

Linguagem Java

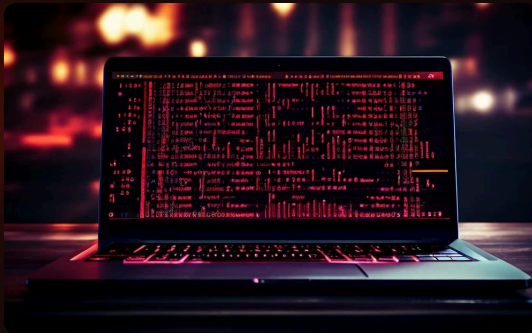
Java é uma linguagem de programação versátil e amplamente adotada, utilizada para desenvolver uma ampla variedade de aplicações, desde jogos e aplicativos móveis até sistemas empresariais complexos. Nesta aula, você aprenderá o que são exceções e como tratá-las para garantir que seu código seja mais robusto e possa lidar com erros de forma eficaz, mantendo a execução dos programas mesmo quando ocorrem situações inesperadas.



by Alexandre de Souza Jr.



O que são exceções em Java?



Exceções em Java

Exceções em Java são eventos inesperados que ocorrem durante a execução de um programa, como erros de divisão por zero, tentativas de acessar um índice inválido de um array, tentativas de abrir um arquivo inexistente, entre outros.



Interrupção do Fluxo Normal

Quando uma exceção é lançada, o fluxo normal de execução do programa é interrompido, evitando que o programa entre em um estado inconsistente ou indesejado.



Tratamento Adequado

É crucial que os desenvolvedores tratem as exceções de maneira adequada, prevenindo erros e garantindo a estabilidade e confiabilidade do software.

Tipos de exceções em Java

Exceções Checked

São exceções que o compilador Java obriga o desenvolvedor a tratar. Exemplos incluem **IOException**, **SQLException** e **ClassNotFoundException**.

Exceções Unchecked

Também conhecidas como exceções Runtime, são exceções que não precisam ser declaradas nem tratadas explicitamente. Exemplos incluem **NullPointerException** e **ArrayIndexOutOfBoundsException**.

Erros

Representam condições de falha graves que normalmente não podem ser recuperadas pelo aplicativo, como falta de memória (**OutOfMemoryError**) ou stack overflow (**StackOverflowError**).

Tratamento de exceções com try-catch



O tratamento de exceções em Java é feito por meio da estrutura **try-catch**. O bloco **try** contém o código que pode gerar uma exceção, enquanto o bloco **catch** define como essa exceção deve ser tratada. Essa abordagem permite que o programa continue a executar, mesmo quando ocorrem erros inesperados.

Bloco finally

O Bloco finally

O bloco **finally** é usado em Java para garantir que um determinado código seja executado, independentemente de ocorrer uma exceção ou não. Ele é adicionado após os blocos **try** e **catch** para assegurar a limpeza de recursos, como fechamento de conexões com bancos de dados ou arquivos.

Execução Garantida

Mesmo que uma exceção seja lançada e capturada pelo bloco **catch**, o bloco **finally** será sempre executado. Isso é útil para garantir que recursos importantes sejam liberados adequadamente, independentemente do resultado da execução do código no bloco **try**.

Múltiplos blocos catch

1

Exceção específica

Trate a exceção de forma precisa.

2

Exceção genérica

Capture exceções inesperadas.

3

Ordem dos blocos

Coloque os blocos **catch** da mais específica para a mais geral.

Ao tratar exceções, é comum precisar lidar com diferentes tipos de exceções que podem ocorrer em um mesmo trecho de código. Para isso, você pode usar múltiplos blocos **catch**, cada um responsável por um tipo específico de exceção. Dessa forma, você pode tomar ações diferentes para cada tipo de problema que possa surgir.

Exceções personalizadas



Flexibilidade

As exceções personalizadas permitem que você crie seus próprios tipos de erro, adaptados às necessidades específicas do seu sistema.



Legibilidade

Ao definir exceções personalizadas, você melhora a legibilidade do seu código, tornando-o mais intuitivo e fácil de entender.



Autonomia

Você tem o controle total sobre o comportamento e a mensagem das suas exceções personalizadas, permitindo uma resposta mais precisa aos problemas.

Propagação de exceções

1 Origem das exceções

As exceções podem ser geradas em qualquer ponto do seu código Java, seja em métodos que você criou ou em bibliotecas externas.

1

2

Propagação de exceções

Quando uma exceção é lançada, ela sobe pela pilha de chamadas de métodos até ser tratada ou chegar ao topo da pilha.

3

Tratamento adequado

É importante tratar as exceções de forma apropriada, evitando que elas cheguem ao topo da pilha e causem a interrupção abrupta do programa.

Boas práticas no tratamento de exceções

1 Identifique exceções esperadas

Antecipe os possíveis erros que podem ocorrer em seu código e trate-os adequadamente com blocos **try-catch**.

2 Evite capturar exceções genéricas

Capture exceções específicas, pois isso permite um tratamento mais preciso e informativo para o usuário.

3 Seja descritivo nas mensagens

Inclua informações relevantes nas mensagens de exceção, como contexto, causas e soluções sugeridas.

4 Documente o tratamento de exceções

Documente claramente as exceções que podem ser lançadas e como elas são tratadas em seu código.

Conclusão

Chegamos ao final desta jornada de aprendizado sobre a linguagem Java. Aprendemos sobre conceitos básicos como variáveis, operadores e estruturas de controle, e tivemos a oportunidade de desenvolver exercícios práticos que reforçaram nosso entendimento e habilidades na programação em Java.

