

© Copyright 2010 tv <thierry.vaira@orange.fr>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License,

Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover.

☞ can obtain a copy of the GNU General Public License : write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

SOMMAIRE

Utilisation de conteneurs de la STL.....	3
Introduction.....	3
Partie 1 : Les figures géométriques.....	3
Partie 2 : Une liste de figures.....	4
Liens.....	4

UTILISATION DE CONTENEURS DE LA STL

Introduction

Dans ce TP, nous allons étudier et manipuler plus en profondeur l'une des classes conteneurs de la bibliothèque standard de patron (STL). Nous allons aussi mettre en oeuvre les notions de typage dynamiques qui ont été vues en cours (fonctions virtuelles, etc). Il est très probable que vous ayez besoin de rechercher de l'information, et ce lien a été préparé dans ce but :

<http://pluton.up.univ-mrs.fr/eremy/Ens/Info1.C++/approfondir.html>

Partie 1 : Les figures géométriques

Développez une hiérarchie de classe destinées à mémoriser ou manipuler les propriétés de différentes figures géométriques. On propose de traiter au moins le cas des rectangles, cercles, triangles, carrés, sphères, parallélépipèdes rectangles, cubes... mais cette liste n'est pas limitative. Chaque classe devra permettre de mémoriser les données qui permettent de définir une instance, par exemple la longueur des deux cotés d'un rectangle, le rayon de la sphère, etc. Chaque instance devra disposer quand cela a un sens :

- d'une méthode double `périmètre()` qui renvoie la valeur du périmètre de l'instance sur laquelle on l'appelle ;
- d'une méthode double `aire()` qui renvoie l'aire de la surface de l'objet concerné ;
- d'une méthode double `volume()` qui renvoie le volume de la forme concernée.

On considérera que lorsque ces trois notions ne sont pas définies mathématiquement, alors la valeur renvoyée sera nulle. Par exemple, le volume d'un carré sera nul, et le périmètre d'une sphère également.

Partie 2 : Une liste de figures

Vous allez utiliser un `std::list<>` de la STL pour constituer une liste hétérogène de figures rassemblant n'importe lesquelles des formes définies plus haut. Vous adapterez votre programme pour pouvoir consulter, ajouter ou supprimer des figures dans cette liste. Ensuite vous ferez en sorte de pouvoir automatiquement enregistrer dans un fichier de texte les informations de ces classes pour pouvoir tout aussi automatiquement les recharger au prochain démarrage de votre programme. La consultation se fera grâce à un parcours des éléments à l'aide d'un itérateur.

Une fois tout ce qui précède opérationnel, on demande la possibilité de trier la liste suivant au choix une des trois possibilités suivantes :

- les valeurs croissantes de périmètre ;
- les valeurs croissantes de l'aire ;
- les valeurs croissantes du volume.

On utilisera la méthode `sort()` du patron de classe `std::list<>` dans sa version recevant un paramètre qui est la fonction de comparaison entre deux éléments. De l'aide peut être trouvée à ce sujet sur le site de la bibliothèque STL développée par SGI.

Liens

- Page du cours de programmation C/C++ d'Éric REMY :

<http://www.iut-arles.up.univ-mrs.fr/eremy/Ens/Info1.C++/index.html>

- Planche de TP n°4 d'Éric REMY :

<http://pluton.up.univ-mrs.fr/eremy/Ens/LPSIL.C++/tp4/index.html>

- Pour approfondir vos connaissances en C et C++ :

<http://www.iut-arles.up.univ-mrs.fr/eremy/Ens/Info1.C++/approfondir.html>

- C++ Reference : <http://www.cplusplus.com/reference/>

- Standard Template Library Programmer's Guide :

<http://www.sgi.com/tech/stl/>

- Aire et périmètre : <http://cedric.beltrami.free.fr/aireperimetre.html>