



Les exceptions

Qu'est ce qu'une exception ?

- Une erreur dans le programme
 - Due à un bug ou un cas « anormal »
 - Gestion complète en java : JVM
 - Dans d'autre langage, pas toujours facile à obtenir
- Plutôt que de quitter le programme
 - Gérer l'exception
 - Retourner à un état stable
 - Continuer l'exécution

Gestion des erreurs

- En C, ou d'autres langage
 - Pas de gestion d'erreur prévu par le langage
 - Gestion d'erreur à l'appel de chaque fonction
 - Valeur de retour spéciale pour signaler un problème
 - Gestion disséminée de partout
 - Lourd à gérer, souvent des « trous »
- En java
 - Gestion des erreurs au cœur du langage
 - En cas d'erreur, la méthode lève une exception
 - L'appelant (ou encore plus haut) capture l'exception

Syntaxe : throw

- Lever une exception

```
if( null == t )  
    throw new NullPointerException();
```

- Créer une exception avec new
 - Lever n'importe quel objet de type Throwable
 - Deux constructeurs dans toutes exceptions java
- Retourne directement au niveau de gestion des exceptions

Syntaxe : try/catch

- Pour capturer une exception

```
try {  
    // ce que vous voulez  
} catch (NullPointerException e) {  
  
} catch (ArithmeticException e) {  
  
}
```

- Le type d'exception vous renseigne
- Dans le bloc catch, il faut corriger l'erreur
 - Pour permettre de continuer l'exécution
 - Ou sinon on peut quitter le programme

Syntaxe : finally

- A effectuer dans tous les cas
- Permet de stopper certains services

```
try {  
    // ce que vous voulez  
} catch (NullPointerException e) {  
  
} catch (ArithmeticException e) {  
  
} finally {  
    // dans tous les cas  
}
```

Exceptions personnalisées

- Créer une classe dérivant de Exception
 - Par convention avec NomClasseException

```
public class NomClasseException extends Exception {
```

- Spécifier à la déclaration d'une méthode qu'elle peut lancer une exception

```
public void setName(String name) throws NomClasseException {
```

Catch : qu'est ce qui est rattrapé ?

- Un catch capture un objet exception
 - Dont la class dérive de la classe indiquée
 - L'ordre des catch importe donc beaucoup !
 - Pour rattraper toute exception il suffit de faire :

```
catch (Exception e) {  
  
}
```

Exception perdue

- On peut perdre une exception importante
- Si on lève une exception dans un catch
 - On est au courant de ce qu'on fait
- Mais encore plus dans un finally
 - On perd l'exception levée au départ