TD n° 1
Licence Informatique (L2)
« Programmation objet avancée »

F. BERTRAND

Année universitaire 2020–2021

Concepts abordés :
- Méthodes de classe
- Principe d'encapsulation
- Utilisation de la relation d'héritage (spécialisation, généralisation)

1 Représentation de nombres complexes

Pour illustrer la notion de méthode de classe, nous allons définir une classe `Complexe` représentant un *nombre complexe* à l'aide d'une représentation cartésienne constituée d'une partie réelle et d'une partie imaginaire ($z = x + iy$).

Travail à effectuer :

1. Définir un ensemble de constructeurs permettant de créer un nombre complexe :
   - en précisant la valeur des parties réelle et imaginaire ;
   - en ne donnant aucun paramètre (les 2 parties sont alors égales à zéro) ;
2. Définir l'addition de 2 nombres complexes, à la fois comme une méthode d'instance et comme une méthode de classe. Le résultat retournera un nouveau nombre complexe correspondant à la somme des 2 nombres additionnés.
3. Définir la multiplication de 2 nombres complexes sous la forme de deux méthodes d'instance. La première prenant un complexe en paramètre et la seconde prenant simplement un réel en paramètre. Une seule méthode (la première) est nécessaire, cependant la présence de la seconde fera que la multiplication d'un complexe avec un réel sera plus simple (pas de nécessité de créer un complexe ayant une partie imaginaire nulle) et plus efficace (moins de calculs).

2 Figures géométriques

Le but de cet exercice est d'illustrer la notion d'héritage en utilisant une hiérarchie de figures géométriques. Soit les classes `Rectangle` et `Cercle` suivantes :

```
1  public class Rectangle {
2          private double largeur;
3          private double longueur;
4          public Rectangle(double larg, double longueur) {
5                  this.largeur = larg;
6                  this.longueur = longueur;
7          }
8          public double donneSurface()  { return this.largeur * this.longueur; }
9          public double donneLongueur() { return this.longueur; }
10         public double donneLargeur()  { return this.largeur;  }
11         public void changeLargeur(double l)  { this.largeur = l;  }
12         public void changeLongueur(double l) { this.longueur = l; }
13 }
```

```java
public class Cercle {
        private double x, y;  // abscisse et ordonnee du centre
        private double rayon;
        public Cercle(double x, double y, double r) {
                this.x = x;
                this.y = y;
                rayon = r;
        }
        public void affiche() {
                System.out.println("centre = (" + this.x + ", " + this.y + ")");
        }
        public double donneX() { return this.x; }
        public double donneY() { return this.y;        }
        public void changeCentre(double x, double y) {
                this.x = x;
                this.y = y;
        }
        public double donneSurface() { return Math.PI * this.rayon * this.rayon; }
        public boolean estInterieur(double x, double y) {
                return (((x - this.x) * (x - this.x) + (y - this.y) * (y - this.y))
                                        <= this.rayon * this.rayon);
        }
        public double donneRayon() { return this.rayon; }
        public void changeRayon(double r) {
                if (r < 0.0)
                   r = 0.0;
                this.rayon = r;
        }
}
```

Travail à réaliser :

1. Définir une classe `RectangleColoré` qui hérite de `Rectangle` et possède un attribut `couleur`.
2. Définir une classe `Figure` contenant :
   — Deux attributs `x` et `y` qui représentent le centre de la figure ;
   — Un constructeur prenant les coordonnées du centre en paramètres.
   Faire ensuite hériter les classes `Cercle` et `Rectangle` de la classe `Figure`.

3 Hiérarchie d'héritage

Le but de cet exercice est d'utiliser la relation d'héritage de manière ascendante (généralisation). À partir de l'ensemble des classes présentées sur la figure 1 concevoir une hiérarchie de classes permettant de de factoriser les définitions d'attributs.

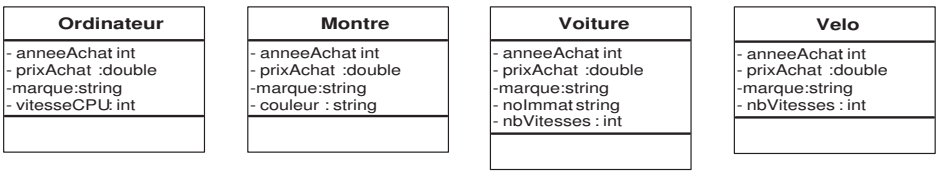

FIGURE 1 – Les classes à restructurer