



Diferencie entre exceções do tipo checked, runtime e erros

Durante a execução de uma aplicação, erros podem acontecer. A linguagem Java oferece um mecanismo para que o programador possa definir as providências apropriadas a serem tomadas na hora em que um *erro de execução* ocorrer.

Os erros de execução são classificados em algumas categorias. É fundamental que você seja capaz de, dado um erro de execução, determinar seu tipo. A classificação das categorias depende exclusivamente da hierarquia das classes que modelam os erros de execução.

A classe principal dessa hierarquia é a `Throwable`. Qualquer erro de execução é um objeto dessa classe ou de uma que deriva dela.

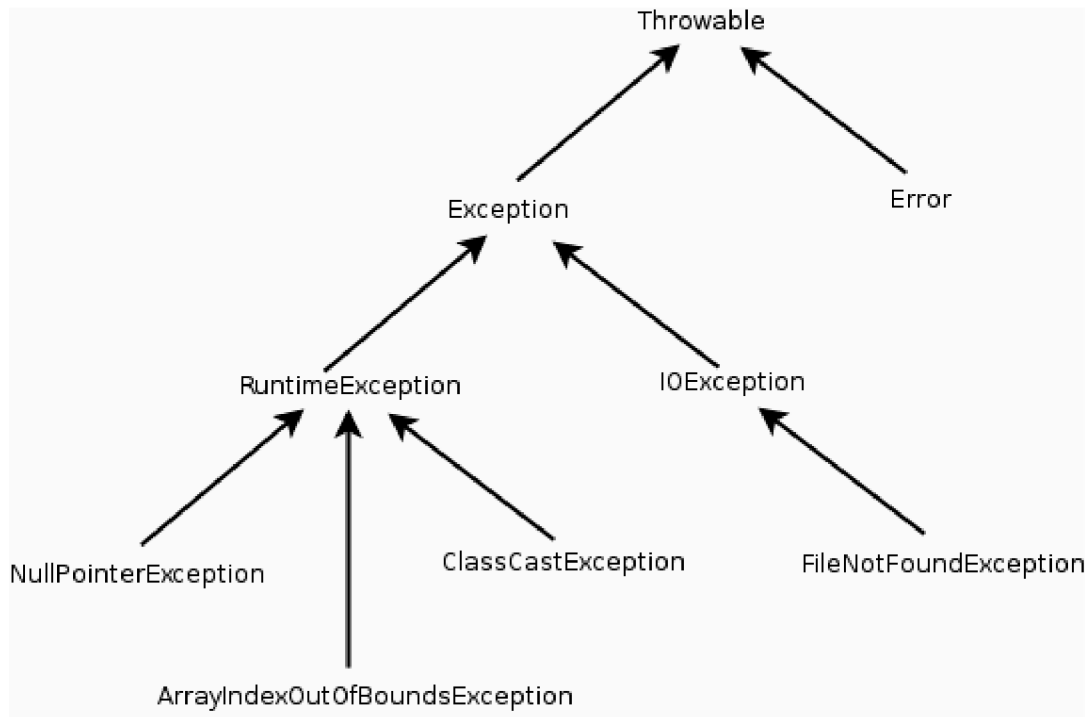
Como filhas diretas de `Throwable` temos: `Error` e `Exception`.

Os `Error`s são erros de execução gerados por uma situação totalmente anormal que não deveria ser prevista pela aplicação. Por exemplo, um `OutOfMemoryError` é gerado quando a JVM não tem mais memória RAM disponível para oferecer para as aplicações. Em geral, esse tipo de erro não é responsabilidade das aplicações pois quem cuida do gerenciamento de memória é a JVM.

Por outro lado, as `Exception`s são erros de execução que são de responsabilidade das aplicações, ou seja, são as aplicações que devem tratar ou evitar esses erros. Por exemplo, um `SQLException` é gerado quando algum erro ocorre na comunicação entre a aplicação e o banco de dados. Esse tipo de erro deve ser tratado ou evitado pela aplicação.

Por sua vez, as `Exception`s são divididas em duas categorias: as **unchecked** e as **checked**. As `unchecked` são exceptions que teoricamente podem ser mais facilmente evitadas pelo próprio programador se ele codificar de maneira mais cuidadosa. As `checked` são exceptions que teoricamente não são fáceis de evitar, de modo que a melhor abordagem é estar sempre preparado para seu acontecimento.

As unchecked são definidas pelas classes que derivam de `RuntimeException`, que por sua vez é filha direta de `Exception`. As outras classes na árvore da `Exception` definem as checked.



Essas diferenças não ficam apenas na teoria. O compilador irá verificar se seu programa pode lançar alguma checked exception e, neste caso, obrigá-lo a tratar essa exception de alguma maneira. No caso das exceptions unchecked, não há nenhuma verificação por parte do compilador pelo tratamento ou não.