

Laboratoire 14: Classe String

Durée du laboratoire: 6 périodes. A rendre le jeudi 7 avril 2016, au début de la séance de laboratoire.

Différentes approches sont possibles pour définir une classe `String` permettant de manipuler les chaînes de caractères de manière plus conviviale qu'en C. La première consiste à utiliser les conventions propres au langage C: une chaîne se termine par le code nul (`'\0'`) et les caractères sont ceux de la table ASCII sur 8 bits. Une autre implémentation pourrait être celle du langage Pascal: un premier caractère désigne le nombre de caractères utiles de la chaîne, il est suivi des caractères ASCII sur 8 bits.

Définir une classe `String` avec la première implantation, en utilisant une approche orientée objets.

1. Séparer la spécification de l'implantation. La spécification se trouvera dans le fichier `cstring.h`, l'implantation dans le fichier `cstring.cpp`.
2. Permettre la création d'instances de la classe `String` dans les situations suivantes:
 - pour représenter une chaîne de caractères vide,
 - à partir d'une chaîne de caractères,
 - à partir d'une instance de la classe `String`,
 - à partir d'un caractère,
 - à partir d'un entier,
 - à partir d'un réel, ou
 - à partir d'un booléen.
3. Définir des méthodes permettant de manipuler une instance de la classe `String`:
 - détermination de sa longueur,
 - représentation sous la forme de `const char*` (afin de pouvoir utiliser une instance de la classe `String` avec des fonctions C/C++ standard, comme `printf`),
 - récupération du *i*-ème caractère de manière à ce que l'éventuelle modification de sa valeur affecte l'instance de la classe `String`,
 - détermination si son état est égal à celui d'une autre chaîne de caractères ou d'une autre instance de la classe `String`,
 - modification de son état à partir d'une chaîne de caractères ou d'une autre instance de la classe `String`.
 - concaténation avec un caractère, une chaîne de caractères ou une instance de la classe `string` (procéder par analogie avec les opérateurs `+` et `+=` en rendant une nouvelle instance de la classe `string` ou en modifiant l'instance manipulée),
 - extraction d'une sous-chaîne entre deux indices donnés,
 - écriture à la console,
 - facultatif: lecture sûre au clavier d'un mot et d'une ligne contenant des espaces.
4. Pour une utilisation plus ergonomique de cette classe, définir des opérateurs surchargés utilisant les méthodes de manipulation du point 3. Par exemple, invoquer la méthode `void append(const char* other)` dans le corps de l'opérateur `void operator +=(const char* other)`.
5. Définir un programme principal (`main.cpp`) testant toutes les fonctionnalités de la classe `String`.

Remarque

- Fonctions utiles pour manipuler des chaînes de caractères en C (disponibles dans `<string.h>` et `<stdio.h>`): `strlen`, `strcpy`, `sprintf`, `strcat`, `memcpy`.