Práctica 4: Colas

Estructuras de Datos– 1º GII

El comportamiento de un TAD Cola viene determinado por su especificación formal, que se muestra a continuación:

ESPECIFICACION Colas
PARAMETROS GENERICOS
TipoElemento
FIN PARAMETROS GENERICOS
TIPOS TipoCola
OPERACIONES
(* CONSTRUCTORAS GENERADORAS *)
CrearColaVacia: → TipoCola
Insertar: TipoElemento x TipoCola → TipoCola

(* OBSERVADORAS SELECTORAS *)
PARCIAL PrimeroCola : TipoCola → TipoElemento
PARCIAL Eliminar : TipoCola → TipoCola

(* OBSERVADORAS NO SELECTORAS *)
EsColaVacia : TipoCola → Booleano
IgualCola: TipoCola x TipoCola → Booleano

(* CONSTRUCTORAS NO GENERADORAS *)
CopiarCola: TipoCola → TipoCola

VARIABLES
cola, cola2 : TipoCola;
elem, elem2 : TipoElemento;
ECUACIONES DE DEFINITUD
DEF(PrimeroCola(Insertar(elem, cola)))
DEF(Eliminar(Insertar(elem, cola)))
ECUACIONES
(* Observadoras selectoras *)
PrimeroCola(Insertar(elem, cola)) = SI EsColaVacia(cola) →
elem
| PrimeroCola(cola)
(* Observadoras no selectoras *)
EsColaVacia (CrearColaVacia) = CIERTO
EsColaVacia (Insertar(elem, cola)) = FALSO

IgualCola(CrearColaVacia, cola2) = EsColaVacia(cola2)
IgualCola(Insertar (elem, cola), CrearColaVacia)= FALSO
IgualCola(Insertar (elem, cola), Insertar (elem2, cola2)) = (elem = elem2) Y IgualCola(cola, cola2)

(* Constructoras no generadoras *)
Eliminar (Insertar(elem,cola)) = SI EsColaVacia(cola) →
cola
| Insertar(e, Eliminar(cola))
CopiarCola (CrearColaVacia) = CrearColaVacia
CopiarCola (Insertar(elem,cola)) = Insertar (elem, CopiarCola(cola))
FIN ESPECIFICACION

Para manejar la implementación de colas, en esta práctica, se pide

a) Implementación de la estructura de datos `Cola` que considere más adecuada, justificando su elección.
b) Realizar la implementación completa de un TAD que simule el funcionamiento de un hospital (`tHospital`) y que disponga de la siguiente información: estructura de datos donde los pacientes puedan ser atendidos según el orden de llegada, el número de camas disponibles en el hospital y un almacén de datos donde se guarda la información de cinco médicos (nombre y apellidos) en plantilla que serán los encargados de dar salida a los pacientes. Las operaciones que debe contemplar el TAD son:

- `crearHospitalVacio`: crea una variable de tipo `tHospital`, con una estructura vacía para dar cabida a los pacientes. Se rellenará la información de los médicos en plantilla (5) y el número de camas disponibles (75).
- `insertarEnHospital`: dada una variable de tipo `tHospital` y un nuevo paciente, lo inserta en el lugar adecuado, teniendo en cuenta que consumirá una cama.
- `consultarMedico`: muestra por pantalla la asignación de un paciente con un médico, teniendo en cuenta que para el siguiente paciente lo tratará el siguiente médico y así sucesivamente hasta empezar de nuevo el proceso de asignación con el primer médico. Nótese que este subprograma debe eliminar el paciente de la correspondiente estructura de datos.
- `consultarPrimero`: dada una variable de tipo `tHospital`, devuelve la información del paciente que será atendido en primer lugar.
- `quitarPrimero`: dada una variable de tipo `tHospital`, elimina la información del paciente que se atendería en primer lugar, actualizando la información de camas y médicos.
- `esHospitalVacio`: dada una variable de tipo `tHospital`, nos indica si la estructura está vacía o no.

Nota 1: Considere para el tipo tPaciente la siguiente estructura:

```
typedef struct paciente{
        int exp;
        char nombre[20];
        int edad;
        char dolecia[50];
}tPaciente;
```

Nota 2: Se considera necesario un nivel elevado de encapsulación y abstracción.

Nota 3: Se hará uso de las normas de estilo dictadas en clase (cabecera del fichero, interfaz de la unidad con precondiciones, postcondiciones, excepciones, implementaciones con el análisis de complejidad de cada operación, nombres coherentes de variables y operaciones,...)

Plantilla de cabecera del fichero:

```
{*****************************************************************************
*                                                                            *
*       Módulo:                                                              *
*       Tipo:  Programa()     Interfaz-Implementación TAD ()       Otros()     *
*       Autor/es:                                                            *
*       Fecha de actualización:                                              *
*       Descripción:                                                         *
*                                                                            *
*****************************************************************************}
```