

## Práctica 1

(miércoles 2 de mayo)

1. Una Pila es un Tipo Abstracto de Datos (TAD) que permite almacenar una secuencia de elementos. Entre los elementos almacenados existe una noción de orden, y los elementos son insertados y eliminados en el mismo extremo de la secuencia; esto es, el elemento a eliminar corresponde al último elemento insertado.

Las operaciones que debe ofrecer este TAD son: (i) **crear**, operación que dado un tamaño máximo, crea una Pila vacía de dicho tamaño; (ii) **empilar**, operación que dado una Pila y un elemento, agrega ese elemento en el tope de la Pila; (iii) **desempilar**, operación que dada una Pila, elimina de ésta el elemento ubicado en su tope; (iv) **tope**, operación que dada una Pila devuelve el elemento ubicado en su tope; (v) **esVacía**, operación que dada una Pila, indica si dicha Pila es vacía.

Este TAD se especificará usando como modelo abstracto de representación la estructura matemática *secuencia*, a continuación se presenta una especificación parcial del **TAD**:

### Especificación A de TAD Pila(T)

#### Modelo de Representación

```
const MAX : int
var contenido : seq T
```

#### Invariante de Representación

$$\text{MAX} > 0 \wedge \# \text{ contenido} \leq \text{MAX}$$

#### Operaciones

```
proc crear (in m : int; out p : Pila)
  { Pre: m > 0 }
  { Post: p.MAX = m  $\wedge$  p.contenido = < > }
```

```
proc empilar (in-out p : Pila; in e : T)
  { Pre: ... }
  { Post: ... }
```

```
proc desempilar (in-out p : Pila)
  { Pre: ... }
  { Post: ... }
```

```
proc tope (in p : Pila; out t : T)
  { Pre: ... }
  { Post: ... }
```

```
proc esVacía (in p : Pila; out ev : boolean)
  { Pre: ... }
  { Post: ... }
```

**Fin TAD**

Se desea que Ud. escriba las especificaciones de cada una de las operaciones propuestas para este TAD; en sus especificaciones haga uso de las operaciones definidas sobre la estructura matemática *secuencia* dadas en el capítulo 9 de [1].

2. Una Cola es un Tipo Abstracto de Datos (TAD) que permite almacenar una secuencia de elementos. Entre los elementos almacenados existe una noción de orden, y los elementos son insertados en un extremo, y eliminados del otro extremo; esto es, el elemento a eliminar corresponde al primer elemento insertado.

Las operaciones que debe ofrecer este TAD son: (i) **crear**, operación que dado un tamaño máximo, crea una Cola vacía de dicho tamaño; (ii) **encolar**, operación que dado una Cola y un elemento, agrega ese elemento en el extremo final de la Cola; (iii) **desencolar**, operación que dada una Cola, elimina de ésta su primer elemento; (iv) **primero**, operación que dada una Cola devuelve su primer elemento; (v) **esVacía**, operación que dada una Cola, indica si dicha Cola es vacía.

Se desea que Ud. especifique este TAD, usando como modelo abstracto de representación alguna(s) estructura(s) matemática(s) presentes en el capítulo 9 de [1]. A continuación se presenta la interfaz que debe ofrecer su TAD:

### Especificación A de TAD Cola(T)

#### Modelo de Representación

#### Invariante de Representación

#### Operaciones

```
proc crear (in m : int; out c : Cola)
  { Pre: ... }
  { Post: ... }
```

```
proc encolar (in-out c : Cola; in e : T)
  { Pre: ... }
  { Post: ... }
```

```
proc desencolar (in-out c : Cola)
  { Pre: ... }
  { Post: ... }
```

```
proc primero (in c : Cola; out p : T)
  { Pre: ... }
  { Post: ... }
```

```
proc esVacía (in c : Cola; out ev : boolean)
  { Pre: ... }
  { Post: ... }
```

Fin TAD

## Referencias

- [1] Carroll Morgan. *Programming from specifications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1990.