

Cours et photocopiés autorisés. Autres documents interdits

On prendra soin d'écrire le code java le plus propre possible, en particulier en ce qui concerne les spécificateurs.

Exercice 1

Soit la classe :

```
Point.java
public class Point {
    private int x;
    private int y;

    public Classe1(int x, int y) {
        this.x= x;
        this.y= y;
    }

    public int getX() {
        return this.x;
    }

    public int getY() {
        return this.y;
    }
}
```

On écrit le code suivant :

```
public class MauvaisCode {
    public Point milieu(Point p, Point q)
    {
        Point m;
        m= new Point(0,0);
        m.x= (p.x + q.x) / 2;
        m.y= (p.y + q.y) / 2;
        return m;
    }

    public static void main(String a[]) {
        Point p= new Point(3, 5);
        Point q= new Point(4, 2);
        Point m= milieu(p,q);
        System.out.println(m);
    }
}
```

Question 1.1

En supposant que la classe `Point` est correcte, Expliquez quelles sont les erreurs contenues dans le code de la classe `MauvaisCode`. Quand c'est possible, proposez des solutions.

Exercice 2

On suppose qu'il existe une classe `InformationsSysteme`, dotée des constructeurs et méthodes suivantes :

```
private InformationsSysteme ()
    constructeur de la classe.

public static InformationsSysteme getInformationSysteme
    ()
    Retourne l'objet InformationsSysteme qui correspond à l'ordinateur sur lequel
    tourne le programme.

public int getMemory ()
    retourne la mémoire.

public int getCharge ()
    retourne la charge.
```

Question 2.1

Écrire un programme `AfficherInfoSysteme.java`, qui affiche la mémoire et la charge du système. On suppose que `InformationNonDisponibleException` descend de `Exception`.

Exercice 3

Soient les classes :

MaConsole	CompilateurJava
+ MaConsole() + lireLigne(): String + ecrire(s: String)	+ CompilateurJava() + compile(fichier: String): Programme

InterfaceTexte
- c: CompilateurJava - console: MaConsole
+ InterfaceTexte() + demandeEtCompile(): Programme

ErreurSyntaxeException
+ numeroLigne: int
+ ErreurSyntaxeException(String msg, int numLigne)

On suppose que toutes les classes sont écrites et fonctionnent, sauf `InterfaceTexte`. On précise que :

- `lireLigne` permet de lire une entrée de l'utilisateur. `ecrire` affiche un message;

- compile a la documentation suivante :

```
public Programme compile (String fname)
    throws ErreurSyntaxeException,
FileNotFoundException
    compile le fichier java dont le nom est fname, et retourne
    un objet Programme. La méthode peut retourner l'exception
    FileNotFoundException si le fichier n'existe pas, et l'excep-
    tion ErreurSyntaxeException si le fichier contient une erreur.
```

- la méthode demandeEtCompile va demander un nom de fichier à l'utilisateur, le compiler et retourner l'objet Programme correspondant. S'il y a un problème, la méthode doit redemander un nom de fichier jusqu'à ce qu'il arrive à créer un objet Programme. Dans le cas d'une erreur de syntaxe, on veut, de plus, afficher le message associé ainsi que le numéro de la ligne concernée.

Question 3.1

Écrire complètement la classe InterfaceTexte, sans utiliser d'autres classes que celles qui sont citées ici.

Exercice 4

Une compagnie de bus vous a demandé d'écrire une application pour permettre à ses utilisateurs de connaître les horaires.

Vous avez pris les décisions suivantes :

- une heure (de passage par exemple) sera représentée par un objet de classe Heure :

Heure
- heures : int - minutes : int
+ Heure(heures : int, minutes : int) + getHeures() : int + getMinutes() : int + toString() : String + estAprès(h : Heure) : boolean

La méthode estAprès permet de comparer deux heures. Elle retourne vrai si l'heure à laquelle elle s'applique est plus tardive que celle qu'elle prend en paramètre.

- les horaires d'un bus seront conservés en mémoire, dans un objet de classe Horaire. En interne, un objet Horaire sera représenté par un tableau ou un Vector d'heures. On désire :
 - pouvoir ajouter une Heure (on peut supposer que les heures seront ajoutés par ordre croissant) ;
 - pouvoir consulter l'horaire. On veut connaître le nombre de passage des bus prévu dans la journée, et avoir accès à chacune des Heures de passage ;

- disposer d'une méthode qui, étant donné une `Heure` passée en paramètre, renvoie l'heure du prochain passage de bus.
- La représentation interne de la classe ne doit pas être accessible.

Question 4.1

écrire la classe `Heure`.

Question 4.2

écrire la classe `Horaire`.