

TAD Cola de Prioridad
Cola de Prioridad = $C = \{\emptyset, N_1, N_2, N_3, \dots, N_n\}$
Invariante: $\{N \in Clase\ nodo = \{Indice = entero, Valor = Objeto\} \wedge \forall i, j \in \mathbb{N} \mid N_i, N_j \in C. N_i \geq N_j \vee N_j \geq N_i\}$
Operaciones primitivas: <ul style="list-style-type: none"> <li>• CrearColadePrioridad: <math>\rightarrow</math> Cola de Prioridad</li> <li>• ObtenerMáximo: <math>\rightarrow</math> ColadePrioridad x Nodo <math>\rightarrow</math> Nodo</li> <li>• ExtraerMáximo: <math>\rightarrow</math> Cola de Prioridad x Nodo <math>\rightarrow</math> Nodo</li> <li>• InsertarNodo: <math>\rightarrow</math> Cola de Prioridad x Nodo <math>\rightarrow</math> Cola de Prioridad</li> <li>• IncrementarLlave <math>\rightarrow</math> Cola de Prioridad x Entero <math>\rightarrow</math> Cola de Prioridad</li> <li>• EstaVacío <math>\rightarrow</math> Cola de Prioridad <math>\rightarrow</math> Boolean</li> </ul>

Operaciones principales.

Operaciones constructoras

CrearColadePrioridad() “Crea una cola de prioridad vacía” {Pre: TRUE} {Post: $C = \emptyset$ }
---

Operaciones modificadoras

ObtenerMáximo() “Retorna el nodo que está al inicio de la cola” {Pre: $\exists C \wedge  C  \geq 1$ } {Post: $ C  = 0$ , retorna null, sino Nodo}
--

ExtraerMáximo() “Retorna el nodo que está al inicio y lo elimina” {Pre: $\exists C \wedge  C  \geq 1. C = \{\emptyset, N_1, N_2, N_3, \dots, N_n\}$ } {Post: Si $ C  = 0$ , lanza excepción, sino $Nodo \wedge C = \{\emptyset, N_1, N_2, N_3, \dots, N_{n-1}\}$ }
---

InsertarNodo(i,k) “Inserta un elemento a la lista y crea un nodo donde i=índice como entero y k=Objeto” {Pre: $\exists C$ } {Post: Si añade TRUE $\wedge C = \{\emptyset, N_1, N_2, N_3, \dots, N_{n+1}\}$ , sino FALSE}
---

IncrementarLlave(i,k) “Incrementa la prioridad de un elemento en la cola de la cola” {Pre: $\exists C \wedge  C  \geq 1$ }
--

{ Post: Si  $|C| = 0 \vee$  si  $i \in \mathbb{N} < 0 \vee i <$  La prioridad del elemento, lanza excepción, sino  $C[k] = i$   
donde  $k \in \mathbb{N} \wedge k \geq 0 \vee k \leq |C| - 1$ }

EstaVacío()

“Determina si el conjunto está vacío o no”

{Pre:  $\exists C$ }

{Post: Si  $|C| = 0$  TRUE, sino FALSE”