

## TD04 Correction

Matière : PROGRAMMATION OBJET

Classes : SEM21

```
package fg;

public class FormeGeometrique {
    private static int nbFormeGeometrique;

    public FormeGeometrique() {
        nbFormeGeometrique++;
    }

    public void afficher() {
        System.out.println("Je suis une forme géométrique");
        System.out.println("Mon numéro dans les formes géométrique est : " +
nbFormeGeometrique);
    }

    public static void afficherNbFormeGeometrique() {
        System.out.println("Le nombre de fomrres géométrique est : " +
nbFormeGeometrique);
    }
}

package fg;

public class FormeGeometriqueOuverte extends FormeGeometrique {
    public void afficher() {
        super.afficher();
        System.out.println("Je suis une forme géométrique ouverte");
    }
}

package fg;

public class FormeGeometriqueFermee extends FormeGeometrique {
    public void afficher() {
        super.afficher();
        System.out.println("Je suis une forme géométrique fermée");
    }
}
```

```

package fg;

public class Ellipse extends FormeGeometriqueFermee {
    protected float petitRayon;
    protected float grandRayon;

    public Ellipse(float petitRayon, float grandRayon) {
        super();
        this.petitRayon = petitRayon;
        this.grandRayon = grandRayon;
    }

    public void afficher() {
        super.afficher();
        System.out.println("Je suis un ellipse");
        System.out.println("        Petit Rayon : " + petitRayon);
        System.out.println("        Grand Rayon : " + grandRayon);
    }

    @Override
    public String toString() {
        return "Ellipse [petitRayon=" + petitRayon + ", grandRayon=" +
grandRayon + "]";
    }

    @Override
    public boolean equals(Object arg0) {
        if (arg0 instanceof Ellipse) {
            Ellipse e = (Ellipse) arg0;
            if (petitRayon == e.petitRayon && grandRayon == e.grandRayon)
                return true;
        }
        return false;
    }
}

package fg;

public class Cercle extends Ellipse {

    public Cercle(float rayon) {
        super(rayon, rayon);
        // un cercle est un ellipse dont les deux rayons sont égaux
    }

    public float getSurface() {
        return (float) (Math.PI * Math.pow(petitRayon, 2));
    }

    public float getPerimetre() {
        return (float) (2 * Math.PI * petitRayon);
    }
}

```



```

public void afficher() {
    super.afficher();
    System.out.println("Je suis un cercle");
    System.out.println("        Rayon : " + petitRayon);
    System.out.println("        Surface : " + getSurface());
    System.out.println("        Perimetre : " + getPerimetre());
}

@Override
public String toString() {
    return "Cercle [rayon=" + petitRayon + "]";
}

@Override
public boolean equals(Object arg0) {
    if (arg0 instanceof Cercle) {
        Cercle c = (Cercle) arg0;
        if (petitRayon == c.petitRayon)
            return true;
    }
    return false;
}
}
package fg;

public class Polygone extends FormeGeometriqueFermee {
    protected int nbCote;

    public Polygone(int nbCote) {
        super();
        this.nbCote = nbCote;
    }

    public void afficher() {
        super.afficher();
        System.out.println("Je suis un polygone");
        System.out.println("        Nb Cote : " + nbCote);
    }

    @Override
    public String toString() {
        return "Polygone [nbCote=" + nbCote + "]";
    }
}
package fg;

public class Rectangle extends Polygone {
    protected float longueur;
    protected float largeur;

    public Rectangle(float longueur, float largeur) {
        super(4); // un rectangle est un polygone qui a 4 cotés
        this.longueur = longueur;
        this.largeur = largeur;
    }
}

```



```

public float getSurface() {
    return longueur * largeur;
}

public float getPerimetre() {
    return 2 * (longueur + largeur);
}

public void afficher() {
    super.afficher();
    System.out.println("Je suis un Rectangle");
    System.out.println("    Surface :" + getSurface());
    System.out.println("    Perimetre :" + getPerimetre());
}

@Override
public String toString() {
    return "Rectangle [longueur=" + longueur + ", largeur=" + largeur +
"]";
}

@Override
public boolean equals(Object arg0) {
    if (arg0 instanceof Rectangle) {
        Rectangle r = (Rectangle) arg0;
        if (longueur == r.longueur && largeur == r.largeur)
            return true;
    }
    return false;
}
}

package fg;

public class Carre extends Rectangle {

    public Carre(float cote) {
        super(cote, cote);
        // un carre est un rectangle qui a longueur = largeur
    }

    public float getSurface() {
        return longueur * longueur;
    }

    public float getPerimetre() {
        return 4 * longueur;
        // optimisation de la formule de calcul
    }

    public void afficher() {
        super.afficher();
        System.out.println("Je suis un Carre");
        System.out.println("    Surface :" + getSurface());
        System.out.println("    Perimetre :" + getPerimetre());
    }
}

```



```

@Override
public String toString() {
    return "Carre [côté=" + longueur + "];"
}

@Override
public boolean equals(Object arg0) {
    if (arg0 instanceof Carre) {
        Carre c = (Carre) arg0;
        if (longueur == c.longueur)
            return true;
    }
    return false;
}
}

package fg;

public class TestFormeGeometrique {
    public static void main(String[] args) {

        FormeGeometrique fg;
        fg = new FormeGeometrique();
        fg.afficher();
        System.out.println(fg.toString());
        System.out.println(fg);
        FormeGeometrique.afficherNbFormeGeometrique();

        FormeGeometriqueFermee fgf;
        fgf = new FormeGeometriqueFermee();
        fgf.afficher();
        System.out.println(fgf.toString());
        System.out.println(fgf);
        FormeGeometriqueFermee.afficherNbFormeGeometrique();

        FormeGeometriqueOuverte fgo;
        fgo = new FormeGeometriqueOuverte();
        fgo.afficher();
        System.out.println(fgo.toString());
        System.out.println(fgo);
        FormeGeometriqueOuverte.afficherNbFormeGeometrique();
    }
}

```



```

Ellipse e1;
e1 = new Ellipse(10, 15);
e1.afficher();
System.out.println(e1.toString());
System.out.println(e1);
Ellipse e2;
e2 = new Ellipse(10, 15);
if (e1.equals(e2))
    System.out.println(e1.toString() + "=" + e2.toString());
else
    System.out.println(e1.toString() + "!=" + e2.toString());
Ellipse e3;
e3 = new Ellipse(10, 14);
if (e1.equals(e3))
    System.out.println(e1.toString() + "=" + e3.toString());
else
    System.out.println(e1.toString() + "!=" + e3.toString());

Cercle c1;
c1 = new Cercle(10);
c1.afficher();
Cercle c2;
c2 = new Cercle(10);
if (c1.equals(c2))
    System.out.println(c1.toString() + "=" + c2.toString());
else
    System.out.println(c1.toString() + "!=" + c2.toString());
Cercle c3;
c3 = new Cercle(15);
if (c1.equals(c3))
    System.out.println(c1.toString() + "=" + c3.toString());
else
    System.out.println(c1.toString() + "!=" + c3.toString());

Polygone p;
p = new Polygone(5);
p.afficher();

Rectangle r;
r = new Rectangle(5, 4);
r.afficher();

Carre cr;
cr = new Carre(5);
cr.afficher();
}
}

```

