 niveles de prioridad, de menor a mayor:

Prioridad	Rando de Edad	Tasa de mortalidad
1	0-9 años	-
2	10-19	0,2%
3	20-29	0,2%
4	30-39	0,2%
5	40-49	0,4%
6	50-59	1,3%
7	60-69	3,6%
8	70-79	8%
9	80 o más	14,8%

El fichero de definición de la clase ColaPri será:

```
//Fichero TADColaPri.h
#include "tadcola.h"
#define MAXcolas 9
class ColaPri {
      cola tabla[MAXcolas];
public:
ColaPri(); //Constructor: crea un objeto ColaPri vacío
~ColaPri(); //Destructor: libera la memoria dinámica del objeto ColaPri
void insertarColaPri(int i, Paciente p);
/*Insertará el paciente p en la cola de prioridad i de la ColaPri*/
void sacarColaPri();
/*Elimina el paciente de mayor prioridad. Si hay varios con la misma
prioridad saldrá aquel que lleve más tiempo en el objeto ColaPri*/
Paciente consultarColaPri();
/*Devuelve el paciente de mayor prioridad almacenado en el objeto ColaPri
original sin quitarlo. Si hay varios con la misma prioridad saldrá aquel
que lleve más tiempo en el objeto ColaPri*/
bool esvaciaColaPri();//Devuelve si el objeto ColaPri está vacío
int longitudColaPri();//Devuelve la longitud del objeto ColaPri
bool cargarfichero (cadena fich);
/*Cargará los datos almacenados en el fichero cuyo nombre se pasa como
parámetro en el objeto ColaPri. Devuelve true si no hay error en la carga*/
bool guardarfichero (cadena fich);
/*Guardará los datos del objeto ColaPri en el fichero cuyo nombre se pase
como parámetro. Devuelve true si no hay error al guardar el objeto*/
};
```

Para gestionar las urgencias haciendo uso de la cola de prioridad anterior, se pretende diseñar una aplicación con el siguiente menú principal:

```
----- Menú Principal -----

1- Cargar datos desde fichero
2- Admisión de un nuevo Paciente
3- Atender un Paciente
```

```
4- Listar Pacientes esperando para atender
5- Listar Pacientes según prioridad
6- Salir
```

Opción 1. Permitirá cargar los datos desde el fichero cuyo nombre se solicite desde teclado.

Opción 2. Pedirá los datos del paciente y lo insertará en la cola de espera para ser atendido según su prioridad.

Opción 3. Para atender un paciente saldrá de la cola el paciente de mayor prioridad que más tiempo lleve esperando.

Opción 4. Mostrará por pantalla todos los pacientes que se encuentren esperando para ser atendidos en urgencias.

Opción 5. Pedirá un valor de prioridad y mostrará los pacientes esperando con esa prioridad.

Opción 6. Antes de dejar la aplicación, debemos volcar toda la información de los pacientes en espera en el fichero cuyo nombre se solicite desde teclado.

Se proporciona el fichero ***urgencias.dat***, que contiene 5 pacientes de ejemplo con los siguientes datos:

```
Identificador: 16
Nombre:        JUAN
Apellidos:     FERNANDEZ
Edad:          28

Identificador: 80
Nombre:        MARIA
Apellidos:     PEREZ
Edad:          71

Identificador: 25
Nombre:        MARISA
Apellidos:     DOMINGUEZ
Edad:          33

Identificador: 70
Nombre:        FRANCISCO
Apellidos:     DOMINGUEZ
Edad:          41

Identificador: 55
Nombre:        IRENE
Apellidos:     GUTIERREZ
Edad:          62
```

Para la elaboración de la práctica el alumno podrá utilizar dicho fichero como datos iniciales o partir de que no hay ningún paciente esperando para ser atendido en las urgencias.

Es obligatorio el uso de diseño modular.

Fecha de finalización

La práctica deberá estar finalizada antes de la realización de la segunda prueba de modificación de prácticas, fecha que será comunicada por los profesores de la asignatura y publicada en el portal de apoyo de la docencia semipresencial de la UHU (aulasvirtuales.uhu.es).