

Programmation Orientée Objet
L3 INFO – L3 MIAGE
Durée : 2h
Documents non autorisés

Exercice 1 (15 pts)

On considère un complexe qui dispose d'un ensemble de salles pour organiser des soirées. On veut implémenter un ensemble de classes permettant de gérer ces soirées qui peuvent être de différents types (soirées de famille, anniversaires d'entreprises, levée de fonds). Une soirée a une thématique et a lieu à une date précise dans une salle dédiée du complexe. Son coût de base spécifique correspond aux frais de location de la salle. Une soirée de gala est une soirée exceptionnelle souvent organisée pour une célébration exceptionnelle. Elle requiert ainsi plus d'engagement et de moyens. On distingue deux types de soirées de gala : les anniversaires d'entreprises et les levées de fonds.

On considère pour cela le fichier main.cpp fourni en annexe (qu'il s'agira de copier et de compléter)

I.

Écrire dans les fichiers Soiree.h et Soiree.cpp la classe Soiree avec les méthodes suivantes :

1. Un constructeur qui permet de construire une soirée à partir de ses caractéristiques.
2. Une méthode d'affichage d'une soirée avec ses spécificités.

II.

Un anniversaire d'entreprise est une soirée de gala dont l'organisation requiert un savoir-faire événementiel essentiel. L'anniversaire d'une entreprise est en effet un événement incontournable à la fois festif et marquant pour chaque collaborateur mais aussi pour les clients et partenaires. C'est ainsi qu'il faut gérer la communication (publicité, etc), le contact et les invitations des clients et partenaires. Le coût d'un anniversaire d'entreprise est égal au coût de base, plus les frais de communication, plus les frais de contact et d'invitation des clients et partenaires (un forfait), plus les frais réservés à cette manifestation (frais_anniversaire_entreprise).

Une levée de fonds est une soirée de gala dont l'objectif est de collecter un montant pour une cause précise. Pour sa réussite, il faut préparer des affiches et des supports de marketing. Le coût d'une levée de fonds est égal au coût de base spécifique plus les frais de communication, plus 2% du montant à collecter, plus les frais réservés à cette manifestation (frais_levee_de_fonds).

Écrire dans les fichiers Anniversaire_Entreprise.h et Anniversaire_Entreprise.cpp la classe Anniversaire_Entreprise avec les méthodes suivantes :

3. Un constructeur d'un anniversaire d'entreprise dont on lui fournit les éléments requis.
4. Une méthode d'affichage d'un anniversaire d'entreprise avec ses spécificités.

Écrire dans les fichiers Levee_de_fonds.h et Levee_de_fonds.cpp la classe Levee_de_fonds et ses méthodes suivantes :

5. Un constructeur d'une levée de fonds dont on lui fournit les éléments requis.
6. Une méthode d'affichage d'une levée de fonds avec ses spécificités.

III.

Vous allez maintenant définir une classe *Complexe* permettant d'organiser un ensemble de soirées.

La classe *Complexe* sera dotée des méthodes suivantes :

- un constructeur prenant en paramètre le nombre de soirées
- un destructeur
- une méthode permettant d'ajouter une soirée au complexe
- une méthode permettant d'afficher l'ensemble des soirées avec leurs spécificités (dont le coût de la soirée).
- Une méthode permettant de calculer le gain total obtenu suite à l'organisation des différentes soirées

Complétez votre main avec :

- une instruction permettant de créer un *Complexe* de 10 soirées et organisant deux fêtes de famille, deux anniversaires d'entreprise et deux levées de fonds;
- une instruction permettant d'afficher l'ensemble des soirées avec leur coût;
- une instruction qui renvoie le gain total du complexe.

Exercice 2 (5 pts)

On souhaite créer une classe *int2d* permettant de représenter des tableaux dynamiques d'entiers à deux indices, c'est-à-dire dont les dimensions peuvent ne pas être connues lors de la compilation. Plus précisément, on prévoira de déclarer de tels tableaux par une déclaration de la forme :

`int2d t(exp1, exp2);`

dans laquelle *exp1* et *exp2* désignent une expression quelconque (de type entier).

On surdéfinira l'opérateur `()`, de manière qu'il permette d'accéder à des éléments d'un objet d'un type *int2d* comme on le ferait avec un tableau classique.

Pour cela, on donne en annexe la définition à compléter de la classe *int2d*.

On ne cherchera pas à résoudre les problèmes posés éventuellement par l'affectation ou la transmission par valeur d'objets de type *int2d*. En revanche, on s'arrangera pour qu'il n'existe aucun risque de débordement d'indice.

- Dans un fichier « *int2d.h* », donnez la définition complète de la classe *int2d* (voir Annexe) ;
- Dans un fichier « *int2d.cpp* », définissez le constructeur, le destructeur et la surdéfinition de l'opérateur `()` ;
- Dans un fichier « *main.cpp* », donnez un exemple d'utilisation de la classe *int2d* dans une fonction « *main* ».

Annexe : la classe *int2d* à compléter

```
class int2d
{
    int nlig; // nombre de lignes
    int ncol; // nombre de colonnes
    int *adr; // adresse emplacement dynamique contenant les valeurs

public:
    // fonctions membres
```

Annexe

```
#include <string>
#include <iostream>
using namespace std;

// Inclusions à completer ici

// =====
// 
int main() {

    Soiree s1(Date(27,04,2024), "liens de famille", "Salle AMY", 100000);
    Soiree s2(Date(13,09,2024), "l'amitié", "Salle BETA", 90000);

    ...

    // affichage des soirées à compléter ici
    cout << s1 << endl;
    cout << s2 << endl;

    // Test des parties suivantes
    cout << "test des parties suivantes ..." << endl;
    // A COMPLETER
    return 0;
}
```