



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
Instituto de Ciências Exatas e Informática  
Prática sobre Listas

Curso : *Engenharia de Software*  
Disciplina : *Algoritmos e Estruturas de Dados II*  
Professora : *Eveline Alonso Veloso*

### **Regras Básicas:**

1. Estude bastante cada par de entrada/saída fornecido.
2. Todos os programas deverão ser desenvolvidos na linguagem de programação Java.
3. Essas práticas poderão ser desenvolvidas em grupos de, no máximo, três integrantes.
4. Cópias, se existirem, serão encaminhadas ao colegiado de coordenação didática do curso.
5. Fique atento ao *charset* dos arquivos de entrada e saída. Recomenda-se a utilização dos métodos da classe MyIO.java para **leitura de dados do teclado**. É necessário definir o *charset* a ser utilizado antes de começar a leitura de dados do teclado, da seguinte forma: `MyIO.setCharset("UTF-8")`.
6. As saídas esperadas, cadastradas no VERDE pela professora, foram geradas empregando-se: `System.out.println()`.
7. Para cada exercício, vocês devem submeter apenas um arquivo (.java). Essa regra será necessária para a submissão de exercícios no VERDE e no identificador de plágios utilizado na disciplina.
8. A resolução (código) de cada exercício deverá ser submetida ao VERDE.
9. A execução do código submetido será realizada automaticamente pelo VERDE, mas o código será analisado e validado pela professora.

### **Base de Dados:**

Você foi contratado para trabalhar em uma empresa que distribui *stream* de séries de televisão na *Web*. Sua tarefa é organizar as informações das séries disponíveis para exibição ao usuário. Os dados que serão exibidos foram obtidos por meio de consultas à base de dados Wikipedia.



Assim, você precisará ler, organizar e armazenar os dados de cada série em memória, utilizando estruturas de dados adequadas.

## Exercícios:

### 1. Lista implementada por meio de vetor

Utilize a classe *Serie* especificada e desenvolvida em prática anterior.

Crie uma **lista, implementada por meio de vetor**, de objetos da classe *Serie*. Lembre-se que, na verdade, temos um vetor de referências para objetos do tipo *Serie*.

Neste exercício, faremos inserções e remoções de itens na lista e, após o processamento de todas as operações, mostraremos seus elementos.

Os métodos de sua lista devem operar conforme descrito a seguir, respeitando-se parâmetros e tipos de retorno:

- Sua classe *Lista* deverá ter **dois construtores**.
- *void inserirInicio (Serie serie)*: insere um objeto do tipo *Serie* na primeira posição da lista, necessitando remanejar todos os demais.
- *void inserir (Serie serie, int posicao)*: insere uma série na posição da lista indicada pelo parâmetro *posicao*, desse método; onde  $0 \leq posicao \leq n$ , sendo  $n$  o número de séries já inseridas na estrutura. Esse método também remaneja os demais objetos da lista.
- *void inserirFim (Serie serie)*: insere um objeto da classe *Serie* na última posição da lista.
- *Serie removerInicio()*: remove e retorna a primeira série da lista, remanejando as demais.
- *Serie remover(int posicao)*: remove e retorna o objeto *Serie* armazenado na posição da lista indicada pelo parâmetro *posicao*, desse método; necessitando remanejar os demais.
- *Serie removerFim()*: remove e retorna a última *Serie* da lista.
- *void mostrar()*: para todos os objetos do tipo *Serie* presentes na lista, exibe a posição do objeto na lista seguida dos valores de seus atributos (observe o formato de cada linha da saída esperada).

Seu programa deve ler um arquivo-texto chamado **data.txt** que, no VERDE, **localiza-se na pasta /tmp**. Você deve preencher um vetor de objetos da classe *Serie* com os dados das diversas séries informadas nesse arquivo.

A **primeira linha** desse arquivo deve ser **descartada**, pois apenas informa os diversos **campos** presentes em cada uma das linhas seguintes. Cada uma das linhas seguintes indica os dados de uma série, separados por um ponto-e-vírgula (;).

A **entrada padrão** é dividida em **duas partes**. A **primeira** contém, em cada linha, uma *string* indicando o **nome da série** que deve ser inicialmente **inserida no final da lista** de séries, na ordem em que são apresentadas. Após a palavra FIM, inicia-se a segunda parte da entrada padrão.

A primeira linha dessa **segunda parte da entrada padrão** apresenta um número inteiro  $n$  indicando a **quantidade de séries que serão** em seguida **inseridas ou removidas** da lista. Nas próximas  $n$  linhas, tem-se  $n$  **comandos**

**de inserção ou remoção** que devem ser processados neste exercício. Cada uma dessas linhas tem uma palavra de comando, conforme descrito a seguir:

- II: inserir no início;
- I\*: inserir em uma determinada posição;
- IF: inserir no final;
- RI: remover do início;
- R\*: remover de uma determinada posição; e
- RF: remover do final.

No caso dos comandos de **inserção**, temos também uma *string* indicando o **nome da série que deve inserida** na lista de séries.

No caso dos comandos de **inserção e remoção "em uma determinada posição"**, temos também um inteiro indicando essa **posição**. No comando de inserção, a posição fica imediatamente após a palavra de comando. Lembre-se que o primeiro item da lista encontra-se na posição 0.

A **saída padrão** deve apresentar uma linha para **cada série removida**, sendo que essa informação será constituída pela *string* "(R)" seguida do atributo nome dessa série.

Em seguida, teremos, ainda na saída padrão, os atributos relativos às **séries armazenadas na lista após o processamento de todas as operações de inserção e remoção** (observe o formato de cada linha da saída esperada).

## 2. Lista encadeada

Refaça o exercício **Lista implementada por meio de vetor** usando **lista encadeada**.

Neste exercício, sua classe **Lista** deverá ter **apenas um construtor**.