

# MOOC Intro POO Java

## Tutoriels semaine 2

Les tutoriels sont des exercices qui reprennent des exemples du cours et dont le corrigé est donné progressivement au fur et à mesure de la donnée de l'exercice lui-même.

Ils sont conseillés comme un premier exercice sur un sujet que l'étudiant ne pense pas encore assez maîtriser pour aborder par lui-même un exercice «classique».

---

### La classe Rectangle pas à pas

Le but de cet exercice est d'illustrer les notions d'objet et de classe au moyen d'un exemple simple.

Il s'agit ici de définir une classe `Rectangle` représentant une abstraction (informatique) de ce qu'est un rectangle : une largeur, une longueur et sa surface.

Commencez par ouvrir un fichier `ManipRectangle.java` et définissez-y la classe :

```
class Rectangle {  
}
```

Bien ! Prévoyons tout de suite d'utiliser notre toute première classe. Pour cela il faut déclarer une variable de type `Rectangle`. Cela se fait comme pour n'importe quelle autre variable utilisée jusque maintenant, la classe `Rectangle` étant simplement un nouveau type. La méthode `main` est mise ici dans la classe `ManipRectangle` qui constitue le programme principal :

```
class ManipRectangle {  
  
    public static void main(String[] args) {  
        Rectangle rect;  
    }  
}
```

Bon, jusque là pas grand chose de bien intéressant. Implémentons maintenant l'abstraction "rectangle" en commençant par les deux attributs largeur et longueur.

Cela se fait comme suit :

```
class Rectangle {  
  
    double largeur;  
    double longueur;  
}
```

Dotons notre classe d'une méthode constructeur construisant nos objets `Rectangle` et initialisant leurs attributs. Notez que le constructeur a le même nom que la classe, et que dans le programme principal la déclaration-initialisation d'un rectangle se fait en utilisant le constructeur.

```
class Rectangle {  
  
    double largeur;  
    double longueur;  
  
    //Le constructeur  
    public Rectangle(double uneLargeur, double uneLongueur) {  
        // largeur et longueur sont les attributs  
        // de la classe  
        largeur = uneLargeur;  
        longueur = uneLongueur;  
    }  
    // largeur et longueur sont les attributs  
    // de la classe  
}
```

Pour le calcul de la surface, nous implémentons une méthode, c'est-à-dire un traitement propre à la classe.

```
class Rectangle {  
  
    double largeur;  

```

```

double longueur;

//Le constructeur
public Rectangle(double uneLargeur, double uneLongueur) {
    largeur = uneLargeur;
    longueur = uneLongueur;
}
// largeur et longueur sont les attributs
// de la classe

// La méthode pour le calcul de la surface
public double surface () {
    return largeur * longueur;
}
}

```

On peut maintenant déjà tester notre programme :

```

class Rectangle {

    double largeur;
    double longueur;

    public Rectangle(double uneLargeur, double uneLongueur) {
        largeur = uneLargeur;
        longueur = uneLongueur;
    }

    public double surface() {
        return largeur * longueur;
    }
}

class ManipRectangle {

    public static void main(String[] args) {
        Rectangle rect = new Rectangle(1.5,12.8);
        System.out.println("Surface : " + rect.surface());
    }
}

```

Compilez et exécutez ce programme.

Supposons maintenant que nous souhaitions modifier les valeurs des attributs dans le programme principal:

```

...
class ManipRectangle {

    public static void main(String[] args) {
        Rectangle rect = new Rectangle(1.5, 12.8);
        System.out.println("Surface : " + rect.surface());

        rect.largeur = 3.2;
        rect.longueur = 6.9;
    }
}

```

Essayons maintenant d'affiner notre "style objet" en rendant privés les attributs et ne laissant publiques que les méthodes :

```

class Rectangle {

    private double largeur;
    private double longueur;

    public Rectangle(double uneLargeur, double uneLongueur){
        largeur = uneLargeur;
        longueur = uneLongueur;
    }

    public double surface() {
        return largeur * longueur;
    }
}

class ManipRectangle {

    public static void main(String[] args) {

```

```

        Rectangle rect = new Rectangle(1.5,12.8);
        System.out.println("Surface : " + rect.surface());

        rect.largeur = 3.2;
        rect.longueur = 6.9;
    }
}

```

Essayez de compiler ce programme. Que se passe-t-il?

Le message est assez clair : tout les attributs sont privés. Cela veut dire qu'on ne peut pas les utiliser hors de l'objet (ils ne font pas partie de l'interface de l'objet).

En particulier, on n'a pas le droit de faire `rect.largeur` ! Ce n'est pas défini.

Pour rendre accessible depuis l'extérieur un attribut, il faut mettre en place les méthodes "get" et "set" correspondantes (qui devront évidemment être publiques !).

Cela se fait simplement, comme pour la méthode du calcul de surface. Dans la méthode `main` l'accès aux attributs se fera désormais en passant par ces méthodes

L'intérêt est notamment que l'on peut faire faire à ces méthodes des contrôles d'intégrité sur les données (de sorte à ne pas mettre n'importe quoi comme valeur aux attributs). L'accès direct aux attributs, sans passer par des méthodes, n'offre pas cette sécurité.

```

class Rectangle {

    private double largeur;
    private double longueur;

    public Rectangle(double uneLargeur, double uneLongueur) {
        largeur = uneLargeur;
        longueur = uneLongueur;
    }

    public double surface() {
        return largeur * longueur;
    }

    public double getLongueur() {
        return longueur;
    }

    public double getLargeur() {
        return largeur;
    }

    public void setLargeur(double l) {
        if (l < 0) {
            l = -1;
        }
        largeur = l;
    }

    public void setLongueur(double l) {
        if (l < 0) {
            l = -1;
        }
        longueur = l;
    }
}

class ManipRectangle {

    public static void main(String[] args) {
        Rectangle rect = new Rectangle(1.5,12.8);
        System.out.println("Surface : " + rect.surface());
        rect.setLargeur(3.2);
        rect.setLongueur(6.9);
        System.out.println("Surface : " + rect.surface());
    }
}

```

Recompilez. Ça fonctionne cette fois-ci et vous avez codé votre premier programme objet !

---

