

Exercícios de Fixação

Java - 02

Grupo: A4

Brian

Caio

Carolina

Gabriel

Rafaella Rodrigues - 243597

Tratamento de Exceções em Java

12. Hierarquia de exceção

A hierarquia de exceções em Java descreve a maneira como diferentes tipos de erros e problemas inesperados, que podem ocorrer durante a execução de um programa, são organizados e tratados no código. Dentro dessa estrutura organizada de classes, existem três principais categorias: exceções verificadas, exceções não verificadas e erros. Cada categoria desempenha um papel específico na criação de programas sólidos e seguros.

- **Exceções Verificadas:**

As exceções verificadas, também chamadas de exceções checadas, são aquelas que o programador é obrigado a tratar ou declarar no cabeçalho do método onde podem ser lançadas. Isso ocorre porque essas exceções derivam da classe `Exception`, mas não da subclasse `RuntimeException`, ou seja, têm previsibilidade e podem ser gerenciadas de maneira eficaz.

Alguns exemplos de exceções verificadas incluem `IOException`, que trata erros de entrada e saída, `SQLException`, que lida com problemas relacionados a bancos de dados, e `ClassNotFoundException`, que ocorre quando uma classe não é encontrada. Essas exceções são verificadas porque o desenvolvedor deve tomar medidas para tratá-las ou manipulá-las corretamente, garantindo que o programa possa se recuperar ou informar o usuário sobre a situação.

- **Exceções Não Verificadas:**

As exceções não verificadas, também conhecidas como exceções não checadas, são aquelas que o compilador não obriga o programador a tratar ou declarar no cabeçalho do método. Elas derivam da classe `RuntimeException` e surgem quando há erros de programação ou situações imprevisíveis.

Exemplos comuns de exceções não verificadas incluem `NullPointerException`, que ocorre quando um objeto é acessado quando ele é nulo, `ArithmeticException`, que trata erros de cálculos matemáticos, e `IndexOutOfBoundsException`, que ocorre quando se tenta acessar um índice inválido em uma coleção. Como não são diretamente controláveis, servem como indicadores de problemas lógicos ou programação falha que devem ser corrigidos durante o desenvolvimento.

- **Erros:**

Os erros são situações não tão frequentes e excepcionais que geralmente estão fora do controle do programador, afetando o ambiente de execução como um todo. Eles derivam da classe `Error`. Erros são frequentemente causados por fatores externos de recursos do sistema, como falta de memória (`OutOfMemoryError`) ou falhas na máquina virtual Java (`StackOverflowError`). Como os erros são normalmente irreversíveis e não podem ser tratados de maneira confiável, os desenvolvedores geralmente não precisam se preocupar em capturá-los ou tratá-los.

Compreender a distinção entre exceções verificadas, exceções não verificadas e erros é essencial para construir programas robustos e evitar surpresas indesejadas. As exceções verificadas garantem que problemas antecipáveis sejam tratados, as exceções não verificadas alertam sobre erros de programação e os erros indicam falhas mais sérias. Em resumo, essa hierarquia de exceção oferece uma estrutura sólida para tratar exceções e erros de forma organizada, contribuindo para aplicações mais estáveis e confiáveis.