



Conteúdo

1. Introdução

2. Listas

3. Pilhas e Filas

4. Árvores

5. Árvores de Pesquisa

- Árvore Binária e Árvore AVL
- Árvore N-ária e Árvore B

6. Tabelas de Dispersão (Hashing)

7. Métodos de Acesso a Arquivos

8. Métodos de Ordenação de Dados



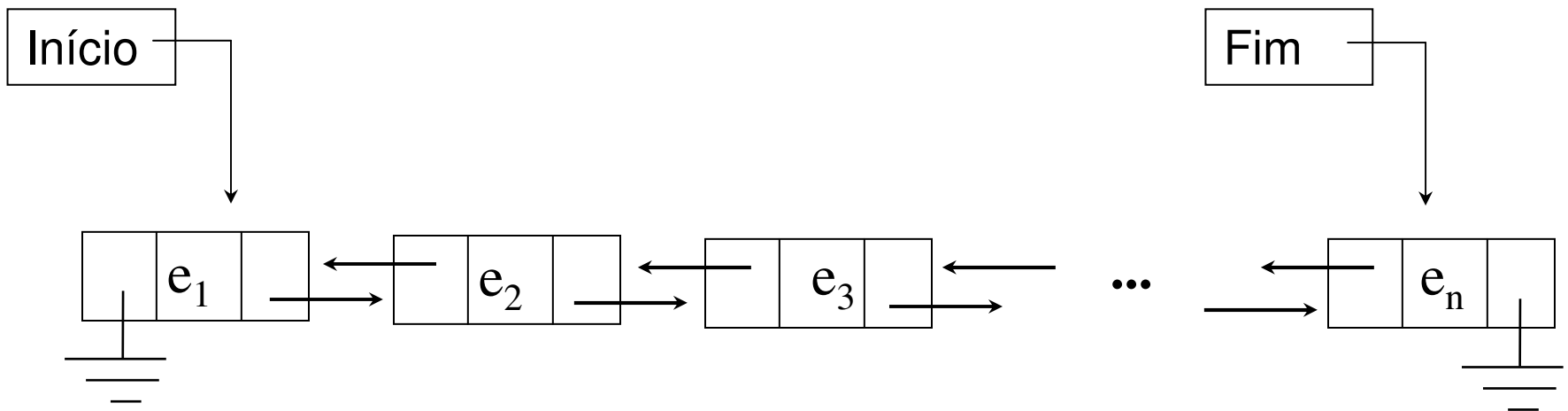


Listas Duplamente Encadeadas

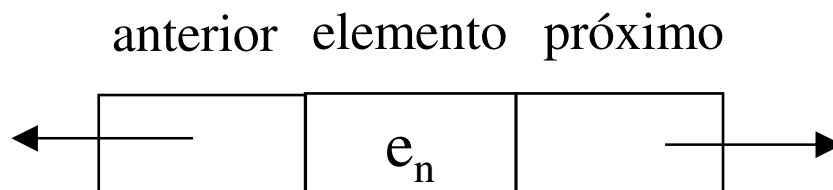


Listas Duplamente Encadeadas

⇒ Os elementos estão associados entre si através de elos.

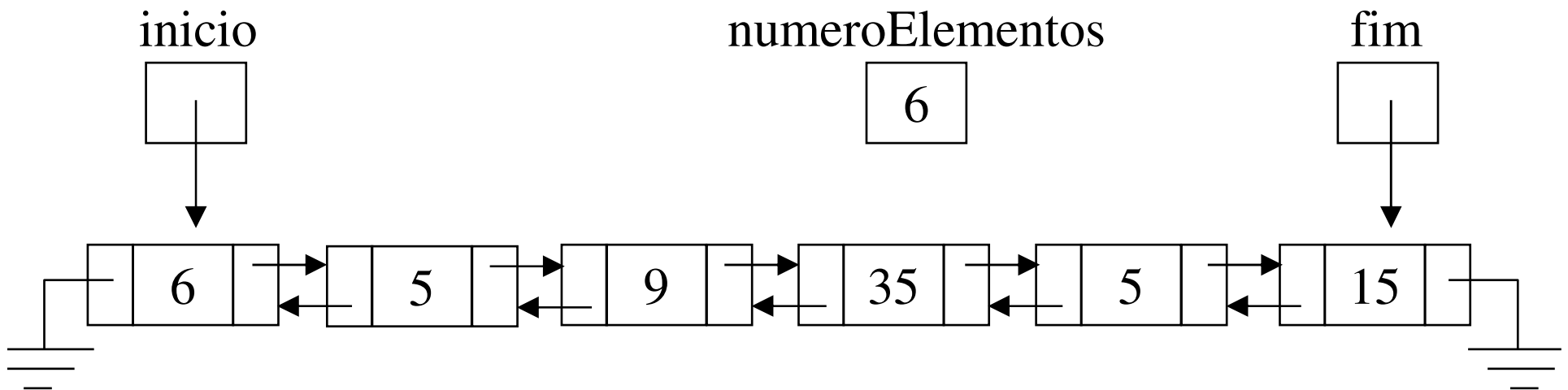


⇒ Cada nodo da lista mantém um elemento (objeto) e uma referência para o nodo anterior e para o nodo seguinte.



Operações sobre a Lista

Exemplo:





Operações sobre a Lista

Operações:

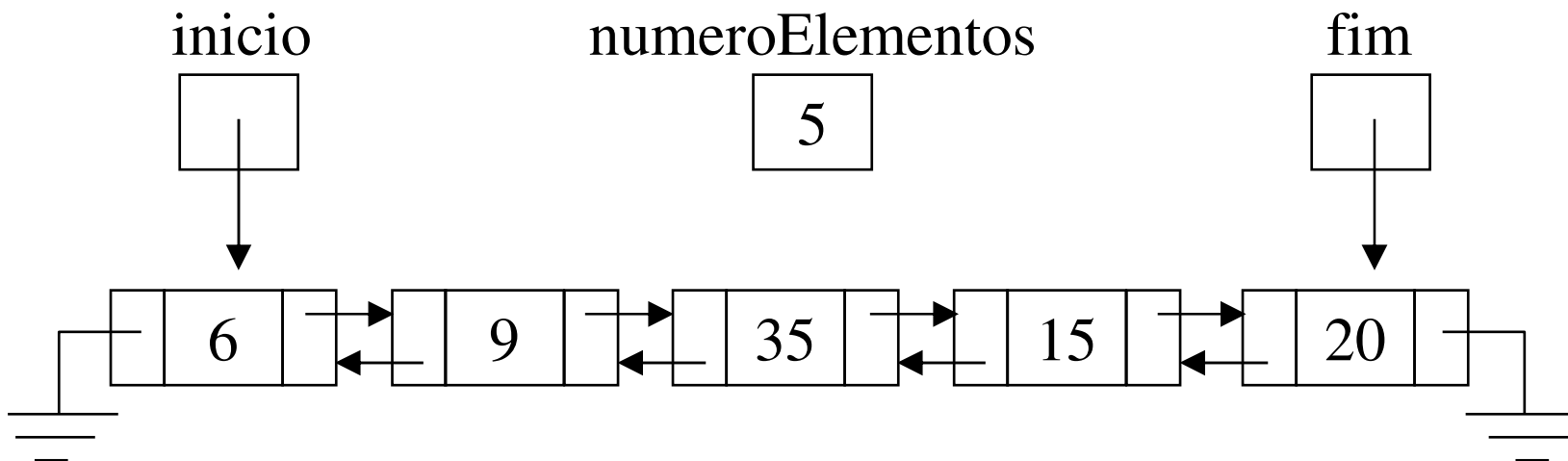
- inserir o elemento 30 na posição 7
- inserir o elemento 15 na posição 3
- inserir o elemento 16 na posição 0
- excluir da posição 9
- excluir da posição 5
- excluir da posição 0
- excluir o elemento 20
- retorna o elemento da posição 3
- retorna a posição do elemento 5



Implementação

➡ Classe **ListaDuplamenteEncadeada** implementa Lista
Atributos

- **inicio** (referência a um objeto da classe **NodoDuplo**)
- **fim** (referência a um objeto da classe **NodoDuplo**)
- **numeroElementos**

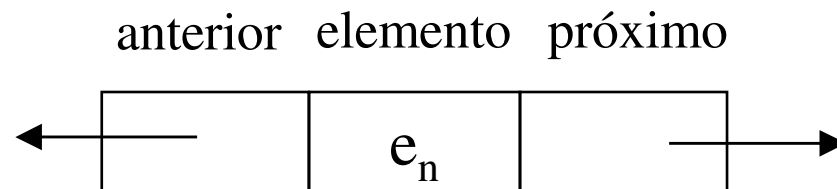


Implementação

⇒ Classe **NodoDuplo**

Atributos:

- **elemento**
- **anterior** (referência a um outro objeto da classe **NodoDuplo**)
- **proximo** (referência a um outro objeto da classe **NodoDuplo**)





Implementação

⇒ Classe NodoDuplo

Atributos:

- private E elemento
- private Nodo anterior
- private Nodo proximo

Métodos:

- public Nodo (E elemento)
- public void atribuiElemento (E elemento)
- public void atribuiProximo (Nodo nodo)
- public void atribuiAnterior (Nodo nodo)
- public E retornaElemento ()
- public Nodo retornaProximo ()
- public Nodo retornaAnterior ()





Implementação

⇒ Classe **ListaDuplamenteEncadeada** implementa Lista

Atributos

- **inicio** (referência a um objeto da classe NodoDuplo)
- **fim** (referência a um objeto da classe NodoDuplo)
- **numeroElementos**

Métodos

- construtor ()
+
• métodos especificados na interface Lista



Implementação

⇒ Método construtor da Lista Duplamente Encadeada

```
public class ListaDuplamenteEncadeada<E> implements Lista<E> {  
  
    private NodoDuplo inicio;  
    private NodoDuplo fim;  
    private int numElementos;  
  
    public ListaDuplamenteEncadeada() {  
        this.inicio = null;  
        this.fim = null;  
        this.numElementos = 0;  
    }  
}
```



Implementação

⇒ Métodos especificados na interface Lista

- public void insere (E elemento);

/** Insere o elemento no final da lista.

* @param elemento Elemento a ser inserido no final da lista. */

- public E remove (int posicao) throws ExcecaoPosicaoInvalida;

/** Remove o elemento que se encontra na posição indicada pelo argumento e retorna o elemento.

* @param posicao Posição do elemento que será removido da lista. A lista começa na posição 0.

* @return O elemento removido.

* @throws ExcecaoPosicaoInvalida se a posição for menor do que zero, ou igual ou maior do que o número de elementos. */

