

Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-06-2-Appel-pt1

Concepts (extraits des sous-titres générés automatiquement) :

Toute première étape. Seconde étape. Forme d'expressions. Paramètre x. Résultat de la première expression. Moment de la seconde étape. Paramètres de la méthode. Corps de la fonction. Exemple du calcul des moyennes. Moment de l'appel. Notion de méthode. Troisième étape. Petit programme principal. Quatrième étape. Séquence précédente.



[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 : appel

(Partie 1)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s





Dans une séquence précédente, nous avons introduit la notion de fonction, méthode

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

0m 1s



.....

.....

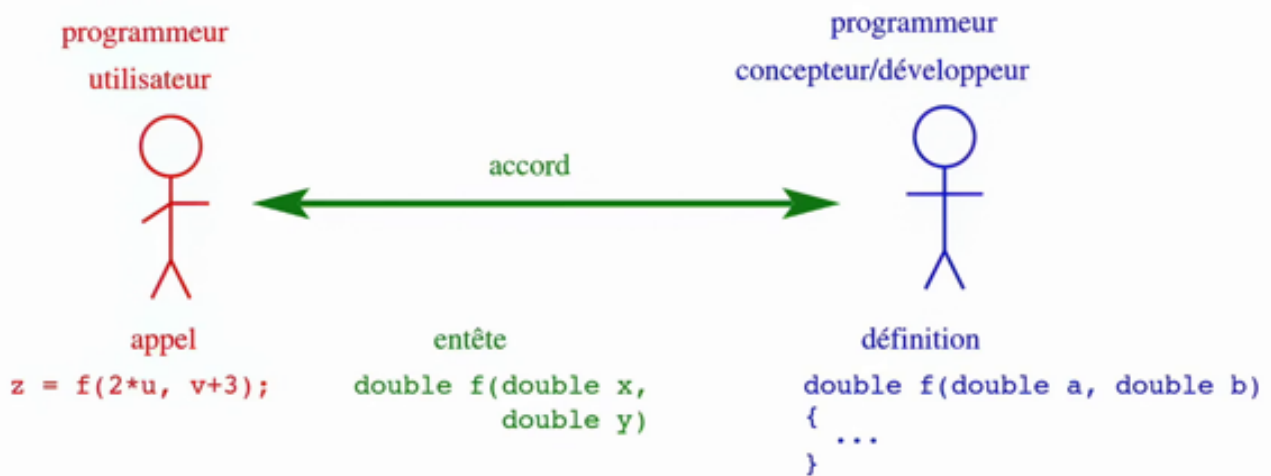
.....

.....

.....

Les « 3 facettes » d'une méthode

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



et son utilité en programmation. Nous allons aujourd'hui nous intéresser à ce qui se passe lorsqu'on utilise, lorsqu'on appelle une méthode. Concrètement à ce stade, nous savons qu'à la notion de méthode sont associées trois facettes : l'entête, la définition et l'appel.

notes

résumé

0m 6s



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;
    }
}
```

Et nous allons aujourd'hui nous intéresser en détail à ce qui se passe au moment de l'appel.

notes

résumé

0m 25s



Évaluation d'un appel de méthode

EPFL

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

Que se passe-t-il lors de l'appel suivant :

→ $z = \text{moyenne}(1.5 + 0.8, 3.4 * 1.25);$

1. $1.5 + 0.8 \rightarrow 2.3$

$3.4 * 1.25 \rightarrow 4.25$

2. $x = 2.3$; $y = 4.25$

3. rien

4. $(x + y) / 2.0 \rightarrow 3.275$

x 2.3

y 4.25



Reprenons notre exemple du calcul des moyennes. Nous avons ici à faire à un petit programme principal, qui demande à l'utilisateur d'introduire deux notes. Ces deux notes sont introduites par l'utilisateur par le biais du clavier, et enfin la moyenne de ces deux notes est affichée. Le calcul de la moyenne est réalisé ici par le biais d'un appel à une méthode, la méthode moyenne. Et nous allons maintenant étudier les mécanismes impliqués par un tel appel. Nous avons ici un exemple concret d'appel de méthode, qu'on appelle la méthode moyenne, et on l'appelle en lui passant deux arguments, ici sous la forme d'expressions qui bien évidemment vont correspondre respectivement aux deux paramètres de la méthode. Vous noterez que l'on appelle habituellement paramètre les données dont a besoin la méthode pour pouvoir fonctionner telle que décrit dans l'entête de la méthode et par argument les données effectives fournies à la méthode au moment de son invocation, de son appel. Donc le but ici est de décortiquer ce qui se passe au moment où un tel appel est exécuté. La toute première étape qui est effectuée consiste en fait à évaluer les différents arguments et en java cette évaluation va se faire systématiquement de gauche à droite. Donc ici concrètement, le premier argument est évalué. Donc l'expression est évaluée, qui va retourner à une valeur de 2.3. La seconde. Ce qui va produire la valeur 4.25. Ensuite, seconde étape, il faudra établir le lien entre, concrètement, ces arguments et les paramètres de la méthode. C'est ce qui se passe au moment de la seconde étape. Donc, lors de la seconde étape, on va apparier le paramètre x avec le résultat de la première expression, x vaudra 2.3. Et apparier y avec le résultat de l'évaluation de la seconde expression : 4.25. Nous nous trouvons donc dans la situation où les paramètres x et y ont des valeurs concrètes avec lesquelles la

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

0m 30s



Évaluation d'un appel de méthode

EPFL

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

Que se passe-t-il lors de l'appel suivant :

→ $z = \text{moyenne}(1.5 + 0.8, 3.4 * 1.25);$

1. $1.5 + 0.8 \rightarrow 2.3$

$3.4 * 1.25 \rightarrow 4.25$

2. $x = 2.3$; $y = 4.25$

3. rien

4. $(x + y) / 2.0 \rightarrow 3.275$

x 2.3

y 4.25



méthode va pouvoir maintenant travailler et s'exécuter. Troisième étape, le corps de la fonction est exécuté. Ici en l'occurrence, le corps consiste en une instruction unique qui est le retour, donc ici l'expression de retour. Donc ici, le corps à proprement parler avant l'expression de retour ne fait rien de particulier. Quatrième étape, c'est l'expression qui va servir de valeur de retour qui maintenant va être évaluée. Donc concrètement, x plus y sur deux va être calculé, ce qui produira le résultat suivant.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

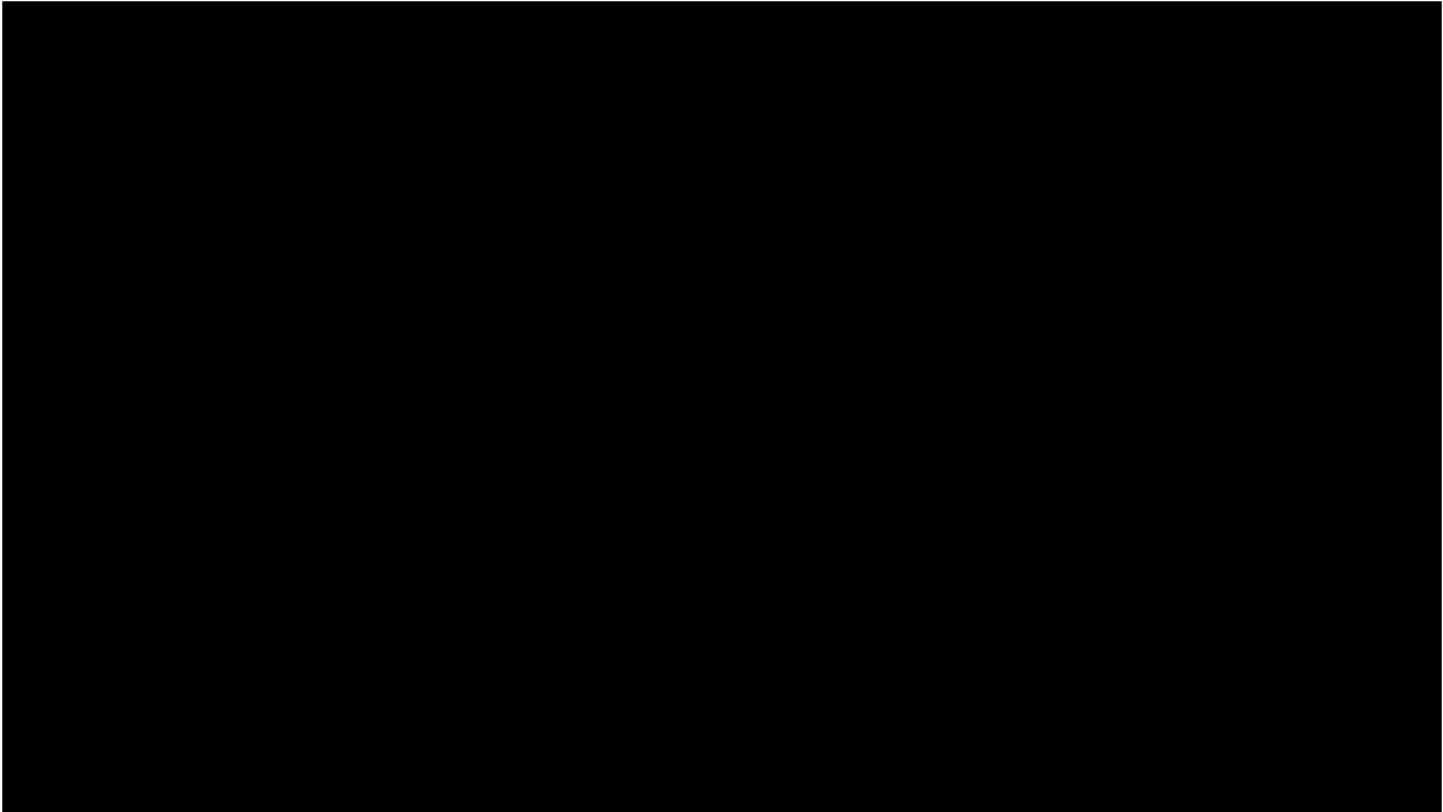
résumé

.....

.....

.....

.....



Et cette valeur est maintenant retournée par la méthode qui va la mettre à disposition du problème, du programme appelant. Donc en fait, tout cet appel de méthode est remplacé par le résultat produit par la fonction. Donc ultime étape, z prend la valeur 3.275. z prend la valeur 3.275.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

3m 37s

