

Exercice 1 (reprises du TD9). Écrire une méthode statique qui prend en argument une chaîne contenant une balise xml (voir exemple ci-dessous), et qui renvoie une `HashMap<String,String>` contenant ses attributs et leurs valeurs. Par exemple la balise

```
<person nom="Marsault" yeux="vert" cheveux="brun clair">
```

a trois attributs `nom`, `yeux` et `cheveux` qui ont pour valeurs respectives les chaînes `Marsault`, `vert` et `brun_clair`. La méthode renvoie donc la table de hachage qui associe `nom` à `Marsault`, etc. (Attention aux guillemets!).

Exercice 2. Écrire une méthode statique

```
String remplaceGroupe(String chaine, int groupNb, Pattern p, String remplacement)
```

qui confronte la chaîne `entree` à l'expression `p` et, si elles correspondent, renvoie la chaîne dans laquelle le groupe `groupNb` est remplacé par `remplacement`.

Par exemple, si les argument valent

```
— chaine= "victor.marsault@marsault.fr"
— groupNb= 1
— p= Pattern.compile("[^@]*@[a-z.]*.fr")
— remplacement= "liafa.univ-paris-diderot"
```

alors le résultat doit être `"victor.marsault@liafa.univ-paris-diderot.fr"`. (Notez que la chaîne extraite par le groupe est `"marsault"`, et que sa première occurrence n'est donc pas remplacée.)

Exercice 3. Écrire une méthode statique `racinePrimitive(String)` qui renvoie la *racine primitive* du mot donné en argument. Un mot est la puissance (entière) d'un autre mot `u` s'il est la concaténation de plusieurs fois ce mot `u`. La racine primitive d'un mot donné est le plus petit mot dont il est la puissance. Exemples :

- La racine primitive de `"aaaa"` est `"a"` celle de `tchatchatcha` est `tcha` et celle de `"agaragar"` est `"agar"`.
- Les mots `aaba` et `victor` ne sont pas les puissances d'autres mots; chacun d'entre eux est donc sa propre racine primitive.

Exercice 4. Écrire une méthode statique qui prend une chaîne quelconque et remplace n'importe quel nombre de caractères blancs (`\\s`) par un seul espace `"_"`.

Exercice 5. Écrire une méthode statique `etoile` qui prend un `Pattern` et une `String`. Cette méthode renvoie une liste de `String` correspondant à toutes les occurrences du pattern s'il était mis à l'étoile. Par exemple, si l'on considère l'expression `"((foo)|(bar))"` et la chaîne `"foobarbar_pasfoo"`, alors cette fonction doit renvoyer la liste `[foo,bar,bar]`.

Exercice 6 (Découpage intelligent). Écrire une méthode statique `smartSplit` qui prend un caractère séparateur et une chaîne et renvoie une liste de chaîne. Cette méthode sépare la chaîne selon le séparateur mais ignore les caractères séparateurs s'ils sont immédiatement précédés d'un antislash; ces antislash doivent être retirés du résultat final (mais seulement s'ils précèdent un caractère séparateur). Par exemple, découper la chaîne

```
liste:avec:des:deux points\\:::\\
```

selon le caractère séparateur `':'` renvoie la liste

```
[ liste, avec, des, deux points:, :\\ ]
```