

## *Tdm 6*

**Exercice 1.** Le but de cet exercice est d'écrire une application Java en ligne de commande qui permet d'afficher les lignes d'un fichier contenant un certain mot. Par exemple, voici la liste des lignes qui contiennent le mot "public" dans un fichier UnRectangle.java...

On lance la commande :

```
java Grep public UnRectangle.java
```

Et on obtient le résultat :

```
(UnRectangle.java,1) : public class UnRectangle
```

```
(UnRectangle.java,4) : (public UnRectangle(UnPoint cig,int l,int h)
```

```
(UnRectangle.java,10) : public void translation(int dx,int dy)
```

Dans la parenthèse (x :y), on trouve en premier x, le nom du fichier et en deuxième y, l'indice de la ligne.

La commande à utiliser a la forme générale : `java Grep mot fichier1 fichier2 ...`

Remarque : lorsqu'un programme Java est lancé avec des arguments en ligne de commande, ces arguments sont disponibles sous forme de chaînes de caractères placées dans le tableau paramètre de la méthode main. Dans l'exemple qui précède, si l'entête de la méthode principale est **public static void main(String[] args)** alors lors de l'exécution, `args.length` vaut 3, ce qui indique qu'il y a trois arguments en ligne de commande, `args[0]` est la chaîne "mot", `args[1]` est la chaîne "fichier1" et `args[2]` est la chaîne "fichier2".

Créer une classe **Grep** contenant un main capable d'effectuer les opérations décrites ci-dessus.

**Exercice 2.** Ecrire une classe sort dont le main prend pour argument un nom de fichier texte et effectue une copie de ce fichier avec les lignes triées en utilisant l'ordre lexicographique.

Exemple :

Le fichier A est comme suit :

```
this is the end
oh, the end
my only friend
```

Exécuter la commande `java Sort A` doit écrire un fichier Acopie comme suit :

```
my only friend
oh, the end
this is the end
```

**Exercice 3.** Parce que jamais deux sans trois... Mais on a déjà bossé suffisamment et je n'ai plus d'idées.

THE END