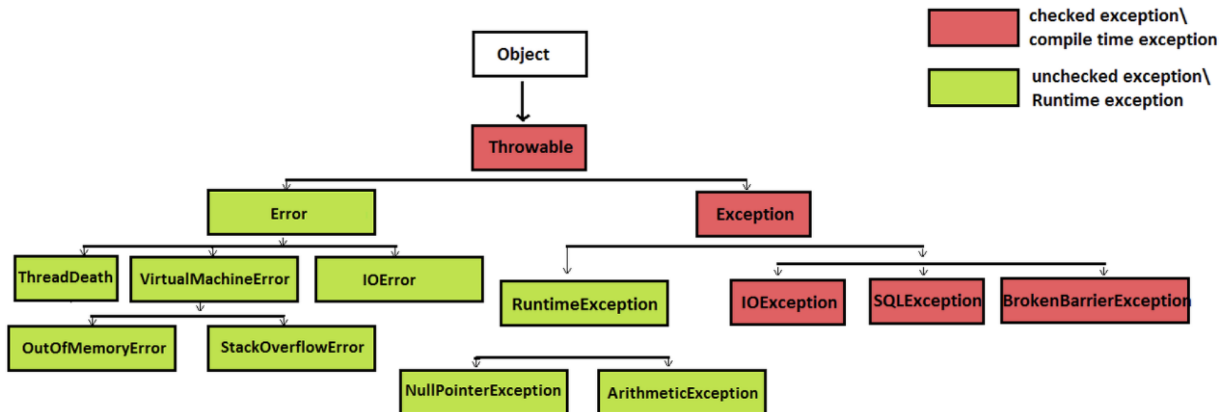


[item74] 메서드가 던지는 모든 예외를 문서화하라



출처 : <https://www.javamadesoeasy.com/2015/05/exception-handling-exception-hierarchy.html>

- 검사 예외가 여러개 발생할 수 있을 때 예외가 발생하는 상황을 각각 따로 분리하여 자바독 @throws 으로 정확히 문서화하라
- 검사 예외는 항상 따로 선언해야 한다. 공통 상위 클래스 하나로 뭉뚱그려 선언하는 일을 삼가자
 - Exception, RuntimeException
 - 클라이언트가 예외를 어떻게 대처해야 할지 모르게 된다.
 - 같은 맥락에서 발생할 여지가 있는 다른 예외들까지 삼켜버릴 수 있다.
 - 단 main() 에서는 Exception 으로 던져도 괜찮다. main() 메서드의 클라이언트는 오직 JVM!
- 비검사 예외도 검사 예외처럼 정성껏 문서화해두면 좋다.
 - 해당 문서화를 보고 클라이언트가 자연스럽게 해당 오류가 나지 않도록 사용하게 된다.

- 사실 public 메서드라면 필요한 전제조건을 명세화해야 한다 (item 56)
그 수단으로 좋은 방법이 바로 비검사 예외들을 문서화하는 것이다.
- 특히 인터페이스 메서드에서 중요하다.
- 검사 예외나 비검사 예외냐에 따라 API 사용자가 해야하는 일이 달라지므로 둘을 확실히 구분하는 것이 좋다.
 - 비검사 예외는 메서드 선언의 throws 목록에 넣지 말자
 - 주석에도 검사 예외와 비검사 예외를 따로 분리해서 문서화 한다.
- 한 클래스에 정의된 많은 메서드가 같은 이유로 같은 예외를 던진다면 메서드 레벨이 아니라 클래스 레벨에서 주석화하는 방법도 있다.
 - NullPointerException 의 경우