Trabalho Prático

Regulamento:

Nº de Alunos por grupo:	2
Peso do Trabalho:	40% da nota final
Data Limite para definição de grupos e temas:	8 de Janeiro de 2016
Data Limite de Entrega:	29 de Janeiro de 2016
Nota importante:	A utilização de material já desenvolvido, ou tentativa de plágio implicará uma classificação final do trabalho de **zero valores.** Aos trabalhos que aparentem ser idênticos será atribuída a classificação final de **zero valores**.

Inscrição no trabalho prático:

Cada grupo deverá efectuar a sua inscrição no trabalho prático, preenchendo o formulário disponível na *Moodle* para o efeito.

Material a Entregar:

1. Código fonte
2. Relatório (max. 2 pág., pdf) com explicação das técnicas utilizadas e justificação das opções tomadas. (fonte: Times New Roman, tam: 11, espaçamento 1,5).

Descrição do Trabalho:

Pretende-se com este trabalho criar um programa de manipulação de texto. O texto é proveniente de ficheiros de texto. O programa deverá englobar as seguintes funcionalidades:

1) Contar o total de palavras existentes no texto (números não contam).
2) Calcular e apresentar a frequência de cada palavra no texto (absoluta e relativa).
3) Detectar e apresentar as palindromes e capicuas existentes no texto.
4) Ordenar alfabeticamente cada palavra do texto e gravar em ficheiro o resultado.

A gestão da leitura e escrita de ficheiros é já fornecida pelo docente, e é implementada pela classe FileUtils.java, que contém dois métodos:

```
public static String lerFicheiroTexto(String caminhoParaFicheiro)
public static boolean guardarEmFicheiro(String caminhoParaFicheiro, String texto)
```

O primeiro permite ler o ficheiro especificado pelo parâmetro de entrada. O segundo permite escrever o texto de uma String para um ficheiro especificado no parâmetro de entrada.
Exemplo de utilização:

```
// lê o ficheiro.txt e guarda o seu conteúdo na String oTexto.
String oTexto = FileUtils.lerFicheiroTexto("c:\\ficheiro.txt");

// gravar o conteúdo da String outroTexto no ficheiro resultado.txt
String outroTexto = "Olá :)";
FileUtils.guardarEmFicheiro("c:\\resultado.txt", outroTexto)
```

Métodos de String úteis para a construção de uma solução para o problema proposto:

`String[] split(String exp)`	Transforma uma String num array de strings, considerando como separador o parâmetro `exp`. Ex: `String [] s = umaString.split(".")`
`int compareToIgnoreCase(String str)`	Compara duas Strings ignorando maiúsculas e minúsculas e devolve `0` se as Strings forem iguais Ex: `int res = umaString.compareToIgnoreCase(outraString);`

`char charAt(int index)`	Retorna o caracter posicionado no índice pelo parâmetro `index`. Ex:
	`String s= "trabalho prático".`
	`char c = s.charAt(2);  // c='a' ;`

`String replace(String target, String replacement)`	Substitui todas as ocorrências de `target` pela string `replacement`.
	Ex:
	`String s= "ola";`
	`s=s.replace("a","e"); // s= ole`

Exemplo de funcionamento:

Dado o seguinte conteúdo do ficheiro `ftexto.txt`:

```
Os sapatos são escuros, os atacadores são azuis, a mensagem é clara. A Ana corta sebes 121.
```

```
******** MENU ***********
1 - Contar Palavras
2 - Frequências globais
3 - Palindromes e Capicuas
4 - Ordenar palavras
5 - Sair
Opção: 1

Total de palavras:16


******** MENU ***********
1 - Contar Palavras
2 - Frequências globais
3 - Palíndromes e Capicuas
4 - Ordenar palavras
5 - Sair
Opção: 2

Os:2(13%)
sapatos:1(6%)
são:2(13%)
escuros:1(6%)
atacadores:1(6%)
azuis:1(6%)
a:2(13%)
mensagem:1(6%)
é:1(6%)
clara:1(6%)
Ana:1(6%)
corta:1(6%)
sebes:1(6%)
121:1(6%)

Prima enter para continuar...
```

```
******** MENU ***********
1 - Contar Palavras
2 - Frequências globais
3 - Palindromes e Capicuas
4 - Ordenar palavras
5 - Sair
Opção: 3

Palíndromes detetadas:
#1- a
#2- é
#3- A
#4- Ana
#5- sebes

Capicuas detetadas:
#1- 121
```

```
******** MENU ***********
1 - Contar Palavras
2 - Frequencias globais
3 - Palindromes e Capicuas
4 - Ordenar palavras
5 - Sair
Opção: 3

Os aaopsst osã ceorssu,  os aaacdeorst osã aisuz, a aeegmmns é  aaclr. A
Aan acort beess 112.
```