



## TP06 Correction

Matière : ATELIER DE PROGRAMMATION OBJET

Classes : SEM2

```
package solide;

public class Solide {
    protected float masse;
    protected String matiere;
    private static int nbSolide = 0;

    public Solide(float masse, String matiere) {
        this.masse = masse;
        this.matiere = matiere;
        nbSolide++;
    }

    public float getMasse() {
        return masse;
    }

    public String getMatiere() {
        return matiere;
    }

    public void afficher() {
        System.out.println("Je suis un Solide");
    }

    @Override
    public String toString() {
        return "Solide [" + masse + "g, " + matiere + "]";
    }

    public static int getNbSolide() {
        return nbSolide;
    }
}
```



```

package solide;

public class Cylindre extends Solide {
    protected float rayon;
    protected float hauteur;
    private static int nbCylindre = 0;

    public Cylindre(float masse, String matiere, float rayon, float hauteur) {
        super(masse, matiere);
        this.rayon = rayon;
        this.hauteur = hauteur;
        nbCylindre++;
    }

    public float getSurface() {
        return (float) (2 * Math.PI * rayon * (hauteur + rayon));
    }

    public float getVolume() {
        return (float) (Math.PI * Math.pow(rayon, 2) * hauteur);
    }

    @Override
    public void afficher() {
        super.afficher();
        System.out.println("Je suis un Cylindre");
    }

    public float getMasseVolumique() {
        return masse / getVolume();
    }

    @Override
    public String toString() {
        return "Cylindre [" + masse + " g, " + matiere + ", " + getVolume() + "cm3, "
+ getMasseVolumique() + "g/cm3]";
    }

    @Override
    public boolean equals(Object arg0) {
        if (arg0 instanceof Cylindre) {
            Cylindre c = (Cylindre) arg0;
            if (matiere.equals(c.matiere) && rayon == c.rayon && hauteur == c.hauteur)
                return true;
        }
        return false;
    }

    public static int getNbCylindre() {
        return nbCylindre;
    }
}

```



```

package solide;

public class Sphere extends Solide {
    protected float rayon;
    private static int nbSphere = 0;

    public Sphere(float masse, String matiere, float rayon) {
        super(masse, matiere);
        this.rayon = rayon;
        nbSphere++;
    }

    public float getSurface() {
        return (float) (4 * Math.PI * Math.pow(rayon, 2));
    }

    public float getVolume() {
        return (float) (Math.PI * Math.pow(rayon, 3) * 4 / 3);
    }

    @Override
    public void afficher() {
        super.afficher();
        System.out.println("Je suis une Sphère");
    }

    public float getMasseVolumique() {
        return masse / getVolume();
    }

    @Override
    public String toString() {
        return "Sphere [" + masse + " g, " + matiere + ", " + getVolume() + "cm3, " +
getMasseVolumique() + "g/cm3]";
    }

    @Override
    public boolean equals(Object arg0) {
        if (arg0 instanceof Sphere) {
            Sphere c = (Sphere) arg0;
            if (matiere.equals(c.matiere) && rayon == c.rayon)
                return true;
        }
        return false;
    }

    public static int getNbSphere() {
        return nbSphere;
    }
}

```



```

package solide;

public class Parallelepipede extends Solide {
    protected float longueur;
    protected float largeur;
    protected float hauteur;
    private static int nbParallelepipede = 0;

    public Parallelepipede(float masse, String matiere, float longueur, float
largeur, float hauteur) {
        super(masse, matiere);
        this.longueur = longueur;
        this.largeur = largeur;
        this.hauteur = hauteur;
        nbParallelepipede++;
    }

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

    public float getVolume() {
        return longueur * largeur * hauteur;
    }

    @Override
    public void afficher() {
        super.afficher();
        System.out.println("Je suis un Parallélépipedè");
    }

    public float getMasseVolumique() {
        return masse / getVolume();
    }

    @Override
    public String toString() {
        return "Parallélépipedè [" + masse + " g, " + matiere + ", " + getVolume() +
"cm3, " + getMasseVolumique()
            + "g/cm3]";
    }

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

    public static int getNbParallelepipede() {
        return nbParallelepipede;
    }
}

```



```

package solide;

public class Cube extends Parallelepipede {
    private static int nbCube = 0;

    public Cube(float masse, String matiere, float cote) {
        super(masse, matiere, cote, cote, cote);
        nbCube++;
    }

    @Override
    public void afficher() {
        super.afficher();
        System.out.println("Je suis un Cube");
    }

    @Override
    public float getSurface() {
        return (float) (6 * Math.pow(longueur, 2));
    }

    @Override
    public float getVolume() {
        return (float) Math.pow(longueur, 3);
    }

    @Override
    public String toString() {
        return "Cube [" + masse + " g, " + matiere + ", " + getVolume() + "cm3, " +
getMasseVolumique() + "g/cm3]";
    }

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

    public static int getNbCube() {
        return nbCube;
    }
}

package solide;
public class TestSolide {
    public static void main(String[] args) {
        // Instanciation des classes et affichage
        Solide s = new Solide(12, "Fer");
        s.afficher();
        Cylindre c = new Cylindre(2, "Bois", 10, 20);
        c.afficher();
        Sphere sp = new Sphere(7, "Plomb", 10);
        sp.afficher();
        Parallelepipede p = new Parallelepipede(18, "Fer", 10, 7, 5);
        p.afficher();
        Cube cb = new Cube(6, "Bois", 3);
        cb.afficher();
    }
}

```



```

// toString()
System.out.println(s.toString());
// ou
System.out.println(s);
System.out.println(c.toString());
System.out.println(sp.toString());
System.out.println(p.toString());
System.out.println(cb.toString());
// instanceof
if (s instanceof Object)
    System.out.println("s est un Object");
if (s instanceof Solide)
    System.out.println("s est un Solide");
if (s instanceof Parallelepipede)
    System.out.println("s est un Parallélépipède");
if (s instanceof Cube)
    System.out.println("s est un Cube");
if (p instanceof Object)
    System.out.println("p est un Object");
if (p instanceof Solide)
    System.out.println("p est un Solide");
if (p instanceof Parallelepipede)
    System.out.println("p est un Parallélépipède");
if (p instanceof Cube)
    System.out.println("p est un Cube");
if (cb instanceof Object)
    System.out.println("cb est un Object");
if (cb instanceof Solide)
    System.out.println("cb est un Solide");
if (cb instanceof Parallelepipede)
    System.out.println("cb est un Parallélépipède");
if (cb instanceof Cube)
    System.out.println("cb est un Cube");
// equals()
if (cb.equals(s))
    System.out.println("cb=s");
if (cb.equals(c))
    System.out.println("cb=c");
if (cb.equals(sp))
    System.out.println("cb=sp");
if (cb.equals(p))
    System.out.println("cb=p");
Cube cb1 = new Cube(4, "Fer", 3);
if (cb.equals(cb1))
    System.out.println("cb=cb1");
Cube cb2 = new Cube(4, "Bois", 2);
if (cb.equals(cb2))
    System.out.println("cb=cb2");
Cube cb3 = new Cube(6, "Bois", 3);
if (cb.equals(cb3))
    System.out.println("cb=cb3");
// méthodes statiques
System.out.println("Nb Solide: " + Solide.getNbSolide());
System.out.println("Nb Cylindre: " + Cylindre.getNbCylindre());
System.out.println("Nb Sphere: " + Sphere.getNbSphere());
System.out.println("Nb Parallélépipède: " +
    Parallelepipede.getNbParallelepipede());
System.out.println("Nb Cube: " + Cube.getNbCube());
}
}

```

