

Durée : 1h00. Documents autorisés. À la fin de la séance, faites une archive (tar.gz ou zip) ne contenant que les sources (pas de fichiers objets ni exécutable) et postez-le sur l'espace Moodle dans le devoir « Contrôle Continu 1 ».

1. La classe demandée ici sera déclarée dans un fichier *horaire.hh* et le code de certaines (ou toutes les) méthodes sera donné dans un fichier *horaire.cc*. Aucune autre classe ne sera définie dans ces 2 fichiers. Écrire une classe *horaire* représentant un horaire d'une réunion. Un horaire est composé de trois informations : le mois (entier positif), le jour du mois, et l'heure de début d'une réunion. Par exemple, le 14 Novembre à 15h est représenté par mois : 11, jour : 14, heure : 15. Munir cette classe des méthodes suivantes : constructeur prenant comme paramètres trois entiers, pas de constructeur par défaut, accesseurs mois, jour, heure, pas de mutateurs.
2. Rajouter une méthode *tostring* retournant une chaîne de la forme 14/11-15h.
3. On veut maintenant représenter les messages échangés entre différentes personnes. Chaque message a un expéditeur (chaîne), des destinataires (vector ou list de chaînes) et un numéro d'identification unique attribué automatiquement à la construction. Il y a trois types de messages :
 Un *texte* contient un texte (comparable à un mél, en quelque sorte).
 Une *proposition* de réunion contient un descriptif de réunion (chaîne) et un **liste** d'horaires représentant les différents horaires possibles pour la réunion.
 Une *réponse* à une proposition contient **un** horaire choisi par une personne et l'identifiant de la proposition concernée.
 En réponse à une *proposition*, chaque destinataire de la proposition enverra un message de *réponse* à l'expéditeur de la proposition. Par exemple, à la proposition 1 expédiée par X à Y et Z dont le descriptif est « CC C++ » et les horaires possibles sont 14/11-15h et 15/11-8h ; Y répondra à X par la réponse 2 concernant la proposition 1 : 14/11-15h et Z répondra à X la réponse 3 à la proposition 1 14/11-15h.
 Écrire une (des) classe(s) pour représenter ces différents types de messages, vous n'écrirez que les constructeurs et accesseurs nécessaires, pas de mutateurs.
4. Écrire une méthode *sortieflux* prenant comme paramètre un flux de sortie et sortant sur le flux le message sous la forme « 1 X->Y Z PROPOSITION CC C++ : 14/11-15h 15/11-8h » pour une proposition, « 2 Y->X RÉPONSE À 1 : 14/11-15h » pour une réponse ou « 4 X->Y TEXTE blabla » pour un texte.
5. Écrire une méthode *reuniondate* prenant comme paramètre un jour et un mois, et retournant vrai si le message concerne une proposition de réunion dans laquelle (au moins) un horaire possible est à cette date ou une réponse avec cette date ; faux sinon.
6. Écrire une classe *basemessages* représentant un ensemble des messages, et munir cette classe (au moins) de méthodes permettant de construire un ensemble vide et ajouter un message.
7. Écrire une méthode *affichermessagesdate* prenant comme paramètre un mois et un jour et affichant la totalité des messages (propositions et réponses) concernant des réunions concernant cette date.
8. Écrire une méthode *suppressioninteractive* prenant comme paramètre un destinataire D, affichant les messages ayant pour destinataire D et permettant de choisir un de ces messages pour le supprimer. La suppression se fera par saisie de l'identifiant du message à supprimer, aucune vérification n'est demandée.
9. Écrire une méthode *repondre* prenant comme paramètre une proposition P, un destinataire D de cette proposition, et un horaire H. Si H fait partie des horaires proposés (dans P), la méthode ajoutera à la base une réponse à P expédiée par D à l'expéditeur de P et contenant l'horaire H. Sinon, la méthode ne fera rien.