

Especificación informal TAD Lista

TAD Lista
VALORES

- Una lista es una secuencia de cero o más elementos de un mismo tipo $(a_1, a_2, a_3, \dots, a_n)$ donde $n \geq 0$:
 - Si $n=0$ se dice que la lista es vacía.
 - Los elementos de la lista están ordenados de forma lineal, no por su contenido, sino por la posición que ocupan. Es decir:
 - a_i precede a a_{i+1} ($\forall i, i=1..n-1$)
 - a_i sucede a a_{i-1} ($\forall i, i=2..n$)

OPERACIONES (SINTAXIS y SEMÁNTICA)

- Generadoras
 - ListaVacía \rightarrow Lista
 - { *Objetivo*: Crear una lista vacía
 - Salida*: Lista vacía
 - PosCondición*: Lista sin datos
 - Inserta (Datos, Posición, Lista) \rightarrow Lista, Boolean
 - { *Objetivo*: inserta un elemento antes de una posición de la lista
 - Entrada*:
 - Datos: contenido del elemento a insertar
 - Posición: posición de inserción
 - Lista: lista donde vamos a insertar
 - Salida*:
 - Lista: lista con el elemento Datos insertado verdadero si se ha podido insertar, falso en caso contrario
 - PreCondición*:
 - Posición es una posición válida de la lista

- Anadir (Datos, Lista) \rightarrow Lista, Boolean
 - { *Objetivo*: Añadir un elemento al final de la lista
 - Entrada*:
 - Lista: lista donde vamos a añadir
 - Datos: contenido del elemento a añadir
 - Salida*:
 - Lista: lista con el elemento añadido verdadero si se ha podido insertar, falso en caso contrario
- Modificadoras
 - Modifica-contenido (Lista, Posición, Datos) \rightarrow Lista
 - { *Objetivo*: modificar el contenido de un elemento de la lista
 - Entrada*:
 - Lista: lista a modificar
 - Posición: dirección del elemento que queremos modificar
 - Datos: nuevo contenido a asignar al elemento en Posición
 - Salida*:
 - Lista: lista con el contenido del elemento modificado
 - Precondición*:
 - La Posición es válida en Lista

- Destructoras

- **Eliminar contenido (Datos, Lista) → Lista, Boolean**
{ Objetivo: Elimina de la lista un elemento con cierto contenido
Entrada:

Datos: contenido del elemento a eliminar

Lista: lista a modificar

Salida:

Lista: lista sin el elemento

verdadero si existe el elemento, falso en caso contrario}

- **Eliminar posición (Posición, Lista) → Lista, Boolean**
{ Objetivo: Elimina de la lista un elemento con cierta posición
Entrada:

Posición: posición del elemento a eliminar

Lista: lista a modificar

Salida:

Lista: lista sin el elemento

verdadero si existe la posición, falso en caso contrario}

- Observadoras

- **Localiza elemento (Datos, Lista) → Posición**
{ Objetivo: busca el primer elemento con cierto contenido en la lista
Entrada:

Datos: contenido del elemento que buscamos

Lista: lista donde realizar la búsqueda

Salida:

Posición: posición del elemento encontrado o nulo si el dato no se encuentra}

- **Existe elemento (Datos, Lista) → Boolean**
{ Objetivo: Determina si existe un elemento con cierto contenido en una lista
Entrada:

Datos: contenido del elemento que buscamos

Lista: lista donde buscar

Salida:

verdadero si se encuentra el elemento, falso en caso contrario}

- **Existe posición (Posición, Lista) → Boolean**
{ Objetivo: Determina si existe un elemento con una cierta posición en una lista
Entrada:

Posición: dirección del elemento que buscamos

Lista: lista donde buscar

Salida:

verdadero si se encuentra la posición en la lista, falso en caso contrario}

- **Es_vacia (Lista) → Boolean**
 { *Objetivo*: determinar si la una lista está vacía
Entrada:
 Lista: lista a comprobar
Salida:
 verdadero si la lista está vacía, falso en caso contrario }
- **Devuelve_contenido (Lista, Posición) → Datos**
 { *Objetivo*: obtener el contenido de un elemento de la lista
Entrada:
 Posición: dirección del elemento que buscamos
 Lista: lista donde buscar
Salida:
 Datos: Contenido del elemento que está en Posición
Precondición:
 La Posición es válida en Lista }

- **Primero (Lista) → Posición**
 { *Objetivo*: devuelve la posición del primer elemento de la lista
Entrada:
 Lista: lista a manipular
Salida:
 Posición del primer elemento
Precondición: La lista no está vacía }
- **Ultimo (Lista) → Posición**
 { *Objetivo*: devuelve la posición del último elemento de la lista
Entrada:
 Lista: lista a manipular
Salida:
 Posición del último elemento
Precondición: La lista no está vacía }
- **Anterior (Posición, Lista) → Posición**
 { *Objetivo*: devuelve la posición del elemento anterior al actual
Entrada:
 Posición: dirección del elemento actual
 Lista: lista a manipular
Salida:
 Posición del elemento anterior o nulo si es el primero
Precondición: La lista no está vacía y la posición es válida }
- **Siguiente (Posición, Lista) → Posición**
 { *Objetivo*: devuelve la posición del elemento siguiente al actual
Entrada:
 Posición: dirección del elemento actual
 Lista: lista a manipular
Salida:
 Posición del elemento siguiente o nulo si es el último
Precondición: La lista no está vacía y la posición es válida }