Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-06-2-Appel-pt2

Concepts (extraits des sous-titres générés automatiquement) :

Paramètres de la méthode. Valeurs correspondantes. Cas de ce petit exemple. Méthode. Première étape. Valeur de retour. Résultat de l'appel. Nombre de point. Besoin de paramètres de données. Guise d'arguments. Quatrième étape. Résultat de l'évaluation de cette expression. Petit schéma récapitulatif. Méthode affichescore. Petit exemple concret.

vers la recherche de séquences vidéo
(dans Initiation à la programmation (en Java).)

vers la vidéo

EPFL

Fonctions : appel
(Partie 2)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

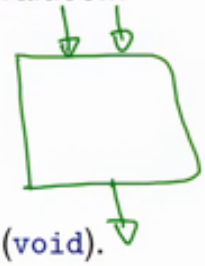
0m 0s

Évaluation d'un appel de méthode (résumé)

EPFL

L'évaluation de l'**appel** d'une méthode s'effectue de la façon suivante :

1. les *expressions* passées en argument sont évaluées
2. les valeurs correspondantes sont **affectées** aux paramètres de la méthode
3. le corps de la méthode est exécuté
4. l'expression suivant la première commande `return` rencontrée est évaluée...
5. ...et retournée comme résultat de de l'appel :
   cette valeur remplace l'expression de l'appel



Les étapes 1 et 2 n'ont pas lieu pour une méthode sans arguments.

Les étapes 4 et 5 n'ont pas lieu pour une méthode sans valeur de retour (`void`).

Nous venons donc de voir les cinq étapes qui se déroulent habituellement lorsque l'on appelle une méthode. Donc, première étape, les expressions qui sont passées en guise d'arguments à la méthode sont évaluées. Deuxièmement, les valeurs correspondantes sont ensuite affectées aux paramètres de la méthode. La méthode dispose à ce moment-là de toutes les informations qui lui sont nécessaires pour pouvoir travailler, son corps peut s'exécuter. Quatrième étape, l'expression qui suit la première commande return rencontrée à l'exécution va être évaluée et le résultat de l'évaluation de cette expression est ensuite retournée comme résultat de l'appel. Ces cinq étapes se déroulent dans le cas le plus général, il existe des situations où le déroulement est un peu simplifié. Donc, nous savons que dans le cas le plus général une méthode a besoin de paramètres de données entrantes et produit un résultat en sortie.

notes

résumé

0m 1s

Appel : autre exemple

Une méthode peut appeler une autre méthode.

```
void afficheScore(int joueur, double points, double temps)
{
 System.out.println("  Joueur " + joueur
        + score(points, temps) + " points");
}

int score (double points, double tempsJeu)
{ // ... comme avant ... }
```

Alors il existe des situations cependant, où la méthode n'a pas besoin de données entrantes et à ce moment-là évidemment les étapes un et deux, qui consistent à évaluer les expressions passées en arguments et les affecter aux paramètres n'ont pas lieu pour une méthode sans arguments. Et puis, il existe des situations où il n'y a pas de données en sortie, la méthode ne fournit aucun résultat, et à ce moment-là les étapes qui consistent à évaluer l'expression après le return et retourner ce résultat, les quatre et cinq n'ont évidemment pas lieu. Une méthode peut parfaitement appeler une autre méthode, c'est le cas de

notes

résumé

1m 1s

Appel : autre exemple

Une méthode peut appeler une autre méthode.

```
void afficheScore(int joueur, double points, double temps)
{
 System.out.println("  Joueur " + joueur
        + score(points, temps) + " points");
}

int score (double points, double tempsJeu)
{ // ... comme avant ... }
```

ce petit exemple-là où vous avez une méthode afficheScore qui prend en

notes

résumé

1m 37s

Appel : autre exemple

Une méthode peut appeler une autre méthode.

```
void afficheScore(int joueur, double points, double temps)
{
 System.out.println("  Joueur " + joueur
        + score(points, temps) + " points");
}

int score (double points, double tempsJeu)
{ // ... comme avant ... }
```

paramètres un joueur identifié par un entier, un nombre de point gagnés par

notes

résumé

1m 43s

Appel : autre exemple

Une méthode peut appeler une autre méthode.

```
void afficheScore(int joueur, double points, double temps)
{
 System.out.println("  Joueur " + joueur
        + score(points, temps) + " points");
}

int score (double points, double tempsJeu)
{ // ... comme avant ... }
```

le joueur, le temps qu'il a passé à jouer et qui va afficher, pour le joueur, son score. Ici, il se trouve que le score est calculé en fonction du nombre de points et du temps passé, et le calcul se fait par le biais d'une méthode. La méthode est définie à un autre endroit et prend en entrée justement le nombre de points et le temps de jeu.

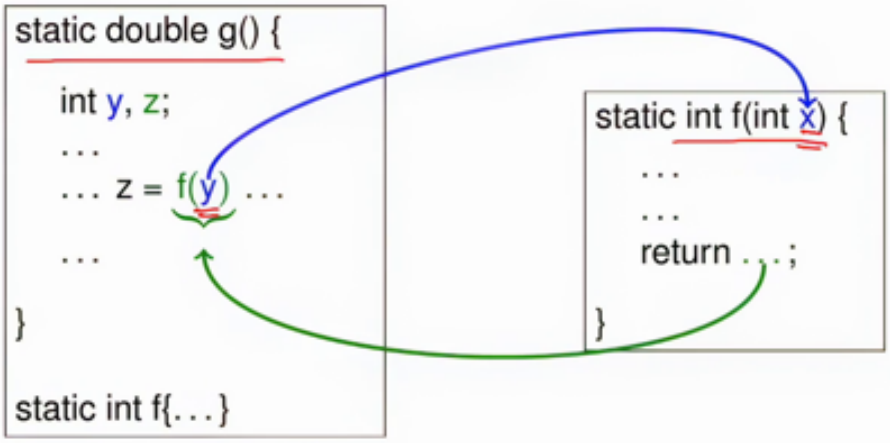
notes

résumé

1m 50s

Appel : résumé

L'évaluation de l'appel d'une méthode peut être schématisé de la façon suivante :



Vous connaissez maintenant les différentes étapes qui s'exécutent lorsqu'une méthode est appelée dans un programme et vous avez maintenant sous vos yeux un petit schéma récapitulatif qui synthétise ces différentes étapes. Donc ici, on a une méthode f qui est appelée par une méthode g. La correspondance entre les paramètres de la méthode et les arguments d'appel se fait au moment de l'appel.

notes

résumé

2m 13s

Donc ici, au moment de l'appel, l'argument est copié dans le paramètre de la méthode qui l'utilise pour exécuter les traitements qu'elle doit faire. Dans le cas le plus général, la méthode calcule une valeur de retour et cette valeur de retour est transmise à la méthode appelante qui va pouvoir l'utiliser pour réaliser les traitements.

notes

résumé

2m 37s

Résumé du jargon

« *Appeler la méthode* `f` » = utiliser la méthode `f` : `x = 2 * f(3);`

« *3 est passé en argument* » = (lors d'un appel) la valeur 3 est copiée dans un paramètre de la méthode :

```
x = 2 * f(3);
```

« *la méthode retourne la valeur de y* » = l'expression de l'appel de la méthode sera remplacée par la valeur retournée

```
{
return y;
}
...
x = 2 * f(3);
```

Autres exemples : « `cos(0)` retourne le cosinus de 0 », « `cos(0)` retourne 1 ».

Vous aurez sans doute constaté tout au long de cette séquence qu'il existe un certain jargon, une terminologie associée à l'appel de méthode, nous allons maintenant résumer un petit peu ce jargon. Vous aurez compris que lorsque je parle d'appeler une méthode, je signifie que j'utilise la méthode, ici je suis entrain d'utiliser la méthode f pour calculer un résultat que j'affecte ensuite à une variable x. Lorsque je parle de passer une valeur en argument à une méthode, je signifie que cette valeur est simplement copiée dans un paramètre de la méthode. Et enfin, lorsque je dis qu'une méthode retourne la valeur de y par exemple, je signifie que l'expression de l'appel de la méthode sera simplement remplacé par le valeur retournée. Donc si on prend un petit exemple concret maintenant, supposons que nous ayons ici une méthode f dont la dernière instruction soit celle-ci.

notes

résumé

3m 0s

Supposons que cette méthode soit appelée de cette façon-là, et qu'au moment de cet appel la valeur de y soit concrètement cinq, dire que la méthode retourne la valeur de y, c'est-à-dire cinq, revient à dire que l'on remplace simplement cet appel-là par la valeur retournée, c'est-à-dire cinq. Ainsi, je peux dire indifféremment cos de zéro retourne le cosinus de zéro ou cos de zéro retourne un, par exemple. ou cos de zéro retourne un, par exemple.

notes

résumé

3m 49s