

Examen AP4A – Automne 2021

Durée 1h30

Préambule :

- Les exercices 1 et 2 sont indépendants et doivent être rédigés sur des feuilles séparées.
 - Pour chaque méthode étudiée, préciser sa déclaration et sa définition.
 - Il est conseillé de lire l'ensemble de l'énoncé d'un exercice avant de commencer à répondre aux questions.
 - Toute tentative de triche entre étudiants ou en réutilisant du code sur internet sera sanctionnée.
-

Exercice 1 (10 points)

Un livre est caractérisé par un numéro unique appelé le numéro international standard du livre, (International Standard Book Number) ou ISBN en abrégé. Le code ISBN de chaque livre est formé de 10 chiffres et comporte 4 parties distinctes, ex. 0-13-146413-2

- La première section indique la zone dans laquelle se trouve l'éditeur.
- La deuxième partie représente la maison d'édition du livre.
- La troisième chaîne de chiffres est un code unique à chaque livre imprimé par l'entreprise.
- La quatrième partie est le dernier chiffre, connu sous le nom de chiffre de contrôle, il peut s'agir de n'importe quel nombre compris entre 0 et 9 ou X (qui est traité comme 10).

Pour l'exemple cité ci-dessus : le (0) indique qu'il provient d'un pays anglophone, le (13) indique qu'il a été publié par Prentice Hall, le (146413) représente le code unique du livre, le chiffre de contrôle (2) indique que la somme des chiffres de l'ISBN est 10.

1. Créer la classe ISBN comportant :

- Les attributs qui forment un ISBN
- Une méthode `getFullISBN()` qui retourne l'ISBN complet en chaîne de caractères sans tenir compte des tirets
- Une méthode `diviserISBN(string isbnNbr)` qui prend en paramètre une chaîne de caractères et la divise en quatre parties, comme expliqué ci-dessus, et affecte chaque valeur à l'attribut correspondant de l'ISBN

2. Créer une classe Livre qui représente des informations pertinentes sur un livre, y compris le titre, l'auteur, l'éditeur, la ville et la date de publication, le prix du livre, et le champ `isbnNbr` de type ISBN. Cette classe doit inclure aussi :

- Un constructeur paramétré qui affecte le titre, l'auteur, l'éditeur, l'ISBN
- Les méthodes getters et setters des attributs titre, auteur, éditeur et `isbnNbr`
- Une méthode `imprimerDetails()` : pour imprimer les informations d'un livre sous la forme :

Titre : Probability and Statistical Inference

Auteur : Elliot Tanis and Robert V. Hogg

Éditeur : Prentice Hall

ISBN : 0131464132

3. Un livre peut avoir différents formats : format papier, format PDF, et format audio. Chaque format est visualisé comme une sous-classe de la classe Livre. Écrire la forme canonique des trois classes : LivrePapier, LivrePDF, LivreAudio, sans tenir compte des attributs spécifiques des classes. Ajouter à chaque classe le constructeur paramétré correspondant qui affecte le titre, l'auteur, l'éditeur, l'ISBN.
4. Modifier les classes nécessaires pour ajouter à la fonction `imprimerDetails()` une ligne qui indique le format correspondant du livre dans le cas où elle est appelée par une classe fille.
5. Définir la fonction `main` dans laquelle :

- On demande de l'utilisateur la saisie des détails de 5 livres différents, en donnant séparément, pour chaque livre, le format, le titre, l'auteur, l'éditeur, et l'ISBN.
- Suivant le format, créer le type de livre correspondant et afficher ses détails, avec le format, avant de passer à l'autre.

Peu importe le format, un livre est toujours une bonne idée !

Exercice 2 (10 points)

Un SSIAD (Service de Soins Infirmiers A Domicile) est une structure qui intervient au domicile de patients pour réaliser des soins infirmiers. Plusieurs catégories d'employés travaillent dans un SSIAD, des administratifs et des soignants, les soignants pouvant être des infirmiers ou des aides-soignants.

On envisage de créer une application pour gérer le personnel d'un SSIAD en utilisant le langage C++. Pour cela, on dispose déjà d'une classe liste générique dont l'interface publique est la suivante :

```
// liste.h
template <class T> class liste {
    private : // ...
    public :
        liste() ;
        liste(const liste&) ;
        liste& operator=(const liste&) ;
        virtual ~liste() ;
        int size() const ;           // taille de la liste
        T& operator[](int) ;         // récupère un objet de la liste avec un indice
        liste& operator+(T&) ;      // ajoute un objet à la liste
};
```

On veut définir une classe C++ représentant un SSIAD et ses employés en réutilisant une seule instance de la classe liste générique. Chaque employé n'aura comme seule information qu'un identifiant de type entier qui sera noté 'id'.

1. Ecrire les classes employés nécessaires en précisant les attributs de classes et un seul constructeur permettant d'initialiser ces attributs.
2. Donner la classe 'SSIAD' en précisant ses attributs et le constructeur par défaut.
3. Ecrire l'opérateur d'ajout (operator+) d'un employé à la classe SSIAD.
4. On veut maintenant pouvoir afficher la liste des employés d'un SSIAD en précisant pour chaque employé sa catégorie et son id. Indiquer quelles modifications il faut apporter à votre code initial pour mener à bien cette opération, et que le code suivant soit opérationnel :

```
SSIAD ssiad ;
// ...
std::cout << ssiad ;
```

5. Ecrire le destructeur de la classe SSIAD. Faut-il appeler le destructeur de la classe liste ?
6. Quel problème peut poser l'opérateur[] défini dans la classe liste ? Peut-on y remédier ?
7. Est-il possible de compléter la forme canonique de Coplien de la classe SSIAD en écrivant le constructeur par recopie et l'opérateur d'affectation ? Si oui, indiquer ce qu'il faudrait faire pour y arriver.