

Module 1 – Informatique – Algorithmique et Programmation Objet

Travaux Pratiques (4), Licence 1ère Année

Les chaînes de caractères

Les exercices suivants ont pour but de vous familiariser à la manipulation des chaînes de caractères et des méthodes de la classe String.

Il vous est recommandé de consulter le cours distribué, ou tout autre document de cours Java, pour de plus amples informations sur leur syntaxe et leur sémantique.

Les méthodes qui seront utiles à la réalisation de ce TP vous sont rappelées ci-dessous :

- **int str.length()** : retourne la longueur de la chaîne de caractères str.
- **char str.charAt(int index)** : retourne le caractère de la chaîne str à l'indice index.
- **String str.substring(int indice)** : retourne une nouvelle chaîne qui commence à l'indice indice de la chaîne str.
- **String str.substring(int indiceDebut, int indiceFin)** : retourne une nouvelle chaîne comprise entre les indices indiceDebut et indiceFin de la chaîne str. Le caractère d'indice indiceFin n'est pas inclus dans la chaîne retournée.
- **int str.indexOf(char c)** : retourne l'indice de la 1ère occurrence du caractère c dans la chaîne str.
- **int str.lastIndexOf(char c)** : retourne l'indice de la dernière occurrence du caractère c dans la chaîne str, sinon -1.
- **int str1.indexOf(String str2)** : retourne l'indice de la 1ère occurrence de la sous-chaîne str2 dans la chaîne str1 en effectuant une recherche de gauche à droite. Si str2 n'est pas incluse dans str1, la méthode retourne -1.
- **int str1.lastIndexOf(String str2)** : retourne l'indice de la dernière occurrence de la sous-chaîne str2 dans la chaîne str1 en effectuant une recherche de gauche à droite. Si str2 n'est pas incluse dans str1, la méthode retourne -1.
- **int str1.indexOf(String str2, int indice)** : retourne la position de la 1ère occurrence de la sous-chaîne str2 dans la chaîne str1, en effectuant une recherche de gauche à droite à partir de l'indice indice. Si str2 n'est pas présente dans str1, à partir de l'indice indice, la méthode retourne -1.
- **boolean str1.startsWith(String str2)** : retourne true si la chaîne str1 débute par la sous-chaîne str2, sinon false.
- **boolean str1.endsWith(String str2)** : retourne true si la chaîne str1 se termine par la sous-chaîne str2, sinon false.
- **boolean str1.equals(String str2)** : compare si les deux chaînes de caractères str1 et str2 sont identiques. Si les deux chaînes sont identiques, la méthode retourne true, sinon elle retourne false.
- **String str1.concat(String str2)** : retourne une nouvelle chaîne constituée de la chaîne str2 concaténée à la fin de la chaîne str1.

- **int str1.compareTo(String str2)** : compare alphabétiquement les deux chaînes str1 et str2.
 - retourne une valeur <0 si la chaîne str1 est alphabétiquement avant str2
 - retourne 0 si la chaîne str1 est identique à str2
 - retourne une valeur >0 si la chaîne str1 est alphabétiquement après str2
- **String [] str1.split(String str2)** : renvoie un tableau de chaînes où chacune correspond à une sous-chaîne de str1 délimitée (séparée) par str2.
- **char [] str.toCharArray()** : retourne un tableau de caractères où chaque élément correspond à un caractère de la chaîne str.
- **String valueOf(Object obj)** : retourne la chaîne de caractères représentant l'objet obj. Ce dernier peut être d'un type de base (int, float, boolean, ...) ou un objet tableau char [] ou String.

Exercices Manipulation de chaînes

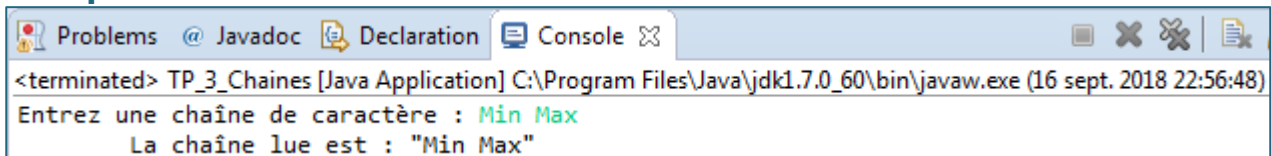
Note : afin d'éviter à chaque fois de saisir une chaîne, vous pouvez l'initialiser une fois pour toute (String s = "contenu de la chaîne" ;).

Remarque : pour conserver certaines instructions que vous ne souhaitez plus exécuter, vous pouvez les mettre en commentaire (sélectionnez les lignes et tapez Ctrl + Shift + c).

1. **Saisie et affichage** : Écrire un programme qui demande à l'utilisateur d'introduire une chaîne de caractères et l'affiche sur la console.

Note : Utiliser la méthode **nextLine()** de la classe **Scanner** pour lire, à partir de la frappe au clavier, une chaîne de caractères qui peut contenir plusieurs mots et des espaces.

Exemple

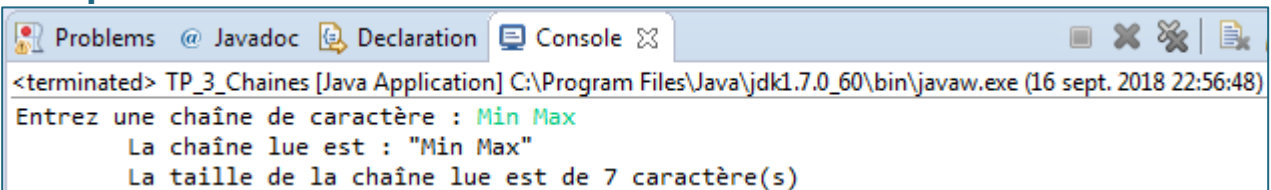


```

<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (16 sept. 2018 22:56:48)
Entrez une chaîne de caractère : Min Max
La chaîne lue est : "Min Max"
  
```

2. **Taille chaîne** : complétez votre programme pour qu'il affiche également la taille de la chaîne lue.

Exemple

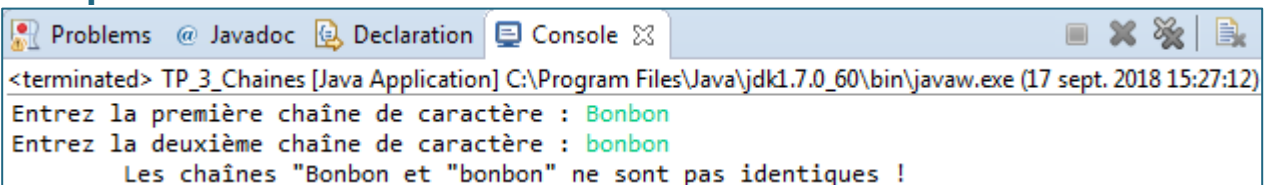


```

<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (16 sept. 2018 22:56:48)
Entrez une chaîne de caractère : Min Max
La chaîne lue est : "Min Max"
La taille de la chaîne lue est de 7 caractère(s)
  
```

3. **Test égalité** : Écrire un programme qui lit deux chaînes de caractères et affiche si elles sont identiques ou pas.

Exemple



```

<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (17 sept. 2018 15:27:12)
Entrez la première chaîne de caractère : Bonbon
Entrez la deuxième chaîne de caractère : bonbon
Les chaînes "Bonbon et "bonbon" ne sont pas identiques !
  
```

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (17 sept. 2018 15:30:37)
Entrez la première chaîne de caractère : Bonbon
Entrez la deuxième chaîne de caractère : Bonbon
    Les chaînes "Bonbon et "Bonbon" sont identiques !
    
```

4. **Comparaison alphabétique** Écrire un programme qui lit deux chaînes de caractères et affiche laquelle précède alphabétiquement l'autre.

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (17 sept. 2018 16:06:18)
Entrez la première chaîne de caractère : Patrick
Entrez la deuxième chaîne de caractère : Nicolas
    La chaîne "Nicolas précède alphabétiquement la chaîne "Patrick" !
    
```

5. **Concaténation** : Écrire un programme qui lit deux chaînes de caractères, les concatène en une troisième chaîne qu'il affiche sur la console.

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (17 sept. 2018 16:19:52)
Entrez la première chaîne de caractère : Le ciel
Entrez la deuxième chaîne de caractère : est bleu
    La concaténation de "Le ciel " et de "est bleu" donne la chaîne "Le ciel est bleu"
    
```

6. **Sélection** : Écrire un programme qui lit une chaîne composée exactement de 6 caractères qu'il décompose en deux autres chaînes, l'une constituée des caractères en positions paires et l'autre constituée des caractères en positions impaires. Ces dernières seront affichées sur la console.

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (17 sept. 2018 16:39:24)
Entrez une chaîne (exactement) de 6 caractère : A1B2C3
    La chaîne constituée des caractères en positions paires est "ABC"
    La chaîne constituée des caractères en positions impaires est "123"
    
```

7. **Sous-chaîne** : Écrire un programme qui lit une chaîne et qui affiche sa deuxième moitié.

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (17 sept. 2018 16:56:41)
Entrez une chaîne de caractère : Chou-fleur
    La deuxième moitié de "Chou-fleur" est "fleur"
    
```

- 8. Sous-chaîne** : Écrire un programme qui lit une chaîne et qui affiche sa sous-chaîne constituée des caractères du deuxième à l'avant-dernier.

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (17 sept. 2018 17:07:03)
Entrez une chaîne de caractère : Panner
    La sous-chaîne intérieure de "Panner" est "anne"
  
```

- 9. Chercher** : Écrire un programme qui lit une chaîne de caractères et un caractère **c** et qui affiche si **c** est présent ou pas dans la chaîne. Si **c** est présent, sa position sera également affichée.

Rappel : pour lire un caractère à partir de l'entrée, il faut lire toute la ligne et prendre le 1er caractère (**nextLine.charAt(0)**).

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (17 sept. 2018 17:22:29)
Entrez une chaîne de caractère : Chapeau
Entrez un caractère : p
    Le caractère 'p' est présent dans "Chapeau" en position 3
  
```

- 10. Chercher** : Écrire un programme qui lit deux chaînes de caractères et qui affiche si la seconde chaîne est présente ou pas dans la première chaîne. Si elle est présente, sa position sera également affichée.

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (17 sept. 2018 17:31:48)
Entrez la première chaîne de caractère : Empoter
Entrez la deuxième chaîne de caractère : pot
    La chaîne 'pot' est présente dans "Empoter" en position 2
  
```

- 11. Vérifier** : Écrire un programme qui lit deux chaînes de caractères et qui affiche si la seconde chaîne correspond ou pas au début de la première chaîne.

Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (17 sept. 2018 17:45:20)
Entrez la première chaîne de caractère : Tourniquet
Entrez la deuxième chaîne de caractère : Tour
    La chaîne 'Tour' correspond au début de "Tourniquet"
  
```

- 12.** Écrire un programme qui lit deux chaînes de caractères et qui affiche si la seconde chaîne correspond ou pas à la fin de la première chaîne.

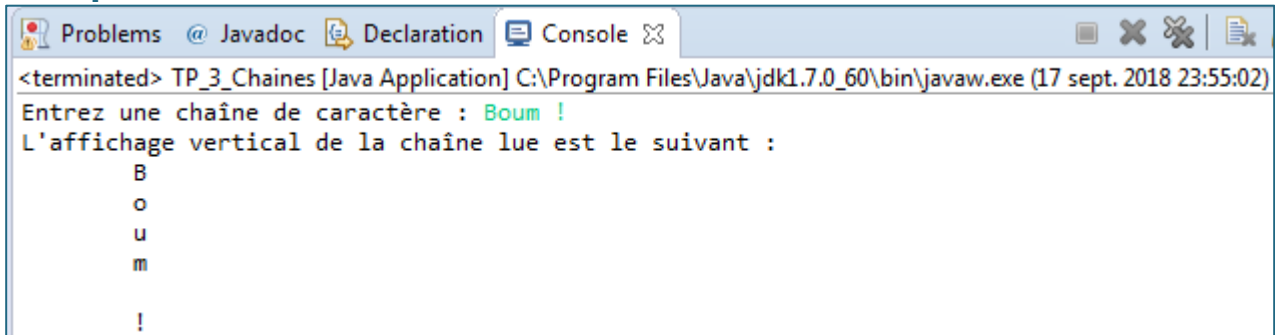
Exemple

```

Problems @ Javadoc Declaration Console
<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (17 sept. 2018 17:52:29)
Entrez la première chaîne de caractère : campagne
Entrez la deuxième chaîne de caractère : pagne
    La chaîne 'pagne' correspond à la fin de "campagne"
  
```

13. Affichage vertical : Écrire un programme qui lit une chaîne de caractères et qui l’affiche à la verticale, un caractère par ligne.

Exemple

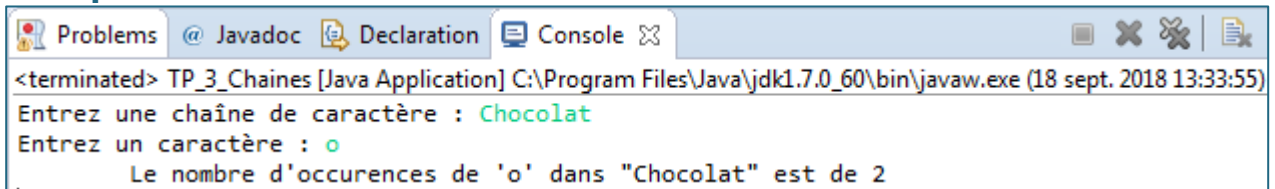


```

<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (17 sept. 2018 23:55:02)
Entrez une chaîne de caractère : Boum !
L'affichage vertical de la chaîne lue est le suivant :
  B
  o
  u
  m
  !
  
```

14. Fréquence d’un caractère : Écrire un programme qui lit une chaîne de caractères et un caractère **c** et qui affiche le nombre d’occurrences de **c** dans la chaîne lue.

Exemple



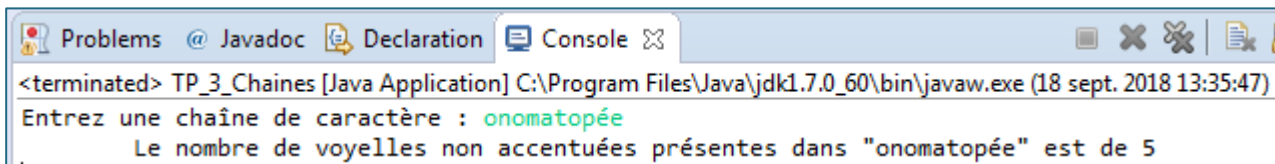
```

<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (18 sept. 2018 13:33:55)
Entrez une chaîne de caractère : Chocolat
Entrez un caractère : o
  Le nombre d'occurrences de 'o' dans "Chocolat" est de 2
  
```

15. Nombre de voyelles : Écrire un programme qui lit une chaîne de caractères et qui affiche le nombre de voyelles non accentuées (a, e, i, o, u, y) présentes dans la chaîne lue.

Remarque : essayez de transformer votre programme de l’exercice précédent en une méthode acceptant en paramètres d’entrées une chaîne de caractère et un caractère et qui rend le nombre d’occurrences de ce dernier dans la chaîne. Vous pourrez ainsi l’utiliser pour réaliser cet exercice.

Exemple

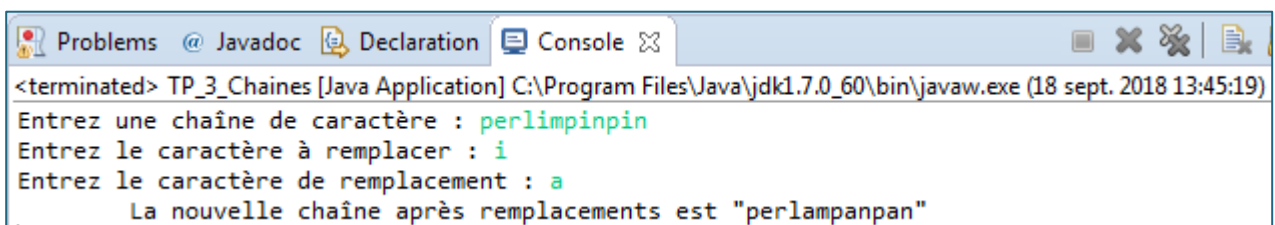


```

<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (18 sept. 2018 13:35:47)
Entrez une chaîne de caractère : onomatopée
  Le nombre de voyelles non accentuées présentes dans "onomatopée" est de 5
  
```

16. Remplacer une lettre : Écrire un programme qui lit une chaîne de caractères et deux caractères **c1** et **c2**, qui construit une nouvelle chaîne à partir de celle lue en remplaçant dans cette dernière les caractères correspondant à **c1** par le caractère **c2**. La nouvelle chaîne sera affichée sur la console.

Exemple



```

<terminated> TP_3_Chaines [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (18 sept. 2018 13:45:19)
Entrez une chaîne de caractère : perlimpinpin
Entrez le caractère à remplacer : i
Entrez le caractère de remplacement : a
  La nouvelle chaîne après remplacements est "perlampnpan"
  
```