

I4 : Programmation Objet - Examen
Durée : 1h30 - Aucun document autorisé

Important : dans toutes les classes que vous écrirez, tout attribut devra être déclaré privé.

1. (10 points) On veut faire un programme qui modélise et affiche des polynômes, c'est-à-dire des expressions de la forme $a_0 + a_1x + a_2x^2 + \dots + a_nx^n$, où n est le degré du polynôme et les a_i sont des coefficients des monômes de degré i .

1.1 *Ecrivez la classe Monome qui représente un monôme, c'est-à-dire une expression de la forme $a.x^b$, où a est le coefficient et b est le degré. Cette classe devra définir entre autres les méthodes suivantes :*

- un constructeur `Monome(float coeff, int degre)` qui crée un monôme de coefficient `coeff` et de degré `degre`.
- la méthode `void additionne(float coeff)` qui additionne un monôme de même degré de coefficient `coeff` (au monôme sur lequel on appelle la méthode), c'est-à-dire, qui ajoute `coeff` au coefficient du monôme.
- la méthode `void additionne(Monome m)` qui additionne le monôme `m` s'il a le même degré (sinon, ne fait rien).
- la méthode `void multiplie(float coeff, int degre)` qui multiplie le monôme par un monôme de coefficient `coeff` et de degré `degre`.
- la méthode `void multiplie(Monome m)` qui multiplie le monôme par un monôme `m`.
- la méthode `void affiche()` qui affiche (sans retour à la ligne) le monôme sous la forme `a x^b`

1.2 *Ecrivez la classe Polynome qui représente un polynôme. Cette classe devra définir entre autres les méthodes suivantes :*

- un constructeur `Polynome(float coeff, int degre)` qui crée un polynôme de degré `degre`, dont tous les coefficients des monômes de degré inférieur à `degre` sont nuls.
- la méthode `void additionne(float coeff, int degre)` qui additionne au polynôme le monôme de coefficient `coeff` et de degré `degre`.
- la méthode `void additionne(Monome m)` qui additionne au polynôme le monôme `m`.
- la méthode `void multiplie(float coeff, int degre)` qui multiplie le polynôme par le monôme de coefficient `coeff` et de degré `degre`. Rappel: $(a_0 + a_1x + a_2x^2 + \dots + a_nx^n) \times (cx^d) = (c.a_0)x^d + (c.a_1)x^{d+1} + (c.a_2)x^{d+2} + \dots + (c.a_n)x^{n+d}$.
- la méthode `void multiplie(Monome m)` qui multiplie le polynôme par le monôme `m`.
- la méthode `void affiche()` qui affiche le monôme sous la forme `a0 x^0 + a1 x^1 + a2 x^2 + ... + an x^n`, en évitant d'afficher les monômes dont le coefficient est nul.

1.3 *Ecrivez la classe TestePolynome qui permet l'exécution du programme suivant : on crée le polynôme $3.6 + 5.9x + 6.1x^2$, on le multiplie par le monôme $2x^3$ et on affiche le résultat (qui sera $7.2 x^3 + 11.8 x^4 + 12.2 x^5$).*

2. (10 points) *On veut faire un programme qui modélise des individus qui pensent et communiquent. Certains sont francs, d'autres sont hypocrites. Certains ont leur propre pensée, d'autres suivent la pensée des autres. On définit d'abord les classes suivantes :*

```
interface Pensant {  
    String pense();  
}
```

```

interface Communicant {
    void parle(Communicant x); // parle à x (et donc x l'écoute)
    void ecoute(String s); // ecoute une parole
}
class Penseur implements Pensant {
    private String avis;
    Penseur(String avis) {this.avis = avis; }
    public String pense() { return avis; }
}

```

Dans cet exercice, il faudra au maximum réutiliser les classes existantes (par composition ou héritage). Dans les questions 2.1 à 2.4, il faudra définir des classes qui permettront d'exécuter la suite d'instructions suivante :

```

PenseurFranc pf = new PenseurFranc("Eh bien à mon avis...");
PenseurHypocrite ph = new PenseurHypocrite("Sans aucun doute, il est vrai que...");
SuiveurFranc s1 = new SuiveurFranc();
SuiveurFranc s2 = new SuiveurFranc();
System.out.println(s1.pense()); // affiche rien
pf.parle(s1);
System.out.println(s1.pense()); // affiche "Eh bien à mon avis..."
s1.parle(ph);
ph.parle(s2);
System.out.println(s2.pense()); // affiche "Eh bien à mon avis..."
System.out.println(ph.pense()); // affiche "Sans aucun doute, il est vrai que..."

```

2.1 Ecrivez la classe `Franc` qui représente les individus francs, c'est-à-dire, qui disent exactement ce qu'ils pensent (à ceux à qui ils parlent). Ce qu'ils écoutent ne leur fait pas d'effet. Cette classe doit définir les méthodes `void parle(Communicant x)` et `void ecoute(String s)`.

2.2 Ecrivez la classe `Hypocrite` qui représente les individus hypocrites, c'est-à-dire, qui disent la dernière chose qu'ils ont entendu (indépendamment de ce qu'ils pourraient penser). Cette classe doit définir les méthodes `void parle(Communicant x)` et `void ecoute(String s)`.

2.3 Ecrivez la classe `SuiveurFranc` qui représente les suiveurs francs, c'est-à-dire, les individus francs qui pensent la dernière chose qu'ils ont entendu (et ne pensent rien s'ils n'ont encore rien entendu).

2.4 Ecrivez les classes `PenseurFranc` et `PenseurHypocrite` qui représentent des penseurs francs et des penseurs hypocrites.

2.5 Ecrivez les classes `Gourou` et `Adeptes` qui sont telles que :

- le gourou est un penseur qui édicte ce qu'il pense et ce que ses adeptes doivent penser, grâce à la méthode `void edicte(String dogme)`. La classe `Gourou` ne peut avoir qu'une instance.
- un adepte pense ce que le gourou pense, même si ça change à chaque édicte. Un adepte n'écoute pas ce qu'on lui dit (il connaît les édits de son gourou en les lisant). La classe `Adeptes` ne contient que les méthodes `String pense()` et `void parle(Communicant x)`.

Ces deux classes doivent permettre d'exécuter la suite d'instructions suivante :

```

Adeptes a = new Adeptes();
System.out.println(a.pense()); // affiche "rien"
Gourou.leGourou().edicte("A bas les épinards");
System.out.println(a.pense()); // affiche "A bas les épinards"

```