



#2.fr - String, StringBuilder, égalité, et expressions régulières

Exercice 1 - Assignment, égalité, référence

1.

```
var s = "toto";
System.out.println(s.length());
```

Quel est le type de 's' ? Comment le compilateur fait-il pour savoir qu'il existe une méthode `length()` sur 's' ?

2. Qu'affiche le code suivant ?

```
var s1 = "toto";
var s2 = s1;
var s3 = new String(s1);

System.out.println(s1 == s2);
System.out.println(s1 == s3);
```

Expliquer.

3. Quelle est la méthode à utiliser si l'on veut tester si le contenu des chaînes de caractères est le même ?

```
var s4 = "toto";
var s5 = new String(s4);

System.out.println(/* comparer contenu de s4 et s5 */);
```

4. Qu'affiche le code suivant ?

```
var s6 = "toto";
var s7 = "toto";

System.out.println(s6 == s7);
```

Expliquer.

5. Expliquer pourquoi il est important que `java.lang.String` ne soit pas mutable.
6. Qu'affiche le code suivant ?

```
var s8 = "hello";
s8.toUpperCase();
System.out.println(s8);
```

Expliquer.

Exercice 2 - En morse. Stop.

Écrire une classe `Morse` qui permet, lors de son exécution, d'afficher les chaînes de caractères prises en argument séparées par des "Stop."

```
$ java Morse ceci est drôle
ceci Stop. est Stop. drôle Stop.
```

1. Utiliser dans un premier temps l'opérateur `+` qui permet la concaténation de chaînes de caractères.

2. A quoi sert l'objet `java.lang.StringBuilder` ?
Pourquoi sa méthode `append(String)` renvoie-t-elle un objet de type `StringBuilder` ?
3. Réécrire la classe `Morse` en utilisant un `StringBuilder`.
Quel est l'avantage par rapport à la solution précédente ?
4. Recopier le code suivant dans une classe de `Test` :

```
public static void main(String[] args) {  
    var first = args[0];  
    var second = args[1];  
    var last = args[2];  
    System.out.println(first + ' ' + second + ' ' + last);  
}
```

Pourquoi peut-on utiliser `' '` à la place de `" "` ?

Compiler le code puis utiliser la commande `javap` pour afficher *le bytecode Java* (qui n'est pas un assembleur) généré.

```
javap -c Test
```

Que pouvez-vous en déduire ?

5. Compiler le code de la question 1, puis utiliser la commande `javap` pour afficher *le bytecode Java* généré.
Que pouvez-vous en déduire ?
Dans quel cas doit-on utiliser `StringBuilder.append()` plutôt que le `+` ?
Et pourquoi est-ce que le chargé de TD va me faire les gros yeux si j'écris un `+` dans un appel à la méthode `append` ?

Exercice 3 - Reconnaissance de motifs

Le but de cet exercice est la manipulation d'expressions régulières en java. Nous utiliserons pour cela les classes du paquetage `java.util.regex`.

1. A quoi servent la classe `java.util.regex.Pattern` et sa méthode `compile` ?
A quoi sert la classe `java.util.regex.Matcher` ?
2. Écrire un programme qui lit des chaînes de caractères sur la ligne de commande et affiche les chaînes qui correspondent à des nombres, c'est-à-dire les chaînes dont tous les caractères sont compris entre '0' et '9'.
3. Modifier le programme pour que, si une chaîne de caractères commence par des caractères qui ne sont pas des chiffres, ces chiffres soient également affichés (par exemple, avec les chaînes "abc", "ab3", "4de", "f6h" et "789", on affiche : "3 789").
4. Écrire une méthode qui prend en paramètre une chaîne de caractères contenant une adresse IPv4 et renvoie un tableau de 4 bytes. Il faut tester qu'il s'agit bien d'une adresse valide.
Vous utiliserez pour cela la notion de groupe.