TP n° 4 de C++

Classe Pile - Vecteur et généricité élémentaire
Vecteur dynamique

« Une classe générique se construit plus facilement à partir d'une classe simple conçue sur un type de base. » B.S.

1) Ecrire le code C++ d'une classe Stack de caractères avec comme attributs une taille _size, un pointeur de pile _sp, et un pointeur sur la base de la pile : _base. Comme méthodes (inline implicite pour cette classe), on prévoit en plus de celles exigées par la forme canonique de Coplien : la méthode push() pour empiler un élément, la méthode pop() et une méthode getNbElements() qui retourne le nombre courant d'éléments présents dans la pile. Ecrire un programme principal qui teste cette classe
2) Ecrire une classe pile générique et implémenter les méthodes de façon déportées. Ecrire un programme principal dans un fichier séparé qui teste plusieurs instances Suivez bien le guide de style et les conseils pour savoir quel type de fichier utiliser et inclure.
3) Pour ceux qui n'ont pas fait le supplément du dernier TP écrire une classe vecteur de réels double précision sur le modèle de la classe String du TP précédent. Etudier le code donné sur la classe FloatVector. Pensez vous qu'il faille ajouter une méthode ? Adaptez ce code pour gérer des réels double précision. Notez bien où vous effectuez les changements. Tester et étudier l'option –MM du compilateur g++ (avec la commande g++ -MM *.cpp) puis proposer un makefile. Tester dans le programme principal les différentes méthodes du DoubleVector une à une. Prenez le temps de les développer et de les tester au fur et à mesure.

DoubleVector
- _size : entier - _pData : tableau de *double*
+ DoubleVector() + DoubleVector (int inSize) + DoubleVector(double * inDArray, int inSize) + DoubleVector(const DoubleVector & inDVector) +~DoubleVector() + opérateur d'affectation + opérateurs d'indexation [] + opérateur d'affichage avec flux << + opérateur d'addition

4) Ecrire le code C++ d'une classe Vector template en réutilisant le code précédent. C'est dans le fichier Main.cpp que l'on réalisera l'instanciation et l'utilisation de la classe instanciée. Vous testerez les méthodes sur des vecteurs d'entiers et de réels.

5) Modifier ce vecteur pour qu'il ait le comportement "dynamique" suivant. Lorsque le vecteur est plein et qu'un utilisateur désire encore ajouter un élément, il y a une nouvelle allocation dynamique de mémoire (Type amortis : nouvelle taille = ancienne-taille + ½ * ancienne-taille). Tester ce nouveau vecteur générique.

TIPS pour les classes génériques

Conseil : Les définitions de classes génériques (partie déclaration des classes et implémentation des méthodes) devraient se retrouver dans un unique fichier d'entête class.hxx.