

Support de cours

Cours:

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

Vidéo:

### W12-02-constrdefault-JAVA-pt2

Concepts (extraits des sous-titres générés automatiquement) :

**Constructeur explicite. Valeurs d'initialisation. Constructeur. Défaut. Liste de paramètres vide. Seul constructeur disponible. Questions suivantes. Seconde tournure. Classe rectangle. Défaut de ce fait. Constructeurs de la classe. Attributs correspondants. Liste de paramètres. Types élémentaires. Valeurs de base.**



[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/>

# Constructeurs par défaut en Java

## (Partie 2)

Introduction à la programmation orientée objet (en Java)

Jamila Sam, Jean-Cédric Chappelier et Vincent Lepetit

...

notes

résumé

0m 0s



Si aucun constructeur n'est spécifié, le compilateur *génère automatiquement* une **version minimale du constructeur par défaut**

qui :

initialise les attributs avec les valeurs par défaut : 0, 0.0, et `false` pour les types de base et `null` pour les objets.

Dès qu'**au moins un constructeur a été spécifié**, ce constructeur par défaut par défaut *n'est plus fourni*.

Si donc on spécifie *un* constructeur sans spécifier de constructeur par défaut, on ne peut plus construire d'objet de cette classe sans les initialiser explicitement (ce qui est voulu !) puisqu'il n'y a plus de constructeur par défaut.

*Leclercq*

Attention, un point très important. Ce constructeur par défaut par défaut n'est plus fourni, disparaît, dès l'instant que l'on définit dans la classe au moins un constructeur explicite, que ce constructeur soit par défaut lui-même ou pas. Ceci signifie concrètement que si dans une classe je spécifie un constructeur qui n'est pas un constructeur par défaut, je n'ai plus aucun moyen de construire l'objet sans fournir de valeurs d'initialisation. Concrètement, je ne peux plus écrire quelque chose comme ceci

notes

résumé

0m 1s





Donc invoquer un constructeur par défaut, cette tournure n'est plus licite Pourquoi ? Parce que tout constructeur par défaut a disparu Ceci correspond bien à quelque chose de voulu. Si nous avons pris le soin de programmer explicitement un constructeur dans la classe nous ne souhaitons pas que Java nous en glisse un par défaut sans que nous l'ayons mentionné explicitement, sans que nous ayons explicitement manifesté le souhait d'utiliser un constructeur par défaut Pour illustrer ceci, nous allons maintenant étudier trois variantes

notes

résumé

0m 37s



A :

```
class Rectangle {  
    private double hauteur;  
    private double largeur;  
    // suite ...  
}
```

d'un même programme qui diffèrent les unes des autres par leur façon de fournir des constructeurs.

notes

résumé

1m 13s



A :

```
class Rectangle {
    private double hauteur;
    private double largeur;
    // suite ...
}
```

B :

```
class Rectangle {
    private double hauteur;
    private double largeur;

    public Rectangle()
    {
        hauteur = 0.0;
        largeur = 0.0;
    }
    // suite ...
}
```

Dans la première variante, la variante A, une classe Rectangle est définie avec deux attributs « hauteur » et « largeur » qui sont des doubles Elle peut y prévoir éventuellement d'autres contenu mais elle ne contient aucun constructeur explicite

notes

résumé

1m 17s



A :

```
class Rectangle {
    private double hauteur;
    private double largeur;
    // suite ...
}
```

B :

```
class Rectangle {
    private double hauteur;
    private double largeur;

    public Rectangle()
    {
        hauteur = 0.0;
        largeur = 0.0;
    }
    // suite ...
}
```

C :

```
class Rectangle {
    private double hauteur;
    private double largeur;

    public Rectangle(double h,
                     double l)
    {
        hauteur = h;
        largeur = l;
    }
    // suite ...
}
```

Dans la variante B, toujours les mêmes attributs mais cette fois-ci un constructeur explicite, qui est un constructeur par défaut car il a une liste de paramètres vide. Ce constructeur va initialiser la hauteur et la largeur aux valeurs 0.0 Dans la variante C enfin, la classe Rectangle dispose d'un

notes

résumé

1m 32s



A :

```
class Rectangle {
    private double hauteur;
    private double largeur;
    // suite ...
}
```

B :

```
class Rectangle {
    private double hauteur;
    private double largeur;

    public Rectangle()
    {
        hauteur = 0.0;
        largeur = 0.0;
    }
    // suite ...
}
```

C :

```
class Rectangle {
    private double hauteur;
    private double largeur;

    public Rectangle(double h,
                     double l)
    {
        hauteur = h;
        largeur = l;
    }
    // suite ...
}
```

constructeur explicite qui cette fois n'est plus un constructeur par défaut puisque sa liste de paramètres n'est plus vide. Donc ici, il s'agit du constructeur qui attend une valeur pour la

notes

résumé

1m 49s





constructeur par défaut	<code>Rectangle r1 = new Rectangle();</code>	<code>Rectangle r2 = new Rectangle(1.0,2.0);</code>
-------------------------	--	---

hauteur et une autre pour la largeur, et qui se charge d'affecter ces valeurs aux attributs correspondants

notes

résumé

2m 0s



	constructeur par défaut	Rectangle r1 = new Rectangle();	Rectangle r2 = new Rectangle(1.0,2.0);
(A)	constructeur par défaut par défaut	0.0 0.0	Illicite !

```
class Rectangle {  
}
```

Pour chacune de ces variantes, nous allons nous poser les questions suivantes:  
Existe-t-il un constructeur par défaut ? Est-il licite d'initialiser un objet selon cette  
tournure ? Est-il licite d'initialiser un objet selon cette seconde tournure ?  
Commençons par la première variante, la variante A, qui pour rappel,

notes

résumé

2m 8s



## Constructeur par défaut : exemples

	constructeur par défaut	Rectangle r1 = new Rectangle();	Rectangle r2 = new Rectangle(1.0,2.0);
(A)	constructeur par défaut par défaut	0.0 0.0	Illicite !

```
class Rectangle {  
}
```

ici on n'indique que les constructeurs de la classe et pas le reste, n'avait aucun constructeur explicite Et bien comme on l'a dit tout à l'heure lorsqu'une classe ne prévoit aucun constructeur explicite, il existe un constructeur par défaut, il s'agit du constructeur par défaut par défaut De ce fait, l'initialisation selon cette tournure est licite On utilise un constructeur sans paramètre Et nous avons vu que le constructeur par défaut par défaut, lorsque les attributs sont de types élémentaires, initialise ces attributs avec des valeurs de base Donc la largeur vaudra 0.0, et la hauteur vaudra 0.0

notes

résumé

2m 25s



## Constructeur par défaut : exemples

	constructeur par défaut	<code>Rectangle r1 = new Rectangle();</code>	<code>Rectangle r2 = new Rectangle(1.0,2.0);</code>
(A)	constructeur par défaut par défaut	<div>0.0 0.0</div>	Illicite !
(B)	constructeur par défaut explicitement déclaré	<div>0.0 0.0</div>	Illicite !

```
class Rectangle {
    public Rectangle()
    {
        hauteur = 0.0;
        largeur = 0.0;
    }
}
```

Par contre, vu que le seul constructeur disponible est le constructeur par défaut par défaut, on ne peut initialiser l'objet que selon cette tournure En utilisant un constructeur sans paramètre Cette tournure qui attend des paramètres est, par conséquent, illicite Pour la variante B, nous avons donc dans la classe Rectangle,

notes

résumé

3m 1s



	constructeur par défaut	Rectangle r1 = new Rectangle();	Rectangle r2 = new Rectangle(1.0,2.0);
(A)	constructeur par défaut par défaut	0.0 0.0	Illicite !
(B)	constructeur par défaut explicitement déclaré	0.0 0.0	Illicite !

```
class Rectangle {
    public Rectangle()
    {
        hauteur = 0.0;
        largeur = 0.0;
    }
}
```

un constructeur explicite, il s'agissait d'un constructeur par défaut A la question de savoir s'il existe un constructeur par défaut, la réponse est oui Il s'agit de ce constructeur explicitement déclaré Comme tout à l'heure, on peut évidemment utiliser ce constructeur par défaut selon cette tournure, un constructeur qui n'attend aucun paramètre

notes

résumé

3m 18s



## Constructeur par défaut : exemples

	constructeur par défaut	<code>Rectangle r1 = new Rectangle();</code>	<code>Rectangle r2 = new Rectangle(1.0,2.0);</code>
(A)	constructeur par défaut par défaut	<input type="text" value="0.0"/> <input type="text" value="0.0"/>	Illicite !
(B)	constructeur par défaut explicitement déclaré	<input type="text" value="0.0"/> <input type="text" value="0.0"/>	Illicite !
(C)	pas de constructeur par défaut	Illicite !	<input type="text" value="1.0"/> <input type="text" value="2.0"/>

```
class Rectangle {
    public Rectangle(double h, double l)
    {
        hauteur = h;
        largeur = l;
    }
}
```

Et ici, ce constructeur aura le même effet que le constructeur par défaut par défaut. Puisqu'il initialise la hauteur et la largeur avec les mêmes valeurs que celles utilisées par le constructeur par défaut par défaut. Selon les mêmes arguments que précédemment, cette tournure demeure illicite, il n'existe en effet dans la classe Rectangle aucun constructeur autre que le constructeur par défaut. Donc aucun constructeur qui attendrait deux valeurs pour pouvoir fonctionner. Enfin, pour la variante C, nous avons donc un constructeur explicite mais qui n'était pas un constructeur par défaut. A la question de savoir si il existe un constructeur par défaut, la réponse est non. Il n'existe plus de constructeur par défaut. Pourquoi ? Nous avons en effet dit tout à l'heure que dès l'instant où on définit un constructeur explicite, qu'il soit par défaut ou non, le constructeur par défaut par défaut disparaît. De ce fait, le seul constructeur disponible dans la classe maintenant est celui-ci. Il n'est plus possible d'initialiser un objet selon cette tournure. Plus de constructeur qui n'attend aucun argument. Ceci est illicite. Par contre la seconde tournure devient licite. Elle va initialiser la largeur et la hauteur au moyen des valeurs passées ici en argument et nous obtiendrons donc un objet rectangle où la hauteur vaudra 1.0 et la largeur vaudra 2.0.

notes

résumé

3m 37s

