

## Interrogation 3 – TimeTracker

### Consignes

- Vous pouvez utiliser **toutes les notes écrites que vous voulez** ainsi que tous les programmes qui se trouvent dans votre dossier.
- A faire **exclusivement sous Linux**. Vous disposez de **2 heures**.
- Chaque fois qu'un point est terminé, copiez la classe dans le **casier** de votre professeur.
- Soyez attentifs, au **style**, à l'**indentation** et la **clarté** du code.

### Le problème

Un projet est constitué de différentes tâches. On voudrait comptabiliser le temps consacré à chaque tâche afin de vérifier qu'elle ne dépassent pas un temps prévu.

### Tâche

(6 points)

Une tâche est composée des attributs suivants :

- Nom et description de la tâche : 2 chaînes
- Durée prévue pour la tâche : un entier (le nombre de minutes)<sup>1</sup>
- Durée actuelle consacrée à la tâche : un entier

Il faut prévoir un constructeur qui reçoit en paramètre : le nom , la description et la durée prévue (le durée actuelle étant initialisée à 0)

En ce qui concerne les méthodes, définir

- Les 4 getters mais pas de setter
- La méthode « void ajouterDurée(int) » qui ajoute une durée à la durée actuelle
- La méthode « boolean estDépassé() » qui indique si le temps prévu est dépassé
- Une méthode « toString() »

### Projet

(9 points)

Un Projet regroupe un ensemble de Tâches. Utilisez une collection pour la retenir et pas un tableau (c'est le seul attribut).

Le constructeur n'a pas de paramètre et crée une liste vide.

Les méthodes sont :

- « ajouterTâche(nom, description, duréePrévue) » qui ajoute une tâche au projet. On peut supposer que la tâche n'existe pas encore.
- « ajouterDurée(nom, durée) » qui ajoute la durée donnée à la durée actuelle de la tâche dont on donne le nom. Pour simplifier le problème, on peut supposer que la tâche existe bien.

<sup>1</sup> Il s'agit bien sûr d'une simplification. Dans un cadre plus réel, on utiliserait une classe Durée par exemple.

- « getDépassés() » qui retourne une liste de chaînes contenant les noms des tâches qui ont pris plus de temps que le temps prévu.
- « nbTâches() » qui retourne le nombre de tâches
- La méthode « toString() »

## Javadoc (2 points)

Ecrivez une partie de la Javadoc de la classe Projet (sauf la méthode toString()).

## Tests unitaires (3 points)

Ecrivez des tests unitaires pour la classe Projet. Vous testerez que

- Le constructeur crée bien une liste de taille 0 (inutile de tester les getters)
- La méthode nbTâches() fonctionne
- La méthode ajouterTâche() fonctionne (en testant la taille après ajout)
- Je ne vous demande pas de tester la méthode getDépassés()