

Introduction aux exceptions

Temps de lecture : 4 minutes

Pourquoi la gestion d'erreurs ?

Prenons le programme suivant :

```
var chaine = "chaine";  
var entier = Integer.parseInt(chaine);
```

Ce programme compile sans problème. Cependant, à l'exécution, on obtient l'erreur suivante :

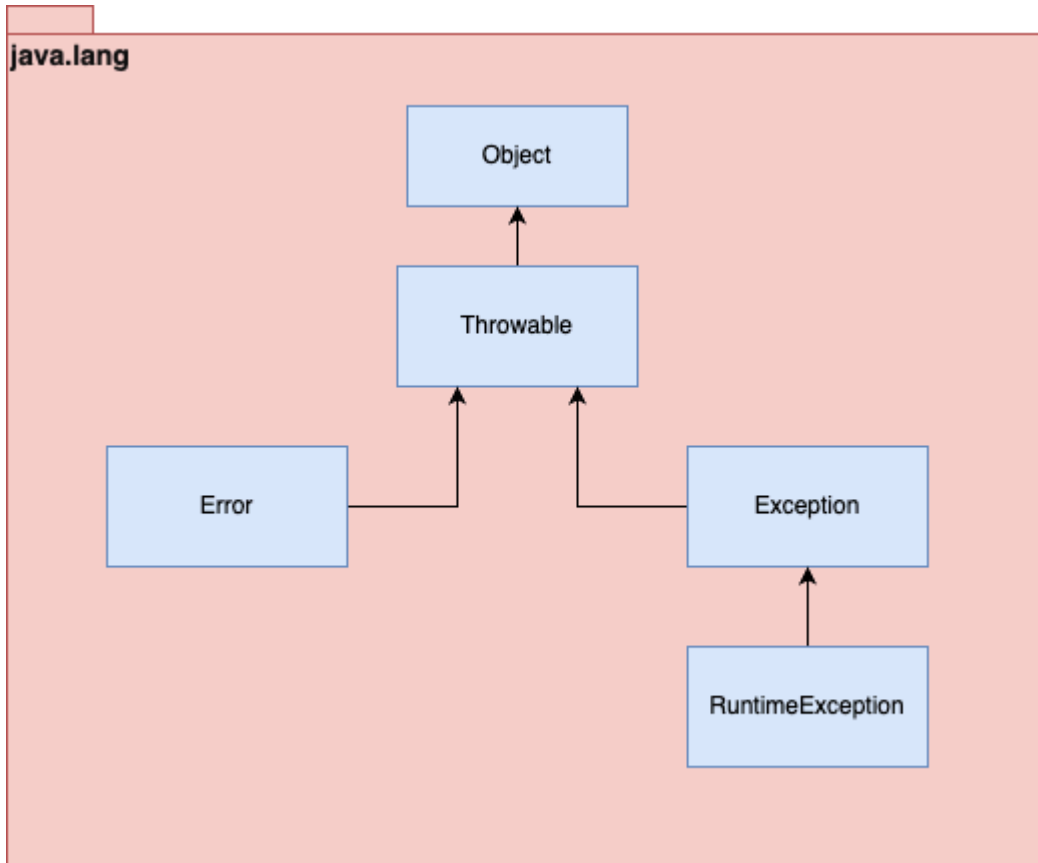
```
Exception in thread "main" java.lang.NumberFormatException: For  
input string: "chaine"
```

C'est ce qu'on appelle une `exception`. Dans notre programme, l'erreur n'est pas gérée, ce qui provoque un bug dans le programme et provoque l'arrêt du programme. Une `exception` non gérée est donc un vrai problème qu'il nous faut apprendre à régler.

Nous allons voir dans ce chapitre comment gérer les erreurs existantes d'une part, mais aussi comment créer nos propres erreurs d'autre part.

Les types d'erreurs en Java

Le diagramme suivant montre comment sont organisées les classes d'erreurs en Java :



Nous avons en réalité trois types d'exceptions :

- La classe `Error` représente une erreur grave intervenue dans la machine virtuelle Java ou dans un sous système Java. L'application Java s'arrête instantanément dès l'apparition d'une exception de la classe `Error`.
 - *Note : Il n'y a pas grand chose à faire pour empêcher ces erreurs dans le code de nos programmes directement. Ce genre d'erreurs peut par exemple arriver quand il n'y a plus de RAM disponible pour faire fonctionner le programme. Ce sont en fait des erreurs techniques.*
- La classe `Exception` représente à l'inverse des erreurs liées à l'exécution du programme, donc liées au code que l'on développe. La particularité de ces exceptions, c'est qu'elles doivent obligatoirement être gérées dans le code, sinon le code ne peut pas compiler. On les appelle checked exception.
 - Nous apprendrons dans la prochaine leçon à gérer les exceptions grâce aux blocs d'instructions `try / catch / finally`.
- La classe `RuntimeException` est similaire à `Exception` (normal puisqu'elle en hérite). Cependant, il n'est pas obligatoire de les gérées dans le code du point de vue de la compilation. C'est pour ça que le code présenté ci-dessus compile mais plante à l'exécution (`NumberFormatException` est une sous-classe de `RuntimeException`). On les appelle unchecked exception.
 - Il est quand même conseillé de gérer ces erreurs, pour éviter de planter au milieu d'un programme.

