

Oficina: Hierarquia de Exceções Java

Autora: Carolina Moraes Quitério - RA: 257041

Palavras-chave: Java

A hierarquia de exceções Java é uma estrutura que organiza as diferentes exceções em grupos, de acordo com suas características e comportamentos. Essa hierarquia é baseada na classe `Throwable`, que é a superclasse de todas as exceções e erros em Java. Existem três principais categorias de exceções na hierarquia de exceções Java: exceções verificadas, exceções não verificadas e erros.

As exceções verificadas são aquelas que são verificadas pelo compilador em tempo de compilação. Se um método lança uma exceção verificada, ele deve declará-la na sua assinatura usando a palavra-chave `throws` ou tratá-la com um bloco `try-catch`. Essas exceções geralmente representam condições inválidas em áreas fora do controle imediato do programa, como problemas com entradas inválidas do usuário, falhas de rede ou arquivos ausentes.

As exceções não verificadas, por outro lado, não são verificadas pelo compilador e representam defeitos no programa (bugs). Elas são subclasses de `RuntimeException` e incluem exceções como `IllegalArgumentException`, `NullPointerException` e `IllegalStateException`. Um método não é obrigado a declarar ou tratar essas exceções. Os erros representam condições irreversíveis que podem ocorrer no programa e interromper sua execução. Eles geralmente são causados pelo ambiente no qual o programa está sendo executado e não podem ser controlados pelos programadores.

A principal diferença entre as exceções verificadas e as não verificadas é que as primeiras devem ser tratadas ou declaradas pelo método que as lança, enquanto as segundas não têm essa obrigação. Isso ocorre porque as exceções verificadas representam condições que podem ser previstas e tratadas pelo programa, enquanto as não verificadas representam defeitos no código que devem ser corrigidos pelo programador. Além dessas três principais categorias, a hierarquia de exceção Java também inclui outras subclasses de `Throwable`, como `IOException`, `SQLException` e `EOFException`. Cada uma dessas subclasses representa um tipo específico de condição inválida que pode ocorrer durante a execução do programa.

É importante notar que a hierarquia de exceções Java é extensível, o que significa que os programadores podem criar suas próprias subclasses de `Throwable` para representar condições específicas em seus programas. Isso permite uma maior flexibilidade na hora de tratar erros e exceções. A hierarquia de exceções Java organiza as diferentes exceções em grupos, de acordo com suas características e comportamentos. As principais categorias são as exceções verificadas, que devem ser tratadas ou declaradas pelo método que as lança; as exceções não verificadas, que representam defeitos no código; e os erros, que representam condições irreversíveis. A principal diferença entre as exceções verificadas e as não verificadas é a obrigatoriedade de tratamento ou declaração das primeiras. Além disso, a hierarquia é extensível, permitindo aos programadores criar suas próprias subclasses para representar condições específicas.