Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-01-5-Expressions-pt3

Concepts (extraits des sous-titres générés automatiquement) :

Variables de types. Variables de type. Valeurs décimales. Exemple de type double. Variable de type. Côté des valeurs entières. Exemple de type. Variable x de type. Variable moyenne. Solution possible. Piège de la division entière. Variable n. Valeur décimale. Valeur de l'expression. Variable n de type.

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

vers la vidéo

EPFL

Expressions
(Partie 3)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s

On dispose aussi des opérateurs `+=` , `-=` , `*=` , `/=`

Par exemple:

Jusqu'ici nous avons toujours vu

notes

résumé

0m 1s

On dispose aussi des opérateurs +=, -=, *=, /=

Par exemple:

a += 5 .

des valeurs décimales et des variables de types "double" d'un côté, et d'un autre côté des valeurs entières et des variables de type "int". Que se passe t-il quand on essaie d'affecter par exemple une valeur décimale

notes

résumé

0m 6s

On dispose aussi des opérateurs +=, -=, *=, /=

Par exemple:

```
a += 5;
a = a + 5;

a *= b;
```

à une variable de type "int" ? Eh bien en JAVA, il est impossible d'affecter une valeur décimale par exemple de type "double" à une variable de type "int".

notes

résumé

0m 19s

Opérateur modulo (%)

Dans le cas des `int`, il existe aussi:

- un opérateur modulo, noté `%`, qui renvoie le **reste** de la division entière:

Par exemple j'ai déclaré ici une variable x de type "double" et j'essaie de mettre la valeur de l'expression 3*x, qui est donc de type "double", dans ma variable n qui est déclarée de type "int" Eh bien, le compilateur va refuser de compiler ce code

notes

résumé

0m 32s

Opérateur modulo (%)

Dans le cas des `int`, il existe aussi:

- un opérateur modulo, noté `%`, qui renvoie le **reste** de la division entière:

11 % 4 → 3 $\quad$ 11 = 2×4+3

12 % 4 → 0 $\quad$ 12 = 3×4+0

et va produire le message d'erreur "possible loss of precision" qui dit que si je fais cette affection, je perdrais la valeur qui est après la virgule. En revanche, il est possible d'affecter une valeur de type "int" à une variable de type décimale, par exemple de type double. J'ai, ici, déclaré une variable n de type "int" et j'essaie de mettre la valeur de l'expression 2*n qui est, elle, de type "int" dans ma variable x de type double. Eh bien dans ce cas, le compilateur va faire la conversion de lui-même, du type "int" vers le type "double", ce qui est parfaitement possible puisque dans ce cas-là je n'ai pas de perte de précision.

notes

résumé

0m 49s

Opérateurs ++ et −−

- deux opérateurs notés ++ et −−, qui permettent respectivement d'incrémenter et de décrémenter, c'est-à-dire d'ajouter et de soustraire 1 à une variable.

Revenons maintenant sur le piège de la division entière.

notes

résumé

1m 37s

Alors j'ai déclaré ici une variable x de type "double" que j'ai représenté ici. Et dans cette affectation, j'affecte la valeur de l'expression 1 / 2 à ma variable x. Alors à votre avis, que va contenir x après cette affectation ? 1 et 2 sont deux valeurs littérales toutes les deux de type "int". Cette division est donc la division entière et 1 / 2 dans le cas de la division entière vaut 0. On va donc affecter la valeur 0 à la variable x Alors notez au passage qu'il y a une conversion automatique de "int" vers "double", puisque ce 0 ci est de type "int" et le 0 affecté à x est évidemment de type "double". Donc le fait que x soit de type "double" n'influence pas l'évaluation de l'expression 1 / 2.

notes

résumé

1m 40s

Affectation d'une valeur entière à une variable décimale

En revanche, il est possible d'affecter une valeur de type `int` à une variable de type décimale, par exemple `double`.

Exemple:

```
int n = 3;
double x = 2 * n;
```

Le problème peut se poser en pratique quand on essaie, par exemple, de calculer la moyenne de deux valeurs entières. Dans cet exemple j'ai note1 qui est une variable de type "int" initialisée à 4, note2 est également de type "int" et initialisée à 5. Si j'essaie de calculer leur moyenne en faisant cette déclaration ; dans cette expression, note1 et note2 sont de type "int" Leur somme est donc de type "int", alors leur somme vaut 9 mais c'est une valeur de type "int". Ce 2 ci pour calculer la moyenne est une valeur littérale de type "int", donc cette division est la division entière.

notes

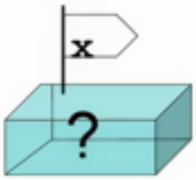
résumé

2m 37s

La division entière

```
double x;

x = 1 / 2;
```



on va donc calculer le résultat de la division entière entre 9 et 2, obtenir la valeur 4 et l'affecter à ma variable moyenne alors que j'aurais voulu avoir 4,5 de moyenne. Une solution possible est de calculer d'abord la somme des deux valeurs de type "int". Donc ici j'ai note1 + note2 qui est affecté à la variable moyenne note1 + note2 et 9 de types "int" qui va être affecté à la variable moyenne

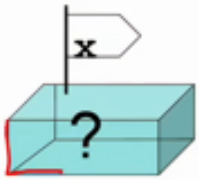
notes

résumé

3m 25s

La division entière

```
double x;
x = 1 / 2;
```



Quand je fais cette instruction, qui est équivalente à, je vous le rappelle, écrire moyenne = moyenne / 2 2 est une valeur littérale de type "int"

notes

résumé

4m 0s

mais moyenne est une variable de type "double" donc cette division est donc la division dont on a l'habitude. Et cette fois-ci je vais bien obtenir 4,5 dans ma variable moyenne dans ma variable moyenne

notes

résumé

4m 13s