



1 Présentation

Dans ce sujet, nous présentons quelques éléments d'un logiciel pour gérer des réunions. Vous vous appuierez sur la correction déposée sur Moodle pour la session 1 qui traite du même sujet.

Les noms des classes, attributs et méthodes vous sont indiqués en police de caractères **TrueType**. Les signatures des méthodes ne vous sont pas toujours données, dans ce cas, vous devez les déterminer.

Une **SalleReunion** accueillera des **Reunions**. Nous considérons ici deux catégories de réunions (qui seront des sous-classes de la classe **Reunion**) : la **ReunionPresentiel** dont tous les participants sont sur place et la **VideoConference** dont une partie seulement des participants sont en présentiel, et les autres participants sont sur des sites distants.

2 Représentation des réunions

On ne créera jamais d'instance directe de la classe **Reunion**, seulement des instances de ses sous-classes.

Toutes les réunions (**Reunion**) sont décrites par :

- un sujet (**sujet**),
- une date au format "AAAAMMJJ" (**date**). Par exemple la chaîne "20191218" représente le 18 décembre 2019,
- une heure de début et une heure de fin, qui sont des nombres entiers entre 8 et 20 (pour représenter des réunions se tenant entre 8h et 20h).

Une réunion en présentiel (**ReunionPresentiel**) est une sorte de réunion avec la particularité suivante :

- un nombre de participants (**nbParticipants**).

Une vidéo-conférence (**VideoConference**) est une sorte de réunion avec les particularités suivantes :

- un nombre de participants présents (**nbParticipantsPresentes**),
- un nombre de participants sur les sites distants (**nbParticipantsDistants**),
- un nombre de sites distants connectés (**nbSitesDistants**).

Question 1. Pourquoi dans la correction a-t-on indiqué que la classe **Reunion** est abstraite ?

Question 2. Ecrivez l'accessor **setDate** en contrôlant au mieux la valeur de l'année (les 4 premiers caractères). Les années peuvent aller de "1970" et au-delà. Pour cela, extrayez la sous-chaîne composée des 4 premiers caractères et vérifiez qu'elle est plus grande que "1970" dans l'ordre lexicographique, sinon, affichez un message.

Question 3. Ecrivez en Java, dans la classe **ReunionPresentiel** uniquement, un constructeur avec paramètres permettant d'initialiser les attributs propres aux instances.

Question 4. Ecrivez en Java, dans les classes **Reunion** et **ReunionPresentiel** uniquement, une méthode **nbPlacesNecessaires** qui retourne le nombre de places nécessaires. Ce nombre de places nécessaires n'est pas défini pour les réunions en général. Pour les réunions en présentiel, il s'agit du nombre de participants.

Question 5. Ecrivez en Java, pour les classes **Reunion** et **ReunionPresentiel** uniquement, une méthode **toString**, retournant une chaîne de caractères représentant l'objet. Indiquez bien dans quelle classe vous écrivez quelle méthode.

- Pour toutes les réunions, cette chaîne contient : la date (gardez le format AAAAMMJJ), l'heure de début, l'heure de fin, la durée, le sujet, et le nombre de places nécessaires.
- Pour les réunions en présentiel, cette chaîne contient comme ci-dessus la date (gardez le format AAAAMMJJ), l'heure de début, l'heure de fin, la durée, le sujet, le nombre de places nécessaires. Est-elle différente de la méthode héritée ? Qu'est-ce que cela vous inspire ?

3 Représentation des salles de réunion

Une salle de réunion (**SalleReunion**) a une adresse, un numéro, une capacité (nombre de personnes qu'elle peut accueillir), et une liste de réunions déjà accueillies, en cours ou à venir (réunions du passé, du présent, du futur) . Vous utiliserez l'entête et les attributs de la classe **SalleReunion** de la correction.

Question 6. Ecrivez en Java, dans la classe `SalleReunion`, une méthode `ajout` permettant de lui ajouter une réunion en vérifiant que cette réunion n'y est pas déjà (sinon afficher un message d'erreur).

Question 7. Ecrivez en Java, dans la classe `SalleReunion`, une méthode `ajoutPendantCovid` permettant de lui ajouter une réunion en vérifiant que cette réunion n'a pas lieu en présentiel (sinon afficher un message d'erreur).

Question 8. Ecrivez en Java, dans la classe `SalleReunion`, une méthode `NbParticipantsCumulé` retournant la somme des nombres de participants des réunions de la salle.

Question 9. Ecrivez en Java, dans la classe `SalleReunion`, une méthode `rechercheNbParticipantsSuperieurA` retournant la liste des réunions de cette salle et ayant un nombre de participants supérieur à un nombre passé en paramètre.