Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-01-3-Variables-pt3

Concepts (extraits des sous-titres générés automatiquement) :

Second exemple. Terme de méthodologie. Symboles spéciaux. Nombre de règles. Ensemble de lettres. Premier caractère. Variables p. Exemple précédent. Règles de nommage. Nombre de conventions usuelles. Données numériques. Langage java. Type entier. Partie des conventions. Déclaration des différentes variables.

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

vers la vidéo

EPFL

Variables
(Partie 3)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s

Déclaration de variables

D'autres exemples de déclaration:

```
int m = 1;
int p = 1, q = 0;
double x = 0.1, y;
```

on peut déclarer plusieurs variables simultanément.
Ne pas en abuser

Il est licite en java de déclarer plusieurs variables sur une même ligne.

notes

résumé

0m 1s

Déclaration de variables

D'autres exemples de déclaration:

```
int m = 1;
int p = 1, q = 0;
double x = 0.1, y;
```

on peut déclarer plusieurs variables simultanément.
Ne pas en abuser

On va déclarer sur une même ligne des variables du même type, ce qui veut dire que l'on spécifie le type qu'une seule fois. Ici, on a la déclaration de deux variables p et q, de type entier, et on sépare la déclaration des différentes variables, au moyen de la virgule.

notes

résumé

0m 6s

Déclaration de variables

D'autres exemples de déclaration:

```
int m = 1;
int p = 1, q = 0;
double x = 0.1, y;
```

on peut déclarer plusieurs variables simultanément.
Ne pas en abuser

pas de valeur initiale

Vous avez ici, un second exemple, où je déclare deux double sur la même ligne. La différence avec l'exemple précédent, est que pour y, je ne donne pas de valeur initiale, ce qui est aussi possible. En terme de méthodologie, gardez à l'esprit qu'il ne faut pas abuser de ce genre de tournure

notes

résumé

0m 25s

Noms de variables

Règles pour nommer les variables:

- Le nom peut être constitué uniquement de lettres, de chiffres, et des deux symboles autorisés: _ (underscore) et $ (pas d'espace);
- Le premier caractère est nécessairement une lettre ou un symbole;
- Le nom ne doit pas être un mot-clé réservé par le langage Java;
- Les majuscules et les minuscules sont autorisées mais ne sont pas équivalentes. Les noms `ligne` et `Ligne` désignent deux variables différentes.

Exemples de noms valides:

`nCarreTotal` `sousTotal98`

Exemples de noms invalides:

`n carre` Contient des espaces;
`n-carre` Contient le symbole – (moins);
`1element` Commence par un chiffre.

car elles peuvent nuire à la lisibilité du programme. Il existe un certain nombre de règles à respecter, lorsque l'on choisit un identificateur pour une variable. Il faudra faire en sorte que cet identificateur soit constitué uniquement de lettres, de chiffres, ou de l'un des symboles spéciaux, souligné (_) ou dollar ($). Le premier caractère doit nécessairement être une lettre, il peut aussi être l'un des symboles spéciaux, souligné (_) ou dollar ($), mais cela ne fait pas partie des conventions usuellement utilisées par les programmeurs java, dont nous reparlerons plus loin. Bien entendu, l'identificateur ne doit pas être un mot-clé réservé du langage java. Et les majuscules et minuscules sont autorisées,

notes

résumé

0m 46s

Conventions en Java pour les noms de variables

En Java, bien que ce ne soit pas requis par le compilateur, la convention est d'écrire le nom des variables en commençant par une minuscule, et commencer les mots suivants par une majuscule.

Par exemple, on utilisera

```
nombreDePoints
```

plutôt que

```
NombreDePoints
```

ou

```
nombre_de_points
```

mais elles ne sont pas équivalentes. Par exemple, si je choisis l'identificateur ligne commençant par un petit l, il n'est pas équivalent à l'identificateur Ligne, commençant avec un grand L. Vous avez donc ici des exemples typiques d'identificateurs valides. Ici, constitués d'un ensemble de lettres uniquement, ici, constitués d'un ensemble de lettres et de chiffres, et ici, vous avez un certain nombre d'identificateurs invalides. Ici, un identificateur ne peut pas contenir d'espace ni le symbole moins (-), et ne peut pas commencer par un chiffre. En plus des règles de nommage imposées par le langage java, et qu'il faut strictement respecter, il existe un certain nombre de conventions usuelles, qui elles, ne sont pas requises par le compilateur mais que néanmoins, la plupart des programmeurs java respectent. Typiquement, vous trouverez le nommage des variables sous cette forme là. Si le nom de la variable, l'identificateur est constitué de plusieurs mots, on va séparer les mots, en commençant chacun des mots par une majuscule, et avec la convention que tout identificateur de variable commence plutôt par une minuscule. Donc vous allez trouver ce genre de tournure, plutôt que celle-ci, qui commence par une majuscule,

notes

résumé

1m 25s

Types de variables

Les principaux types élémentaires sont:

- `int`, pour les valeurs entières (pour *integer*, entiers en anglais);
- `double`, pour les nombres à virgule, par exemple 0.5

et aussi:

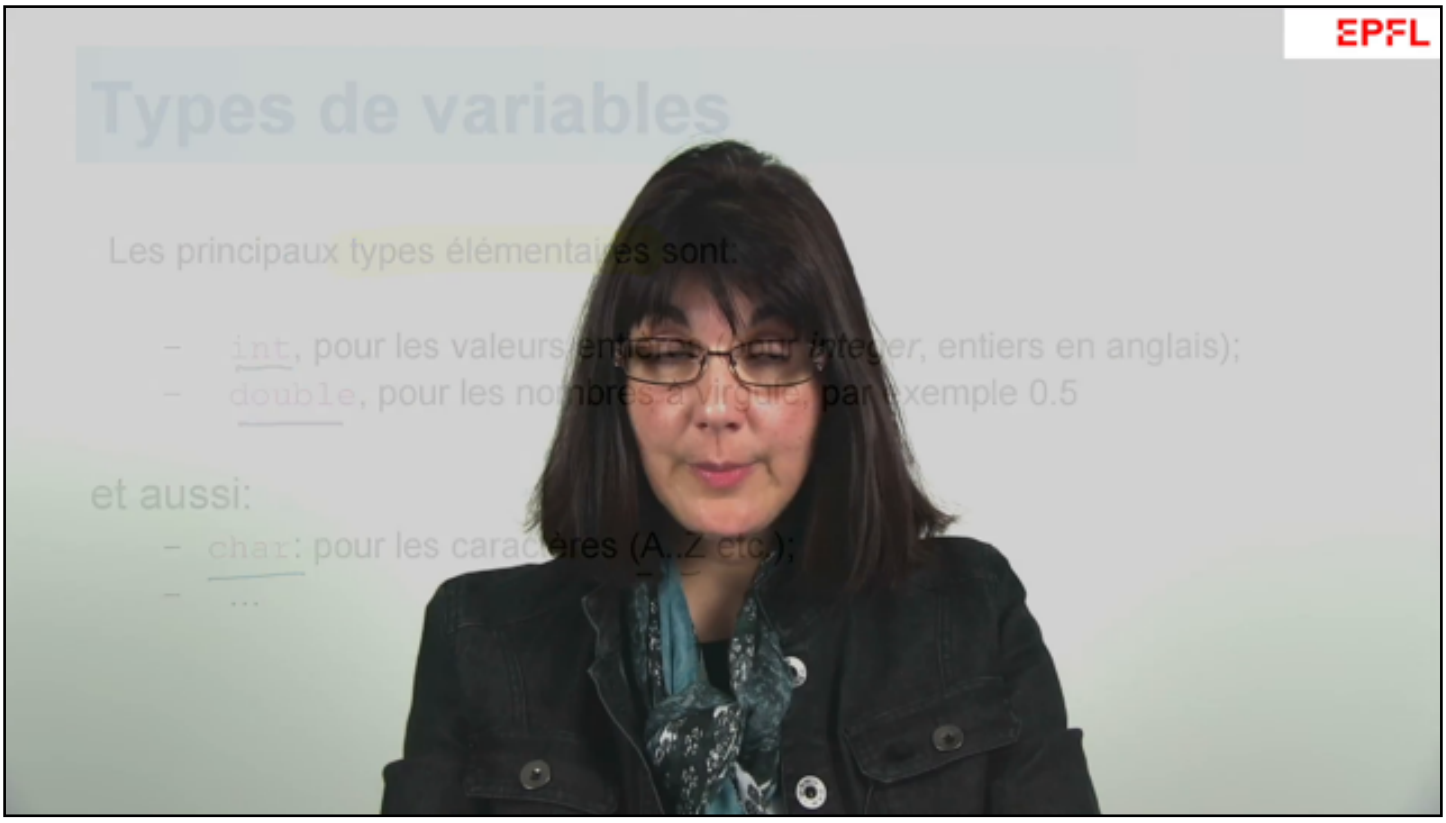
- `char`: pour les caractères (A..Z etc.);
- ...

ou celle-là, où on utilise des soulignés (_). Nous avons donc vu que la notion de type était essentielle pour déclarer une variable en java. Les deux types élémentaires à disposition pour manipuler des données numériques, c'est-à-dire pour déclarer des variables de type numérique, sont int et double, que nous avons eu l'occasion de croiser, dans divers exemples précédemment. Bien évidemment, il existe de nombreux autres types prédéfinis.

notes

résumé

2m 37s



A titre d'exemple, si dans un programme, j'ai besoin de manipuler les caractères usuels de l'alphabet, comme par exemple A et Z, je peux avoir recours au type prédéfini char. Nous aurons l'occasion tout au long du cours de revenir sur les types prédéfinis en java.

notes

résumé

3m 1s

Affectations

La ligne:

```
nCarre = n * n;
```

est une **affectation**.

Attention, ce **n'est pas** une égalité mathématique: Une affectation est une instruction qui permet de **changer** une valeur à une variable.

Vous savez maintenant déclarer une variable, ce qui va vous permettre de mémoriser une donnée, utilisable dans un programme. Que faire maintenant si l'on souhaite changer la valeur d'une variable, qu'on a au préalable déclarée ? Pour cela, il faut utiliser la notion d'affectation. Nous avons déjà eu l'occasion d'en croiser quelques exemples informels.

notes

résumé

3m 18s

Pour réaliser une affectation, je vais utiliser le symbole égal (=), en respectant la syntaxe suivante. A gauche de l'expression, je spécifie l'identificateur de la variable, ensuite évidemment mon symbole égal (=), et finalement la nouvelle valeur que je veux désormais stocker dans ma variable. Il est très important de comprendre, nous aurons l'occasion d'en discuter un peu plus loin, qu'une affectation n'est pas une égalité mathématique. Son rôle est de changer la valeur d'une variable en cours d'exécution d'un programme. en cours d'exécution d'un programme.

notes

résumé

3m 37s