Héritage comme généralisation

Exercice 1 : Formaliser le schéma

Dans la question 3.2 du TP 5, nous avons défini le schéma comme plusieurs objets (points, points nommés et segments) qui sont référencés par des variables différentes. Il serait plus *logique* et *pratique* d'avoir une seule variable qui représente le schéma (on l'appellera naturellement `schema`). Comme un schéma est constitué d'un nombre variable d'éléments, on peut le représenter par un tableau. Si nous appelons X le type des éléments de ce tableau, nous pouvons alors écrire le code du listing 1

1.1. Indiquer à quelles conditions sur X les lignes suivantes compilent.

```
schema[nb++] = s12;
schema[nb++] = barycentre;
schema[i].afficher();
schema[i].translater(4, -3);
```

1.2. Quel code sera exécuté pour `x.afficher()` et `x.translater(4, -3)` ?

1.3. Indiquer les autres éléments à définir sur X ? Justifier la réponse.

1.4. Donner un nom plus significatif à X.

Exercice 2 : Écrire la classe X et adapter l'application

2.1. Est-ce que l'on sait écrire le code des méthodes `afficher` ou `translater` de X ?

2.2. Peut-on créer des instances de X ?

2.3. Quels constructeurs définir sur X ?

2.4. Quand ces constructeurs seront-ils appelés ?

2.5. Écrire le code de la classe X.

2.6. Lister et effectuer les modifications à apporter aux autres classes de l'application.

Exercice 3 : Construire le schéma en utilisant les listes

Au lieu d'utiliser un tableau comme dans l'exercice 2, on veut utiliser l'interface `List` et sa réalisation `ArrayList` du paquetage `java.util` (en particulier la méthode `add` et `foreach`).

3.1. Indiquer les avantages et inconvénients des listes par rapport aux tableaux.

3.2. Construire le schéma en utilisant une liste.

Exercice 4 : Définir un groupe

Dans un éditeur de schémas mathématiques, il serait pratique de pouvoir grouper plusieurs X pour les manipuler comme un seul et leur appliquer à tous, en une seule fois, la même opération (translater, afficher, etc.).

4.1. Sachant que la classe `X` est abstraite, la classe `Groupe` est-elle abstraite ou concrète ?

4.2. Écrire la classe Groupe et l'utiliser (`ExempleSchemaGroupe`).

4.3. On souhaite pouvoir mettre un groupe dans un groupe. Par exemple, on souhaite grouper les trois segments, puis ce groupe et le barycentre. Comment faire ?

Listing 1 – La classe `ExempleSchemaTab` (extraits)

```
1
2  public class ExempleSchemaTab {
3
4     public static void main(String[] args)
5     {
6        // Créer les trois segments
7        Point p1 = new PointNomme(3, 2, "A");
8        Point p2 = new PointNomme(6, 9, "S");
9        Point p3 = new Point(11, 4);
10       Segment s12 = new Segment(p1, p2);
11       Segment s23 = new Segment(p2, p3);
12       Segment s31 = new Segment(p3, p1);
13
14       // Créer le barycentre
15       double sx = p1.getX() + p2.getX() + p3.getX();
16       double sy = p1.getY() + p2.getY() + p3.getY();
17       Point barycentre = new PointNomme(sx / 3, sy / 3, "C");
18
19       // Définir le schéma (vide)
20       X schema[] = new X[10]; // le schéma
21          // 10 : capacité suffisante ici, non contrôlée dans la suite.
22       int nb = 0;  // Le nombre d'éléments dans le schéma
23
24       // Peupler le schéma
25       schema[nb++] = s12;
26       schema[nb++] = s23;
27       schema[nb++] = s31;
28       schema[nb++] = barycentre;
29
30       // Afficher le schéma
31       System.out.println("Le␣schéma␣est␣composé␣de␣:␣");
32       for (int i = 0; i < nb; i++) {
33          schema[i].afficher();
34          System.out.println();
35       }
36
37       // Translater le schéma
38       System.out.println("Translater␣le␣schéma␣de␣(4,␣-3)␣:␣");
39       for (int i = 0; i < nb; i++) {
40          schema[i].translater(4, -3);
41       }
42
43       // Afficher le schéma
44       System.out.println("Le␣schéma␣est␣composé␣de␣:␣");
45       for (int i = 0; i < nb; i++) {
46          schema[i].afficher();
47          System.out.println();
48       }
49    }
50
51 }
```