Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-05-4-tabdyn-pt1

Concepts (extraits des sous-titres générés automatiquement) :

Cas des tableaux de taille. Tableaux dynamiques. Type particulier. Moment de l'initialisation du tableau. Nombre de fonctionnalités de méthodes. Nom de la variable. Variable de type. Chaines de caractères. Tableaux statiques. Élément d'indice. Type arraylist. Exemple concret. Syntaxe suivante. Nom de la méthode de la fonctionnalité. Moyen d'un tableau vide.

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

vers la vidéo

EPFL

Tableaux dynamiques
(Partie 1)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s

Dans une séquence vidéo précédente nous avons étudié le cas des tableaux de taille fixe en Java. Nous avons vu que pour ces tableaux, une fois la taille fixée au moment de l'initialisation du tableau, il n'est plus possible de la modifier lorsque le programme s'exécute. Nous allons aujourd'hui étudier les tableaux dynamiques qui nous affranchiront de cette contrainte. Les tableaux dynamiques ont la particularité de pouvoir voir leur taille croître ou décroître pendant que le programme s'exécute. Leur utilisation nécessite en réalité un certain nombre de concepts liés à l'orienté objet. Cependant tout comme pour les chaines de caractères, il n'est pas nécessaire d'avoir une connaissance préalable de ces concepts pour pouvoir utiliser les fonctionnalités requises au préalable. Nous avons choisi de les présenter très tôt dans le cadre de ce cours car les tableaux dynamiques sont d'une utilisation très naturelle pour la résolution de nombreux problèmes

notes

résumé

0m 1s

qui ne sont pas forcément liés à l'orienté objet.

notes

résumé

0m 49s

Déclaration d'un tableau dynamique

EPFL

Une variable correspondant à un tableau dynamique se déclare de la façon suivante :

```
ArrayList<type> identificateur;
```

où *identificateur* est le nom du tableau et *type* correspond au type des éléments du tableau.

Le type des éléments doit nécessairement correspondre à un **type évolué**.

Exemple :

```
ArrayList<String> tableau;
```

Comme pour les tableaux statiques, les tableaux dynamiques sont une collection de données homogènes. Pour pouvoir les utiliser, on a recours à un type particulier pré défini en Java qui s'appelle « ArrayList », ce type n'est pas défini par défaut, pour l'utiliser dans un fichier il faut inclure au début du fichier une directive d'importation qui va rendre accessible le type ArrayList. On peut déclarer en Java une variable de type « ArrayList » en ayant recours à la syntaxe suivante : on met le nom de la variable et en guise de type, il faudra mettre le nom réservé « ArrayList » suivi des deux symboles ouvrant et fermant inférieur et supérieur, et entre ces deux symboles le type de données

notes

résumé

0m 52s

Initialisation d'un tableau dynamique

Un tableau dynamique initialement vide (sans aucun élément) s'initialise comme suit :

```
ArrayList<type> identificateur = new ArrayList<type>();
```

où *identificateur* est le nom du tableau et *type* correspond au type des éléments du tableau.

Exemple :

```
ArrayList<String> tableau = new ArrayList<String>();
```

qu'on veut mettre dans le tableau. Notez que le type doit être impérativement un type évolué ; voici un exemple concret : ici on est en train de déclarer une variable tableau, le type de cette variable est un tableau dynamique de chaîne de caractères. Vous noterez que la variable ainsi déclarée n'est pas initialisée, on ne lui affecte pas de valeurs initiales. Comment faire pour initialiser un tableau dynamique ? Une façon courante d'initialiser un tableau dynamique est de l'initialiser au moyen d'un tableau vide, un tableau sans aucun élément. Ceci se fait par la tournure suivante : donc à la variable associée au tableau nous allons affecter un tableau vide qui est créé par la tournure suivante, donc une tournure à respecter exactement sous cette forme là le mot réservé « new », le mot réservé « ArrayList », ici il faudra mettre exactement le même type que celui qu'on a mis au moment de la déclaration du tableau, suivi des deux parenthèses à ne pas oublier. Ici nous avons recours en fait à une tournure liée à l'orienté objet mais il n'est pas nécessaire d'en comprendre les tenants et aboutissants pour l'employer. Ce qu'il faut retenir ici c'est qu'au moyen de cette instruction, vous avez construit un tableau dynamique sans aucun élément. Ici vous avez un exemple concret de déclaration initialisation d'un tableau dynamique de chaîne de caractères, la variable tableau contient la référence à un tableau dynamique de chaîne de caractères sans aucun élément. Puisque ce tableau ainsi déclaré ne contient aucun élément, nous allons naturellement nous poser la question

notes

résumé

1m 37s

Méthodes spécifiques

Un certain nombre d'opérations sont **directement attachées** au type `ArrayList`.

L'utilisation de ces opérations spécifiques se fait avec la syntaxe suivante :

nomDeTableau.nomDeMethode(arg1, arg2, ...);

Exemple :

```
ArrayList<String> prenoms = new ArrayList<String>();

System.out.println(prenoms.size()); // affiche 0
```

de comment mettre des valeurs dans ce tableau.

notes

résumé

3m 13s

Méthodes spécifiques

EPFL

Un certain nombre d'opérations sont directement attachées au type `ArrayList`.

L'utilisation de ces opérations spécifiques se fait avec la syntaxe suivante :

nomDeTableau.nomDeMethode(arg1, arg2, ...);

Exemple :

```
ArrayList<String> prenoms = new ArrayList<String>();

System.out.println(prenoms.size()); // affiche 0
```

Comme pour les chaîne de caractères, il existe un certain nombre de fonctionnalités de méthodes spécifiques liées au type « ArrayList » ; pour pouvoir utiliser ces fonctionnalités il faut recourir à une syntaxe particulière, nom du tableau, suivi du point, suivi du nom de la méthode de la fonctionnalité et ensuite on doit indiquer entre parenthèses les données dont a besoin la fonctionnalité, la méthode pour pouvoir s'exécuter. Vous avez ici un exemple concret je déclare un tableau dynamique de chaîne de caractères qui s'appelle « prenoms » et j'initialise ce tableau au moyen d'un tableau vide. Ensuite j'ai recours à une fonctionnalité particulière définie pour les « ArrayLists » qui s'appelle « size ». Cette fonctionnalité retourne simplement la taille courante du tableau dynamique. J'ai eu recours à la syntaxe que je viens d'évoquer

notes

résumé

3m 19s

Méthodes spécifiques

Quelques fonctions disponibles pour un tableau dynamique nommé `tableau`, de type `ArrayList<type>` :

`tableau.size()` : renvoie la taille de `tableau` (un entier)

à savoir, le nom du tableau, suivi de point, suivi du nom de la fonctionnalité. « Size » n'a besoin d'aucune information extérieure pour pouvoir fournir la taille du tableau, donc je dois mettre une liste de parenthèses vides ; évidemment l'exécution de cette instruction va afficher 0 puisque j'ai initialisé le tableau au moyen d'un tableau vide.

notes

résumé

4m 1s

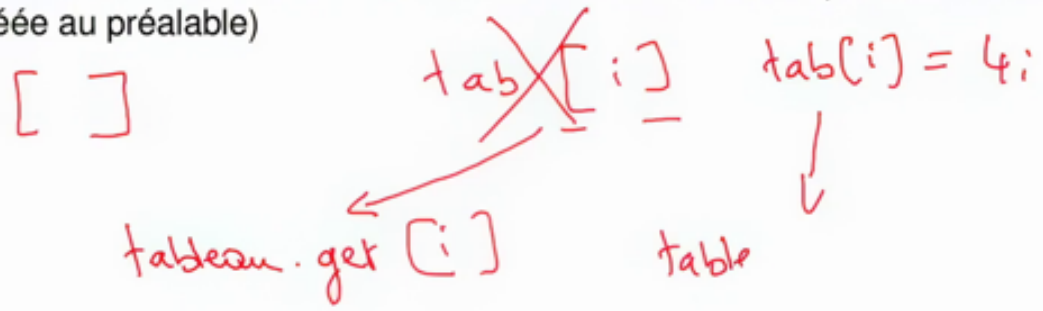
Méthodes spécifiques

Quelques fonctions disponibles pour un tableau dynamique nommé `tableau`, de type `ArrayList<type>` :

`tableau.size()` : renvoie la taille de `tableau` (un entier)

`tableau.get(i)` : renvoie l'élément à l'indice `i` dans le tableau (`i` est un entier compris entre 0 et `tableau.size() - 1`)

`tableau.set(i, valeur)` : affecte `valeur` à la case `i` du tableau (cette case doit avoir été créée au préalable)



Voici maintenant une liste non exahaustive de quelques fonctionnalités typiques associées aux tableaux. Nous venons juste de parler de la méthode « size » qui permet de connaître la taille d'un tableau et qui le retourne sous la forme d'un entier. Il existe d'autres méthodes comme « get » et « set » que je vais employer exactement selon le même principe, à savoir nom du tableau, point, suivi du nom de la fonctionnalité et je constate qu'ici, contrairement à la méthode « size », j'ai besoin de fournir des données à la méthode « get » ou « set » pour que celles-ci puissent fonctionner. Les méthodes « get » et « set » sont l'équivalent du mécanisme d'indexation pour les tableaux statiques. Pour un tableau statique simple, si je souhaite accéder à l'élément d'indice 'i' j'utilise l'indexation ; ce mécanisme n'existe pas pour les tableaux dynamiques ; à la place il faut avoir recours à des méthodes telles « get » ou « set ». Pour un tableau dynamique, si je veux accéder à l'élément d'indice 'i', j'utiliserai cette notation plutôt que celle-ci. De même si je veux modifier un tableau statique je vais utiliser cette notation, donc si je veux modifier la i ème case d'un tableau statique je vais utiliser cette notation, ceci n'existe pas pour les tableaux dynamiques,

notes

résumé

4m 22s

il faudra à la place avoir recours à ce type de notation, donc nom du tableau, point set, entre parenthèses l'indice de la case dont je veux modifier le contenu et la valeur correspondante, donc cette notation n'existe pas non plus pour les tableaux dynamiques. Les restrictions sur les bornes de l'indice restent cependant les mêmes que pour un tableau de taille fixe, à savoir l'indice varie entre 0 et taille du tableau -1, taille du tableau -1 s'écrit pour un tableau dynamique comme ceci, par exemple. Donc, comme pour les tableaux de taille fixe, toute tentative d'accès avec un indice qui serait en dehors de ces bornes va se traduire par une erreur à l'exécution du programme. à l'exécution du programme.

notes

résumé

5m 37s