

Université Tunis El-Manar	Faculté des Sciences de Tunis
Module : Programmation.O.O	Année universitaire : 2024-2025
Section : MI 2 – PI 2	Hajer Dammak

Série N° 1 (suite) : Initier avec Java

Exercice 4 : Palindrome

Il s'agit d'écrire un programme qui, étant donnée une chaîne de caractères (une instance de la classe `String`) :

- Calcule la chaîne inverse.
- Indique s'il s'agit ou non d'un palindrome.

Premier temps En un premier temps, la chaîne à traiter ne comportera pas de caractère d'espacement et sera indiquée sur la ligne de commande.

Exemples

- Si la commande est :
`> java Palindrome bonjour`
La réponse pourrait être :
L'inverse de bonjour est ruojnob
bonjour n'est pas un palindrome
- Si la commande est :
`> java Palindrome laval`
La réponse pourrait être :
L'inverse de laval est laval
laval est un palindrome

On définira une seule classe qui n'aura aucun attribut et deux méthodes statiques :

- Une méthode dont l'en-tête sera : `static String inverse(String s)` qui retournera un objet de type `String` contenant la chaîne inverse de la chaîne `s`. Cette méthode aura entre autres variables locales une variable nommée `envers`, de type `char[]`, pour référencer un tableau de caractères comportant autant de case que le nombre de caractères de la chaîne `s`. Ce tableau est destiné à contenir les caractères de `s` en sens inverse de `s`.
- Une méthode `main`; dans cette méthode, on récupérera la chaîne par la ligne de commande, on calculera son inverse grâce à la méthode `inverse`, on affichera cet inverse dans la fenêtre de commande, on indiquera aussi dans la fenêtre de commande si la chaîne est ou non un palindrome.

Indications

- La classe `String` possède une méthode `length()` qui retourne le nombre de caractères de la chaîne concernée. Par exemple, si on a :

```
String s = "java" ;  
s.length() vaut 4
```

- La classe `String` possède une méthode `charAt(int i)` qui retourne le $(i + 1)$ ème caractère de la chaîne concernée. Par exemple, si on a :

```
String s = "java" ;  
s.charAt(2) vaut v
```

- La classe `String` possède une méthode `equals(String c)` qui retourne `true` la chaîne concernée est identique à la chaîne `c` et `false` sinon. Par exemple, si on a :

```
String s = "java", t = "lundi", u = "ja" + "va",  
s.equals(t) vaut false et s.equals(u) vaut true
```

- La classe `String` possède un constructeur ayant un paramètre de type `char[]` et retourne un objet de type `String` correspondant au tableau reçu en paramètre. Par exemple, si on a :

```
char[] table = {'j', 'a', 'v', 'a'};  
s = new String(table) revient à s = "java"
```

Second temps : En un second temps, la chaîne à traiter sera indiquée par la fenêtre d'exécution mais la saisie se fera après le lancement du programme. Cela permettra d'avoir des caractères d'espacement dans la chaîne de caractères à traiter.

Exemples

```
> java Palindrome2  
>Indiquez la chaîne de caractères  
>je programme avec java  
>L'inverse de "je programme avec java" est "avaj ceva emmargorp  
ej"  
>"je programme avec java" n'est pas un palindrome
```

A part ce qui concerne la saisie, le programme sera analogue à celui développé dans le premier temps.

Indication

Pour effectuer une saisie par la fenêtre d'exécution, on peut utiliser l'instruction suivante :
`Scanner entree = new Scanner(System.in);`

L'objet nommé ci-dessus `entree` permet de faire des saisies ; en particulier, l'instruction `String chaine = entree.nextLine();` met dans chaîne le contenu de la ligne indiquée dans la fenêtre d'exécution par l'utilisateur (et "envoyée" par la touche d'entrée).