

## 1 Objectivos

Pretende-se o uso e implementação de listas ligadas. Devem os alunos apresentar uma implementação de listas ligadas, e posteriormente, usar a própria implementação para criar um pequeno editor de texto. Não devem os alunos sem autorização do professor fazer uso de classes do package "java.util", excepto para input/output. O editor permitirá um conjunto reduzido de operações, tais como inserir texto no fim duma linha, avançar com o cursor para a linha anterior, procurar texto, etc.

## 2 O Trabalho

O trabalho consiste na apresentação de 2 classes Java: uma com a implementação do editor a outra com a implementação das listas duplamente ligadas. Os nomes das classes devem respeitar o definido na tabela abaixo.

Funcionalidade	Interface	Implementação
Listas duplamente ligadas iteráveis	-	DoubleLinkedList.java
Editor de Texto	-	LineEditor.java

Table 1: Classes a implementar

### 2.1 Listas

Apesar das listas duplamente ligadas serem apresentadas numa única classe, devem providenciar a implementação dum iterador. O iterador deverá providenciar iteração para a frente e para trás e deve a iteração ser compatível com *ListIterator*. Após a implementação das listas duplamente ligadas, nas condições enunciadas, pode avançar para a implementação do editor de linhas. É uma decisão de implementação(sua!) implementar as operações opcionais de *ListIterator*. Consulte a api de *ListIterator* em <https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html>.

### 2.2 O Editor

O Editor de linhas irá permitir a edição de linhas de texto, devendo portanto apresentar as funcionalidades habituais a este tipo de programas. O editor será implementado com uma *DoubleLinkedList*: cada linha de texto será um nó da lista ligada, existindo um cursor, que indica qual a linha a que serão aplicados os comandos, quando não se indica o nº de linha. Os comandos aos quais deve responder o editor são os que seguidamente se apresentam:

1. insertEnd "text": insere o texto "text" no fim do documento
2. insert [n] "text" : insere o texto "text" na n-ésima linha, se for omissa o nº de linha, usa a linha correspondente ao cursor

3. delete [n] : apaga a n-ésima linha de texto, se for omissa o nº de linha, usa a linha correspondente ao cursor
4. edit [n] "text": substitui o conteúdo da linha n pelo texto "text", se for omissa o nº de linha, usa a linha correspondente ao cursor
5. print : lista o documento, com os números das linhas, e a posição do cursor de linha
6. search "text" : lista o número da linha e o texto da linha que contem o texto "text".
7. lineUp() : o cursor de linha vai para a linha acima
8. lineDown() : o cursor de linha vai para a linha abaixo

Todos os comandos em que os números de linha saiam fora dos limites existentes devem ser ignorados.

## **2.3 Entrega**

O trabalho pode ser realizado individualmente ou em grupos de 2 alunos. Os alunos devem manter os mesmos grupos que formaram para a realização do trabalho 1 (sempre que possível). A data limite para a submissão do trabalho é dia 6 de Novembro de 2018, sendo realizada a submissão pelo moodle, nos moldes habituais. Os trabalhos serão apresentados e avaliados na aula prática de dia 6. Todos os ficheiros deverão ser "zipados" e submetidos num único ficheiro com o número dos alunos que realizam o trabalho.