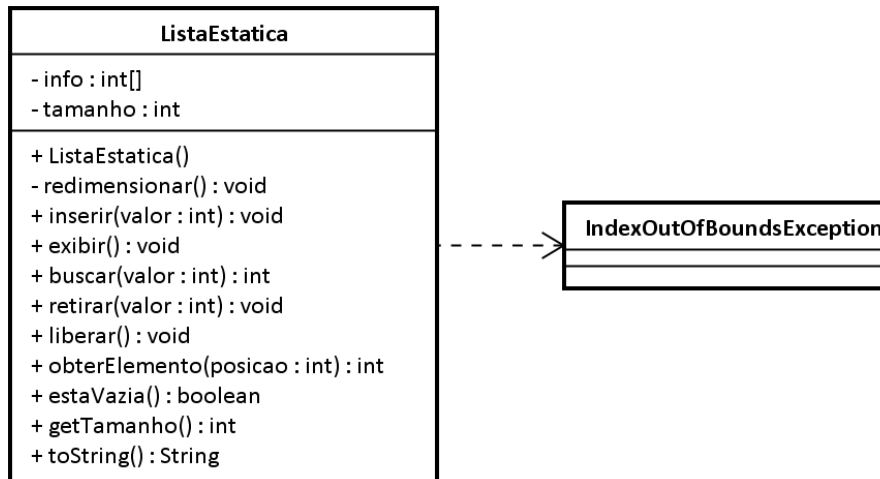


Lista de Exercícios 01

Questão 1

Traduza o diagrama de classes abaixo para exercitar a implementação estática de uma lista capaz de armazenar números inteiros.



A descrição dos métodos da classe **ListaEstatica** consta a seguir:

- ListaEstatica()**: construtor da classe. Deve criar um vetor para guardar os dados e estabelecer que a lista está vazia;
- redimensionar()**: este método deverá aumentar a capacidade de armazenamento da lista, criando um novo vetor com capacidade de armazenamento expandida em 10 novas posições e copiar os dados do vetor original para o novo vetor criado. Por fim, o método **redimensionar()** deverá assumir que o novo vetor **info** é o vetor recentemente criado;
- inserir(int)**: Deve inserir o número fornecido como argumento no vetor encapsulado pela lista. Caso o vetor encapsulado **info** não tenha mais posições livres, deve invocar o método privado **redimensionar()**, para expandir a capacidade de armazenamento da lista;
- exibir()**: Deve exibir o conteúdo armazenado na lista, apresentando na tela o valor do primeiro número até o último número armazenado, nesta ordem;
- buscar(int)**: Deve procurar se há um número igual ao fornecido como argumento. Caso seja encontrado, este método deverá retornar a posição do número no vetor (índice no vetor). Se não for localizado, deverá retornar **-1**;
- retirar(int)**: Deve retirar da lista o dado fornecido como argumento, deslocando todos os elementos das posições seguintes, uma posição para esquerda;
- liberar()**: Deverá limpar a estrutura de dados de forma que o vetor **info** seja redimensionado novamente para que tenha capacidade de armazenar apenas 10 elementos;
- obterElemento(int)**: este método deverá retornar o número armazenado na posição fornecida como argumento. Caso o valor do parâmetro corresponda a uma posição inexistente ou não ocupada, este método deverá lançar a exceção **IndexOutOfBoundsException**.
- estaVazia()**: este método deverá retornar **true** se a lista não possuir nenhum dado armazenado e **false** se a lista estiver armazenando algum dado;
- getTamanho()**: método *getter* da variável **tamanho**;
- toString()**: deve retornar os valores armazenados na lista, desde o primeiro até o último, separando-os por vírgula

Questão 2

Implemente o seguinte plano de testes:

Plano de testes PL01 – Validar funcionamento da implementação estática de lista			
Caso	Descrição	Entrada	Saída esperada
1	Testar método de inclusão de dados na lista	Criar uma lista. Adicionar os dados 5, 10, 15 e 20.	toString() deve resultar em "5,10,15,20"
2	Testar método de obtenção de tamanho da lista	Criar uma lista. Adicionar os dados 5, 10, 15 e 20.	getTamanho() deve resultar em 4
3	Testar método buscar() com elemento existente	Criar uma lista. Adicionar os dados 5, 10, 15 e 20, nesta ordem.	buscar(15) deve resultar em 2.
4	Testar método buscar() com elemento inexistente	Criar uma lista. Adicionar os dados 5, 10, 15 e 20.	buscar(30) deve resultar em -1.
5	Testar método retirar()	Criar uma lista. Adicionar os dados 5, 10, 15 e 20, nesta ordem. Em seguida, retirar o elemento 10 – retirar(10).	toString() deve resultar em "5 15 20". getTamanho() deve resultar em 3.
6	Testar inclusão que provoque redimensionamento	Criar uma lista. Adicionar 15 números na lista (de 1 à 15).	toString() deve resultar em "1 2 3 4 5 6 7 8 9 10 11 12 13 14 15". getTamanho() deve resultar em 15.
7	Testar método obterElemento()	Criar uma lista. Adicionar os dados 5, 10, 15 e 20.	obterElemento(3) deverá resultar em 20.
8	Testar lançamento de exceção no método obterElemento()	Criar uma lista. Adicionar os dados 5, 10, 15 e 20.	obterElemento(5) deverá lançar uma exceção.
9	Certificar que liberar() remove todos os elementos	Criar uma lista. Adicionar os dados 5, 10, 15 e 20. Invocar o método liberar().	estaVazia() deverá resultar em true.