

Support de cours

Cours:

## Introduction à la programmation orientée objet (en Java)

Vidéo:

### W11-04-resume-JAVA-pt2

Concepts (extraits des sous-titres générés automatiquement) :

**Valeurs particulières. Moyen d'interactions clavier. Code du programmeur. Concepteur de la classe rectangle. Niveau type de données. Cadre de ce cours. Premières choses. Moyen de la méthode sethauteur. Déclaration de variables de types. Terme de l'exécution de cette instruction. Objet de type rectangle. Stade du cours. Hauteur de l'objet rect. Lignes de codes. Côté du programmeur.**



[vers la recherche de séquences vidéo](#)

(dans Introduction à la programmation orientée objet (en Java).)



[vers la vidéo](#)

Center for Digital Education. Plus de matériel de soutien pédagogique ici :

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>

# Encapsulation et abstraction : résumé

(Partie 2)

Introduction à la programmation orientée objet (en Java)

Jamila Sam, Jean-Cédric Chappelier et Vincent Lepetit

...

notes

résumé

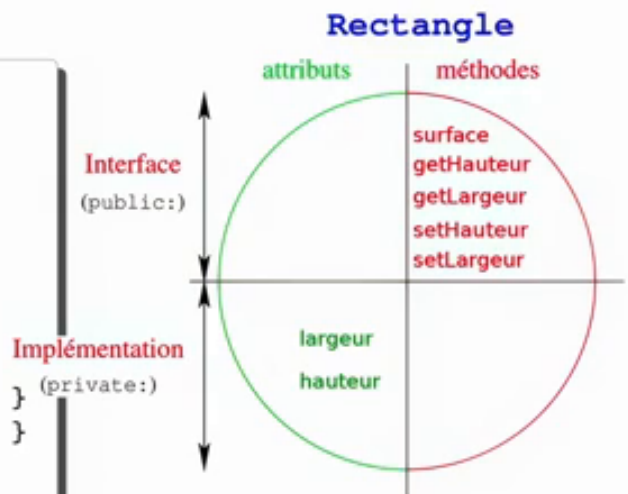
0m 0s



## Un exemple complet de classe (1/2)

```
// définition de la classe
class Rectangle {
    // déclaration des attributs
    private double hauteur;
    private double largeur;

    // définition des méthodes
    public double surface()
    { return hauteur * largeur; }
    public double getHauteur() { return hauteur; }
    public double getLargeur() { return largeur; }
    public void setHauteur(double h)
    { hauteur = h; }
    public void setLargeur(double l)
    { largeur = l; }
}
```



Ce que vous avez sous les yeux, ici, comme code, est le code du programmeur/concepteur de la classe rectangle, celui qui conçoit ce niveau type de données.

notes

résumé

0m 1s



## Un exemple complet de classe (2/2)

```
//utilisation de la classe
class Geometrie
{
    private final static Scanner CLAVIER = new Scanner(System.in);

    public static void main(String[] args)
    {
        → Rectangle rect = new Rectangle();    int i;
        double lu;
        System.out.print("Quelle hauteur? ");
        lu = CLAVIER.nextDouble();
        rect.setHauteur(lu);
        System.out.print("Quelle largeur? ");
        lu = CLAVIER.nextDouble();
        rect.setLargeur(lu);

        System.out.println("surface = " + rect.surface());
    }
}
```

Voyons maintenant ce qui se passe du côté du programmeur/utilisateur qui n'est pas forcément la même personne. Cela peut être le même programmeur mais pas forcément. Le code que vous avez ici sous les yeux, est le code l'implémentation d'une classe Geometrie qui, elle est cliente du type rectangle de la classe rectangle qui était défini précédemment. Ce qui veut dire qu'elle peut désormais l'utiliser comme n'importe quel autre type je suis en train de déclarer une variable rect de type Rectangle. La déclaration a exactement la même forme que celle que j'utilisais pour la déclaration de variables de types prédéfinis. Nous avons vu également dans le cadre de ce cours

notes

résumé

0m 13s

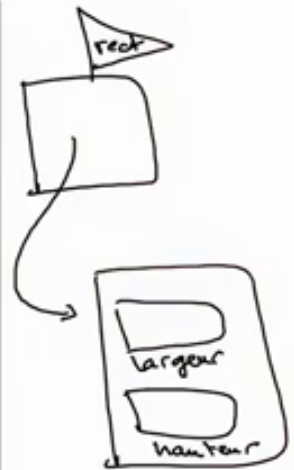


## Un exemple complet de classe (2/2)

```
//utilisation de la classe
class Geometrie
{
    private final static Scanner CLAVIER = new Scanner(System.in);

    public static void main(String[] args)
    {
        → Rectangle rect = new Rectangle();    int i;
        double lu;
        System.out.print("Quelle hauteur? ");
        lu = CLAVIER.nextDouble();
        rect.setHauteur(lu);
        System.out.print("Quelle largeur? ");
        lu = CLAVIER.nextDouble();
        rect.setLargeur(lu);

        System.out.println("surface = " + rect.surface());
    }
}
```



que pour initialiser un objet, à ce stade du cours, il fallait passer par ce genre de tournure en java. Nous allons apprendre à expliciter ce genre de tournure dès le cours suivant. Pour l'instant, c'est un passage obligé. Au terme de l'exécution de cette instruction, nous aboutissons à la situation suivante en mémoire. Nous avons un objet qui s'appelle rect, une variable qui s'appelle rect qui contient la référence, l'adresse, un objet de type rectangle qui a deux champs. Un champ pour la largeur et un champ pour la hauteur, et évidemment, l'une des premières choses que l'on souhaiterait faire ici,

notes

résumé

0m 49s

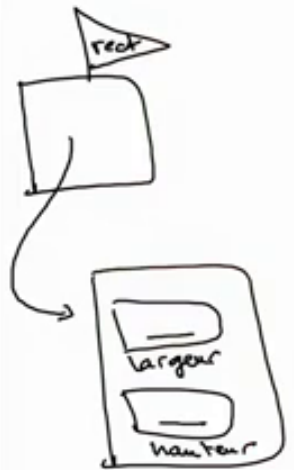


## Un exemple complet de classe (2/2)

```
//utilisation de la classe
class Geometrie
{
    private final static Scanner CLAVIER = new Scanner(System.in);

    public static void main(String[] args)
    {
        → Rectangle rect = new Rectangle();    int i;
        double lu;
        System.out.print("Quelle hauteur? ");
        lu = CLAVIER.nextDouble();
        rect.setHauteur(lu);
        System.out.print("Quelle largeur? ");
        lu = CLAVIER.nextDouble();
        rect.setLargeur(lu);

        System.out.println("surface = " + rect.surface());
    }
}
```



c'est de mettre dans les champs largeur et hauteur, des valeurs particulières afin que ces 2 champs contiennent autre chose que 0. Ceci peut être fait au moyen d'interactions clavier, on peut donc demander à l'utilisateur d'introduire des valeurs pour la largeur et la hauteur. C'est ce qui est fait ici. Une fois que l'utilisateur a introduit une valeur, on peut mettre cette valeur dans la hauteur de l'objet rect

notes

résumé

1m 21s

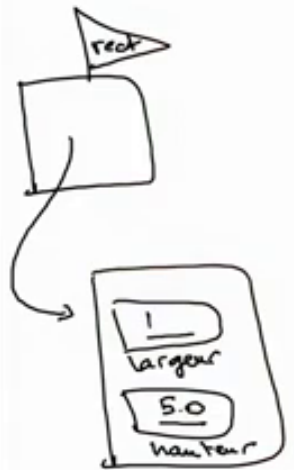


## Un exemple complet de classe (2/2)

```
//utilisation de la classe
class Geometrie
{
    private final static Scanner CLAVIER = new Scanner(System.in);

    public static void main(String[] args)
    {
        → Rectangle rect = new Rectangle();    int i;
        double lu;
        System.out.print("Quelle hauteur? ");
        lu = CLAVIER.nextDouble();
        rect.setHauteur(lu);
        System.out.print("Quelle largeur? ");
        lu = CLAVIER.nextDouble();
        rect.setLargeur(lu);

        System.out.println("surface = " + rect.surface());
    }
}
```



que nous venons de créer, au moyen de la méthode setHauteur. Nous utilisons ici la notation pointée qui veut dire que nous appliquons la méthode setHauteur à l'instance à l'objet rect pour mettre la valeur lue à l'intérieur. Imaginez que l'utilisateur ayant introduit la valeur 5.0, nous nous trouvons dans la situation suivante. Pareil pour ces lignes de codes qui ont pour charge d'initialiser la largeur du rectangle. Nous disposons d'un objet qui est proprement initialisé. Nous pouvons désormais utiliser d'autres fonctionnalités sur ce rectangle comme par exemple, le calcul de la surface. Pareil, on utilise la notation pointée. Ici, je suis en train de dire je calcule la surface de cet objet rectangle déclaré ici et initialisé ici. rectangle déclaré ici et initialisé ici.

notes

résumé

1m 49s

