
Option Informatique – U.E. POO – Session 1

Examen – 2 heures – Documents de Cours autorisés**L'agenda**

Vous allez devoir participer à la conception et à la programmation d'une application de gestion d'un agenda. Un agenda électronique permet de mémoriser des événements, ponctuels (comme un examen de P.O.O. le mercredi 22 mai de 14h à 16h) ou chroniques (comme un cours de P.O.O. tous les lundis, mardis, mercredis à partir du 18 janvier et pendant 6 semaines). Certains choix ont déjà été faits pour l'application.

Un agenda électronique appartient à une personne dont on mémorise le nom et le prénom. Ensuite, il gère essentiellement la structure de données qui lui sert à mémoriser les événements. Il s'agit d'un dictionnaire (une `Map`) dont les clés sont les années et la valeur l'agenda de l'année. À une année est donc associée un autre dictionnaire (une autre `Map`) dont la clé est le numéro d'un jour dans l'année, et la valeur tous les événements qui auront lieu ce jour-là. Pour simplifier notre prototype, les dates sont représentées par le numéro du jour dans une année (le 23 mai 2007 est le 143ème jour de l'année 2007), et les horaires par des flottants (vous n'aurez pas à vérifier, dans vos réponses, que les horaires sont des flottants entre 0. et 24.).

Une partie de l'application a été écrite :

```
// tous les import nécessaires
public class Agenda {

    private Map<Integer,Map<Integer,Collection<Evenement>>>> monAgenda
        = new HashMap<Integer,Map<Integer,Collection<Evenement>>>>();
    private String nom;
    private String prenom;

    Agenda(String nom, String prenom){
        this.nom = nom;
        this.prenom = prenom;
    }
    public void creerEvenementPonctuel(int annee,int jour,
                                       float heureDebut,float heureFin,
                                       String lieu,String titre){

        // A_COMPLETER
    }
    public void creerEvenementChronique(int annee,int jour,
                                       float heureDebut,float heureFin,
                                       String lieu,String titre,
                                       int periode,int nbreFois){

        // A_COMPLETER
    }
    public void afficheEvenementsDuJour(int annee,int jour){
        // A_COMPLETER
    }
}
```

Le titre contient l'information sur le contenu de l'événement (par exemple *cours de P.O.O.*, ou bien *Examen de P.O.O.*). Un événement chronique est caractérisé par les mêmes informations qu'un événement ponctuel, plus la périodicité (le nombre de jours auxquels un événement revient; par exemple, 7 pour un cours hebdomadaire) et le nombre de fois où la périodicité est renouvelée.

Il est de la responsabilité des événements, chroniques ou ponctuels, de gérer leur(s) enregistrement(s) sur un agenda. L'enregistrement consiste en l'ajout de l'événement dans la structure de données principale de la classe **Agenda**. Il est demandé par l'agenda lors de la création d'un événement. La suppression est le retrait de cet événement dans cette même structure de données. La suppression est demandée directement à l'événement (une interface graphique sera ensuite ajoutée à notre application, permettant à l'utilisateur d'accéder directement aux événements créés). Un même événement, s'il est chronique, doit être enregistré plusieurs fois, pour chacun des jours où il aura lieu. Ainsi, un utilisateur qui veut représenter un cours qui aura lieu 12 fois sur 13 semaines consécutives, devra créer un événement chronique qui démarre le jour du premier cours, de périodicité 7, 13 fois, et ensuite supprimer le cours de l'agenda pour le jour où le cours n'a pas lieu. Par souci de simplification, un événement chronique ne peut avoir des occurrences que pour une seule année : si on crée un événement qui démarre le 300ème jour de l'année 2006, qui doit se répéter 5 fois tous les 30 jours, l'événement ne sera enregistré que pour les 300ème, 330ème et 360ème journées de 2006. La suite sera oubliée. On considérera que toutes les années font 365 jours (pas d'année bissextile).

Voici l'interface qu'un événement doit implémenter :

```
interface IEvenement {
    // pour s'inscrire dans un agenda
    public void enregistreToi(Map<Integer,Collection<Evenement>> agendaUnAn);
    // pour supprimer l'événement de l'agenda
    public void supprimeToi(int jour);
}
```

Questions

L'ordre des questions est celui dans lequel je vous conseille d'aborder le problème. Il vous indique également ce que j'attends dans votre copie et le barème qui sera employé.

Q 1 . 2 points – Décrivez les classes **Evenement**, **EvenementChronique** et **EvenementPonctuel** (signatures de méthodes, sans implémentation). Vous pourrez compléter par (et implémenter) d'autres méthodes ou d'autres attributs qui vous sembleront utiles, au moment où vous en aurez besoin dans les questions suivantes.

Q 2 . 3 points – Écrivez tous les constructeurs nécessaires pour les événements, ainsi que les méthodes `toString`.

Q 3 . 2 points – Complétez la méthode `afficheEvenementsDuJour` de la classe **Agenda**.

Q 4 . 2 points – Complétez la méthode `creerEvenementPonctuel` de la classe **Agenda**.

Q 5 . 2 points – Complétez la méthode `creerEvenementChronique` de la classe **Agenda**.

Q 6 . 5 points – Implémentez la méthode `enregistreToi` pour les événements.

Q 7 . 4 points – Implémentez la méthode `supprimeToi` pour les événements.