

# Java POO Challenges :

## Exercice 1 :

on veut développer un logiciel pour générer des sujets d'examen. Un programmeur a écrit la classe suivante avant de démissionner (il a gagné au loto). Vous devez continuer le développement.

```
class Question{  
  
    String enonce;  
  
    int difficulte=50; // la difficulte varie de 0 à 100  
  
    Question(String e){  
  
        this.enonce = e;  
  
    }  
  
    void setDifficulte(int d){  
  
        this.difficulte = d;  
  
    }  
  
    int getDifficulte(){  
  
        return this.difficulte;  
  
    }  
  
    String getEnonce(){  
  
        return this.enonce;  
  
    }  
  
}
```

On veut représenter les questions de QCM (Questionnaire à Choix Multiples). Chaque question à choix multiple a, en plus d'un énoncé et d'une difficulté, un ensemble de réponses possibles, chaque réponse ayant un énoncé et une valeur de vérité (une réponse peut être vraie ou fausse).

**a.** Proposez du code pour représenter les questions à choix multiples.

**b.** Un sujet d'examen est une liste de questions (de QCM ou autre) avec un barème. Chaque question du sujet a donc un nombre de points. Sans modifier les classes que vous avez écrites avant, écrivez du code objet pour représenter les sujets, avec en particulier une méthode qui permet d'obtenir la difficulté moyenne du sujet. La difficulté moyenne est la somme des difficultés des questions du sujet, divisée par le nombre de questions.

## Exercice 2 :

on veut réaliser un programme qui représente des montres et les gens qui les portent. une montre donne l'heure et les minutes. On peut initialiser une montre soit à partir d'un couple heure/minute donné, soit en utilisant l'heure affichée par une autre montre. Il doit être possible de faire avancer l'heure d'une montre en ajoutant une minute (attention, les minutes sont limitées à 60 et les heures à 24).

a. Écrivez une classe qui représente les montres telles que décrites au dessus et une méthode principale qui crée une montre affichant 13h45 et une autre montre qui est un clône de la première. Une personne a un nom et peut éventuellement porter une montre. On peut faire porter une montre donnée à une personne, si elle n'en a pas déjà une. On peut aussi lui enlever sa montre si elle en a une. Une personne peut demander l'heure à une autre, qui lui donne l'heure sous forme d'une chaîne de caractères, en consultant sa montre si elle en a une (sinon elle peut retourner une chaîne vide).

b. Écrivez une classe qui représente les personnes telles que décrites au dessus.

c. On veut faire en sorte que chaque montre ne soit portée que par une seule personne. Proposez des ajouts/modifications des deux classes précédentes pour garantir cela.

## Exercice 3:

on veut réaliser un programme de gestion d'une bibliothèque. Un livre possède un titre et un ensemble de thèmes (qui sont des mots-clés). Une bibliothèque est une liste de livre et possède un nom.

a. Écrivez du code objet pour représenter les livres et les bibliothèques. On veut pouvoir rechercher des livres en fonction d'un thème. On va utiliser un mécanisme d'itérateur, c'est-à-dire des objets qui implémentent l'interface suivante :

```
public interface Iterator<E>{  
    // renvoie true s'il existe encore un élément à parcourir  
    public boolean hasNext();  
    // renvoie l'élément courant et passe au suivant  
    public E next();  
}
```

Il faut pouvoir créer un itérateur avec un thème, et le parcours d'une bibliothèque avec un tel itérateur ne va parcourir que les livres ayant pour thème celui de l'itérateur.

b. Écrire du code objet pour représenter les itérateurs sur bibliothèque