TP4 : Les chaînes de caractères

Enseignants : Madhi Kamel, Majdoub Manel

Groupes : Prepa A02
Objectifs : Apprendre à manipuler les chaînes de caractères en utilisant les méthodes associées au type String.

Rappel : Les chaînes de caractères en Java sont définies par le type String. Parmi les opérations intéressantes sur les chaînes de caractères, on trouve les fonctions length, charAt, equals et compareTo.

NB : dans les exemples ci-dessous,chaine et chaine1 sont deux variables de type String.

int length() : pour obtenir la longueur d'une chaîne (nombre de caractères). Exemple :chaine.length()retourne la longueur de la chaîne chaine.

char charAt(int index) : pour obtenir le caractère à la position donnée. Exemple : chaine.charAt(0) retourne le premier caractère de la chaîne chaine.

boolean equals(Object anObject) : pour vérifier si deux chaînes sont égales. Exemple :chaine.equals(chaine1)retournetruesi les chaînes chaine et chaine1 sont égales, et false sinon.

int compareTo(String anotherString) : pour comparer deux chaînes suivant l'ordre lexicographique. Retourne un nombre négatif si la chaîne est inférieure à celle passée en paramètre, un nombre positif si elle est supérieure, et 0 si les deux chaînes sont égales.

Exercice 1

On se propose de définir une fonction qui évalue une expression de calcul donnée sous forme d'une chaine. On suppose que l'expression ne contient que des nombres et les deux opérateurs d'addition(+) et de multiplication(*).

Exemples :
Ch='18' la fonction retourne 18.
Ch='12*1+4*3+12' la fonction retourne 36.
Ch='22+8' la fonction retourne 30.

Exercice 2

Créer une classe nommée ManipulationDeChaines qui contiendra les méthodes demandées par la suite. Pour être testée, chacune de ces méthodes sera appelée depuis la méthode main, qui aura ensuite à charge d'afficher son résultat.

1. Écrire une méthode nommée parcourirChaine qui parcourt et affiche l'ensemble des caractères contenus dans la chaîne passée en paramètre de la méthode, en sautant une ligne après chaque caractère.

```
public static void parcourirChaine (String s )
```

2. Écrire une méthode nommée compterLaLettre qui compte combien de fois on trouve une lettre donnée dans une phrase. La phrase et la lettre seront passés en paramètres. Le résultat est un entier.

   public static int compterLaLettre (String s , char c)

3. Écrire une méthode nommée supprimerCar qui supprime toutes les occurences d'un caractère c dans la chaine s.

   public static String supprCar (String s , char c)

4. Écrire la méthode enleverNonAlphanumerique qui renvoie en résultat une chaîne de caractères privée de ses caractères non alphanumériques à partir d'une chaine donnée en paramètre.
5. Écrire une méthode nommée supprimerDoublons qui supprime les instances multiples d'un caractère (passé en paramètre) dans une chaîne de caractères. La chaîne modifiée constitue le résultat de votre méthode.
6. Écrire une méthode nommée compterLesMots qui compte le nombre de mots dans une phrase. On considère que les mots ne peuvent être séparés que par des espaces (Pas de saut de ligne ou de tabulation). Cette méthode prend la chaîne de caractères en paramètre et renvoie un résultat entier.

   public static int compterLesMots (String s)

7. Écrire une methode inverserChaine qui, à partir d'une chaîne donnée, produit son inverse. La chaîne fournie en résultat est une nouvelle chaîne construite à partir de la première.
8. Écrire une méthode estUnPalindrome qui détecte si une phrase est un palindrome ou non, c'est à-dire si elle peut se lire dans les deux sens. La phrase sera prise en paramètre et le résultat renvoyé est un booléen.

Exercice 3

Réaliser une classe Repertoire permettant de gérer un répertoire téléphonique associant un numéro de téléphone (chaîne de caractères) à un nom. Pour faciliter les choses, on prévoira une classe Abonne destinée à représenter un abonné et disposant des fonctionnalités indispensables.

La classe Repertoire devra disposer des fonctionnalités suivantes :

— constructeur recevant un argument de type entier précisant le nombre maximum d'abonnés que pourra contenir le répertoire (cette particularité évite d'avoir à se soucier d'une gestion dynamique du répertoire),
— méthode addAbonne permettant d'ajouter un nouvel abonné ; elle renverra la valeur false si le répertoire est plein, la valeur true sinon,
— méthode getNumero fournissant le numéro associé à un nom d'abonné fourni en argument,
— méthode getNAbonnes qui fournit le nombre d'abonnés figurant dans le répertoire,
— méthode getAbonne fournissant l'abonné dont le rang est fourni en argument,

— méthode getAbonnesTries fournissant un tableau des références des différents abonnés, rangés par ordre alphabétique (pour simplifier, on supposera que les noms sont écrits en minuscules, sans caractères accentués).

Écrire un petit programme de test