Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-05-4-tabdyn-pt2

Concepts (extraits des sous-titres générés automatiquement) :

Tableau dynamique. Tableau vide. Méthode clear. Élément d'indice i. Valeur de type booléen. Fin de l'exécution de cette instruction. Tableau d'entiers. Fin du tableau. Tableau dynamique d'entiers. Instruction suivante. Cours d'exécution. Boucle for. Tableau. Éléments d'un tableau dynamique. Chaîne de caractères.

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

vers la vidéo

EPFL

Tableaux dynamiques
(Partie 2)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s

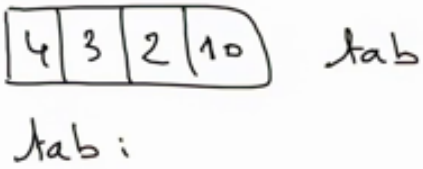
Méthodes spécifiques

EPFL

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

`tableau.isEmpty()` : détermine si `tableau` est vide ou non (`boolean`).

`tableau.clear()` : supprime tous les éléments de `tableau` (et le transforme donc en un tableau vide).



Il est possible de tester si un tableau dynamique est vide en utilisant la fonctionnalité « isEmpty ». C'est une fonctionnalité qui n'a pas besoin de données pour pouvoir fonctionner, donc elle va nous retourner une valeur de type booléen, true ou false dépendamment du fait que le tableau contienne des valeurs ou pas. La valeur retournée sera true si le tableau ne contient aucune valeur et false dans le cas contraire. La méthode clear a pour vocation de vider un tableau dynamique, ce qui veut dire qu'elle en supprime tous les éléments, et elle le transforme en tableau vide ; si on imagine par exemple qu'à un moment donné j'ai dans un programme un tableau dynamique qui est dans cet état de remplissage, on imagine qu'il s'agit d'un tableau d'entiers appelé « tab », si j'invoque la fonctionnalité « isEmpty »

notes

résumé

0m 1s

Méthodes spécifiques

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

`tableau.remove(i)` : supprime l'élément d'indice `i`

il est clair que le résultat sera le booléen false puisque le tableau n'est pas vide. Maintenant je peux appliquer la fonctionnalité « .clear() », par exemple, qui aura pour vocation de supprimer tous les éléments et de le transformer en tableau vide, il n'y a plus d'élément à l'intérieur, donc si à nouveau j'appelle ma fonctionnalité « isEmpty » cette fois-ci le résultat sera true puisque le tableau est désormais vide.

notes

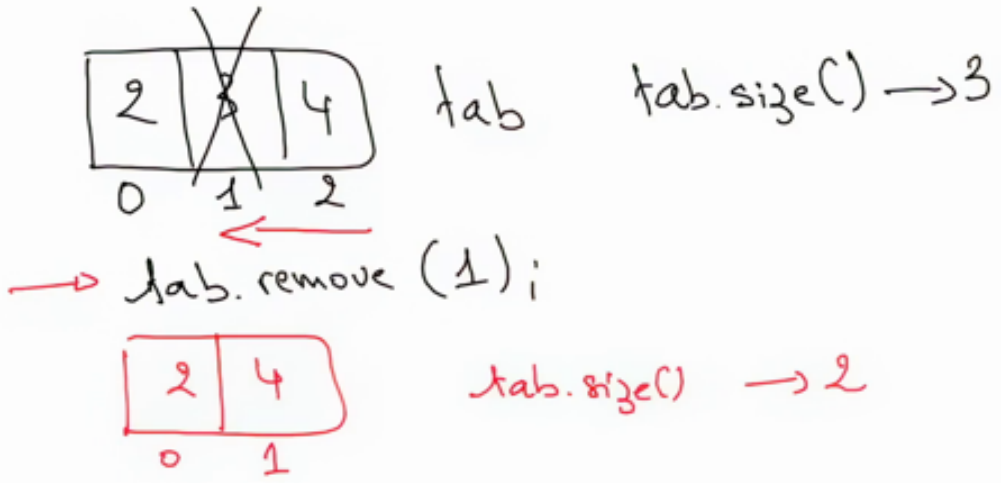
résumé

0m 49s

Méthodes spécifiques

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

`tableau.remove(i)` : supprime l'élément d'indice `i`



La méthode « remove » permet de supprimer l'élément d'indice i dans un tableau dynamique ; donc si on imagine par exemple qu'à un moment donné nous ayons un tableau dynamique d'entiers à trois cases qui ait cette allure particulière, les indices varient entre 0 et taille -1 ce qui nous donne ceci, supposons que le tableau s'appelle « tab » si maintenant j'appelle la fonctionnalité « size » sur ce tableau, le retour sera 3 puisque mon tableau contient trois cases ; si maintenant j'appelle la méthode « remove » en lui passant pour indice 1 c'est cet élément qui doit être supprimé donc tous les éléments qui viennent à la suite de cet élément vont être décalés vers la gauche et au final je vais aboutir à un tableau à deux cases, et si j'applique ma fonctionnalité « tab.size() » le retour sera évidemment 2 ; on voit ici que pour un tableau dynamique la taille peut effectivement varier en cours d'exécution au gré des appels à des méthodes de cette nature là. L'indice de l'élément qu'on veut supprimer doit bien évidemment exister dans le tableau au moment où on invoque la fonctionnalité « remove ».

notes

résumé

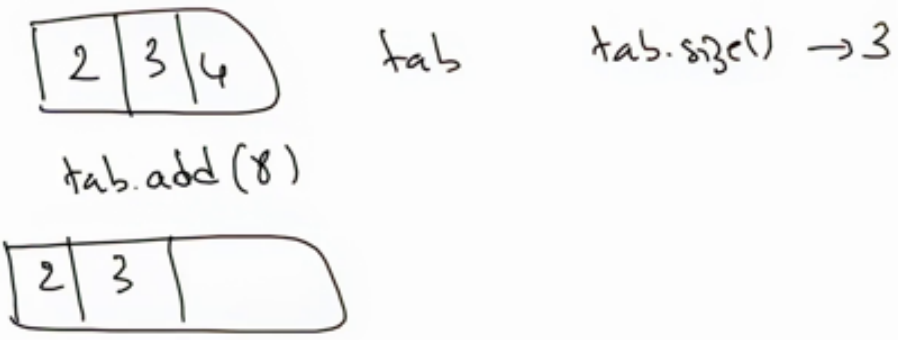
1m 18s

Méthodes spécifiques

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

`tableau.remove(i)` : supprime l'élément d'indice `i`

`tableau.add(valeur)` : ajoute le nouvel élément `valeur` à la fin de `tableau`. Pas de retour.



Supposons par exemple qu'au lieu de faire un « tab.remove(1) » j'essaie de faire un « tab.remove(4) » ceci se traduira par une erreur à l'exécution du programme. S'il est possible de supprimer des éléments d'un tableau dynamique, il est aussi possible d'ajouter des valeurs dans un tel tableau, pour ceci on va utiliser la méthode « add » et il faudra fournir la valeur que l'on souhaite ajouter au tableau, la valeur sera systématiquement ajoutée à la fin du tableau. Prenons un exemple concret d'un tableau d'entiers, donc ici « tab.size() » retourne 3 j'ajoute à mon tableau la valeur 8 en utilisant la méthode « add » le résultat sera un tableau avec une case supplémentaire

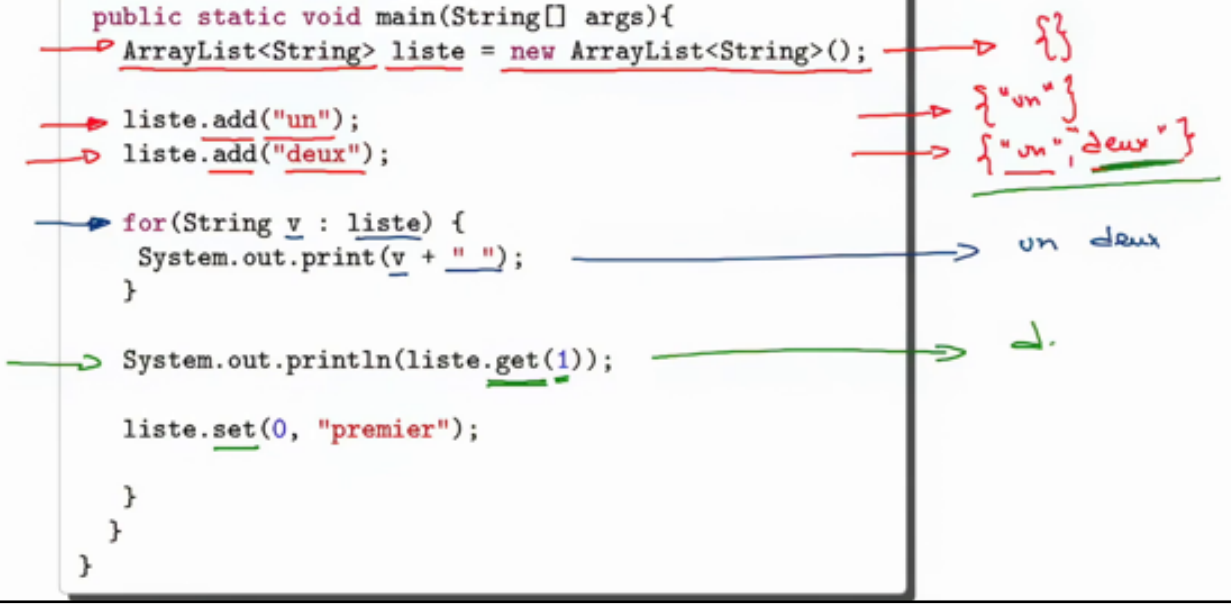
notes

résumé

2m 25s

Exemple de quelques manipulations de base

```
import java.util.ArrayList;

class ArrayListExemple {
 public static void main(String[] args){
   ArrayList<String> liste = new ArrayList<String>();

   liste.add("un");
   liste.add("deux");

   for(String v : liste) {
     System.out.print(v + " ");
    }

   System.out.println(liste.get(1));

    liste.set(0, "premier");

    }
  }
}
```



cette case est à la fin donc j'aurai ceci dans tab et évidemment « tab.size » me retourne cette fois 4, on voit que mon tableau a grandi d'une case. Voici maintenant un exemple avec quelques manipulations de base que l'on peut classiquement faire sur un tableau dynamique donc nous avons vu que pour pouvoir utiliser le type « ArrayList » il était nécessaire de commencer le fichier par une directive d'importation qui aura pour vocation de rendre accessible le type ArrayList dans le programme. Je peux désormais utiliser le type « ArrayList » pour déclarer une variable et ici je suis en train de déclarer une variable de type tableau dynamique de chaîne de caractères. Nous avons également vu que pour initialiser le tableau à un tableau vide, il faut utiliser ce genre de tournure, donc à la fin de l'exécution de cette instruction j'obtiens en liste la référence à un tableau dynamique vide ; j'exécute ensuite l'instruction suivante j'appelle la méthode « add » qui me permet d'ajouter un élément particulier, donc ici la chaîne de caractères "un" en fin de tableau dynamique, comme le tableau est vide, ça ne fait pas beaucoup de différence, je vais avoir au final un tableau dynamique liste qui aura cette allure. J'exécute l'instruction suivante qui ajoute une chaîne de caractères la chaîne de caractères "deux" en fin de tableau donc ici je vais obtenir le contenu suivant. Si je veux faire afficher le contenu de mon tableau dynamique à ce stade, j'utilise une « boucle for ». Nous avons vu que nous pouvions utiliser par exemple une « boucle for » itérant sur un ensemble de valeurs. Ici ma « boucle for » va prendre en séquence chacun des éléments de la liste et les imprimer en les séparant par un espace. Donc ici l'affichage obtenu au terme de l'exécution de

notes

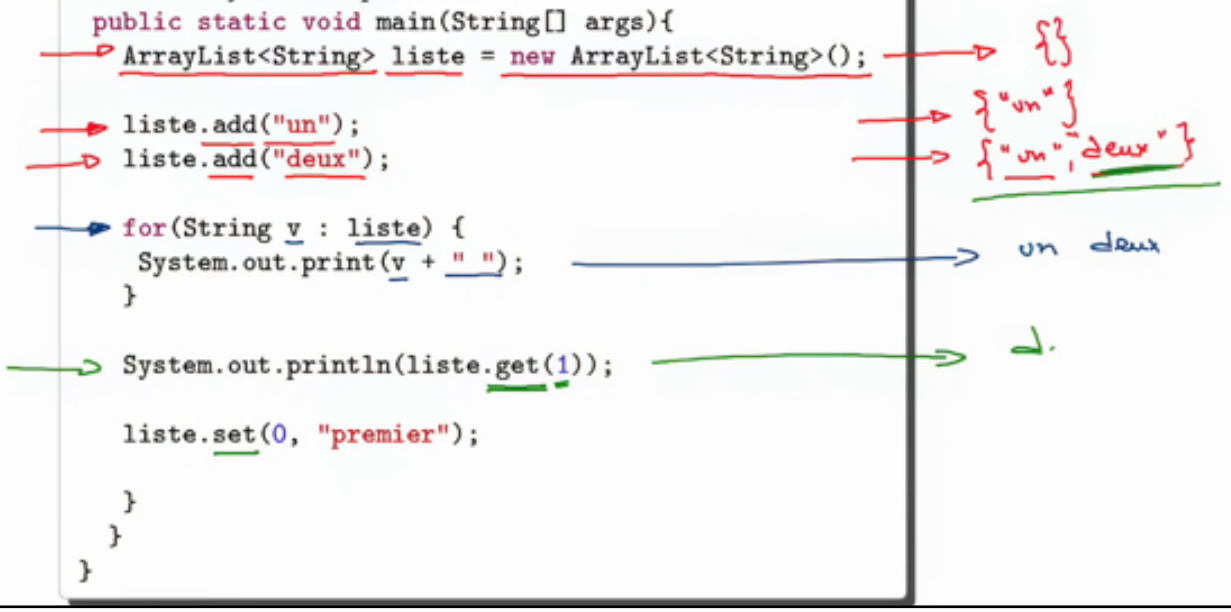
résumé

3m 13s

Exemple de quelques manipulations de base

EPFL

```
import java.util.ArrayList;

class ArrayListExemple {
 public static void main(String[] args){
   ArrayList<String> liste = new ArrayList<String>();

   liste.add("un");
   liste.add("deux");

   for(String v : liste) {
     System.out.print(v + " ");
    }

   System.out.println(liste.get(1));

   liste.set(0, "premier");

   }
  }
}
```



cette instruction va être la chaîne "un" séparée par un espace de la chaîne "deux" qui vont s'afficher de cette manière à l'écran. Si je souhaite accéder à la i-ème case d'un tableau dynamique soit pour consulter son contenu, soit pour le modifier, je devrais utiliser les méthodes « get » et « set ». Donc si j'exécute cette instruction, sachant qu'à ce stade le remplissage du tableau est celui-ci, je vais utiliser la méthode « get » qui va accéder à l'élément d'indice 1 dans le tableau, cet élément d'indice 1 est la chaîne de caractères "deux"

notes

résumé

et à ce moment là je vais afficher la chaîne de caractères "deux". Lorsque j'exécute l'instruction suivante je vais modifier l'élément qui se trouve à l'indice 0 pour faire en sorte que sa nouvelle valeur soit la chaîne de caractères "premier". Donc mon tableau va désormais ressembler à ceci, après l'exécution de cette instruction. L'élément d'indice 0 qui était "un" a désormais été changé en "premier". a désormais été changé en "premier".

notes

résumé

5m 37s