

## Contrôle de conception et programmation par objets

Durée : 1h15 (1h35 pour les étudiants disposant d'un tiers temps). Le barème est donné à titre indicatif.

On s'intéresse à la modélisation d'un site proposant l'organisation de forums de discussion. Les personnes inscrites sur le site (appelées ici membres) peuvent participer à plusieurs forums en s'y inscrivant. Une personne peut participer avec des rôles différents dans un forum :

- organisateur : l'organisateur d'un forum en détermine les caractéristiques principales, et peut supprimer le forum ; il peut également écrire des messages dans le forum ;
- modérateur : le modérateur peut censurer les messages qui lui paraissent inappropriés ; il peut également écrire des messages dans le forum ;
- simple participant : le simple participant écrit des messages dans le forum.

Un même membre peut jouer des rôles différents dans différents forums, il peut par exemple modérer un forum et être un simple participant dans un autre.

La modélisation de la figure 1 est proposée pour représenter les rôles des membres inscrits dans les forums.

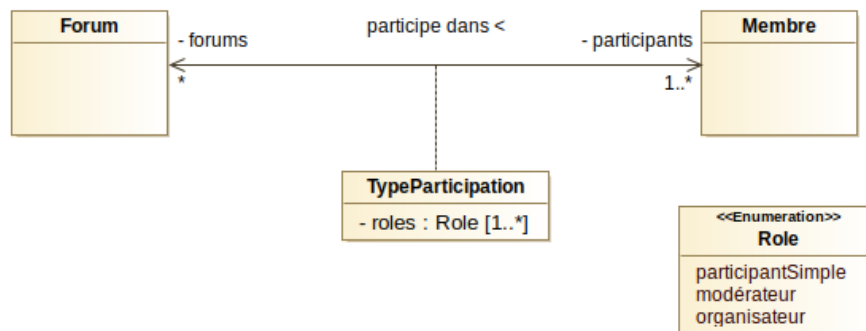


FIGURE 1 – Un diagramme de classes pour les rôles dans les forums

**Question 1.** (1 point) D'après ce diagramme, un membre peut-il participer dans un forum en étant à la fois son modérateur et son organisateur ? Justifier.

**Question 2.** (2 points) Avant d'implémenter ce modèle, il faut réfléchir à une modélisation où la classe d'association n'apparaîtrait plus. Proposez un modèle (un diagramme de classes UML) sans classe d'association, et permettant d'exprimer les rôles joués dans les forums, comme l'exprime la figure 1.

On s'intéresse maintenant aux interventions dans les forums. On propose la modélisation de la figure 2.

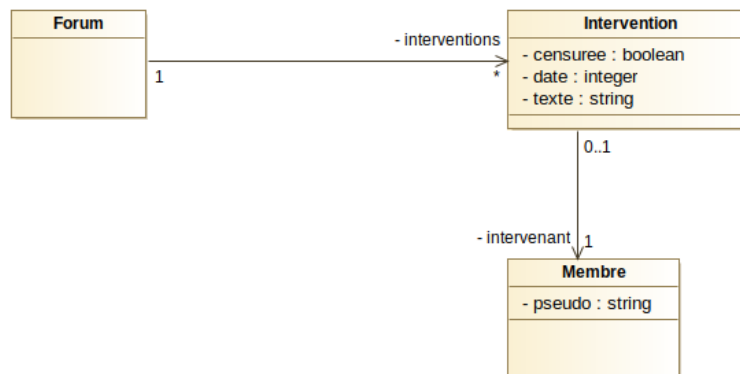


FIGURE 2 – Un diagramme de classes pour les interventions

Dans un forum, il y a plusieurs interventions. Chaque intervention est réalisée par un membre, qui joue alors le rôle d'intervenant. Chaque intervention a une date, qui est ici représentée pour simplifier par un entier (nombre de millisecondes écoulées depuis le 1er janvier 1970). Une intervention est censurée ou pas. Une intervention a un texte.

On propose l'extrait de code Java suivant pour la classe Intervention.

---

```

public class Intervention{

    // attributs
    private boolean censuree;
    private long date; // un long est un entier long
    private String texte;
    private Membre intervenant;

    // constructeur
    public Intervention(String texte, Membre intervenant){
        censuree=false;
        date=new Date().getTime(); // la date du jour
        this.texte=texte;
        this.intervenant=intervenant;
    }

    // accesseurs
    public boolean isCensuree() {
        return censuree;
    }

    public long getDate() {
        return date;
    }

    public String getTexte() {
        return texte;
    }

    public Membre getIntervenant() {
        return intervenant;
    }
}

```

---

**Question 3.** (1 point) Donner une ligne d'instruction Java permettant de créer une intervention non censurée, à la date du jour, de texte "hop" et d'intervenant m (où m est un membre déjà créé). L'instance sera affectée à ("rangée" dans) une variable que vous déclarerez.

Dans la classe Forum, on représente en Java l'extrémité d'association navigable de rôle **interventions** par une ArrayList d'interventions. On prévoit un certain nombre de méthodes :

- **nbInterventions()** qui retourne le nombre d'interventions existant dans le forum ;
- **supprPlusAncienneIntervention()** qui supprime la plus ancienne des interventions du forum ;
- **datePlusRecenteIntervention()** qui retourne sous forme d'un entier (un long ici) la date de la plus récente intervention du forum, ou -1 s'il n'y a actuellement pas d'interventions dans le forum.
- **diffusion()** qui diffuse aux participants du forum qu'une nouvelle participation a été ajoutée.

La classe Forum correspondante vous est donnée en Java ci-après.

---

```
public class Forum{
    private ArrayList<Intervention> interventions = new ArrayList<Intervention>();

    public int nbInterventions() {
        return interventions.size();
    }

    public void supprPlusAncienneIntervention() {
        if (!interventions.isEmpty()) interventions.remove(0);
    }

    public long datePlusRecenteIntervention() {
        if (!interventions.isEmpty()) return interventions.get(nbInterventions()-1).getDate();
        else return -1;
    }

    private void diffusion() {
        // diffuse aux participants le fait qu'il y a une nouvelle intervention
    }
    ...
}
```

---

**Question 4.** (3 points) Ecrire en Java dans la classe Forum une méthode qui compte et retourne le nombre d'interventions dans le forum d'un membre de pseudo donné (on prendra le pseudo en paramètre).

**Question 5.** (3 points) Ecrire en Java dans la classe Forum une méthode qui retourne les interventions non censurées du forum ayant eu lieu entre une date d1 et une date d2 (représentées toutes les deux par des entiers représentant le nombre de millisecondes écoulées depuis le 1er janvier 1970 ; donc comparer deux dates revient à comparer deux entiers).

**Question 6.** (2 points) Décrire comment, en Java, trier l'ArrayList **interventions** par ordre de date croissante, en utilisant la méthode de classe (static) **sort** de la classe Collections, prenant en paramètre une collection à trier.

On souhaite mettre en place en plus des forums précédemment décrits, des forums dits lents pour lesquels une intervention ne peut être ajoutée au forum que s'il s'est écoulé un certain temps (identique pour tous ces forums) depuis la précédente intervention, et des forums dits à petite mémoire, pour lesquels les interventions ne sont pas toutes mémorisées, seules les **n** dernières interventions sont mémorisées (où **n** est le nombre maximum d'interventions, qui est le même pour tous ces forums).

**Les deux questions suivantes sont à réaliser conjointement et en cohérence.**

**Question 7.** (3 points) Modélisez dans un diagramme de classes UML les forums dans une hiérarchie, dans laquelle vous placerez :

- les attributs qui vous semblent judicieux au vu de l'énoncé précédent
- une/des méthode(s) `ajoutIntervention(i:Intervention)` permettant d'ajouter **i** à **interventions**. Pour les forums normaux, cette méthode ajoute **i** à **interventions**, sans condition puis appelle la méthode `diffusion()`. Pour les forums lents, cet ajout et cette diffusion n'ont lieu que si le laps de temps depuis la dernière intervention est respecté (on utilisera la méthode `datePlusRecenteIntervention()`). Pour les forums à petite mémoire, cet ajout et cette diffusion sont sans condition, mais l'ajout peut entraîner au besoin la suppression de l'intervention la plus ancienne si le nombre maximum d'interventions mémorisées est atteint (on utilisera la méthode `supprPlusAncienneIntervention`).

**Question 8.** (5 points) Donnez la/les implémentation(s) de la (des) méthode(s) `ajoutIntervention(i:Intervention)` introduites précédemment. Pour obtenir la date courante sous forme d'entier, vous utiliserez : `new Date().getTime()`. Vous pouvez par exemple écrire : `long now=new Date().getTime();`