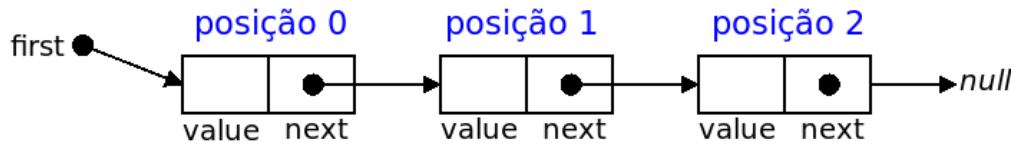


Para efeitos da nota atribuída à resolução de exercícios ao longo do semestre - **Submeter até 23:59 de 10 de Abril** (o problema continuará depois disponível para submissão, mas sem contar para a nota)
[para perceber o contexto do problema deve [ler o guião da aula #06](#)]

[ED190] Cópia de uma lista

Neste problema deverá apenas submeter uma classe **SinglyLinkedList<T>** (e não um programa completo).



Código Base

Use como base a classe **SinglyLinkedList<T>** ([ver código](#) | download de [Node.Java](#) e [SinglyLinkedList.Java](#)), que representa 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 singlyLinkedList<T> copy()** que **cria e devolve uma nova lista que é uma cópia exacta da lista para a qual foi chamado o método**. Note que o retorno tem de ser uma **nova lista**, contendo um novo conjunto de objectos do tipo **Node<T>** que guardam valores iguais aos da lista inicial (não precisa de criar cópias dos valores, cada nó da nova lista deve apontar para a referência respectiva da lista inicial).

Submissão

Deverá submeter apenas a classe **SinglyLinkedList<T>**, acrescentando o método **copy** 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	Nova Lista
list = {2,4,6,8}	newList = list.copy()	newList = {2,4,6,8}
list = {}	newList = list.copy()	newList = {}
list = {"estruturas","de","dados"}	newList = list.copy()	newList = {"estruturas","de","dados"}