

Support de cours

Cours:

Introduction à la programmation orientée objet (en Java)

Vidéo:

W17-02-affichage-JAVA-pt3

Concepts (extraits des sous-titres générés automatiquement) :

Surcharge des constructeurs. Valeur de base nulle. Produit ait. Façons possibles. Existence concrète. Surcharge. Vu de l'existence de ces constructeurs. Valeur de base. Moyen du constructeur. Classe produit. Long de son existence. Valeur nulle. Valeur de base constante. Variable de type produit. Instance de produit.



[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/>

Etude de cas : affichage polymorphique

(Partie 3)

Introduction à la programmation orientée objet (en Java)

Jamila Sam, Jean-Cédric Chappelier et Vincent Lepetit

...

notes

résumé

0m 0s



Supposons enfin que :

- ▶ un produit n'existe pas en tant que tel : c'est une *abstraction*
- ▶ l'on fixe la valeur de base d'un produit au départ et que l'on ne puisse pas en changer
- ▶ un produit ait par défaut une valeur de base nulle

```
abstract class Produit {  
  
    private final double valeur;  
  
    public Produit(double uneValeur) {  
        valeur = uneValeur;  
    }  
  
    public Produit() {  
        valeur = 0.0;  
    }  
}
```

Supposons enfin que l'on souhaite imposer le fait qu'un produit ait, par défaut, une valeur de base nulle, ceci peut être réalisé en ajoutant à la classe Produit un constructeur par défaut qui prendrait le soin d'initialiser la valeur à la valeur nulle. La surcharge, et notamment ici la surcharge des constructeurs est une façon assez typique de faire en Java pour donner des valeurs

notes

résumé

0m 1s



```
abstract class Produit {  
    private final double valeur;  
  
    public Produit(double uneValeur) { valeur = uneValeur; }  
  
    public Produit() { valeur = 0.0; }  
  
    public double prix() { return valeur; }  
  
    public String toString() { return Double.toString(prix()); }  
}
```

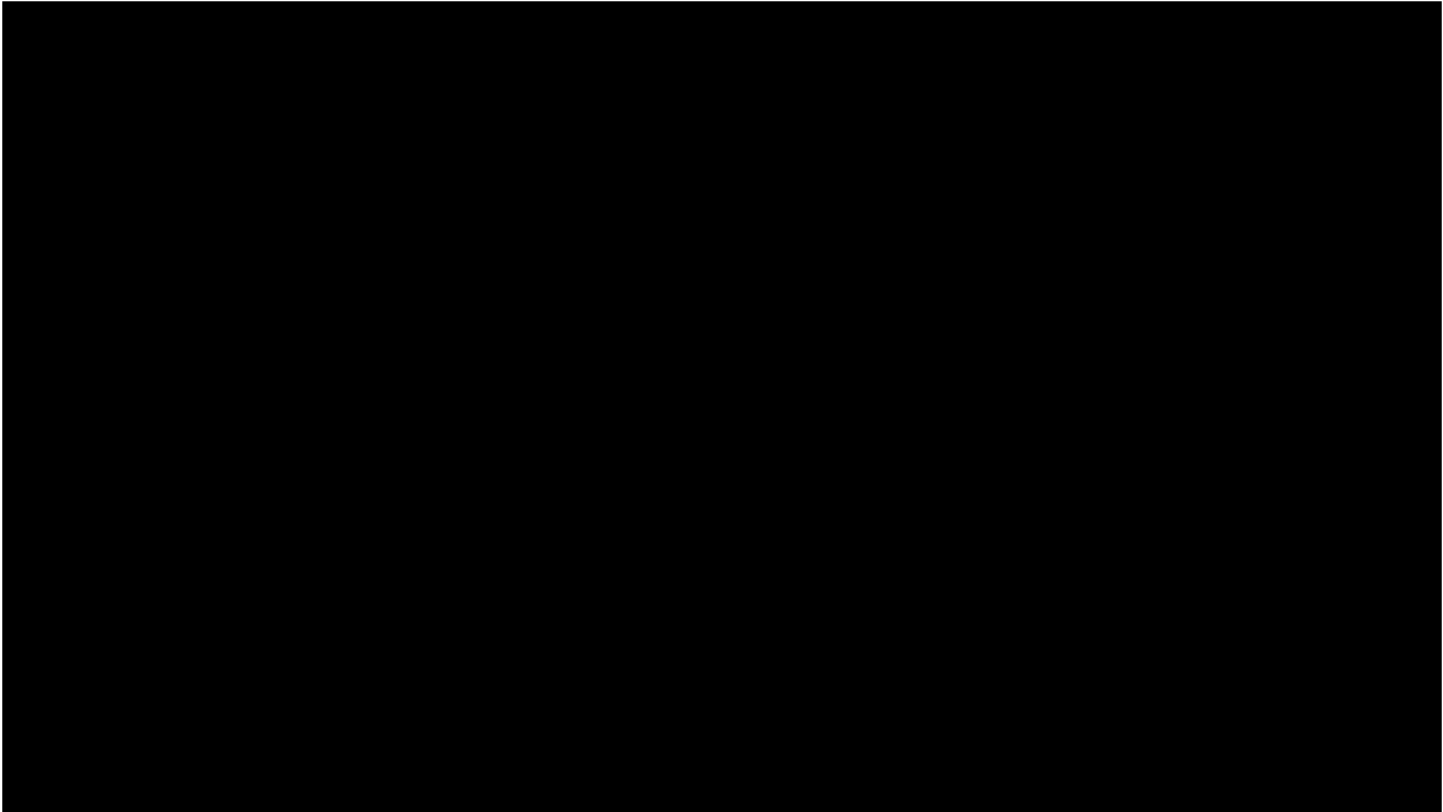
par défaut à certains paramètres. Voici donc à quoi ressemble notre classe Produit au final. Nous l'avons déclarée comme "abstraite" parce que nous ne voulons pas que des produits soient instanciables en tant que tels, ils n'ont pas d'existence concrète en tant que tels : un bracelet sera instanciable, une montre sera instanciable, mais pas un produit en tant que tel. Une instance de Produit est caractérisée par sa valeur de base. Nous avons étiqueté cet attribut comme final, ce qui signifie qu'une fois qu'il sera initialisé par l'un des constructeurs, cet attribut ne pourra plus changer de valeur. Donc un produit a une valeur de base constante tout au long de son existence. Un produit peut donc être initialisé de deux façons possibles : soit en utilisant un constructeur prenant en argument une valeur de base à donner à l'attribut, soit au moyen du constructeur par défaut, auquel cas la valeur de base est mise à zéro.

notes

résumé

0m 25s





Au vu de l'existence de ces constructeurs, est-il possible d'écrire quelque chose comme ceci : donc déclarer une variable de type Produit, et l'initialiser comme ceci, par exemple ? et l'initialiser comme ceci, par exemple ?

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

1m 13s



.....