

Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-06-5-Definitions-pt1

Concepts (extraits des sous-titres générés automatiquement) :

Facettes des méthodes. Ensemble des instructions. Instructions return. Expression particulière. Appel d'une méthode. Spécification du type de retour. Entête d'une méthode. Sens java du terme. Simple instruction return. Expression return. Notion de définition des méthodes. Exemple de la méthode moyenne. Vidéo précédente. Définition de la méthode moyenne. Point de vue de la syntaxe.



[vers la recherche de séquences vidéo](#)
(dans Initiation à la programmation (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/>

Fonctions : définitions

(Partie 1)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

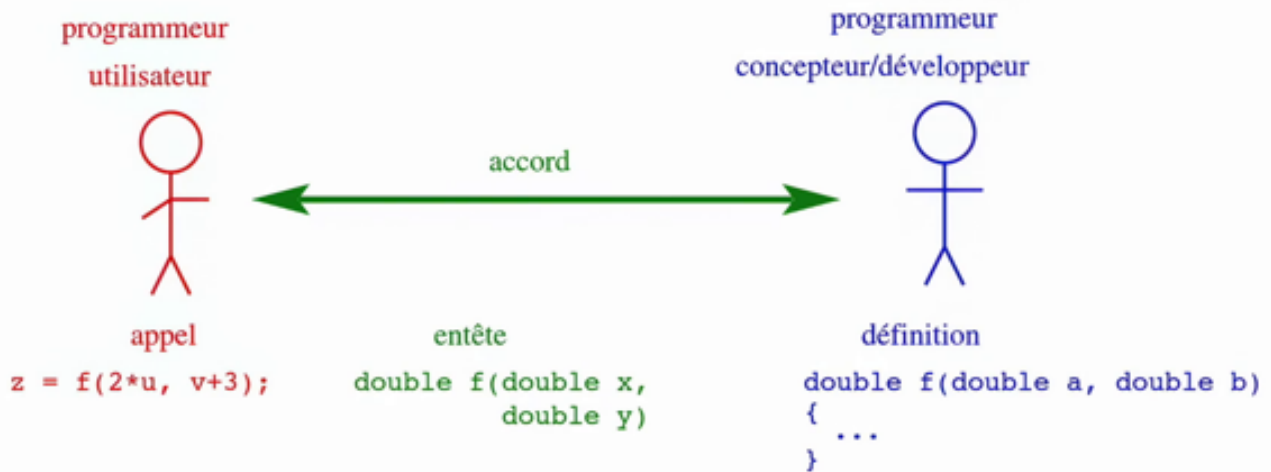
résumé

0m 0s



Les « 3 facettes » d'une méthode

- ▶ Résumé / Contrat (« entête »)
- ▶ **Création / Construction (« définition »)**
- ▶ Utilisation (« appel »)



Dans une vidéo précédente, nous avons vu ce que sont les trois facettes des méthodes.

notes

résumé

0m 1s



Exemple complet

```
class Exemple
{
    private static Scanner clavier= new Scanner(System.in);
    public static void main(String[] args)
    {
        double note1 = 0.0;
        double note2 = 0.0;
        System.out.println("Entrez vos deux notes : ");
        note1 = clavier.nextDouble();
        note2 = clavier.nextDouble();
        System.out.println("Votre moyenne est : "
            + moyenne(note1, note2));
    }

    static double moyenne(double x, double y)
    {
        return (x + y) / 2.0;
    }
}
```



Nous avons vu ce qu'est l'appel d'une méthode, nous avons aussi vu ce qu'est l'entête d'une méthode. Dans cette vidéo-ci, nous allons plus particulièrement nous concentrer sur la notion de définition des méthodes.

notes

résumé

0m 10s



définition d'une méthode :

- spécification du **corps** de la méthode

Syntaxe :

```
type nom ( liste de paramètres )
{
    instructions du corps de la méthode;
    return expression;
}
```

Exemple :

```
static double moyenne (double x, double y)
{
    return (x + y) / 2.0;
}
```

Si je reprends l'exemple courant du calcul de la moyenne, la définition de la méthode moyenne, ce sont toutes ces lignes ici qui commencent par l'entête, ainsi que toutes ces lignes ici qui suivent l'entête entre accolades. La définition d'une méthode sert donc, comme son nom l'indique, à définir ce que doit faire la méthode, c'est-à-dire à spécifier le corps de la méthode, c'est-à-dire l'ensemble des instructions qui ont fait

notes

résumé

0m 25s



Définition des méthodes

définition d'une méthode :

- spécification du **corps** de la méthode

Syntaxe : \rightarrow type, nom (liste de paramètres)
 {
 instructions du corps de la méthode;
 return expression;
 }

Exemple :

```
static double moyenne (double x, double y)
{
    return (x + y) / 2.0;
}
```

justement que l'on a décidé d'isoler cette méthode, de créer cette méthode. Du point de vue de la syntaxe on va donc commencer par donner l'entête, je vous rappelle c'est la spécification du type de retour suivi du nom de la méthode, suivi entre parenthèses rondes de la liste des paramètres, mais dans la définition cette entête va être suivie donc ici d'un bloc qui va contenir les instructions qui font que la méthode va pouvoir fonctionner, qui décrivent le déroulement de la méthode. Parmi ces instructions, nous aurons une ou plusieurs instructions return qui vont mettre fin à l'exécution du corps de la méthode.

notes

résumé

0m 49s



Le corps de la méthode est donc un **bloc** dans lequel on peut utiliser les paramètres de la méthode.

Si je reprends ici l'exemple de la méthode moyenne, on va retrouver l'entête de la méthode, comme on l'a déjà vu dans les exemples précédents, suivi par ici l'ensemble des instructions qui permettent à la moyenne d'effectivement effectuer ce qu'elle doit faire, ici réduit à une simple instruction return qui va calculer la somme de ces deux paramètres x et y et diviser par deux.

notes

résumé

1m 25s



Corps de méthode

Le corps de la méthode est donc un **bloc** dans lequel on peut utiliser les paramètres de la méthode.

La valeur retournée par la méthode est indiquée par l'instruction :

`return expression;`

où l'*expression* a le même *type* que celui retourné par la méthode.

```
static double moyenne (double x, double y)
{
    return (x + y) / 2.0;
}
```

L'instruction `return` fait deux choses:

- ▶ elle précise la valeur qui sera fournie par la méthode en résultat
- ▶ elle met fin à l'exécution des instructions de la méthode.

L'expression après `return` est parfois réduite à une seule variable ou même à une valeur littérale, mais ce n'est pas une nécessité.

return x;

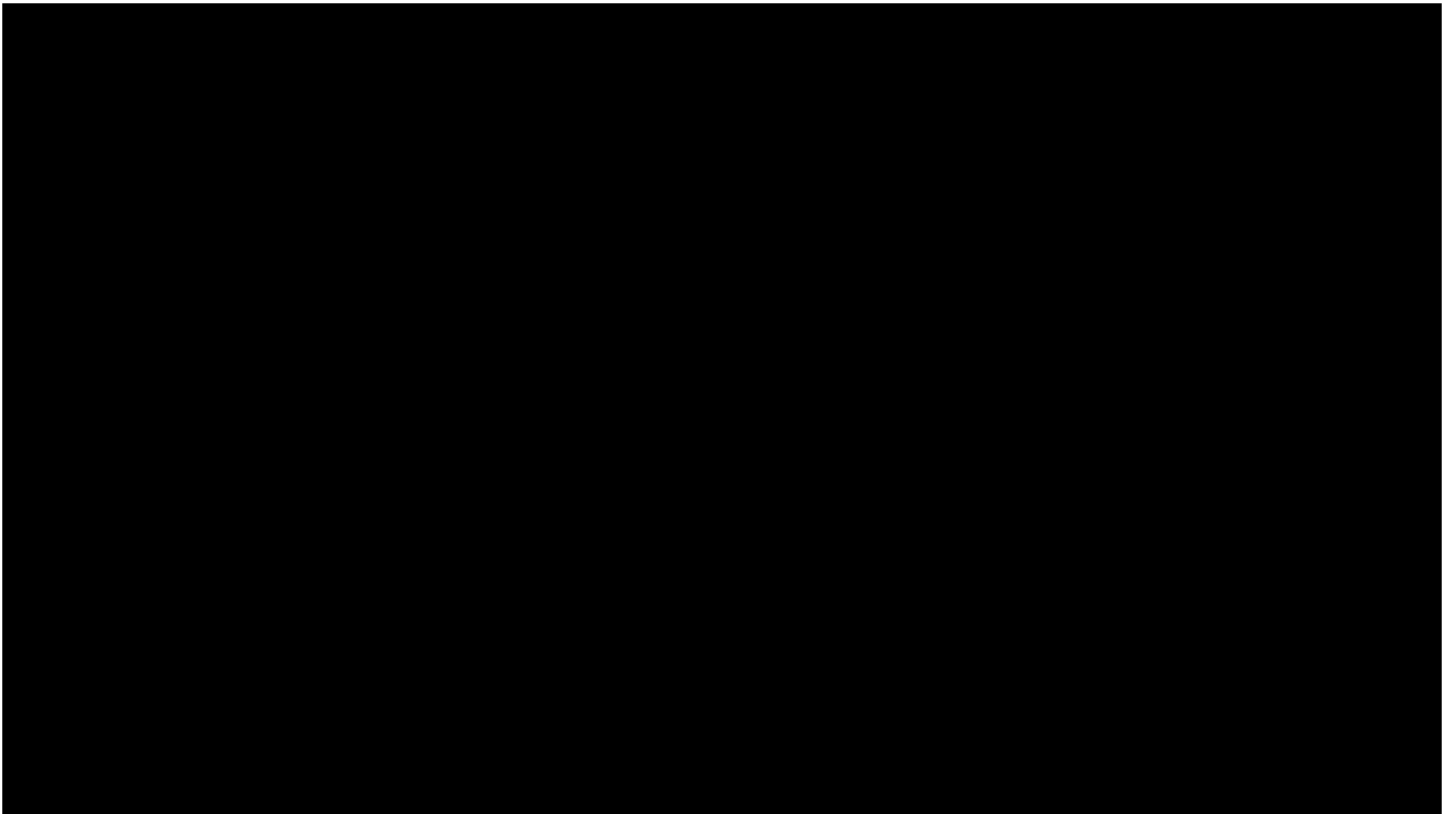
Le corps de la méthode est donc simplement un bloc au sens Java du terme, c'est-à-dire un ensemble d'instructions comprises entre accolades qui permettent de définir ce que doit faire la méthode et dans ce bloc ici on peut utiliser comme variables les paramètres reçus par la méthode. La valeur retournée par la méthode est indiquée par une expression particulière, qui est ce qu'on appelle une expression `return`, qui commence par le mot-clé `return` et ensuite est suivie d'une expression Java qui est évaluée et qui va définir donc la valeur de retour de la méthode. Elle doit donc être naturellement de même type que le type de retour de la méthode. Si je reprends l'exemple du calcul de la moyenne, nous avons ici comme type de retour pour la moyenne un `double` et donc nous verrons derrière l'instruction `return` ici une expression Java de type `double`, qui sera la valeur retournée par l'appel de la méthode `moyenne`. L'instruction `return` fait donc en fait deux choses : elle commence par préciser la valeur de retour de la méthode et ensuite la première instruction `return` rencontrée met fin à l'exécution de la méthode dans laquelle elle se déroule. L'expression `return` est parfois réduite à une seule variable, comme par exemple

notes

résumé

1m 49s





on pourrait rencontrer des expressions du type `return x` ou même simplement une valeur, comme par exemple `return 3`, ceci n'est qu'un cas très particulier d'expressions simples, une seule variable ou une valeur particulière, mais ça n'est pas du tout le cas général, on l'a vu dans l'exemple précédent, le cas général c'est une expression complète au sens Java du terme qui se trouve derrière l'instruction `return`. l'instruction `return`.

notes

résumé

3m 13s