

Java

CM4-1 : Java, exceptions

Mickaël Martin Nevot

V1.0.0



Cette œuvre de [Mickaël Martin Nevot](#) est mise à disposition selon les termes de la [licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Partage à l'Identique 3.0 non transposé](#).

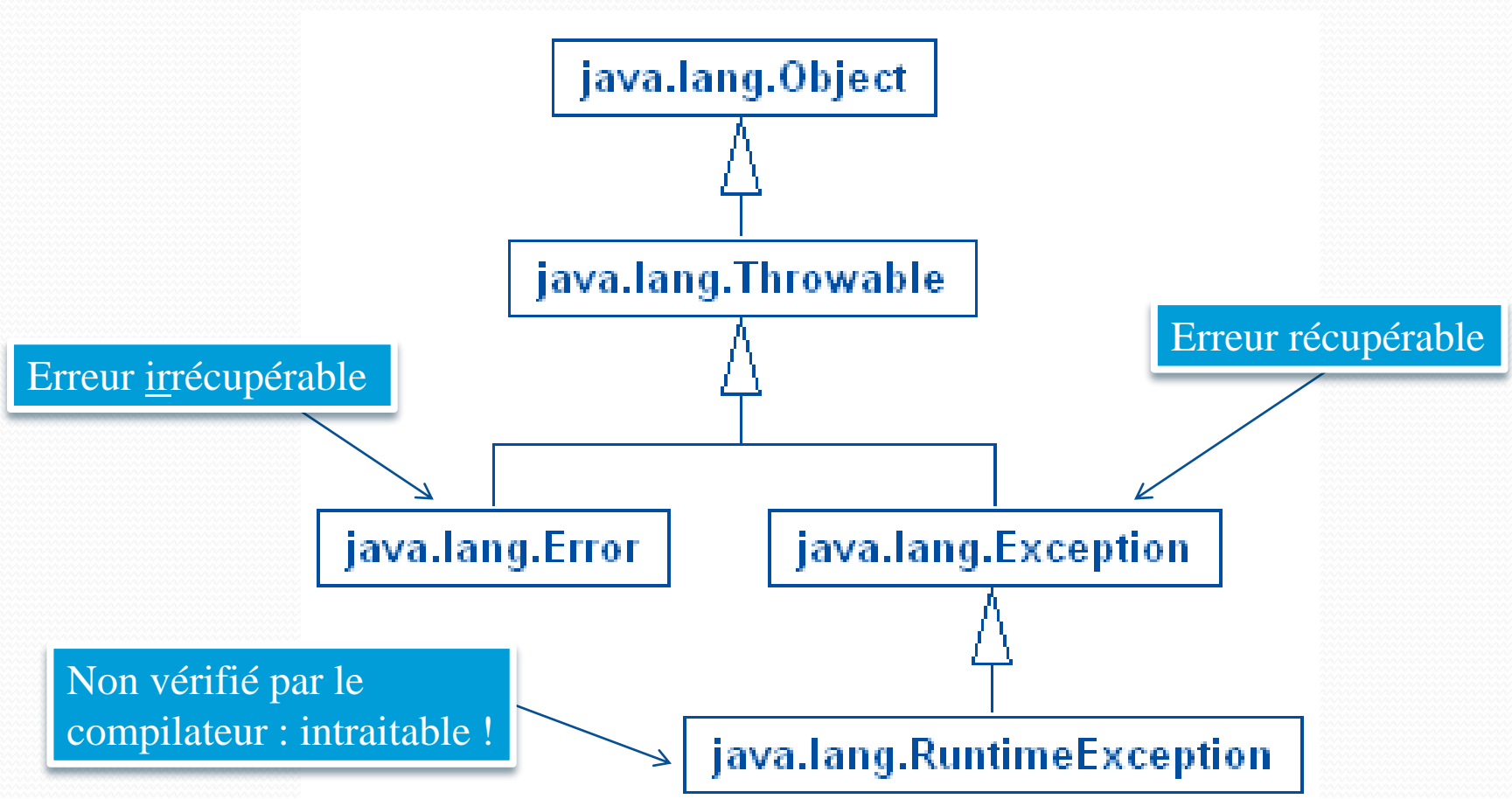
Java

- I. Prés.
- II. POO
- III. Objet
- IV. Java
- V. Types
- VI. Héritage
- VII. Outils
- VIII. **Exceptions**
- IX. Polymorphisme
- X. Thread
- XI. Avancé

Exception

- Objet (sous-classe de `java.lang.Throwable`)
- Signal indiquant un cas exceptionnel :
 - Erreur : irrécupérable (arrêt de l'application)
 - Exception : récupérable (traitable)
- Interrompt le flot d'exécutions normales
- Traitement d'erreur :
 - Séparation du code normal/exceptionnel (lisibilité)
 - Récupération à un autre niveau (propagation dans la pile)
- Si propagée jusqu'en haut de la pile : arrêt de l'application
- `RuntimeException` (implicite) : non traitable

Exceptions : hiérarchie objet



Exceptions : modélisation

- Attributs :

- `String message` ← Contient le message de description de l'exception

- Méthodes :

- `Exception()`
- `Exception(String)` } Constructeur avec ou sans paramètre
- `getMessage() : String` ← Renvoie message
- `printStackTrace()` ← Affiche la liste des appels de méthode ayant conduit à l'exception

Exceptions : mots clefs

- Mot clef `try` : délimitation « d'usabilité » d'exceptions
- Mot clef `catch` : capturer l'exception (traitement)
- Mot clef `finally` : traiter les erreurs non traitées
- Mot clef `throw` : lancer l'exception (signalement)
- Mot clef `throws` : lancement d'exceptions possible

Il n'y a pas de cas d'exceptions implicites

Exceptions : exemple

1

```
// Classe d'exception simple.
```

```
public class MyException extends Exception { ... }
```

2

```
...  
public MyClass() throws MyException {  
    // N'importe quelle méthode peut lancer des  
    // exceptions, y compris un constructeur.  
    ...  
    throw new MyException(val1); // Utilisation optionnelle d'arguments.  
}
```

3

```
...  
public static void main(String[] argv) {  
    try {  
        new MyClass();  
    } catch (MyException e) { // L'ordre des blocs a une importance.  
        e.printStackTrace(); // Équivalent à : System.out.println(e.getMessage()).  
    } finally {  
        ... // On traite ici toutes les exceptions explicites ou implicites  
    } // non traitées jusqu'ici.  
}
```

Crédits

Auteur

Mickaël Martin Nevot

mmartin.nevot@gmail.com



Carte de visite électronique

Relecteurs

Cours en ligne sur : www.mickaël-martin-nevot.com

