Université Paris Diderot **Langages à objets avancés**
M1 **Année 2019-2020**

TP n° 11

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

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 votre IDE.

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 facon suivante : il importe dans un premier temps suffisament de mots pour avoir un texte qui depasse 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 remplis par un nouveau mot il interagit 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.

Vous écrirez vos boucles `for` d'abord « à l'ancienne » (en utilisant des itérateurs), puis avec la syntaxe *for-each* introduite dans C++-11 (`for(elemType elem: collection){ ... }`) et enfin en utilisant la fonction `for_each` de la bibliothèque `algorithm`.

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 intéragit 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` ...)