

Évaluation de TP module M2207 (POO & Java)

avril 2019
durée : 2h
responsable : M.Munier

Sujet 4 - Planning

Documents autorisés : Pour cette évaluation vous êtes autorisés à utiliser tous les documents pédagogiques qui vous ont été fournis : supports de cours "M2207 & Java" (éventuellement annotés) distribués par M.Munier, corrections des TD/TP, notes de TD et notes de TP. Tout autre document (documentations récupérées sur le web, photocopies de livres, etc. . .) est formellement interdit. Dans le même ordre d'idée, aucune assistance électronique (calculatrice, ordinateur de poche, téléphone portable, PDA, . . .) n'est autorisée. Toutes les notions vues en cours sont supposées acquises et, le cas échéant, doivent être mises en œuvre.

Restitution de vos travaux : Vous développerez toutes les classes de cette évaluation dans un seul et unique répertoire dont le nom est **votre nom de famille en minuscules**. À la fin de la séance je passerai avec une clé USB pour récupérer ce répertoire qui ne devra contenir **que les classes du présent exercice**. Tout manquement aux directives sera sanctionné sur la note.

Conseils : Durant cette évaluation de TP (qui ne dure que trois heures), votre but n'est pas de tout faire, mais plutôt de nous prouver que vous êtes capables de vous débrouiller en programmation Java devant une machine. Cela signifie bien évidemment savoir écrire des classes correctes (héritage, constructeurs, méthodes, attributs, exceptions, . . .), mais également savoir interpréter les erreurs de compilation et/ou d'exécution, les corriger, tester vos classes, etc. . . Dit de manière un peu plus directe, **il est plus "intéressant" pour vous d'avoir un programme fonctionnel validant une partie des classes que d'avoir écrit toutes les classes mais sans en avoir compilé aucune !**

★ ★ ★

L'objectif de ce sujet est de programmer en Java des classes nécessaires à la réalisation d'une application de gestion d'un emploi du temps : des heures, des créneaux horaires, un planning, . . .

1. Écrire la classe **Heure** permettant de représenter une heure sous la forme **HH:MM**. Outre les constructeurs et les méthodes de base pour l'accès aux attributs, il faudra veiller à définir les méthodes suivantes :
 - la méthode **String toString()** qui renvoie une chaîne de caractères représentant l'heure et qui permet également d'imprimer une heure directement avec un **System.out.println(heure)**,
 - la méthode **int compareTo(Heure)** qui retourne :
 - une valeur négative si l'heure sur laquelle est invoquée cette méthode est "inférieure" à l'heure passée en paramètre,
 - une valeur nulle (i.e. l'entier 0) si les deux heures sont égales,
 - une valeur positive si l'heure sur laquelle est invoquée cette méthode est "supérieure" à l'heure passée en paramètre
2. Écrire la classe **Creneau** permettant de gérer des créneaux horaire. Outre les constructeurs et les méthodes de base pour l'accès aux attributs, il faudra définir la méthode **boolean conflictWith(Creneau)** qui teste si oui ou non le créneau horaire passé en paramètre entre en conflit (i.e. chevauchement) avec

le créneau horaire sur lequel est invoquée cette méthode. Vous implémenterez également une méthode `double duree()` qui retournera la durée en minutes de ce créneau horaire.

3. Écrire maintenant la classe `Planning` représentant un emploi du temps. Pour simplifier cet exercice, nous nous limiterons à l'emploi du temps **d'une personne sur une journée** uniquement. La principale méthode de cette classe sera `void ajouterCreneau(Creneau c)` permettant, comme son nom l'indique, d'ajouter un nouveau créneau horaire au planning. La classe `Planning` devra également disposer d'une méthode `double dureeTotale()` calculant la durée totale de tous les créneaux horaire insérés dans un planning.
4. Dériver la classe `Creneau` en trois sous-classes : `CreneauCours`, `CreneauTD` et `CreneauTP`. La particularité de ces trois sous-classes est que le résultat de leur méthode `duree` ne sera plus en **heures réelles** mais en **heures équivalent TD (ETD)** selon le calcul suivant : 1 heure de TD vaut 1 heure ETD, 1 heure de cours vaut 1,5 heure ETD et 1 heure de TP vaut 2/3 heure ETD.

Afin de pouvoir valider votre travail, il vous est naturellement demandé d'écrire un programme de test créant plusieurs créneaux horaire de différents types et de les ajouter à un planning.

* * *

Afin de vous aider dans votre tâche, vous trouverez la documentation en ligne du Java SE 7 à l'URL suivante : <https://docs.oracle.com/javase/7/docs/api/>.