Suponha que você necessite chegar até o penúltimo nodo de uma lista encadeada para substituir seu conteúdo pelo recebido por parâmetro.

Mostre como vocês poderia realizar esse procedimento sobre uma lista duplamente encadeada e qual a vantagem em relação à lista simplesmente encadeada, especificamente para este procedimento.

Poste sua solução e comente a dos colegas!

Para dúvidas, utilize o fórum de dúvidas deste módulo.

Em uma lista simplesmente encadeada precisaríamos iterar através da lista inteira até chegarmos ao penúltimo elemento para trocar seu conteúdo. Na lista duplamente encadeada esse procedimento é mais performático já que nela os nodos possuem referência ao nodo anterior da lista. Com isso podemos iniciar a iteração pelo último nodo e chegar ao penúltimo mais rapidamente. O seguinte método, se incluído à classe List, pode realizar essa tarefa.

public void trocaPenultimo(Object obj) throws EmptyListException{

if ( isEmpty() ) {

throw new EmptyListException( name );

}

Node node = lastNode;

node = node.getPrevious();

if ( node == null ) {

throw new EmptyListException( name );

}

node.setData( obj );

}

Obs: o correto seria criar outro tipo exceção para tratar o segundo erro, já que nesse caso a lista não está vazia. O UnderflowException visto nos módulos anteriores seria mais apropriado.