
Département : Mathématiques & Informatique
Filière : Master en Réseaux et Systèmes Informatiques
Module : Programmation Orientée Objet avancée
Mardi 07 novembre 2023

Épreuve de contrôle N° 1

- La durée de l'épreuve est de 2 heures.
- La consultation des supports de cours sous format papier est autorisée.
- Les téléphones portables et les montres connectées sont formellement interdits.
- Vous devez écrire les codes commentés en C++.

Exercice 1 :

On voudrait gérer et manipuler une chaîne de caractères à l'aide d'une application C++.

1. Définir une classe `chaineCara` qui possède :
 - Deux attributs `longueur` de la chaîne (`int`) et `adr` (`char *`) d'une zone allouée dynamiquement.
 - Constructeur `chaineCara()` initialise une chaîne vide.
 - Constructeur `chaineCara(char *texte)` initialise avec la chaîne passée en argument.
 - Constructeur par recopie `chaineCara(chaineCara &ch)`.
 - Surcharge des opérateurs affectation (`=`), comparaison (`==`), concaténation (`+`) et accès à un caractère de rang donné (`[]`).
 - Une fonction membre de prototype `void affiche()` ; qui affiche la chaîne de caractères.
2. Définir une classe `chaineDerivee` qui hérite de la classe `chaineCara`.

```
1 Class chaineDerivee : public chaineCara {  
2     private:  
3         int type; float val;  
4     public:    //...  
5 }
```

`type` prendra deux valeurs : 0 ou 1.

- 1 si la chaîne désigne un nombre, par exemple "256" ou "256.9", exploitable par `atof` la valeur retournée sera `val`. `double atof(const char * ch)` ; permet de transformer une chaîne de caractères, représentant une valeur flottante, en une valeur numérique de type double.
- 0 dans les autres cas, par exemple "Examen" ou "Examen1".

La classe `chaineDerivee` possède aussi :

- Un constructeur de prototype `chaineDerivee()` ; qui initialise les 3 nombres à 0.
- Un constructeur de prototype `chaineDerivee(char *)` ; qui initialise les 3 nombres à 0 ainsi que la chaîne de caractères.
- Une fonction `void affiche()` qui appelle la fonction `affiche` de `chaineCara` et qui affiche les valeurs des 3 nombres.
- Un constructeur par recopie pour `chaineDerivee` ; qui initialise les 3 nombres à 0.
- Une fonction `void calcul()` qui donne les bonnes valeurs à `type`, `val`.

3. Définir la classe principale pour tester l'application.

Exercice 2 :

1. Un crayon a une longueur et un diamètre dont les valeurs par défaut sont resp. 10 cm et 2 cm.
 - (a). Définir la classe `crayon` dans un fichier `crayon.h`, comprenant les déclarations :
 - Un constructeur usuel, de copie et un destructeur.
 - Une surcharge de l'opérateur d'affectation.
 - Les fonctions membres doivent permettre d'initialiser un crayon, de consulter ses 2 attributs et de les mettre à jour, de l'afficher via une surcharge d'opérateur (`friend ostream & operator<< (ostream & out, const crayon & cray);`) sous la forme (8 4) pour un crayon de 8 cm de longueur et de 4 cm de diamètre.
 - (b). Ecrire les définitions des fonctions membres précédentes dans un fichier `crayon.cpp`.
 - (c). Ecrire la fonction principale dans un fichier `crayonTest.cpp`, réalisant les services suivants :
 - Création d'un premier crayon de 8 cm de longueur et de 4 cm de diamètre.
 - Création d'un deuxième crayon de 12 cm de longueur et de 3 cm de diamètre.
 - Changement de la longueur du deuxième crayon à 9 cm.
 - Affichage des deux crayons.
2. On s'intéresse maintenant aux crayons de couleur. Un crayon de couleur est un crayon possédant une couleur, représentée par une chaîne de caractères. Un crayon de couleur a une couleur par défaut, le noir.
 - (a). Ecrire une classe `crayonCouleur`, dans un fichier `crayonCouleur.h`, comprenant les déclarations :
 - Un constructeur usuel et un destructeur.
 - Une surcharge de l'opérateur d'affectation.
 - Les fonctions membres doivent permettre d'initialiser un crayon de couleur, de consulter son attribut et de le mettre à jour, de l'afficher via une surcharge d'opérateur sous la forme (8 4 rouge) pour un crayon de 8 cm de longueur, de 4 cm de diamètre et de couleur rouge.
 - (b). Ecrire les définitions des fonctions membres précédentes dans un fichier `crayonCouleur.cpp`.
 - (c). Ecrire la fonction principale dans un fichier `crayonCouleurTest.cpp`, réalisant les services suivants :
 - Création d'un crayon couleur de 12 cm de longueur, de 3 cm de diamètre et de couleur bleu.
 - Changement de la couleur du crayon à la couleur rouge.
 - Affichage du crayon couleur.
3. On s'intéresse maintenant aux fournitures scolaires d'un élève d'école primaire. Dans sa trousse, sont rangées des crayons, des crayons de couleurs, des gommes, des stylos billes, un taille-crayon, un double-décimètre.
 - (a). Définir le diagramme de classes afin de répondre à la problématique ci-dessus, en précisant les relations entre les classes ?
 - (b). Soit la classe suivante :

```
1 class fourniture {  
2 public:  
3 virtual void afficher(ostream & os) const = 0;  
4 };
```

Une trousse contient une collection de fourniture scolaires. Définir la classe `trousse`, dans un fichier `trousse.h`, comprenant les déclarations :

- d'un constructeur par défaut.
 - d'une seule fonction d'ajout permettant d'ajouter aussi bien un crayon qu'un crayon de couleur, qu'une gomme, qu'un taille-crayon, qu'un stylo bille, qu'un double décimètre.
 - d'une fonction d'affichage via une surcharge d'opérateur.
- (c) Ecrire les définitions des 3 fonctions précédentes dans un fichier `trousse.cpp`.