

Programmation 1

Examen

La classe *StringTokenizer* permet à une application de découper une chaîne de caractères en *tokens* séparés par des délimiteurs. Un *token* est la plus grande séquence de caractères qui ne sont pas des délimiteurs ou un délimiteur lui-même (en fonction du paramètre *returnDelims*). Les délimiteurs peuvent être définis soit à la construction d'une instance de la classe *StringTokenizer*, soit à chaque acquisition d'un nouveau *token*.

Un exemple d'utilisation pourrait être :

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

Qui affiche la séquence de *tokens* :

```
ceci  
est  
un  
test
```

Question 1. (3 Points)

Ecrire la définition des constructeurs de la classe *StringTokenizer* :

- *StringTokenizer* (*String str*, *String delimiters*, *bool returnDelims*) qui construit un *StringTokenizer* pour la chaîne *str*, les délimiteurs étant contenus dans la chaîne *delimiters*. Le booléen *returnDelims* indique si chaque délimiteur doit être retourné comme un *token* ou non.
- *StringTokenizer* (*String str*, *String delimiters*) qui construit un *StringTokenizer* pour la chaîne *str*, les délimiteurs étant contenus dans la chaîne *delimiters*. Le délimiteur n'est pas retourné comme un *token*.
- *StringTokenizer* (*String str*) qui construit un *StringTokenizer* pour la chaîne *str*, les délimiteurs étant " \t\n\r\f". Le délimiteur n'est pas retourné comme un *token*.

Question 2. (2 Points)

Ecrire la définition de la méthode *estDelimiteur(char c)* qui retourne *true* si *c* est un délimiteur et *false* sinon.

Question 3. (3 Points)

Ecrire la définition de la méthode *sauterDelimiters* qui fait avancer *indiceCourant* jusqu'à ce que *laChaine.charAt(indiceCourant)*, s'il existe, ne soit pas un délimiteur.

Question 4. (3 Points)

Ecrire la définition de la méthode *hasMoreToken()* qui renvoie *true* s'il y a encore des *tokens* dans la chaîne et *false* sinon.

Question 5. (4 Points)

Ecrire la définition de la méthode *nextToken()* qui renvoie le *token* suivant s'il y en a (en fonction du paramètre *returnDelim*) et lève une exception du type *NoSuchElementException* sinon. La valeur de *indiceCourant* est modifiée et désigne le premier caractère qui suit le dernier caractère du *token*.

Question 6. (2 Points)

Ecrire la méthode de la fonction *nextToken(String delimiters)* qui renvoie le *token* suivant s'il y en a (en fonction du paramètre *returnDelim*) et lève une exception du type *NoSuchElementException* sinon. La méthode remplace les délimiteurs par *delimiters* avant de commencer le travail.

Question 7. (3 Points)

Ecrire la définition de la méthode *int countTokens()* qui calcule le nombre de fois que la méthode *nextToken()* peut être appelée à partir de l'indice courant sans lever une exception.

```
class StringTokenizer{
    private String laChaine;
    private String lesDelimiters;
    private int indiceCourant;
    private boolean returnDelims;

    public StringTokenizer(String s, String delimiters, boolean returnDelims)
    {...}
    public StringTokenizer(String s, String delimiters) {...}
    public StringTokenizer(String s) {...}

    private boolean estDelimiteur( char c){...}
    private void sauterDelimiters(){...}

    public boolean hasMoreTokens(){...}
    public String nextToken(){...}
    public String nextToken(String delimiters) {...}
    public int countToken(){...}
}
```