

Les objets StringBuffer

Le StringBuffer est une chaîne de caractère présentant certains avantages de manipulations et de performances. Un StringBuffer est une chaîne de caractère modifiable. La classe StringBuffer possède en effet des méthodes permettant d'effectuer des modifications sur l'objet lui-même. Par exemple, ce qui était impossible pour le String (modifier un caractère de la chaîne) est possible pour le StringBuffer grâce à sa méthode *setCharAt*.

Principales méthodes de la classe StringBuffer

Méthode	Description
<i>char charAt(int index)</i>	Renvoie le caractère à cette position
<i>void delete(int iDebut,int iFin)</i>	Supprime les caractères de la chaîne de <i>iDebut</i> à <i>iFin-1</i>
<i>StringBuffer deleteCharAt(int iIndex)</i>	Supprime le caractère à cette position
<i>void setCharAt(int iIndex,char cChar)</i>	Fixe le caractère de la chaîne à l'index <i>iIndex</i> au char donné en argument.
<i>boolean equals(String sOtherString)</i>	Renvoie vrai si le String sur lequel est appliqué la méthode <i>equals</i> est égal à <i>sOtherString</i>
<i>StringBuffer insert(int iIndex, String str)</i>	Insère la chaîne <i>str</i> à l'index <i>iIndex</i> .
<i>StringBuffer replace(int iDebut, int iFin, String str)</i>	Remplace la chaîne par la chaîne <i>str</i> de l'index <i>iDebut</i> à l'index <i>iFin-1</i> .
<i>String substring(int iDepart,int iFin)</i>	Ne conserve de la chaîne que les caractères de l'index <i>iDebut</i> à l'index <i>iFin-1</i> . Plus simplement, <i>iFin-iDebut</i> vaut la longueur de la sous-chaîne.
<i>String toString()</i>	Renvoie le String correspondant à cette chaîne.

La classe StringTokenizer

Cette classe du package *java.util* sert à couper une chaîne en sous-chaînes de façon rapide. Le constructeur le plus courant est:

```
StringTokenizer(String sInput,String sSeparator){}
```

N'importe quelle chaîne peut servir de séparateur (un ou des espaces, des slashes, tirets...). La méthode *nextToken()* renvoie un *String* contenant la sous-chaîne suivante. La méthode *hasMoreTokens()* renvoie *true* s'il reste des chaînes. Exemple:

```
String sMonText="toto titi tata";
StringTokenizer st1=new StringTokenizer(sMonText," ");
String s1=st1.nextToken(); //s1 vaut "toto"
String s2=st1.nextToken(); //s2 vaut "titi"
String s3=st1.nextToken(); //s3 vaut "tata"
```

```

import java.io.*;
import java.util.*;

class SaisieClavier
{
    public static void main (String[] argv) throws IOException,
        NumberFormatException
    {
        int somme = 0;
        String ligne;
        StringTokenizer st;
        BufferedReader entree = new BufferedReader
            (new InputStreamReader(System.in));

        ligne = entree.readLine();
        while(ligne.length() > 0)
        {
            st = new StringTokenizer(ligne);
            while(st.hasMoreTokens())
                somme += Integer.parseInt(st.nextToken());
            ligne = entree.readLine();
        }
        System.out.println("La somme vaut : "+somme);
    }
}

```

Voici une execution :

```

3 1
2

```

La somme vaut : 6

Autre exemple :

1- Instanciez un *StringBuffer* à partir d'une chaîne valant "toto"

```
StringBuffer sb=new StringBuffer("toto");
```

2- Modifiez cette chaîne en remplaçant les 'o' par des 'i'.

```

for (int i=0;i<sb.length();i++){
    if (sb.charAt(i) == 'o' ){
        sb=sb.replace(i,i+1,"i"); //attention, c'est le String "i" et non le char 'i'
    }
}

```

3- Affichez la nouvelle chaîne sous la forme : "Nouvelle chaîne =" et la nouvelle chaîne.

```
System.out.println("Nouvelle chaîne = "+sb.toString());
```