

LES EXPRESSIONS RATIONNELLES

Les expressions rationnelles (aussi appelées expressions régulières qui est une traduction de *regular expressions* (regex)) représentent des ensembles de chaînes de caractères à l'aide de caractères ordinaires et de symboles spéciaux appelés méta-caractères.

Elles nous serviront essentiellement dans deux cas :

- recherche de chaînes de caractères correspondant à un motif dans des fichiers, base de données, etc.
- validation de données saisies par l'utilisateur c'est à dire que les expressions régulières nous serviront à vérifier que les données saisies respectent un certain format (nombre, date, etc.)

Pour les mettre en œuvre nous utiliserons la méthode *matches* (qui signifie correspondre) de la classe *String* qui prend en paramètre le motif à vérifier et qui renvoie vrai si le motif est contenu dans la chaîne.

Les expressions rationnelles sont très puissantes mais elles peuvent être complexes à écrire et à comprendre.

Exemple :

```
String chaine = new String("le ciel est bleu");
System.out.println(chaine.matches(".*est.*")); // renvoie vrai
```

Les caractères a, b, A, B, 0, 1, etc. représentent ce qu'ils sont sensés représenter c'est-à-dire les caractères a, b, A, B, 0, 1.

```
String chaine1 = new String("azET45");
System.out.println(chaine1 + " et azET45 => " +
chaine1.matches("azET45")); // renvoie vrai (il y a correspondance
System.out.println(chaine1 + " et AZER45 => " +
chaine1.matches("AZER45")); // renvoie faux
```

Le caractère . (point) représente n'importe quel caractère.

```
System.out.println(chaine1 + " et azE.45 => " + chaine1.matches("azE.45")); //
renvoie true
```

En fait, les chaîne « azE145 », « azEx45 », « azE=45 », etc. correspondent à l'expression régulière « azE.45 ».

Le caractère * (étoile) signifie que le caractère ou le groupe de caractères précédent l'étoile peut être présent 0, 1 ou plusieurs fois.

"", "g", "gg", etc. correspondent à l'expression régulière "g*".

Le caractère + signifie que le caractère ou le groupe de caractères précédent l'étoile peut être présent 1 ou plusieurs fois.

Le caractère ? signifie que le caractère ou le groupe de caractères précédent le point d'interrogation peut être présent 0 ou 1 fois.

Pour grouper des caractères, on utilise les parenthèses :

```
String chaine6 = new String("abab");
```

```
System.out.println(chaine6 + " et (ab)+ => " + chaine6.matches("(ab)+"));
```

On peut représenter symboliquement un ensemble de caractères : cet ensemble est appelé classe (ne pas prendre ici le mot classe dans le sens de la programmation objet). Exemples de classes prédéfinies :

- `\d` représente tout chiffre ;
- `\s` représente tout caractère blanc : espace, tabulation, retour chariot ;
- `\w` représente tout caractère de mot : a à z, A à Z, 0 à 9, _ ;
- etc.

Les crochets permettent de définir une classe de caractères personnalisée :

- `[abc2]` : correspond à l'un des caractères a, b, c ou 2 ;
- `[^abc2]` : correspond à tout caractère sauf a, b, c ou 2 ;
- `[a-z]` : correspond à tout caractère de a à z ;
- etc.

Il existe un paquet classe `java.util.regex` qui regroupe deux classes qui permettent d'utiliser des expressions rationnelles : *Pattern* (dont la documentation est très détaillée au sujet de expressions rationnelles) et *Matcher*.

Exercice 1 :

Dans la classe *Menu* que nous avons écrite précédemment, on suppose que l'utilisateur ne se trompe pas lorsqu'il saisie son choix : il doit saisir un entier. Dans le cas contraire, une exception est levée et le programme s'arrête.

Créez une nouvelle version de la classe *Menu* de telle sorte qu'une saisie erronée soit signalée par un message plutôt que de provoquer l'arrêt du programme.

La manière de procéder peut être :

- faire des saisies de chaînes au clavier ;
- valider les données à l'aide d'expressions rationnelles ;
- convertir éventuellement la chaîne validée en nombre grâce à la méthode statique *decode* de la classe *Integer* (voir aussi la méthode *valueOf*).

Exercice 2 :

Créez une nouvelle version du programme MasterMind en utilisant la nouvelle version de la classe *Menu*.