

step by step + explication de ma méthode:

Ce fichier PDF est important pour la compréhension rapide du code et aussi pour s'approcher à ma manière de réfléchir .

step by step :

- une/des classe(s) pour représenter différents types de personnes qui interviennent dans un des projets donnés dans le cadre d'une formation universitaire :
etudiantM1 (on mémoriserà numéro d'étudiant unique attribué automatiquement à la construction, nom, langages de programmation maîtrisés (au moins deux parmi cplusplus, javascript, php, python) ;
etudiantM2 (on mémoriserà numéro d'étudiant unique attribué automatiquement à la construction, nom, niveau d'expertise en gestion de projets (un entier compris entre 1 et 10) ;
enseignant (nom, numéro de bureau).
Pour cette/ces classe(s), on doit avoir un constructeur permettant de fixer toutes les informations sur la personne. Il n'y aura pas de mutateurs : les valeurs des attributs seront fixées à la construction.
- une méthode typepers retournant le type de personne(M1,M2,Enseignant)
- pour les étudiants de M1 une méthode maitriselangage prenant comme paramètre un langage et retournant vrai si l'étudiant maîtrise ce langage, faux sinon.
- un opérateur de sortie sur flux permettant de sortir sur le flux l'ensemble des informations mémorisées sur une personne, y compris le type de la personne.
- Un expert est soit un enseignant, soit un M2 ayant un niveau d'expertise supérieur à 7, soit un M1 maîtrisant au moins 3 langages.on doit coder une méthode expert retournant vrai si la personne est un expert, faux sinon.
- une classe exceptionuniversite, sous-classe de std::exception dont le constructeur prendra comme argument une chaîne et dont la méthode what retournera la chaîne en question.
- une classe université contenant un ensemble de personnes avec une méthode ajout permettant d'ajouter une personne. Cette méthode refusera d'ajouter une personne s'il existe déjà une personne de même nom dans l'université, dans ce cas-là, elle lèvera une exceptionuniversite avec le message « Nom en double : le nom ».
- une méthode rechercherdevs prenant comme paramètre un langage et retournant l'ensemble des étudiants qui maîtrisent ce langage.
- une méthode saisiem2 demandant à l'utilisateur de saisir les informations permettant de créer un étudiant de M2 et de l'ajouter à l'université. Si la création de l'objet est impossible

(niveau non compris entre 1 et 10, présence d'une personne de même nom), un message sera affiché, et l'utilisateur recommencera sa saisie².

→ Dans le cadre d'un enseignement de M1-M2, nous organisons des projets composés d'étudiants de M1, M2, encadrés par des enseignants. il faut une classe projet pour représenter cela. Une personne peut faire partie de plusieurs projets. À la construction, un projet sera « vide ».

→ une méthode ajout permettant d'ajouter une personne à un projet en respectant les conditions suivantes : il doit y avoir plus (ou autant) d'étudiants de M1 que de M2 ; il ne peut y avoir plus de deux enseignants. En cas d'erreur lors de l'ajout, la méthode lèvera une exception universite avec un message décrivant le problème.

→ Modifier la classe université pour y rajouter un ensemble de projet.
Écrire une méthode ajout permettant de rajouter un projet à une université. (aucune vérification à faire)

→ une méthode verifierprojets qui vérifie que les projets d'une université sont composés uniquement de personnes qui font partie de l'université. Cette méthode retournera la liste des personnes impliquées dans au moins¹ un projet et qui ne sont pas membres de l'université.

explication (pourquoi ces choix ?) :

Le code proposé ici utilise au maximum des algorithmes de la bibliothèque standard. Cela ne veut pas dire que c'était la seule façon de faire, mais c'est sans doute la façon la plus courte de le faire, une fois qu'on les connaît. Le main proposé est d'un intérêt (très) limité, il a pour unique but de tester les méthodes , j'ai essayé d'avoir une bonne gestion de mémoire. en c++ je considère l'utilisation de const/& très importante mais aussi l'utilisation du std::cout dans le main par exemple au lieu d'utiliser "using namespace std".

Merci pour votre attention.