

Funções do TAD Lista

Prof. D.Sc. Saulo Ribeiro

Funções do TAD Lista

- Lista inicLista(void) : cria e retorna uma lista vazia
- Void anxLista(Lista lst, TipoL elem): adiciona um elemento depois do iterador
- void insLista (Lista lst, TipoL elem): adiciona um elemento antes do iterador
- void elimLista(Lista lst): elimina o elemento que está sob o iterador
- void primLista(Lista lst): coloca o iterador sobre o primeiro elemento da lista
- void ultLista(Lista lst) : coloca o iterador sobre o último elemento da lista

Funções do TAD Lista

- void segLista(Lista lst) : avança o iterador uma posição
- void posLista(Lista lst, int pos): coloca o iterador sobre a posição pos
- TipoL infoLista(Lista lst): retorna o elemento sob o iterador
- Int longLista(Lista lst): retorna a quantidade de elementos da lista
- Int fimLista(Lista lst): retorna verdadeiro se o iterador estiver indefinido

Notação

- Se define o **iterador** de uma lista como o **lugar** da sequencia sobre a qual serão realizadas as operações que se aplicam ao objeto abstrato.
- **Lista com o iterador definido** (iterador aponta para um elemento da lista):
 - $\langle e_1, e_2, \dots, [e_i], \dots, e_n \rangle$
 - 1 2 i n

Notação

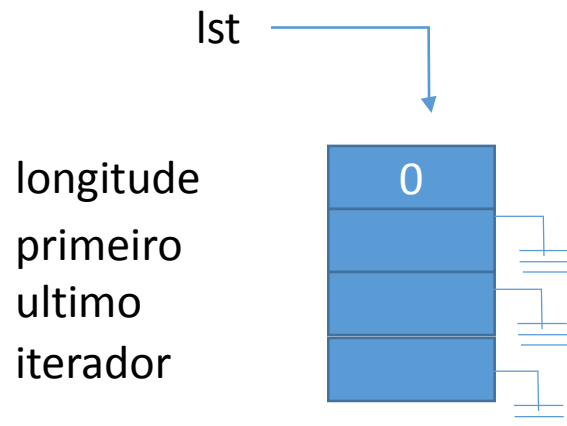
- **Lista vazia**: o **iterador** está indefinido: não aponta para ninguém (NULL).
- < > []

Notação

- **Lista com o iterador indefinido** (iterador NÃO aponta para um elemento da lista):
 - $\langle e_1, e_2, \dots, e_i, \dots, e_n \rangle []$
 - 1 2 i n

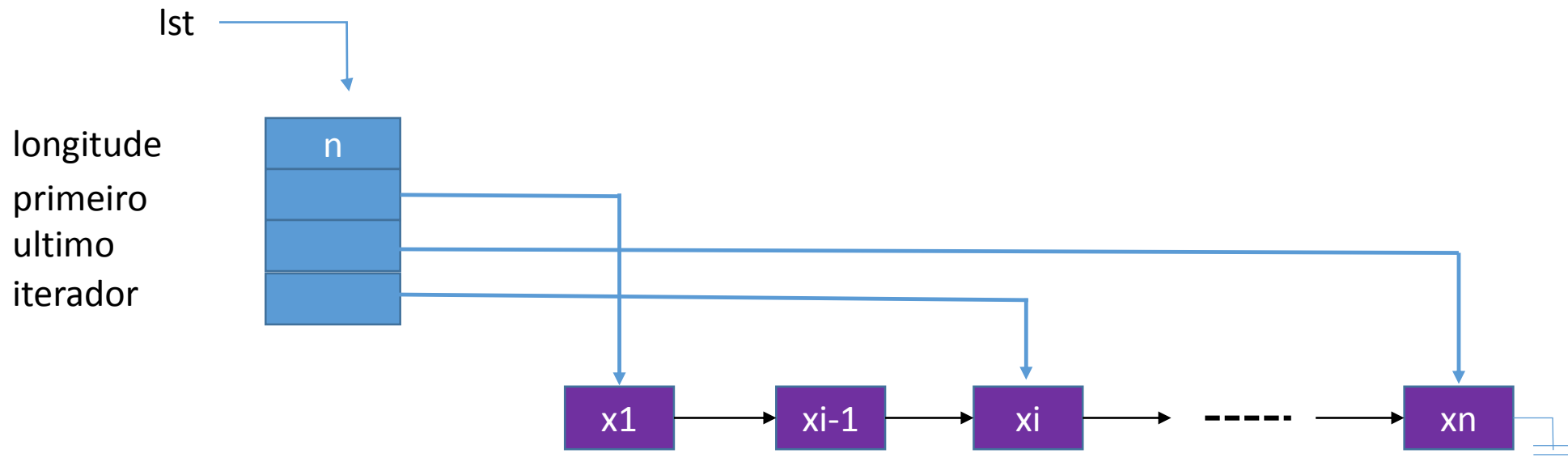
Notação

- **Lista vazia**: o **iterador** está indefinido: não aponta para ninguém (NULL). Primeiro e Ultimo também estão indefinidos (NULL).
- $lst = < > []$



Notação

- **Lista com o iterador definido** (iterador aponta para um elemento da lista). Primeiro aponta para o 1º nó e Ultimo para o ultimo nó.
- $lst = \langle x1, \dots, [xi], \dots, xn \rangle$



typedef: ListaNo, pListaNo, Tlista, Lista

```
typedef int TipoL;
```

```
typedef struct ListaNo{  
    TipoL info;  
    struct ListaNo *prox;  
} *pListaNo;
```

```
typedef struct{  
    pListaNo primeiro, ultimo, iterador;  
    int longitude;  
} Tlista, * Lista;
```

Funções do TAD Lista: inicLista e anxLista

- Lista inicLista(void) : cria e retorna uma lista vazia

{pos: inicLista = < > [] }

- Void anxLista(Lista lst, TipoL elem): adiciona um elemento depois do iterador

{pre: lst = LST, lst = < > [] ou lst = <x1, ... , [xi], ... , xn> }

{pos: (LST = < > [] \rightarrow lst = <[elem] >) ou

(LST=<x1,...,[xi],...,xn>) \rightarrow lst=<x1,...,xi,[elem],...,xn>)

}

Funções do TAD Lista: insLista

- void insLista (Lista lst, TipoL elem): adiciona um elemento antes do iterador

{pre: lst = LST, lst = < > [] ou lst = <x1, ... , [xi], ... , xn> }

{pos: (LST = < > [] \rightarrow lst = <[elem] >) ou

(LST=<x1,...,[xi],...,xn>) \rightarrow lst=<x1,..., xi-1, [elem], xi,...,xn>)

}

Funções do TAD Lista: elimLista

- void elimLista(Lista lst): elimina o elemento que está sob o iterador

{pre: lst = LST= <x1, ... , [xi], ... , xn> }

{pos: (LST = < [x1]> [] \rightarrow lst = < >[]) ou

(LST=<x1,...,[xn]>) \rightarrow lst=<x1,..., xn-1>[] ou

(LST=<x1,...,[xi],...,xn>) \rightarrow lst=<x1,..., xi-1, [xi+1],...,xn>)

}

Funções do TAD Lista: primLista

- void primLista(Lista lst): coloca o iterador sobre o primeiro elemento da lista

{pre: lst = LST }

{pos: (LST = < > [] \rightarrow lst = < > []) ou
(LST=<x1,...,xn>) \rightarrow lst=<[x1],..., xn>)
}

Funções do TAD Lista: ultLista

- void ultLista(Lista lst) : coloca o iterador sobre o último elemento da lista

{pre: lst = LST }

{pos: (LST = < > [] \rightarrow lst = < > []) ou
(LST=<x1,...,xn>) \rightarrow lst=<x1,..., [xn]>)
}

Funções do TAD Lista: segLista

- void segLista(Lista lst) : avança o iterador uma posição

{pre: lst = LST= <x1, ... , [xi], ... , xn> }

{pos: (LST=<x1,...,[xi],...,xn>) → lst=<x1,..., xi, [xi+1],...,xn> ou

(LST=<x1,...,[xn]>) → lst=<x1,...,xn>[])

}

Funções do TAD Lista: posLista

- void posLista(Lista lst, int pos): coloca o iterador sobre a posição pos

{pos: (pos < 1 ou pos > n \rightarrow lst=<x1,...,xn>[]) ou
(1 =< pos <=n \rightarrow lst=<x1,...,[xpos],...,xn>)
}

Funções do TAD Lista: infoLista

- TipoL infoLista(Lista lst): retorna o elemento sob o iterador

{pre: lst = <x1, ... , [xi], ... , xn> }

{pos: infoLista = xi}

Funções do TAD Lista: longLista

- Int longLista(Lista lst): retorna a quantidade de elementos da lista

{pos: longfoLista = n}

Funções do TAD Lista: fimLista

- Int fimLista(Lista lst): retorna verdadeiro se o iterador estiver indefinido

{pos: (lst=<x1,...,xn>[]) \rightarrow fimLista = TRUE ou
(lst=<x1,...,[xi],...xn>) \rightarrow fimLista = FALSE)
}