Les `char` d'un `String`

- L'instruction `chaine.charAt(index)` donne le caractère occupant la position `index` dans la `String chaine`
- L'instruction `chaine.indexOf(caractere)` donne la position de la première occurence du `char caractere` dans la `String chaine`, et -1 si `caractere` n'est pas dans `chaine`.
- `chaine1.length()` donne la taille (c'est-à-dire le nombre de caractères) de `chaine1`. **Attention :** il y a une paire de parenthèses ; différent des tableaux !

Exemple :

```
String s1 = "abcmbx";
int longueur = s1.length();         // 6
char c1 = s1.charAt(0);             // a
char c2 = s1.charAt(longueur - 1); // x
int i = s1.indexOf('b');            // 1
```

☞ Les caractères sont numérotés comme les éléments d'un tableau (à partir de 0)

Les `chars` d'un `String` (2)

Exercice : qu'affichera le programme suivant :

```
String essai = "essai";
String test = "";
for (int i = 1; i <= 3; ++i) {
 test = test + essai.charAt(6-2*i);
 test = essai.charAt(i) + test;
}
System.out.println(test);
```

Pas de `nextChar()` dans la classe `Scanner` ! ?

Pour récupérer un caractère (`char`) avec la classe `Scanner`, il faut faire :

```
// Lire la ligne qui contient un caractère
Scanner keyb = new Scanner(System.in);
String s = keyb.nextLine();

// Prendre comme caractère le premier élément de la String
char c = s.charAt(0);
```

Littéraux introduits par l'utilisateur

Un littéral introduit par l'utilisateur suite à une instruction de lecture n'est pas dans le pool des littéraux

Pour qu'il y soit, il faut l'y mettre explicitement au moyen de `intern`

Exemple

```
Scanner s = new Scanner(System.in);
String response;
do {
   response = s.nextLine();
   //on met le littéral lu dans le pool
   response = response.intern();
   System.out.println("Read: " + response);
   // sans le intern, la boucle ne s'arrête pas!
} while (response != "oui");
```

Traitements spécifiques aux chaînes

Nous avons vu que certains traitements sont **spécifiques** aux `String`.

Ils s'utilisent en fait tous avec la syntaxe particulière suivante :

`nomDeChaine.nomDeTraitement(arg1, arg2 ...);`

Ces traitements s'appellent des **méthodes** en Java.

☞ Ils produisent toujours une nouvelle chaîne de caractères

replace

`chaine.replace(char1, char2)` : construit une nouvelle chaîne valant `chaine` où `char1` est remplacé par `char2`.

Exemple :

```
String exemple = "abracadabra";
String avecDesEtoiles = exemple.replace('a', '*');
```

construit la nouvelle chaîne `"*br*c*d*br*"`.
`exemple` vaut toujours `"abracadabra"`.

substring

`chaine.substring(position1, position2)` : donne la sous-chaîne comprise entres les indices de `position1` (compris) et `position2` (non-compris)

Exemple :

```
String exemple = "anticonstitutionnel";
String racineMot = exemple.substring(4,16);
```

construit la nouvelle chaîne `"constitution"`.