

Enquadramento
Aulas Teóricas
Aulas Práticas
Apontamentos
Útil
Teste Prático



[ED193] Removendo todas as ocorrências de um elemento



Código Base

Use como base a classe `SinglyLinkedList<T>` (descrita no exercício 1 da aula prática 06 - não esquecer da classe `Node`) que implementa uma lista ligada simples e tem disponíveis métodos para adicionar ou remover um elemento no início ou no final, devolver o tamanho, saber se a lista está vazia ou retornar representação em *string* para escrita (tal como dado nas aulas).

O problema

Acrescente à classe dada um novo método `public void removeAll(T value)` que **remove da lista todas as ocorrências do valor *value***. Note que para comparar os elementos deverá usar o método `equals` e não um simples `==` (pode assumir que os objectos da lista têm implementada uma versão correcta do `equals`).

Submissão no Mooshak

Se submeter no Mooshak, deverá submeter apenas a classe `SinglyLinkedList<T>`, acrescentando o método `removeAll` como pedido (**e sem apagar nenhum dos outros métodos dados como base**). Pode assumir que terá acesso no Mooshak à classe `Node<T>` (não a pode mudar) e se precisar pode criar outros métodos auxiliares. O Mooshak irá criar várias instâncias da sua classe e irá fazer uma série de testes ao método por si implementado.

Exemplos de Input/Output

Lista inicial	Chamada	Estado da lista depois da chamada
list = {1,2,2,2,1,3,4,2,1}	list.removeAll(1)	list = {2,2,2,3,4,2}
list = {1,2,2,2,1,3,4,2,1}	list.removeAll(2)	list = {1,1,3,4,1}
list = {1,2,2,2,1,3,4,2,1}	list.removeAll(3)	list = {1,2,2,2,1,4,2,1}
list = {1,2,2,2,1,3,4,2,1}	list.removeAll(5)	list = {1,2,2,2,1,3,4,2,1}
list = {"cc","cc","cc","cc","cc"}	list.removeAll("cc")	list= {}

Estruturas de Dados (CC1007)

list = {1,2,2,2,1,3,4,2,1}	list.removeAll(2)	list = {1,1,3,4,1}
list = {1,2,2,2,1,3,4,2,1}	list.removeAll(3)	list = {1,2,2,2,1,4,2,1}