



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Departamento de Sistemas e Computação

Graduação em Ciência da Computação

Exercício sobre listas ligadas

Objetivo: Praticar a implementação de listas ligadas **sem recursão**.

Relembre o conceito de lista simplesmente e duplamente ligada visto em sala de aula.

Atividades necessárias antes de iniciar o exercício:

1. Crie um projeto no Eclipse chamado LEDA, por exemplo (pode ser qualquer outro nome que lhe convier);
2. Descompacte o arquivo baixado (exceto o PDF) na pasta dos fontes (normalmente **src**) do seu projeto LEDA criado no seu workspace. O arquivo baixado tem a seguinte estrutura:
 - adt
 - linkedList
 - SingleLinkedListNode.java (CLASSE REPRESENTANDO UM NÓ DE UMA LISTA SIMPLES)
 - DoubleLinkedListNode.java (CLASSE REPRESENTANDO UM NÓ DE UMA LISTA DUPLA)
 - LinkedList.java (INTERFACE DA LISTA LIGADA)
 - DoubleLinkedList.java (INTERFACE DA LISTA DUPLAMENTE LIGADA)
 - SingleLinkedListImpl.java (IMPLEMENTAÇÃO PARCIAL DA LISTA SIMPLES)
 - DoubleLinkedListImpl.java (IMPLEMENTAÇÃO PARCIAL DA LISTA DUPLA)
 - stack
 - Stack.java
 - StackOverflowException.java.
 - StackUnderflowException.java.
 - StackDoubleLinkedListImpl.java (IMPLEMENTAÇÃO PARCIAL DE UMA PILHA USANDO UMA LISTA DUPLA)
 - queue
 - Queue.java
 - QueueOverflowException.java.
 - QueueUnderflowException.java.
 - QueueDoubleLinkedListImpl.java (IMPLEMENTAÇÃO PARCIAL DE UMA FILA USANDO UMA LISTA DUPLA)

No Eclipse, selecione a pasta dos fontes no projeto LEDA e faça um refresh (apertar F5). Note que deve aparecer os arquivos mencionados acima.

Agora você está pronto para começar a trabalhar nas seguintes atividades:

1. Observe a interface `LinkedList.java`. Ela descreve os serviços de uma lista genérica.
2. Observe a interface `DoubleLinkedList.java`. Ela descreve os serviços de uma lista duplamente encadeada genérica.
3. Observe também a existência implementação incompleta `SingleLinkedListImpl`. Você precisa implementar os métodos incompletos (*sem recursão*). Use a abordagem iterativa na implementação.

4. Observe também a existência implementação incompleta `DoubleLinkedListImpl`. Você precisa implementar os métodos incompletos (*sem recursão*). Use a abordagem iterativa na implementação. Note a herança existente entre as classes.
5. Observe também a existência implementação incompleta `StackDoubleLinkedListImpl`. Ela representa uma pilha implementada usando uma lista dupla internamente. Você precisa implementar os métodos incompletos.
6. Observe também a existência implementação incompleta `QueueDoubleLinkedListImpl`. Ela representa uma fila implementada usando uma lista dupla internamente. Você precisa implementar os métodos incompletos.
7. Concentre-se em implementar conforme descrito na interface e pense em cenários para testar suas implementações de lista simples e dupla antes de enviar. Alguns cenários interessantes são: testes em uma lista vazia e em uma lista não vazia, nas inserções e remoções verificar se o tamanho da lista é alterado corretamente, após operações de inserção e remoção verificar se a estrutura interna (envolvendo apontadores) está correta, etc.

Instruções para o envio

Ao terminar o exercício, faça os seguintes passos:

1. Compacte a pasta **adt** que existe nos fontes de seu projeto LEDA (**src**) e retire a classe `TestLists.java` desse arquivo compactado. A compactação DEVE ser feita a partir do diretório raiz de seus fontes de forma a preservar a estrutura de pastas que refletem a estrutura dos pacotes (package) Java. Por exemplo, você deve ter um arquivo compactado `NOME_COMPLETO_DO_ALUNO.ZIP` com a seguinte estrutura:
 - adt
 - linkedList
 - `SingleLinkedListImpl.java`
 - `DoubleLinkedListImpl.java`
 - stack
 - `StackDoubleLinkedListImpl.java`
 - queue
 - `QueueDoubleLinkedListImpl.java`
2. Envie esse arquivo com sua solução para o sistema de submissão e verifique que o contador de submissões será alterado.

Observações finais:

- **A interpretação do exercício faz parte da atividade.**
- **A atividade é individual. A conversa entre alunos é proibida.**
- **É proibido coletar códigos prontos e adaptar. Implemente as questões. Isso é para seu aprendizado.**
- **Caso você observe qualquer problema no sistema de submissão, contate o professor imediatamente.**
- **Se você não compactar o arquivo seguindo a estrutura de diretórios a compilação não terá sucesso e o sistema mostrará isso. Erro de compactação serão de responsabilidade do aluno. O professor não ajudará o aluno nesse item. É só seguir as instruções deste arquivo.**