

IFT 1015 - Objets String

Professeur:
Stefan Monnier

B. Kégl, S. Roy, F. Duranleau, S. Monnier
Département d'informatique et de recherche opérationnelle
Université de Montréal

hiver 2006

Au programme

Approfondissement sur les chaînes de caractères.

[Tasso:7]

Nature des chaînes de caractères

Chaîne de caractères en tant qu'objet

La classe `String` de Java

Nature des chaînes de caractères

Rappel: Une chaîne de caractères est une simple suite de caractères quelconques.

Ex.: "Bonjour!"

⇒ c'est fondamentalement un *tableau* de caractères!

De fait, c'est le cas pour certains langages (notamment C/C++).

Ex.: On pourrait représenter la chaîne "allo!" ainsi:

```
char[] str = { 'a', 'l', 'l', 'o', '!'
};
```

Les chaînes de caractères sont très fréquemment employées

⇒ tout langage inclut un ensemble de fonctionnalités pour les manipuler, en particulier aussi une écriture directe pour les instancier.

Ex.: En Java: `String str = "Allo!";`

En C: `char str[] = "Allo!";`

[Voir **Variables**, diapos 23]

Chaîne de caractères en tant qu'objet

Une chaîne de caractères peut aussi être vue comme un objet i.e. c'est une suite de caractères sur laquelle on peut effectuer des opérations (requêtes, commandes).

Ex.: longueur, i^e caractère, produire une sous-chaîne, ...

De fait, tout langage orienté objet (Java, C++, ...) fournit une *classe* pour manipuler les chaînes de caractères.

Classe de chaînes de caractères en Java

Rappel: Dans le cas de Java, cette classe est `String`.

- méthode d'instanciation spéciale supplémentaire: les littéraux entre guillemets doubles. [Voir **Variables**, diapo 23]
- opérateur spécial de concaténation. [Voir **Entrée/Sortie**, diapo 12]

```
Ex. String str = "string";  
str = "sa" + "lut";
```

Hors mis ces particularités, c'est une classe comme les autres.

⇒ On manipule des *références* sur des `Strings`!

Constructeurs de `String`

`String` a des constructeurs particuliers:

```
char[] tab = {'s', 't', 'r', 'i', 'n', 'g'};  
String str = new String(tab);
```

→ Construit une `String` à partir d'un tableau de caractères.

```
String str = new String();
```

→ Équivalent à `String str = "";`

```
String str2 = new String(str);
```

→ Crée une copie de `str`.

Méthodes de `String`

Soit `str` une *expression* de type `String`, voici quelques méthodes fréquemment utilisées avec les chaînes de caractères:

- `str.length()` → retourne la longueur de la chaîne
- `str.charAt(i)` → retourne le i^e caractère de la chaîne. Comme pour les *tableaux*, $i \in 0 \dots str.length() - 1$.
- `str.equals(str2)` → indique si `str` et `str2` sont identiques ou pas.
- `str.compareTo(str2)` → comparaison lexicographique entre `str` et `str2`. Retourne un nombre négatif si `str < str2`, 0 si `str = str2` et un nombre positif si `str > str2`.

Pour en savoir plus...

La liste des méthodes et constructeurs est trop grande pour tout mentionner ici.

Consultez la documentation de l'API pour en savoir plus!