Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI-2692 - Laboratorio de Algoritmos y Estructuras de Datos II
Trimestre Enero-Marzo 2020

TAD Cola y TAD Pila

1. Introducción

Se desea que realice la implementación concreta del TAD Cola y del TAD Pila. En la representación de la especificación de ambos TADs, se va hacer uso de la estructura **secuencia**. Una **secuencia** se define como una colección de elementos, en la cual el orden y la multiplicidad son importantes [2]. Para detalles de la definición, características y operaciones de las **secuencias**, ver [2]. La Figura 1 se muestra las operaciones de las **secuencias**. La Figura 2 muestra ejemplos de las operaciones de las **secuencias**.

$\#q$	The number of elements in q.
$e{:}q$	The sequence whose first element is e, and whose subsequent elements are those of q. We have $(e{:}q)[0] = e$ and for $0 < i \leq \#q$, $(e{:}q)[i] = q[i-1]$.
$q1 +\!\!+ q2$	The sequence that begins with $q1$ and carries on with $q2$. We have $(q1 +\!\!+ q2)[i]$ equals $q1[i]$, if $0 \leq i < \#q1$, and equals $q2[i - \#q1]$ if $0 \leq i - \#q1 < \#q2$.
$\mathsf{hd}\, q$	The first element of q, provided q is not empty. We have $\mathsf{hd}(\langle e\rangle +\!\!+ q) = e$.
$\mathsf{tl}\, q$	The second and subsequent elements of q, provided q is not empty. We have $\mathsf{tl}(\langle e\rangle +\!\!+ q) = q$.
$\mathsf{fr}\, q$	All but the last element of q, provided q is not empty. We have $\mathsf{fr}(q +\!\!+ \langle e\rangle) = q$.
$\mathsf{lt}\, q$	The last element of q, provided q is not empty. We have $\mathsf{lt}(q +\!\!+ \langle e\rangle) = e$.

Figura 1: Definiciones de las operaciones de la **secuencia**. Fuente [2]

Una de las estructuras que se va a usar para la implementación del TAD Cola y del TAD Pila, es la **lista doblemente enlazada**. La Figura3 muestra la representación de la **secuencia** $\langle 9, 16, 4, 1\rangle$ como una **lista doblemente enlazada**.

2. Actividades a realizar

La Figura 4 muestra la especificación del TAD Cola. Debe realizar la implementación concreta del TAD Cola en un módulo llamado `ColaArreglo.py`. Este módulo contiene la

$$
\begin{aligned}
\#\langle\rangle &= 0 \\
\langle 1,2\rangle +\!\!+ \langle\rangle &= \langle 1,2\rangle \\
\langle\rangle +\!\!+ \langle 1,2\rangle &= \langle 1,2\rangle \\
1{:}\langle 2,3\rangle &= \langle 1,2,3\rangle \\
\langle 1\rangle +\!\!+ \langle 2,3\rangle &= \langle 1,2,3\rangle \\
\mathsf{hd}\langle 1,2,3\rangle &= 1 \\
\mathsf{tl}\langle 1,2,3\rangle &= \langle 2,3\rangle \\
\mathsf{fr}\langle 1,2,3\rangle &= \langle 1,2\rangle \\
\mathsf{lt}\langle 1,2,3\rangle &= 3
\end{aligned}
$$

Figura 2: Ejemplos de las operaciones de la **secuencia**. Fuente [2]

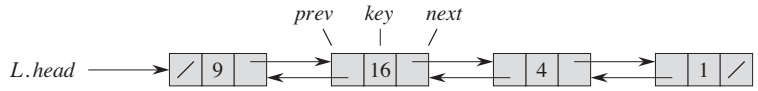

Figura 3: Ejemplo de una lista enlazada, con los elementos 9, 16, 4 y 1. Fuente [1]

clase `ColaArreglo`, en donde la **secuencia** de elementos de la representación del TAD Cola, va a ser implementada como un arreglo. También debe realizar implementación del TAD Cola, en un módulo llamado `ColaLista.py`. Este módulo contiene la clase `ColaLista` en donde la **secuencia** de la representación del TAD Cola, debe ser implementada como una **lista doblemente enlazada**.

También se quiere hacer dos implementaciones concretas del TAD Pila. La Figura 5 muestra la especificación del TAD Pila. Debe realizar la implementación concreta del TAD Pila en un módulo llamado `PilaArreglo.py`. Este módulo contiene la clase `PilaArreglo`, en donde la **secuencia** de elementos de la representación del TAD Pila, va a ser implementada como un arreglo. También debe realizar implementación del TAD Pila, en un módulo llamado `PilaLista.py`. Este módulo contiene la clase `PilaLista` en donde la **secuencia** de la representación del TAD Pila, debe ser implementada como una **lista doblemente enlazada**.

3. Condiciones de entrega

La versión final del código del laboratorio y el informe deben estar contenidos en un archivo comprimido, con formato *tar.xz*, llamado *LabSem7_X.tar.xz*, donde X es el número de carné del estudiante. La entrega del archivo *LabSem7_X.tar.xz*, debe hacerse al profesor del laboratorio por email, antes de las 9:00 pm del día domingo 23 de febrero de 2020.

Especificación $\mathbb{A}$ de TAD *Cola* (T)

Modelo de Representación

const $MAX : \text{int}$
var *contenido* $: \text{seq}\, T$

Invariante de Representación

$MAX > 0 \ \ \wedge \ \ \#\, contenido \leqslant MAX$

Operaciones

proc *crear* (**in** $m : \text{int}$; **out** $c :$ *Cola*)
{ **Pre** : $m > 0$ }
{ **Post** : $c.MAX = m \ \ \wedge \ \ c.contenido = \langle\rangle$ }

proc *encolar* (**in-out** $c :$ *Cola* ; **in** $x : T$)
{ **Pre** : $\#\, c.contenido < c.MAX$ }
{ **Post** : $c.contenido = c_0.contenido +\!\!+ \langle x \rangle$ }

proc *desencolar* (**in-out** $c :$ *Cola*)
{ **Pre** : $c.contenido \neq \langle\rangle$ }
{ **Post** : $c.contenido = \text{tl}\ c_0.contenido$ }

proc *primero* (**in** $c :$ *Cola* ; **out** $x : T$)
{ **Pre** : $c.contenido \neq \langle\rangle$ }
{ **Post** : $x = \text{hd}\ c.contenido$ }

proc *vacia* (**in** $c :$ *Cola* ; **out** $v : \text{boolean}$)
{ **Pre** : true }
{ **Post** : $v \equiv (c.contenido = \langle\rangle)$ }

Fin TAD

Figura 4: Especificación del TAD Cola. Fuente [3]

Referencias

[1] Cormen, T., Leirserson, C., Rivest, R., and Stein, C. *Introduction to Algorithms*, 3ra ed. McGraw Hill, 2009.

[2] Morgan, C. *Programming from Specifications*. Prentice Hall, 1998.

[3] Ravelo, J. Especificacion e implementacion de tipos abstractos de datos. `https://ldc.usb.ve/~jravelo/docencia/algoritmos/material/tads.pdf`, 2009.

Especificación TAD *Pila* (T)

REPRESENTACIÓN

const $MAX : int$
var $sp : seq(T)$

INVARIANTE

$\#sp \leq MAX$

OPERACIONES

proc *vacia* (**in** $tammax : int$; **out** $p : Pila$)
{ Pre: $tammax > 0$ }
{ Post: $p.sp = \langle\rangle \wedge p.MAX = tammax$ }

proc *esVacia* (**in** $p : Pila$; **out** $vacia : boolean$)
{ Pre: $true$ }
{ Post: $vacia \equiv (p.sp = \langle\rangle)$ }

proc *empilar* (**in out** $p : Pila$; **in** $x : T$)
{ Pre: $\#p.sp < p.MAX$ }
{ Post: $p.sp = p'.sp +\!\!+ \langle x \rangle$ }

proc *desempilar* (**in out** $p : Pila$)
{ Pre: $p.sp \neq \langle\rangle$ }
{ Post: $p.sp = fr(p'.sp)$ }

proc *tope* (**in** $p : Pila$; **out** $y : T$)
{ Pre: $p.sp \neq \langle\rangle$ }
{ Post: $y = lt(p.sp)$ }

end TAD *Pila*(T)

Figura 5: Especificación del TAD Pila.