C++
TP n°2

Master Informatique IFI-RIF-MBDS 1ère année
Université de Nice-Sophia Antipolis

1 Constructeur et destructeur

Le but de ce TP est de bien comprendre la création d'objet en C++.
On va donc créer une classe Str qui est une classe faite pour gérer des chaines de caractères. Cette classe aura un tableau de caractère et une longueur. On pourra construire une instance à partir:

- d'un constructeur vide. Dans ce cas on créera juste une chaine contenant le caractère '\0'
- d'une chaine de caractère
- d'une autre instance de Str (constructeur par recopie)

Ecrivez les fonctions suivantes:

- `void print()` qui affiche le contenu de la chaine de caractère
- `void printDebug()` qui affiche l'adresse de l'instance (this)
- les constructeurs
- le destructeur
- `char* ch()` qui renvoie la chaine de caractère de l'objet
- une fonction `concat` qui prend en paramères une taille et une chaine de caractère. cette fonction rajoute à la chaîne courante, la chaîne passée en paramètre.

Modifiez les constructeurs et destructeurs pour qu'ils appellent les fonction print et printDebug sur this sur sur les paramètres. Utilisez aussi la macro

`__FUNCTION__`

qui contient le nom de la fonction courante.
Créez un petit exemple d'appel du genre

```
Str a(``tot'');
Str b(``truc'');
Str c(``machin'');

a.concat(b.ch());
a.concat(c.ch());
a.print();
```

Implémentez les différentes fonctions et répondez aux questions:

1. Ecrivez la fonction `void concat2(Str b)` qui prend en parametre une string b et qui ajoute son contenu à l'objet courant.

2. Modifiez votre code pour appeler `a.concat2(b)`. Que se passe t'il au niveau des appels de constructeurs ?
3. Ecrivez la fonction `Str concat3(Str b)` qui prend en parametre une string b et qui crée une nouvelle string qui est la concaténation de l'objet courant avec b. La fonction renvoie la nouvelle string.
4. Modifiez votre code pour appeler `a.concat3(c)`. Que se passe t'il au niveau des appels de constructeurs ?
5. Modifiez votre code et ajoutez`b=a.concat3(c)`. Que se passe t'il au niveau des appels de constructeurs ?
6. Ecrivez la fonction `Str concat4(Str& b)` qui prend en parametre une string b et qui crée une nouvelle string qui est la concaténation de l'objet courant avec b. La fonction renvoie la nouvelle string.
7. Remplacez les appels de `concat3` par `concat4`. Que se passe t'il au niveau des appels de constructeurs ?
8. Essayez de retourner un `Str&`. Que se passe t'il ?

Surchargez les opérateurs suivants pour la classe Str

1. l'opérateur $+$ en tant que fonction membre. Essayez avec Str et Str& comme paramètres
2. l'opérateur $+$ en tant que fonction amie. Essayez avec Str et Str& comme paramètres
3. l'opérateur $<$ en tant que fonction amie.

Testez l'appel suivant:

```
Str s=a+b;
Str t=''hum `` + c;
Str u=''un `` + ``peu'';
```

Voici une documentation sur la surcharge de l'opérateur $=$. Surchargez cet opérateur pour la classe Str.

```
/* le constructeur par copie est automatiquement appelé */
MaClasse& MaClasse::operator =(MaClass Other)
{
    /* à faire pour chaque membre */
    std::swap(membre,Other.membre); // permutons le membre de l'objet en cours et de la copie
    /* renvoyons l'objet en cours */
    return *this;
}// Other (qui contient maintenant les membres d'origine de l'objet courant)
 // est automatiquement détruit et ses membres correctement détruits
```

Expliquez pourquoi la ligne de code: `a=b=c;` fonctionne