

La **bibliothèque string** du C++ définit une classe de manipulation des chaînes de caractères.

Un objet de type string contient une chaîne de caractères classique (attribut privé) ainsi que des informations et des méthodes publiques permettant de le manipuler.

1 – Instanciation et méthodes publiques membres

instanciation d'un objet string

```
string str1;  
string str2= "bonjour";  
string str3="E"; // un seul caractère  
string chaine("Valeur initiale");
```

Les méthodes publiques de la classe

●length() :retourne le nombre de caractères de la chaîne
prototype: valeur length()

```
int long;  
long=str2.length() ; // long = 7
```

●empty(): retourne vrai si la chaîne est vide
prototype: booléen empty()

```
int etat;  
etat= str1.empty(); // vrai donc etat différent de zéro
```

●c_str(): retourne le tableau contenant la chaîne
prototype: chaîne c_str()

```
char nom[12];  
nom=str2.c_str(); // nom contient bonjour
```

●insert(): insertion d'une chaîne dans une autre à l'emplacement spécifié
prototype de la méthode:
chaîne insert(position, chaîne à insérer)

```
str1=str2.insert(3,str3); //str1 contient bonEjour
```

●erase(): efface une partie de la chaîne
prototype: chaîne erase(position, nombre de caractères)

```
str1=str2.erase(2,2); //str1 contient boour
```

●replace(): efface une partie de la chaîne et la remplace par une autre chaîne

prototype: chaîne replace(position, nombre de caractères, chaîne)

```
str1= str2.replace(2,3,"ne nuit"); // str1 contient bonne nuit
```

●find(): retourne la position dans la chaîne de la première occurrence d'une chaîne de caractères dans une autre à partir d'une position donnée
prototype: chaîne find(chaîne, position à partir de laquelle on recherche)

```
str1=str2.find("o",3); //str1=4
```

```
str1= str2.find("o",0); //str1=2
```

●substr() retourne une partie de la chaîne
prototype: chaîne substr(position, nombre de caractères)

```
str1=str2.substr(3,2); //str1 contient la chaîne jo
```

2 - Les opérateurs

Les opérateurs déjà étudiés s'appliquent aux chaînes de caractères

- affectation = vu dans les exemples
- plus + concatène 2 chaînes

```
str1= str2+str3; //str1 contient bonjourE  
str1= str2+"xyz"+str3; //str1 contient bonjourxyzE  
str2+=str3 ; str2 contient bonjourE
```

●Les opérateurs de comparaison sont utilisables(==, !=, <, >, <=, >=) et retournent un vrai ou faux

●Lecture de chaque caractère (comme pour les tableaux)
char ch;
ch = str2[1]; //ch contient le caractère o

Une coopérative agricole souhaite automatiser les phases d'épandages en utilisant une machine agricole dirigée à l'aide d'un GPS.

L'application finale située sur un PC embarqué sur la machine agricole, doit traiter des positions dans des zones géographiques de formes circulaires.

En tant que programmeur, votre chef de projet vous demande de créer une classe '**Cercle**' afin d'être utilisée dans l'application finale.

Cette classe devra contenir toutes les données membres nécessaires à caractériser un cercle.

De plus cette classe doit:

- permettre de vérifier si deux cercles ont des surfaces égales à 5% près.
- permettre de vérifier si les coordonnées du centre de deux cercles sont égales à 5% près.
- Disposer des constructeurs nécessaires.

L'application devant tourner sur un PC embarqué (ni écran ni clavier), la classe ne doit pas faire appel à des fonctionnalités d'affichage ou de saisie clavier.

Cependant un programme de test doit être mis en œuvre.

- Lister les caractéristiques essentielles à un cercle.
- En déduire les données membres de la classe (nom type).
- Ecrire la déclaration de la classe.
- Ecrire la définition des méthodes.
- Ecrire un programme de test.
- Tester la classe en utilisant le debugger.

RMQ :

Un GPS fournit un certain nombre d'informations telles que la position en degré/minute pour la latitude et la longitude.

Ces données et bien d'autres, sont fournies sous forme de trames NMEA émises toutes les secondes environ, composées de la manière suivante : (tous les caractères sont écrits en ASCII)

\$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47

ou:

GGA	Global Positioning System donnée fixe
123519	heure de la mesure 12:35:19 UTC
4807.038,N	Latitude 48 deg 07.038' N
01131.000,E	Longitude 11 deg 31.000' E
1	qualité de mesure:
	0 = invalide
	1 = GPS fixe (SPS)
	2 = DGPS fixe
	3 = PPS fixe
	4 = Real Time Kinematic
	5 = Float RTK
	6 = estimated (dead reckoning) (2.3 feature)
	7 = Manual input mode
	8 = Simulation mode
08	nombre de satellite utilisés.
0.9	Horizontal dilution of position
545.4,M	Altitude
46.9,M	Height of geoid (mean sea level) above WGS84 ellipsoid
	(empty field) time in seconds since last DGPS update
	(empty field) DGPS station ID number
*47	the checksum data, always begins with *