PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Disciplina: Algoritmos e Estruturas de Dados
Entrega: **A lista deverá ser entregue até dia 04/05 às 18h. Após esse prazo, o sistema irá permitir a entrega da Lista 8 até 11/05 às 18h, valendo metade dos pontos.**

Cópias serão desconsideradas, ou seja, a nota será igual a 0 (zero).

Lista 8

Observações:

- **Nos exercícios que serão refeitos das listas 6 e 7 não devem ser considerados os limites de tamanho para as estruturas de dados. Uma vez que, as estruturas de dados flexíveis não possuem limitação de tamanho.**
- **Nessa lista de exercícios não deve ser utilizada nenhuma Estrutura de Dados nativa do Java**

Pilha flexível

1) Refaça o exercício 4 da Lista 7 usando pilha flexível (Será necessário implementar as classes: Celula, Pilha e Teste)
2) Refaça o exercício 5 da Lista 7 usando pilha flexível (Será necessário implementar as classes: Celula, Pilha e Teste)
3) Escreva um programa que leia uma sequência (String) de parênteses e colchetes e verifique se essa sequência está bem-formada, ou seja, se os parênteses e colchetes são fechados na ordem inversa àquela em que foram abertos. Utilize uma pilha flexível para auxiliar nessa verificação.

   Exemplos:
   - (( ) [( )]) a sequência está bem-formada
   - ([ )] a sequência está malformada

   Será necessário implementar as classes: Celula, Pilha e Teste.

Fila flexível

1) Refaça o exercício 1 da Lista 7 usando fila flexível (Será necessário implementar as classes: Celula, Fila e Teste)
2) Refaça o exercício 3 da Lista 7 usando fila flexível (Será necessário implementar as classes: Cliente, Celula, Fila e Teste. A classe Celula terá um atributo public Cliente elemento;).
3) Escreva um programa que simule o controle de uma pista de decolagem de aviões em um aeroporto. Neste programa, o usuário deve ser capaz de realizar as seguintes tarefas:
   1. Listar a quantidade de aviões que estão aguardando na fila de decolagem
   2. Autorizar a decolagem do primeiro avião da fila de decolagem
   3. Adicionar um avião à fila de colagem
   4. Listar o identificador de todos os aviões que estão na fila de colagem
   5. Listar o nome e o identificador do primeiro avião da fila de colagem

Nesse exercício devem ser criadas as classes: Aviao, Celula, Fila e Teste. A classe Celula terá um atributo public Aviao elemento;. Considere que cada avião possui um nome e um número como identificador. Adicione outros atributos caso julgue necessário.

Lista flexível simples

1) Crie um programa que permita gerenciar as temperaturas diárias registradas em Belo Horizonte. Para tanto, o programa deve apresentar o seguinte menu de opções para o usuário:
   1) Inserir uma temperatura no início da lista
   2) Inserir uma temperatura no final da lista
   3) Inserir uma temperatura numa posição específica da lista
   4) Remover a primeira temperatura da lista (Imprimir a temperatura removida)
   5) Remover a última temperatura da lista (Imprimir a temperatura removida)
   6) Remover uma temperatura de uma posição específica na lista (Imprimir a temperatura removida)
   7) Mostrar todos as temperaturas da lista
   8) Mostrar o número de dias que uma temperatura específica foi registrada
   9) Verificar se a lista está vazia
   10) Encerrar o programa

O programa deve ser executado até que a opção 10 seja escolhida pelo usuário.

Para o desenvolvimento desse programa deve ser utilizada a estrutura de dados Lista flexível simples. Além disso, devem ser criados dois novos métodos na classe Lista:

- pesquisar: recebe como parâmetro uma temperatura (double). O método deve retornar a quantidade de dias (int) que aquela temperatura foi registrada (isto é, a quantidade de vezes que a temperatura aparece na lista).
- isVazia: método sem parâmetros. O método deve retornar *true* caso a lista esteja vazia e *false*, caso contrário.

Nesse exercício será necessário implementar as classes: Celula, Lista e Teste

2) Crie uma classe Site que terá como atributos nome (String) e link (String), implemente os métodos que julgar necessários. Crie uma classe Lista (lista flexível simples), para representar uma Lista de Sites (isto é, a classe Celula terá um atributo public Site elemento;). Implemente na classe Lista todos os métodos de inserção, remoção e também o método mostrar. Além disso, crie um novo método na classe Lista:
    - pesquisarLink: esse método deve receber como parâmetro o nome de um site (String), e deve pesquisar na lista e retornar o link do site (String). Além disso, o método deve mover esse Site para o início da lista, de forma que ele possa ser encontrado mais rapidamente na próxima vez que for pesquisado.

   O programa deve apresentar o seguinte menu de opções para o usuário:
    1. Inserir um Site no início da lista
    2. Inserir um Site no final da lista
    3. Inserir um Site numa posição específica da lista
    4. Remover o primeiro Site da lista (Imprimir o nome do site removido)
    5. Remover o último Site da lista (Imprimir o nome do site removido)
    6. Remover um Site de uma posição específica da lista (Imprimir o nome do site removido)
    7. Mostrar o nome e o link de todos os sites da lista
    8. Pesquisar o link de um site
    9. Encerrar o programa

   O programa deve ser executado até que a opção 9 seja escolhida pelo usuário.
   Nesse exercício será necessário implementar as classes: Site, Celula, Lista e Teste

3) Implemente um programa que permita que a universidade Acme gerencie duas listas de alunos: 1) lista de mestrandos, 2) lista de doutorandos. Na universidade Acme cada aluno tem um código único (int). Deve ser armazenado nas listas apenas os códigos dos alunos. O programa deverá apresentar um menu com as seguintes opções:

   Menu:
   1) Inserir um aluno no final da lista de mestrandos
   2) Inserir um aluno no final da lista de doutorandos
   3) Remover um aluno específico da lista de mestrandos
   4) Remover um aluno específico da lista de doutorandos
   5) Listar os códigos dos alunos que estão na lista de mestrandos
   6) Listar os códigos dos alunos que estão na lista de doutorandos
   7) Pesquisar se um aluno específico está na lista de mestrandos
   8) Pesquisar se um aluno específico está na lista de doutorandos
   9) Encerrar o programa

O programa deverá ler a opção informada pelo usuário e executar a operação selecionada. O programa deve encerrar quando o usuário escolher a opção 9.

Para o desenvolvimento desse programa deve ser utilizada a estrutura de dados Lista flexível simples e dois objetos Lista deverão ser instanciados no método main. Além disso, devem ser criados dois novos métodos na classe Lista:

- remover: o método deve recebe como parâmetro o código de um aluno (int), e deve removê-lo da lista. O método deve retornar *true*, se o código do aluno for removido. Caso o código não conste na lista, o método deve retornar *false*.

- pesquisar: recebe como parâmetro o código de um aluno (int). O método deve retornar *true*, se o código constar na lista, e retornar *false*, caso contrário.

  Nesse exercício será necessário implementar as classes: Celula, Lista e Teste

Lista flexível duplamente encadeada

1) Mostre o passo a passo da execução dos seguintes métodos (similar ao passo a passo apresentado nos slides que mostram a execução do método *inserirInicio*):
   a) método removerInicio() – Exercício do Slide 20
   b) método removerFim() – Exercício do Slide 23
   c) método inserir(int x, int pos) – Exercício do Slide 26
   d) método remover(int pos) – Exercício do Slide 29
2) Crie um programa que permita que um usuário gerencie sua lista de músicas. Para tanto, o programa deverá apresentar para o usuário um menu com as seguintes opções:

Menu:
1. Inserir uma música no final da lista
2. Inserir uma música no início da lista
3. Inserir uma música numa posição específica da lista
4. Remover a música do início da lista
5. Remover a música do final da lista
6. Remover uma música de uma posição específica da lista
7. Remover uma música específica
8. Listar todas as músicas da lista
9. Listar todas as músicas da lista na ordem inversa
10. Pesquisar uma música na lista
11. Pesquisar música anterior
12. Pesquisar música posterior
13. Encerrar o programa

O programa deverá ler a opção informada pelo usuário e executar a operação selecionada. Em seguida o programa deverá apresentar novamente o menu para o usuário, ler e executar a operação selecionada. Esse processo deverá ser repetido até que o usuário digite a opção 13.

Para implementar esse programa, implemente as classes CelulaDupla, ListaDupla e Teste.

Os seguintes métodos devem ser incluídos na classe ListaDupla:

- remover: esse método deverá receber como parâmetro o nome da música a ser removida da lista. Caso, a música conste na lista, a remoção deve ser feita e o método deverá retornar ***true***, caso contrário deverá retornar ***false***.
- mostrarInverso: método sem parâmetros, que deverá imprimir a lista na ordem inversa, isto é, da última música inserida até a primeira.
- pesquisarMusica: esse método deverá receber como parâmetro o nome da música a ser pesquisada na lista. Caso, a música conste na lista, o método deverá retornar ***true***, caso contrário deverá retornar ***false***.
- pesquisarAnterior: esse método deverá receber como parâmetro o nome da música a ser pesquisada na lista. O método deverá retornar a música anterior a música pesquisada (isto é, a música da lista que está logo antes da música pesquisada). Caso a música pesquisada não conste na lista, ou caso essa música seja a primeira da lista, deve ser retornada uma String vazia.
- pesquisarPosterior: esse método deverá receber como parâmetro o nome da música a ser pesquisada na lista. O método deverá retornar a música posterior a música pesquisada (isto é, a música da lista que está logo depois da música pesquisada). Caso a música a ser pesquisada não conste na lista, ou caso essa música seja a última da lista, deve ser retornada uma String vazia.