

## TP n° 10

La librairie STL est en général installée dans toutes les distributions de c++, en tout cas vous pouvez travailler sans difficultés sur lucien.

Sur vos portables, il est possible que vous deviez ajouter une option de compilation pour indiquer où se trouve la librairie. `g++ -I/usr/include/stl yourfile.cpp` ou faire un paramétrage équivalent sur netbean ou codeblocks selon votre environnement de travail.

**Exercice 1** Nous allons manipuler quelques classes de la STL en prenant pour exemple l’affichage de messages sur un écran LCD. Ces petits écrans ne peuvent afficher qu’un nombre limité de caractères, et pour afficher les longs messages on procède donc à une animation qui fait défiler le texte : seule une portion est affichée (elle correspond à la taille de l’écran), puis une seconde après, on procède comme si le texte était décalé d’un caractère vers la gauche, et c’est cette nouvelle portion qui est affichée, donnant ainsi l’impression d’une animation.

On souhaite que l’écran LCD interagisse avec les messages de la façon suivante : il importe dans un premier temps suffisamment de mots pour avoir un texte qui dépasse sa capacité d’affichage. Puis chaque seconde il affiche les premiers caractères de la façon dont nous l’avons décrite, tant que son buffer le lui permet. Dès que son buffer doit être rempli par un nouveau mot il interragit avec l’objet message pour lui en demander un. De façon symétrique, lorsque sur la gauche de son buffer il vient de terminer d’afficher un mot, ce dernier est remis à la fin du message, ce qui permettra de faire des affichages en boucle.

1. Définissez une classe Message qui regroupe un ensemble de mots :
  - à partir de la classe list de STL
  - Testez là en y ajoutant des méthodes d’ajout de mots en fin, de retrait en tête, et la redéfinition de l’opérateur d’affichage de tout le message en utilisant un itérateur.
  - Ajoutez une méthode statique `exempleMessage()` qui retourne un message type que vous aimeriez afficher par la suite.
2. Définissez une classe Ecran qui interagit avec le message de la façon décrite, et qui contient une méthode `loop()` qui affiche le message qui serait attendu chaque seconde.
3. Ajouter le traitement du temps en utilisant la bibliothèque appropriée. (Cherchez `sleep_for` ...)

**Exercice 2** On considère un héritage un peu théorique issu d’une classe mère O qui contient un champ entier public `o`, toutes les classes suivantes n’auront pas de champs propres, et les héritages sont publics. On ajoute les classes suivantes :

- A qui hérite de O

- B1 et B2 qui héritent virtuellement de O .
  - C qui hérite de A, B1, B2
1. Combien de variables de champs de nom *o* contiennent les objets des classes C ?
  2. Codez et vérifiez
  3. Ajoutez une classe D qui hérite de C
  4. Pouvez vous définir une classe qui hérite de O et de A ? de O et de B1 ?