

Support de cours

Cours:

Introduction à la programmation orientée objet (en Java)

Vidéo:

W16-02-exceptionsyntaxe-JAVA-pt1

Concepts (extraits des sous-titres générés automatiquement) :

Objet de type exception. Message d'erreur. Mots du langage. Séquence vidéo précédente. Mot clé throw. Mots tour. Type exception. Avantage de la gestion des exceptions. Api java. Tel objet. Objets particuliers. Quatrième chose. Chaîne de caractères. Nombreuses sous-classes. Gestion des erreurs.



[vers la recherche de séquences vidéo](#)

(dans Introduction à la programmation orientée objet (en Java).)



[vers la vidéo](#)

Center for Digital Education. Plus de matériel de soutien pédagogique ici :

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>

Gestion des exceptions : syntaxe

(Partie 1)

Introduction à la programmation orientée objet (en Java)

Jamila Sam, Jean-Cédric Chappelier et Vincent Lepetit

...

notes

résumé

0m 0s



On cherche à remplir 4 tâches élémentaires :

1. signaler une erreur
2. marquer les endroits réceptifs aux erreurs
3. leur associer (à chaque endroit réceptif) un moyen de gérer les erreurs qui se présentent
4. éventuellement, « faire le ménage » après un bloc réceptif aux erreurs

On a donc 4 mots du langage Java dédiés à la gestion des exceptions :

`throw` : indique l'erreur (i.e. « lance » l'exception)

`try` : indique un bloc réceptif aux erreurs

`catch` : gère les erreurs associées (i.e. les « attrape » pour les traiter)

`finally` : (optionel) indique ce qu'il faut faire après un bloc réceptif

Dans la séquence vidéo précédente, vous avez vu que lorsque l'on veut gérer les exceptions en Java on cherche à faire fondamentalement quatre choses. A signaler les erreurs aux endroits où elles se produisent. A marquer des endroits que l'on veut rendre réceptifs aux erreurs et leur associer, à chaque endroit, un moyen de gérer ces erreurs. Ça c'est vraiment un avantage de la gestion des exceptions. On sépare d'un côté la gestion des erreurs, de l'autre côté l'endroit où elle se produit. Enfin quatrième chose que l'on peut vouloir faire, c'est éventuellement faire le ménage après chaque bloc réceptif aux erreurs. On va donc naturellement avoir quatre mots du langage qui vont être dédiés à chacune de ces tâches élémentaires

notes

résumé

0m 1s



`throw` est l'instruction qui **signale l'erreur** au reste du programme.

Syntaxe : `throw exception`

`exception` est un objet de type `Exception` qui est « lancé » au reste du programme pour être « attrapé »

Exemple :

```
throw new Exception("Quelle erreur !");
```

que l'on souhaite réaliser. Nous allons dans cette vidéo, présenter comment utiliser chacun de ces quatre mots tour à tour En commençant bien sûr par le mot clé `throw` qui permet d'indiquer une erreur. On va dire que `throw` lance une exception. La syntaxe de `throw` pour lancer une erreur, pour signaler une erreur est la suivante, c'est `throw` puis une exception. Qu'est-ce qu'une exception ?

notes

résumé

0m 49s



`throw` est l'instruction qui **signale l'erreur** au reste du programme.

Syntaxe : `throw exception`

`exception` est un objet de type `Exception` qui est « lancé » au reste du programme pour être « attrapé »

Exemple :

```
throw new Exception("Quelle erreur !");
```

`Exception` est une classe de `java.lang` qui possède de nombreuses sous-classes et qui hérite de `Throwable`.

C'est un objet de type `Exception`. Le type `Exception` est un type qui existe dans l'api Java et qui est fourni par la bibliothèque `java.lang`

notes

résumé

1m 13s





qui contient de nombreuses sous-classes que nous allons présenter dans un instant. Donc une exception est un objet de type exception lequel lorsque l'instruction throw est exécutée, est, entre guillemets, lancé. C'est à dire que l'on va avoir un mécanisme qui va créer un tel objet qui pourra être géré par les autres mots clés du mécanisme de gestion des exceptions. C'est ce que signifie "lancer une exception".

notes

résumé

1m 25s



`throw` est l'instruction qui **signale l'erreur** au reste du programme.

Syntaxe : `throw exception`

`exception` est un objet de type `Exception` qui est « lancé » au reste du programme pour être « attrapé »

Exemple :

```
throw new Exception("Quelle erreur !");
```

`Exception` est une classe de `java.lang` qui possède de nombreuses sous-classes et qui hérite de `Throwable`.

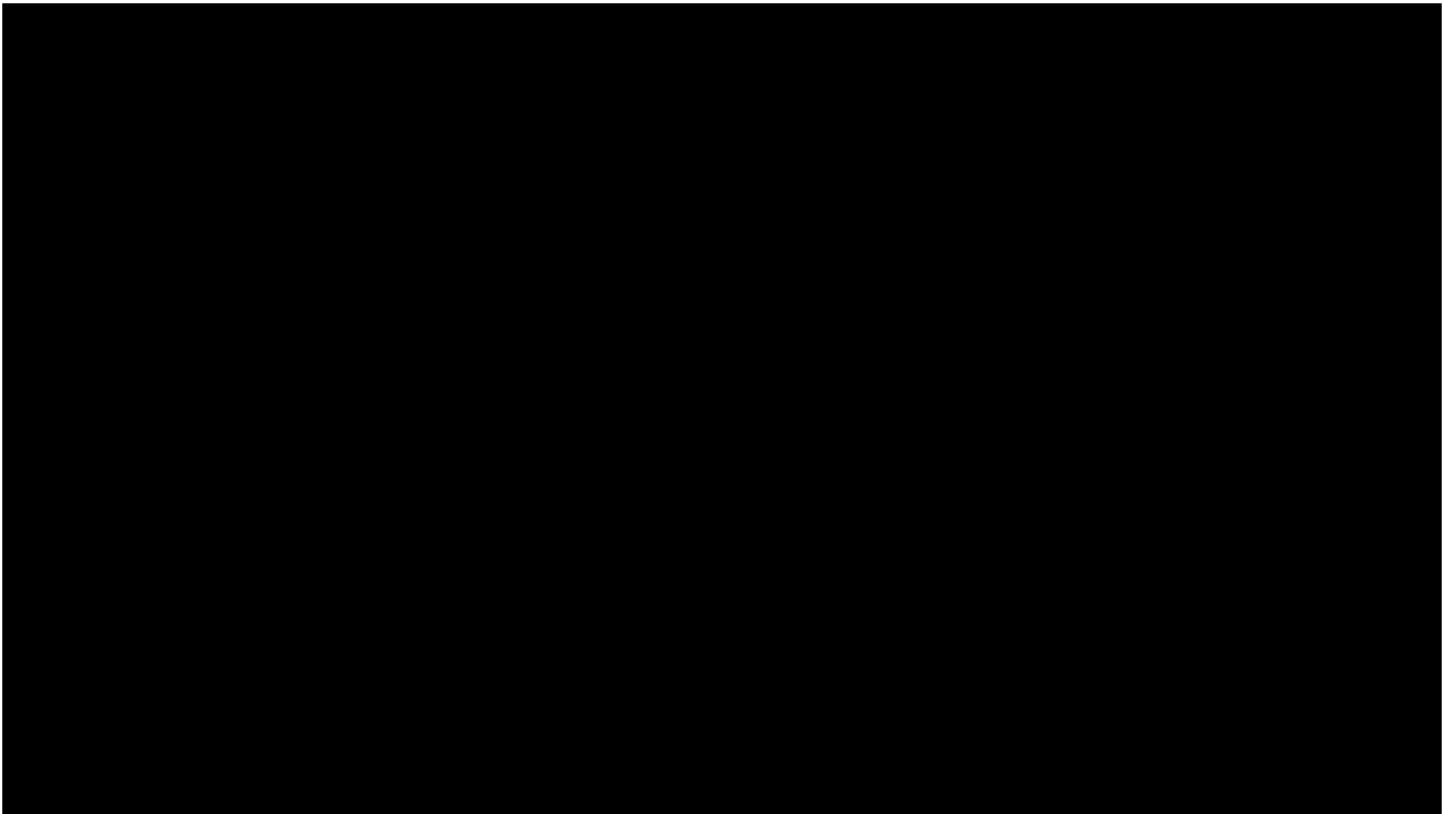
On a un traitement un peu particulier de ces objets particuliers qui sont les erreurs à gérer au niveau du programme. Par exemple, on pourrait tout à fait avoir une ligne comme celle-ci :

notes

résumé

1m 49s





throw new Exception, donc on crée une nouvelle exception. Et puis par exemple, en appelant son constructeur qui prend comme paramètre une chaîne de caractères qui indique un message d'erreur : "Quelle Erreur !", message, par exemple, associé à cette nouvelle instance de la classe Exception. de la classe Exception.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

2m 1s



.....

.....

.....

.....

.....