

UNIVERSIDADE AUTÓNOMA DE LISBOA LUÍS DE CAMÕES
Departamento de Ciências e Tecnologias
Programação Orientada a Objetos 2020/2021
Laboratório 1

Objetivos

Pretende-se a implementação da estrutura de dados *lista simplesmente ligada*, em Java.

Uma lista simplesmente ligada é suportada por um conjunto de nós que guardam os vários elementos da lista, e se associam entre si. Cada nó conhece o próximo nó na lista.

A lista mantém o primeiro nó (i.e., raiz), o número de elemento, e, eventualmente, o último nó.

Datas

- Enunciado: 12 de outubro de 2020;
- Entrega no *e-learning*: 25 de outubro de 2020, 23:59;
- Publicação de nota: 01 de novembro de 2020.

Regras

O trabalho deve ser realizado por grupos de 4 elementos, previamente registados no *e-learning*.

A entrega do trabalho é feita no *e-learning*. Deve ser entregue um ficheiro **zip** com o seguinte conteúdo:

- Diretório **src** com todo o código fonte;
- Diretório **doc** com eventuais diagramas e outra documentação que considerar relevante.

A entrega não será válida se for utilizado outro formato de compressão além do zip, ou se não for possível abrir o ficheiro.

Tarefas

Parte 1

1. Construir a classe nó de lista simplesmente ligada, com o nome `SingleNode`;
 - (a) Deve incluir os seguintes métodos:
 - i. Construtor;
 - ii. Obter elemento (`getElement()`);
 - iii. Obter o próximo nó (`getNext()`).

Parte 2

1. Construir a classe lista simplesmente ligada, com o nome `SinglyLinkedList`;
 - (a) Deve incluir os seguintes métodos:
 - i. Construtor;
 - ii. Obter o primeiro elemento (`getFirst()`);
 - iii. Obter o último elemento (`getLast()`);
 - iv. Obter o elemento na posição n (`get(int position)`);
 - v. Inserir elemento no início da lista (`insertFirst(Object element)`);
 - vi. Inserir elemento no final da lista (`insertLast(Object element)`);
 - vii. Inserir elemento na posição n da lista (`insert(Object element, int position)`);
 - viii. Remover elemento do início da lista (`removeFirst()`);
 - ix. Remover elemento do final da lista (`removeLast()`);
 - x. Remover elemento da posição n da lista (`remove(int position)`).

Parte 3

1. Construir testes unitários para o nó e lista.
 - (a) Deve testar todos os métodos das partes 1 e 2.

Parte 4

1. Construir uma classe para o iterador da lista, com o nome `SinglyLinkedListIterator`;
 - (a) Deve incluir os seguintes métodos:
 - i. Verificar se existe próximo elemento (`hasNext()`);
 - ii. Avançar para o próximo elemento (`next()`);
 - iii. Recomeçar (`rewind()`);
2. Adicionar método à lista para obter um iterador.

Parte 5

1. Atualizar os testes unitários da lista para incluir o método gerador de iterador;
2. Construir testes unitários para o iterador.