
Exercice - 1 Java et les chaînes

Le but de cet exercice est vous faire explorer la classe `java.lang.String`. Il s'agit d'écrire une classe exécutable `TestChaines` dont la méthode principale effectue les opérations suivantes :

1. Lire un nombre et le convertir en chaîne

le nombre 12345 devient la chaîne "12345".

2. Lire une chaîne entièrement formée de chiffres et la convertir dans le nombre entier qu'elle représente

la chaîne "12345" devient le nombre 12345).

La solution se trouve parmi les méthodes statiques de la classe `java.lang.Integer`. Notez que lorsqu'une lecture de chaîne fait suite à une lecture de nombre il est nécessaire de nettoyer l'entrée en plaçant entre ces deux opérations la lecture d'une chaîne à fonds perdu.

3. Même question que ci-dessus, mais avec un nombre flottant

la chaîne "0.12345e4" devient le nombre 0.12345e4.

4. Lire une chaîne représentant un nom de ville, lui enlever les éventuels blancs au début et à la fin et l'afficher entièrement en majuscules.

5. Lire deux chaînes `s1` et `s2` et afficher la réponse à la question :

- Commencent-elles par le même caractère? Utilisez la méthode d'instance `charAt`.
- Evaluer les expressions `s1==s2`, `s1.equals(s2)`, `s1.compareTo(s2)` et `s1.compareToIgnoreCase(s2)`. Entre autres, essayer les couples "abcd" et "abcd", puis "abcd" et "AbcD".
- `s1` commence-t-elle par `s2`? , `s1` finit-elle par `s2`? et `s1` contient-elle `s2`?
- `s1` contient `s2`, renvoyer `s1` privée de `s2` (s'intéresser à `substring`), sinon renvoyer `s1`.

Exercice - 2 : Un palindrome

Un palindrome est un mot, une phrase ou un texte qui se lit indifféremment de gauche à droite ou de droite à gauche, la séquence des lettres étant symétrique. La ponctuation, les espaces, la casse et les accents ne sont pas significatifs lors du test d'égalité. Voici un exemple de palindromes.

Eve, Madam I'm Adam,
Tu l'as trop ecrase, Cesar, ce Port-Salut !

1. Ecrire la fonction `nettoyer` qui enlève les délimiteurs d'une chaîne. Elle renvoie la chaîne nettoyée.
2. Ecrire la fonction `estunPalindrome` qui teste si une chaîne est un palindrome.
3. Ecrire la fonction `main` qui saisit une chaîne quelconque et qui teste si c'est un palindrome.

Indications :

```
StringTokenizer st = new StringTokenizer("ceci est\nun\ttest"," \\t\\r\\n");  
while (st.hasMoreTokens())  
System.out.println(st.nextToken());
```

Permet de d'afficher les mots suivants sans les délimiteurs

```
ceci  
est  
un  
test
```

Exercice - 3 : Conjugaison

Ecrire un programme qui lit un verbe régulier en "er" au clavier et qui en affiche la conjugaison au présent de l'indicatif de ce verbe. Contrôlez s'il s'agit bien d'un verbe en "er" avant de conjuguer. Utiliser les fonctions. Exemple :

```
Verbe : chanter
je chante
tu chantes
il chante
nous chantons
vous chantez
ils chantent
```

Exercice - 4 : Adresse IP valide

Une adresse ip est donnée par quatre nombres compris entre 0 et 255, séparés par des points. Par exemple : 212.85.150.134. L'objectif de cet exercice est de vérifier si une chaîne de caractères donnée définit bien une adresse ip. Pour cela il faut vérifier qu'elle comporte 3 points ; elle commence et se termine par un nombre et les nombres situés entre les points sont compris entre 0 et 255. Les exemples suivant permettent de s'assurer que le code traite bien tous les cas :

```
"127.0.0.1",---> true
"127.231.1.1", --->true
"1.2.3.4"--->true
"12.2.3",---> false
"12.3.213.123" ---> false
"1231.12.2.3",---> false
".1.2.3"---> false
"1.2.3."--->false
"1.2..3"--->false
```

1. Ecrire un programme qui convertit une chaîne de caractères de plusieurs mots séparés par des '.' en un tableau de mots. On utilisera la méthode split de la classe String dont le fonctionnement est décrit ci-dessous :

```
public class ExempleSplit {
    public static void main(String[] a) {
        String s = "Bonjour.les.amis";
        String separateur = "\\.";
        String[] mot = s.split(separateur);
        for (String m : mot) {
            System.out.println(m);
        }
    }
}
```

La sortie du programme est :

```
Bonjour
les
amis
```

On pourra utiliser les fonctions suivantes :

- `int Integer.parseInt(String s)` pour convertir une chaîne de caractères en un nombre (si et seulement si la chaîne de caractères correspond à un nombre ;
- `bool isNumeric(String s)` pour vérifier si la chaîne de caractères représente un nombre.