

## I52-Programmation Orientée Objet C++

Examen de TP

20 novembre 2020

Durée environ 2h30 - Documents non autorisés

Barème indicatif sur 20 : 1 (2 pts), 2 (5 pts), 3 (5 pts) , 4 (8 pts)

### CONSIGNES

Lire et exécuter les consignes suivantes :

- Avant d'entamer votre réflexion, créez à la racine de votre répertoire de travail l'arborescence I52/EXAMTP/
- A la fin de l'examen exécutez la commande  
    /home/partage/finexam I52 202011
- Sur la plateforme moodle se trouvent quatre fichiers, les en-têtes des classes à implanter et le programme principal qui teste ces classes
  - `musique.h`
  - `musiqueCompressee.h`
  - `albumMP3.h`
  - `main.cc`

### 1. MODULARITÉ

**EXERCICE 1.** Créer un Makefile qui produira un exécutable `main` à partir des quatre fichiers cités précédemment et des fichiers `musique.cc`, `musiqueCompressee.cc` et `albumMP3.cc`.

### 2. CLASSE MUSIQUE

#### EXERCICE 2.

- (1) Créer et compléter le fichier `musique.cc` en y implantant toutes les méthodes de la classe `Musique` déclarées dans `musique.h` : les constructeurs et accesseurs
- (2) Définir la méthode `float taille_mem()const` qui calcule et retourne la taille en Méga octets (Mo) d'un morceau de musique à partir de sa durée, sachant que le taux d'encodage est de 192 Kilo octets (Ko) par seconde de musique.
- (3) Définir la méthode `void affiche()`, de la classe `Musique`, qui affiche tous les attributs de la classe et la taille mémoire occupée par le morceau de musique en Mo.

### 3. CLASSE MUSIQUECOMPRESSEE

#### EXERCICE 3.

- (1) Créer et compléter le fichier `musiqueCompresse.cc` en y implantant toutes les méthodes de la classe `MusiqueCompressee`.
- (2) Surdéfinir la méthode `float taille_mem()const` pour la classe `MusiqueCompressee` : le taux d'encodage/compression est donné par l'attribut `taux`.
- (3) Surcharger de l'opérateur d'insertion dans le flux « pour cette classe pour afficher le `titre`, l'`auteur` de la classe `MusiqueCompressee` et la taille mémoire occupée par le morceau de musique en Méga octets (Mo).
- (4) Trouver et désigner dans fichier `main.cc` les appels à la surcharge de «

### 4. CLASSE ALBUMMP3

#### EXERCICE 4.

- (1) Créer et compléter le fichier `AlbumMP3.cc` en y implantant toutes les méthodes de la classe `AlbumMP3`.
- (2) Surcharger de l'opérateur d'affectation pour la classe `AlbumMP3`
- (3) Surcharger de l'opérateur `+` pour qu'il retourne une nouvelle instance de la classe `AlbumMP3` obtenue en ajoutant l'objet courant de la classe `AlbumMP3` et un morceau de musique de type `MusiqueCompressee` (l'objet courant ne doit pas être modifié).

```
AlbumMP3 operator+ (const MusiqueCompressee &);
```

- (4) Surcharger de l'opérateur `+` afin qu'il retourne une nouvelle instance de la classe `AlbumMP3` en ajoutant un morceau de musique de type `MusiqueCompressee` et une instance de la classe `AlbumMP3`.
- (5) **Surcharge de l'opérateur [] pour la classe `AlbumMP3`** : trouver et désigner en commentaires dans le fichier `main.cc` l'utilisation de cette surcharge.
- (6) Définir la méthode `float taille_album()` qui retourne la taille de l'album en Méga octets (Mo).

VALÉRIE GILLOT, UTLN, DÉPARTEMENT D'INFORMATIQUE-L3 INFORMATIQUE