Lista Duplamente Encadeada

Classe No

1. Atributos:

- **valor**: armazena o valor do nó.
- **proximo**: referência para o próximo nó na lista.
- **anterior**: referência para o nó anterior na lista.

2. Construtor:

- Inicializa o nó com um valor e define as referências **proximo** e **anterior** como `null`.

Classe ListaDuplamenteEncadeada

1. Atributos:

- **inicio**: referência para o primeiro nó da lista.
- **fim**: referência para o último nó da lista.
- **tamanho**: número de nós na lista.

2. Construtor:

- Inicializa a lista como vazia (**inicio** e **fim** = `null`, **tamanho** = 0).

3. Método estaVazia:

- Verifica se a lista está vazia com base no tamanho.

Métodos de Inserção

4. inserirNoInicio:

- Adiciona um nó no início.
    - Se a lista estiver vazia, o nó torna-se o único elemento (**inicio** e **fim** apontam para ele).
    - Caso contrário:

        - Ajusta o ponteiro **proximo** do novo nó para o atual **inicio**.
        - Ajusta o ponteiro **anterior** do antigo **inicio** para o novo nó.
    - Atualiza **inicio** e incrementa o tamanho.

5. inserirNoFim:

- Adiciona um nó no final.
    - Similar ao método anterior, mas ajusta os ponteiros do **fim**.

6. inserirEmPosicao:

- Insere um nó em uma posição específica.
    - Valida a posição.
    - Insere no início ou no fim, se for o caso.
    - Para posições intermediárias, ajusta os ponteiros do nó anterior e do próximo.

Métodos de Remoção

7. removerDoInicio:

- Remove o primeiro nó.
    - Ajusta **inicio** para o próximo nó e redefine o ponteiro **anterior** do novo **inicio**.
    - Se a lista tiver apenas um elemento, redefine **inicio** e **fim** como `null`.

8. removerDoFim:

- Remove o último nó.
    - Similar ao método anterior, ajustando o ponteiro **fim**.

9. removerDePosicao:

- Remove um nó de uma posição específica.
    - Valida a posição.
    - Remove do início ou fim, se necessário.
    - Para posições intermediárias, ajusta os ponteiros dos nós adjacentes.

Métodos de Consulta e Atualização

10. pesquisar:

- Procura um valor na lista.
    - Percorre a lista do **inicio** ao **fim** e imprime a posição se o valor for encontrado.

11. atualizar:

- Altera o valor de um nó em uma posição específica.
    - Valida a posição e percorre até o nó desejado para atualizá-lo.

Métodos de Percurso

12. percorrerDoInicioAoFim:

- Percorre a lista do início ao fim, imprimindo os valores.

13. percorrerDoFimAoInicio:

- Percorre a lista do fim ao início, imprimindo os valores.

Classe Main

1. Criação do Menu:

- Oferece opções para manipular a lista.
- Cada operação corresponde a um método da classe **ListaDuplamenteEncadeada**.

2. Entrada do Usuário:

- Utiliza a classe **Scanner** para capturar valores e posições.
- Validações e chamadas aos métodos da lista são feitas com base na opção escolhida.

3. Laço do-while:

- Mantém o menu ativo até o usuário escolher a opção de saída (0).

Validações e Mensagens

- Mensagens claras informam o usuário sobre operações inválidas (e.g., posição fora do intervalo ou lista vazia).