

NOTA: Esta ficha está dividida em duas partes (Parte I e Parte II), pretende-se que a primeira parte seja seguida com os slides da aula teórico-prática. Na segunda parte pretende-se que o aluno consiga realizar os exercícios pondo em prática a matéria abordada nos slides e praticada na Parte I. Devem ser consultados os slides: 2019_ED_Aula06.pdf para a realização desta ficha prática.

Parte I

Exercício 1

Implementar uma Lista Ordenada (`ArrayOrderedList`) recorrendo a um *array*. Lembre-se que terá de implementar a interface adequada.

Preste atenção aos slides da aula teórica principalmente o número 15, e veja o diagrama de interfaces. Quantas classes serão necessárias para implementar o diagrama que lá está presente?

Outro problema muito importante com que se irá debater será na criação do iterador. Para este exercício não considere as modificações concorrentes à estrutura de dados subjacente.

Demonstre a utilização da `ArrayOrderedList` para um cenário à sua escolha.

Exercício 2

Implementar uma Lista Ordenada recorrendo a uma lista duplamente ligada (`DoubleLinkedOrderedList`). Demonstre a utilização da `DoubleLinkedOrderedList` para um cenário à sua escolha.

Exercício 3

Os cenários usados nos exercícios anteriores (1 e 2) forma adicionados elementos não comparáveis? O que aconteceu?

Exercício 4

Definir a interface `UnorderedListADT` e implementar uma Lista Não Ordenada recorrendo a um *array* (`ArrayUnorderedList`). Demonstre a utilização da `ArrayUnorderedList` para um cenário à sua escolha.

Exercício 5

Implementar e testar uma Lista Não Ordenada recorrendo a uma lista duplamente ligada (`DoubleLinkedUnorderedList`).

Parte II

Exercício 1

Porquê que não existe uma método `add(Element)` na interface `ListADT`?

Exercício 2

Dada a versatilidade de uma lista porquê que não é sempre a escolhida em vez de uma `Stack` ou de uma `Queue`?

Exercício 3

Criar um método na `DoubleLinkedListOrderedList` que inverta e devolva todos os elementos no tipo de dados abstracto que entender.

Exercício 4

Suponha que apenas tem uma `DoubleLinkedListUnorderedList` e que pretende desenvolver um programa que precisa obrigatoriamente de um simples *array* para operar. Demonstre este simples cenário.

Exercício 5

A fórmula 1 é a prova de automobilismo mais conhecida e com mais prestígio a nível mundial. O campeonato mundial é constituído por diversas provas, equipas e corredores. Cada um dos corredores mediante a classificação em cada uma das provas irá obter pontos que irão formar a tabela classificativa e no final apurar o vencedor. Pretende-se que crie um programa que faça a simulação de uma *pole position*.