

UEF4.3. Programmation Orientée Objet

CONTROLE FINAL

2h- Documents et téléphones portables interdits-

Page 5 à remettre avec le cahier d'examen

Exercice 1

1. Soient les classes Moteur et Vehicule dont le code java est donné ci-dessous

```
class Moteur{
public double puissance;
public int vitesse;
public Moteur(double p){puissance = p; vitesse = 0 ;}
}

class Vehicule{
public Moteur moteur;
public static int nbVehicules;
public int matricule;
public Vehicule(Moteur m){ moteur= m ; nbVehicules ++;matricule =
nbVehicules;}
public void accelerer(double vitesse){moteur.vitesse = vitesse;}
public void afficherVitesse(){
System.out.println("Le véhicule portant le matricule "+matricule+" roule à
une vitesse de "+moteur.getVitesse()+ "Km/h");}
}
```

- Quel est le principe fondamental de la POO qui n'est pas respecté dans ce code ? Expliquez.
 - Définissez ce principe et expliquez comment son application peut améliorer la qualité d'un programme ?
 - Donnez une version améliorée du programme.
2. On crée la classe Voiture ci-dessous.

```
public class Voiture extends Vehicule{
    private final int vitesseMaximale;
    public Voiture(int vmax){
        vitesseMaximale = vmax;

        public void accelerer(double vitesse){
            System.out.println("Vous devez entrer une valeur entière ! ");}
}
```

- Le code du constructeur de la classe Voiture est-il correct ? Si la réponse est « non » expliquez pourquoi et corrigez-le.
- On veut ajouter à la classe Voiture la méthode `accelerer` ayant comme argument un entier représentant la vitesse à atteindre. Cette méthode augmente progressivement la vitesse du moteur jusqu'à atteindre la vitesse souhaitée. Si la vitesse maximale est atteinte la méthode lance et traite une exception de type `DepassementVitesseException` en affichant l'alerte : « Impossible de dépasser la vitesse Maximale ! »
 - Est-ce une surdéfinition (surcharge) ou une redéfinition ?
 - Donnez le code de cette méthode en indiquant les éventuelles modifications à apporter au code existant.

- c. On veut ajouter à la classe voiture une méthode `afficherVitesse` qui affiche la vitesse courante à l'écran et ajoute un rappel comme dans l'exemple suivant :

Le véhicule portant le matricule xxxxxx roule à une vitesse de 80 Km/h.
Vitesse à ne pas dépasser : 110 Km/h

c.1) Est-ce une surdéfinition ou une redéfinition ?

c.2) Donnez le code de cette méthode.

3. Lisez attentivement le code suivant extrait de la méthode `main`

```

Vehicule vehicules[] = new Vehicule[2];
vehicules[0] = new Vehicule(new Moteur(1.2));
vehicules[1] = new Voiture(new Moteur(2.0), 160);

double v1 = 200; int v2 = 200;

for (Vehicule h : vehicules) {
    h.accelerer(v1);
    h.afficherVitesse();
    System.out.println("-----");
}
System.out.println("*****");
for (Vehicule h : vehicules) {
    h.accelerer(v2);
    h.afficherVitesse();
    System.out.println("-----");
}

```

a) Qu'affiche ce code ?

b) Il existe une incohérence dans le résultat retourné par ce code. A quel niveau se trouve-elle et à quoi est-elle due ?

4. Soient les instructions suivantes :

```

Voiture v1 = new Voiture(new Moteur(1.2), 120);
Voiture v2 = new Voiture(new Moteur(1.8), 140);
Vehicule h1 = new Vehicule(new Moteur(1.2));
Vehicule h2 = new Voiture(new Moteur(0.8), 100);

```

Remplissez le tableau ci-dessous pour chacune des instructions suivantes (Ces instructions sont indépendantes les unes des autres)

`v1 = h1; v2 = h2; h1 = v1; h2 = h1; v1 = h2;`

Instruction	Erreur à la compilation ?	Correction possible ?	Correction	Erreur à l'exécution ?
.....

5. Etant donné que dans ce programme seule la classe `Vehicule` utilise la classe `Moteur`, quelle modification pourriez-vous apporter à la conception et pourquoi ?

Exercice 2

On désire concevoir et réaliser, suivant l'approche orientée objet, un programme permettant d'évaluer automatiquement les réponses des étudiants à un quiz (test). Un quiz contient un ensemble de questions et un total de points. Les questions peuvent être de différents types : certaines offrent un ensemble de propositions parmi lesquelles l'étudiant doit choisir une seule réponse (QCU : question à choix unique) ou plusieurs réponses correctes (QCM : question à choix multiples) alors que d'autres n'offrent aucune proposition et demandent à l'étudiant de saisir une réponse courte (QO : question ouverte). Chaque question comporte un énoncé de type `String` et un attribut `nbPoints` de type `double` correspondant au nombre de points qui lui sont attribués. Elle comporte aussi une solution de type `String` correspondant à la bonne réponse. Dans le cas d'une QCM, la solution est la concaténation de toutes les propositions correctes séparées par des virgules. En plus de ces trois attributs, une question à choix contient aussi des propositions parmi lesquelles l'étudiant doit choisir une ou plusieurs réponses selon qu'elle soit une QCU ou une QCM. Dans le cas d'une QCM, une mauvaise réponse est pénalisée par le retrait d'un certain nombre de points défini par l'enseignant afin d'éviter que l'étudiant coche toutes les cases.

1. Proposez une modélisation orientée objet pour implémenter les différents types de questions en respectant au mieux les principes fondamentaux de la POO.
 - a. Tracez le diagramme des classes en précisant le type des classes et les liens entre elles.
 - b. Précisez pour chacune des classes ses attributs et ses méthodes (hormis les constructeurs, les getters et les setters) en utilisant le tableau suivant.

Type-class nom-class			
Liste des attributs			
Modificateur d'accès	Type de l'attribut	nom de l'attribut	signification
...			...
Liste des méthodes			
Modificateur d'accès	Valeur de retour	Signature de la méthode	Rôle
...			...

2. Soit la classe `Quiz` comportant une collection appelée `questions` destinée à contenir les questions du quiz et un attribut `TotalPoints`, de type `double`.
 - a. Quel est le concept Orienté objet qui permet de regrouper toutes les questions au sein de la même collection ?
 - b. En sachant qu'une question ne doit pas figurer plus d'une fois dans le même quiz, quel est le type de collections le plus adéquat ? Donnez l'instruction permettant de créer cette collection. Que faut-il faire pour garantir que le type de collections choisi fonctionne correctement ? Expliquez en donnant le code java nécessaire.
3. Pour évaluer les réponses d'un étudiant au quiz, nous utilisons une classe appelée `EvaluationQuiz` dont le code source incomplet est donné ci-après. L'objet `reponses` contient les réponses de l'étudiant selon l'ordre d'apparition des questions dans le quiz : la réponse à la première question se trouve à l'indice 0 et la réponse à la dernière question se trouve à l'indice (n-1). Si un étudiant ne répond pas à une question, un blanc (« ») lui est associé comme réponse afin de garantir la bonne correspondance entre les questions et les réponses.

```
import java.util.*;

public class EvaluationQuiz{
    private Quiz quiz;
    private ArrayList <String> reponses;

    public EvaluationQuiz(Quiz quiz, ArrayList<String> reponses ){
        this.quiz = quiz;
        this.reponses = reponses
    }

    public double evaluerQuiz(){
        double note = 0;
        ArrayList<Question> questions = new ArrayList(quiz.getQuestions()); // ligne 14

        // code à ajouter

        return note;
    }
}
```

- A quoi sert l'instruction à la ligne 14 ?
 - Est-ce que la collection choisie en réponse à la question 2.b garantit que le résultat retourné par la méthode `evaluerQuiz` est correct ? Expliquez.
 - Si votre réponse à la question précédente est non, proposez un autre type de collections plus adéquat en expliquant les modifications à apporter à votre programme.
 - Complétez la méthode `evaluerQuiz`.
4. Si votre conception respecte les concepts fondamentaux de la POO, vous avez certainement utilisé l'un d'eux pour simplifier l'écriture de la méthode `evaluerQuiz`. Quel est ce concept? Expliquez à quel niveau il a été utilisé et son principe de fonctionnement.
5. Ci-dessous la fenêtre qui s'affiche à l'écran lorsque l'étudiant répond à une question ouverte. Complétez le code source donné en page 5 afin d'obtenir cette fenêtre en sachant qu'un click sur le bouton « soumettre » provoque l'affichage de la réponse de l'étudiant sur la console.

