

EXAMEN CPP – ZZ2 – 05 DECEMBRE 2019

MISE EN GARDE 1 : le compte de M. Yon a été piraté par la liste BDE GangsterZZ. Toutefois pour éviter une mauvaise note aux membres de notre équipe une fois qu'elle aura été complètement révélée, nous avons juste changé le thème de l'examen...

MISE EN GARDE 2 : inscrivez vos nom/prenom dans le fichier gangster.hpp. Par sécurité, cela vous sera aussi demandé à la première exécution du programme.

Voici votre examen de C++ ZZ2. Lisez tout le sujet avant de commencer : un certain nombre de tests ont été écrits pour être indépendants ou avec peu d'influence sur les autres.

A la fin de l'examen, vous devez rendre un programme qui compile.

Pour plus de simplicité, vous **DEVEZ** utiliser trois fichiers en plus de main.cpp : tests.cpp, gangster.hpp et gangster.cpp. Ne changez pas la casse et n'ajoutez pas de fichiers, cela casserait la chaîne de compilation et engendrerait une note nulle.

Toutes les déclarations seront dans l'entête, les définitions/implémentations seront dans le cpp.

La ligne de compilation est la suivante :
make

Si besoin, vous pouvez faire un make clean. La première compilation de la bibliothèque Catch est un peu longue.

Ne jamais recommencer un test qui a réussi. Tous les tests décommentés doivent compiler dans le même temps (sauf mention contraire).

Pour garder une trace exécutable de votre programme sans stress supplémentaire :

* Vous pouvez utiliser GIT. Un rappel des commandes GIT est fait à la fin de l'énoncé. La correction s'intéressera uniquement à la version disponible.

* Si vous préférez faire des copies de répertoires, tous les sous-répertoires de votre \$HOME sont analysés et seul le code menant à une meilleure note que les autres est gardé.

1/ Créer une classe Personne qui représente une personne avec un nom et les accesseurs qui vont bien.

[Personne1, Personne2]

2/ Une instance particulière de Personne est à créer : INCONNU. Elle sera utilisée plus tard

3/ Créer une classe Gangster avec deux attributs entiers : un identifiant et une influence. Pour un gangster de base, l'influence est de 1. L'identifiant est calculé automatiquement à chaque fois qu'un objet est créé en utilisant un attribut de classe non mentionné dans les tests. [Gangster]

4/ Créer une classe Chef, un chef est un gangster qui gère une équipe de ... gangsters. La méthode diriger() permet d'ajouter un membre à l'équipe. L'influence d'un chef se mesure à celle de son équipe avec un bonus. [Chef]

5/ Créer une classe InconnuException avec un message spécifique [Inconnu]

6/ Cette exception est levée quand on essaie de connaître la personne qui se cache derrière un gangster dont l'identité n'est pas connue [Exception]

7/ On dote la classe Personne d'une méthode equals() pour savoir si deux personnes portent le même nom [Revelation]

8/ On veut maintenant comparer deux gangsters : un gangster sera dit "inférieur à un autre" si le premier a une influence plus petite que l'autre ou, à influence identique, si le gangster est plus jeune (c'est-à-dire créé après l'autre) [Comparaison]

Vous allez maintenant créer une classe Famille qui est un ensemble de Gangsters.

9/ On veut également pouvoir lister la famille en listant les membres dont l'identité est connue par ordre alphabétique [Famille 1A]. Cette liste sera naturellement disponible avec << [Famille 1B]. Du code est laissé en commentaire pour vous aider à déboguer le cas échéant.

10/ On veut également pouvoir avoir une liste ordonnée du gangster le plus influent au moins influent [Famille 2]. Pour cela, on propose de créer un foncteur de comparaison entre deux gangsters connus par adresse si nécessaire. [Foncteur]

11/ Si vous lancez valgrind, vous devriez faire quelque chose de plus...

Commandes GIT

Préparation du compte (1 fois, au début)

```
git config --global user.name "un_nom"  
git config --global user.email "un_mail"
```

Préparation du répertoire

```
git init
```

A chaque nouveau fichier

```
git add fichier
```

```
git add *
```

Sauvegarder / Commiter

```
git commit -am "commentaire"
```

git log permet d'avoir la liste des sauvegardes avec un numéro de hachage

```
git checkout numero_de_hash
```

```
OU git revert numero_de_hash
```

Bonne courage !!!

