Lab I.

Objetivos

Os objetivos deste trabalho são:

- Rever e aplicar conceitos de programação adquiridos anteriormente: arrays bidimensionais, genéricos, ciclos for-each, tipos enumerados.
- Rever e praticar técnicas de desenvolvimento de software: implementar uma especificação de classe, programa com múltiplos componentes, e ficheiros JAR

I.1 Word Search Solver

O objetivo deste trabalho é escrever um programa em JAVA para resolver *Sopas de Letras*. A entrada do programa é um único ficheiro de texto contendo o puzzle e as palavras a encontrar. Exemplo (poderá pesquisar outros online):

```
stackjcpaxlf #isto é um comentário
ylkwuggtestl
lnjsuncuxzpd
etofqikicfng
senilmjfumrk
zbuuomsbskcy
sumtrasarzix
rbmwwrjdaxvf
jejhqgsdraib
acwezolmzolt
viuqvramdgwh
agftwpjzwumh
Programming;Java;Words Lines Civil
Test;Stack;
```

A saída é a lista de palavras, bem como a posição em que se encontram no puzzle.

(a) Requisitos de Entrada

O programa deve verificar se:

1. O puzzle é sempre quadrado, com o tamanho exato de 15x15.
2. As letras do puzzle estão em minúsculas.
3. Na lista, as palavras tem a primeira letra em maiúsculas.
4. O caracter # sinaliza um comentário.
5. As palavras são compostas por caracteres alfabéticos.
6. No puzzle e na lista de palavras, o ficheiro não pode conter linhas vazias.
7. Cada linha pode ter mais do que uma palavra, separadas por vírgula, espaço ou ponto e vírgula.
8. As palavras têm de ter pelo menos 3 caracteres.
9. Todas as palavras da lista têm de estar no puzzle e apenas uma vez.
10. A lista de palavras pode conter palavras com partes iguais (por exemplo, pode conter FARO e FAROL). Nestes casos deve ser considerado apenas a maior (FAROL).

(b) Requisitos de Saída

A lista de palavras do puzzle retornadas pelo WSSolver tem que estar na mesma ordem das palavras passadas na lista. As palavras têm de estar em maiúsculas.

(c) Exemplo de Execução

O programa deverá ser testado com vários ficheiros, verificando os requisitos. Abaixo, mostra-se um exemplo de execução com os dados anteriores:

```
$ java WSSolver sdl_01.txt

programming     11      12,6    Up
java            4       9,1     Down
words           5       11,11   UpLeft
lines           5       5,5     Left
civil           5       6,11    Down
test            4       2,8     Right
stack           5       1,1     Right

S T A C K _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ G _ T E S T _ _ _ _
_ _ _ _ _ N _ _ _ _ _ _ _ _ _
_ _ _ _ _ I _ _ _ _ _ _ _ _ _
S E N I L M _ _ _ _ _ _ _ _ _
_ _ _ _ _ M _ _ _ _ C _ _ _ _
_ _ _ _ _ A S _ _ _ I _ _ _ _
_ _ _ _ _ R _ D _ _ V _ _ _ _
J _ _ _ _ G _ _ R _ I _ _ _ _
A _ _ _ _ O _ _ _ O L _ _ _ _
V _ _ _ _ R _ _ _ _ W _ _ _ _
A _ _ _ _ P _ _ _ _ _ _ _ _ _
```

Juntamente com o código, deverão ser entregues 3 exemplos de execução, i.e., 3 ficheiros de entrada (*sopa01.txt,* ..) e os respetivos ficheiros de saída (*sopa01_result.txt,* ..).

I.2 Word Search Generator

Escreva o programa WSGenerator, que crie uma *Sopa de Letras* de acordo com o formato e requisitos anteriores. O programa deve receber como parâmetro de entrada um ficheiro com a lista de palavras, e o nome de um ficheiro para guardar a *Sopa de Letras*.

(a) Exemplo de Execução

Assumindo que o ficheiro "*wordlist1.txt"* contém a lista de palavras (uma por linha, ou uma lista por linha).

```
$ java WSGenerator -w wordlist1.txt
kfbofyzkzuycztp
xowythojvmztavw
iwckephiafzwsio
rmjnsrkmroxvtlj
shnhtopfbmnmasa
mkwirgbtswmocyv
ihqdxroebvhfkfa
yvwgyargicuxkst
naolmmlxcirdyaa
```

```
nirqimiqbvjhyif
nhdqlinkvilazgx
dssurneztcmlhya
hiyhzgsdbpkhazs
hvuvbgznqjhjdyp
zdsbtmlqfkjsjtj
Programming
Java
Words
Lines
Civic
Test
Stack
```

```
$ java WSGenerator -w wordlist1.txt -s wordlist1_result.txt
```

O resultado é o mesmo do anterior, mas guardado no ficheiro "*wordlist1_result.txt*". Junto com o código, deve entregar 3 exemplos de ficheiros de palavras (*wl01.txt, wl02.txt, wl03.txt*) e os respetivos resultados (*wl01_result.txt,* ...).

Nota importante: para cada guião prático, deverá ser usada no *git* uma nomenclatura uniforme (*lab01, lab02, lab03,…*) para permitir uma identificação mais fácil dos projetos.

Bom trabalho!