

ISIMA 3^{ème} année, Examen d'Objets Avancés 2003

durée 2 heures

Questions de compréhension (1 h sans documents, 10 points)

Répondre aux questions suivantes, en les justifiant :

1. Concepts de base C++

- a. Justifier le recours systématique à la forme de Coplien
- b. Soit une classe Tableau encapsulant un tableau T de n valeurs réelles (*double*). Donner la définition sous forme de Coplien de cette classe : contenus complets des fichiers de déclaration et de définition. Inclure la surcharge de l'opérateur d'accès [] dans les deux formes possibles.
- c. Prévoir en plus deux méthodes permettant d'afficher d'une part le contenu de tout le tableau sur le flux *ostream*, et d'autre part d'afficher toujours sur *ostream* une seule composante identifiée par son index, en gérant une exception éventuelle de type *String*.
- d. Hériter une classe TableauSeuil à partir de la classe Tableau et ayant comme attribut supplémentaire une valeur seuil (réelle) sur les coefficients. Comment s'assurer à l'initialisation des valeurs que le seuil n'est jamais dépassé : pour les constructeurs et pour les accesseurs [] ? Pourquoi l'opérateur [] n'est-il pas adapté au contrôle de validité ? que proposez-vous ? Donnez l'ensemble du code de la classe TableauSeuil.
- e. Réécrire la définition de la classe Tableau sous la forme générique (*Template*) en spécifiant les méthodes que doit implémenter le type générique.

2. STL

- a. Quels sont les critères de choix pour préférer un *Vector*, une *List*, une *Deque* ?
- b. Quelles sont les différences entre les 4 conteneurs associatifs ?
- c. Quelles sont les quatre méthodes nécessaires à l'utilisation d'un itérateur dans une boucle ?

3. Conception objet

- a. Qu'est-ce qu'un Design Pattern ? Quelle est leur utilité ? (3 lignes max)
- b. Citez 3 exemples de Design Pattern avec le minimum utile et nécessaire de description.
- c. Avantages et inconvénients de la POO pour le génie logiciel ?

Conception - développement (1 h avec documents, 10 points)

Documents autorisés : notes de cours et supports de cours (différents cours de Bruno Garcia, support PowerPoint Nils Ferrand) uniquement.

Un jeu d'échec

On rappelle que les échecs se jouent sur un plateau 8x8 avec les pièces suivantes : 8 pions, 1 roi, 1 dame, 2 cavaliers, 2 fous, 2 tours pour chaque joueur. L'objectif fonctionnel est de représenter une situation de jeu pour les 2 joueurs, ainsi que son évolution au cours du temps de jeu. On ne codera pas les déplacements des pièces.

On conseille de lire toutes les questions avant de commencer...

1. Proposer une structure objet pour le jeu, le plateau et les pièces du jeu (de préférence en UML).
2. On fournit un tableau 8x8 de caractères qui décrit une configuration des pièces sur le damier (blancs en minuscule, noirs en majuscule : avec les lettres p,r,d,c,f,t). Proposer la méthode permettant de générer dynamiquement les pièces de la configuration et de les placer sur le damier.
3. On affiche à l'écran en mode texte la configuration courante (utilisez les mêmes lettres)
4. Définir une classe Coup sachant qu'il peut s'agir d'un déplacement avec ou sans prise d'une pièce adverse
5. Proposer un conteneur adapté pour stocker une liste de coups successifs
6. On fournit une instance de ce conteneur, ainsi qu'une configuration initiale (supposés compatibles). Proposer le processus de lecture itérative des coups, de mise à jour de la configuration, et de visualisation. On veillera à gérer proprement les nettoyyages.
7. Lorsqu'un pion atteint l'autre extrémité du damier, il y a promotion du pion : le joueur peut choisir de remplacer son pion par une autre pièce de son choix. Comment implémenter ce mécanisme de manière élégante ?
8. Analyser dans la conception précédente les Design Patterns qui ont été utilisés parmi : Singleton, Factory, Adapter, Observer, Iterator, Strategy, Prototype