

PRÁTICA 4

- 1) Na classe `ListaEstatica` que desenvolvemos em sala, implementamos o método `adicionar`, que recebe um elemento para ser adicionado na Lista.

O método `adicionar` sempre coloca o elemento a ser inserido no final da Lista.

O atributo `nroElem` indica quantos elementos tem na Lista e também é o valor do índice onde será inserido o próximo elemento (no final da Lista).

Implementar na classe `ListaEstatica`, o método `adicionarOrdenado`.

Esse método receberá um valor a ser inserido na Lista.

Porém, a inserção será feita de modo que a lista permaneça ordenada.

Supor que desde o primeiro elemento adicionado, este método é utilizado na lista para que ela permaneça ordenada.

Atenção, pois o método deverá deslocar os elementos da lista para "abrir um espaço na lista" se o valor tiver que ser inserido no meio ou no início da lista.

Devolve -1 se não conseguiu inserir e 0 se conseguiu.

Testar o método `adicionarOrdenado`

No método `Main`, teste esse método `adicionarOrdenado`, inserindo vários números e depois exibindo a lista.

- 2) Crie uma classe chamada **`ListaEstaticaString`**, semelhante à classe **`ListaEstatica`**, porém para armazenar strings. Altere o que for necessário para que os métodos funcionem adequadamente.

Teste a classe no método `Main`.

- 3) Na classe `ListaEstatica` que desenvolvemos em sala, implementamos o método `Adicionar`, que recebe um elemento para ser adicionado na Lista.

O método `Adicionar` sempre coloca o elemento a ser inserido no final da Lista.

O atributo `nroElem` indica quantos elementos tem na Lista e também é o valor do índice onde será inserido o próximo elemento (no final da Lista).

Implementar na classe `ListaEstatica`, o método `AdicionarInicio`.

Esse método receberá um valor a ser inserido na Lista.

Porém, a inserção será feita sempre no **início** da Lista, deslocando os elementos da lista para "abrir" um espaço no início do vetor.

Devolve -1 se não conseguiu inserir e 0 se conseguiu.

Por exemplo (imaginando que a lista está vazia):

`AdicionarInicio(2)`

Após a execução desse método, a lista terá o elemento 2. (2 estará no índice zero da lista)

`AdicionarInicio(10)`

Após a execução desse método, a lista terá os elementos 10, 2 (10 estará no índice zero, e 2 no índice 1)

`AdicionarInicio(15)`

Após a execução desse método, a lista terá os elementos 15, 10, 2 (15 estará no índice zero, 10 estará no índice 1, e 2 no índice 2)

Testar o método `AdicionarInicio`

No método `Main`, crie um objeto da classe `ListaEstatica` e adicione valores utilizando esse método `AdicionarInicio`. Conforme insere, chame o método `Exibir` para conferir se está funcionando corretamente.

- 4) Na classe `ListaEstatica` que desenvolvemos em sala, implementamos o método `RemoverElemento`, que remove um elemento da lista.

Da maneira como implementamos os métodos que inserem valores na lista (métodos `Adicionar`, `AdicionarOrdenado` ou `AdicionarInicio`), nós permitimos que o usuário insira valores repetidos.

Então, o usuário poderia inserir várias vezes o número 3, por exemplo.

Nesse caso, supondo que existam várias ocorrências do número 3 na lista, os métodos `Buscar(3)` e `RemoverElemento(3)` só seriam aplicados ao primeiro número 3 que for encontrado na lista.

Ilustrando, se a lista tivesse os valores abaixo:

4 3 5 3 10 30 3 40

O método `Buscar(3)` devolveria o índice 1.

O método `RemoverElemento(3)` removeria apenas o primeiro 3 da lista.

Implementar na classe `ListaEstatica`, o método `RemoverTodosElem`

Esse método receberá um valor a ser removido da Lista.

Porém, esse método não removerá apenas a primeira ocorrência do elemento na lista, e sim todas as ocorrências.

Devolve -1 se não conseguiu remover e 0 se conseguiu.

Por exemplo:

Imaginando que a lista tenha os valores abaixo:

4 3 5 3 10 30 3 40

`RemoverTodosElem(3)` removeria todos os elementos 3 da lista e a lista ficaria assim:

4 5 10 30 40

Testar o método `RemoverTodosElem`

No método `Main`, crie um objeto da classe `ListaEstatica` e adicione valores utilizando qualquer um dos métodos `Adicionar` que foram implementados. Procure inserir valores repetidos, alternados com outros valores. Chame o método `RemoverTodosElem` para testar seu funcionamento, sempre chamando o método `Exibir` para conferir o resultado.