

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.Aula05.pdf para a realização desta ficha prática.

Parte I

Exercício 1

Implementar a `LinkedList` como sugerido nos slides da aula teórica (*slide 17*). Demonstre a utilização da `LinkedList` para um cenário à sua escolha.

Exercício 2

Implementar a `CircularArrayQueue` como é sugerido no *Slide 29*. Demonstre a sua utilização no mesmo cenário usado no exercício anterior.

Exercício 3

O que foi necessário fazer para adaptar o cenário do exercício 1 para o cenário do exercício 2? Enumere todas as alterações.

Exercício 4

Implementar o programa para a codificação de mensagens com recurso a uma `CircularArrayQueue` como é sugerido no *Slide 13*.

Exercício 5

Implementar o mesmo programa para a codificação de mensagens (do exercício anterior) com recurso a uma `LinkedList`.

Parte II

Exercício 1

Porquê que usamos uma implementação em *array* circular? Porque não um *array* normal como numa *stack*?

Exercício 2

Qual o *Big O* para a operação *dequeue* nas diferentes implementações de um *Queue*: *array*, *array* circular e lista ligada?

Exercício 3

Implementar um programa que dadas duas *Queues* ordenadas as junte numa também ordenada.

Exercício 4

Implementar uma nova *Queue* que obedeça à interface `QueueADT` através de duas *Stacks*.

Exercício 5

Dadas **N** *strings*, criar **N** *Queues* cada uma contendo uma das *strings*. De seguida criar uma *Queue* das **N** *Queues*. Repetidamente aplicar uma operação de junção ordenada às primeiras duas *Queues* e reinserir a nova *Queue* no final. Repetir até que a *Queue* contenha apenas uma *Queue*.