**Bloco Finally try-catch**

Aprendemos que quando ocorre uma exceção, o Java permite tratar o erro usando a declaração try-catch. Entretanto, existe ainda o bloco finally, que é opcional, mas pode ser útil em certas situações.

O finally é usado para executar um bloco de código independentemente de ocorrer uma exceção ou não, ou seja, ele sempre é executado. Isso pode ser útil quando precisamos executar um código tanto no try, caso não ocorra uma exceção, quanto no catch, caso uma exceção seja lançada. Por exemplo, suponha que você tenha o seguinte código:

try {

metodoQuePodeLancarExcecao();

System.out.println("Executou");

System.out.println("Finalizou!");

} catch (Exception e) {

System.out.println("Deu erro!");

System.out.println("Finalizou!");

}COPIAR CÓDIGO

Perceba no código anterior que a instrução System.out.println("Finalizou!"); deve ser sempre executada, independente de ter acontecido ou exception ou não. Mas o problema é que ela acabou tendo de ser duplicada tanto no try quanto no catch. O bloco finally nos ajuda justamente a evitar essa duplicação de código:

try {

metodoQuePodeLancarExcecao();

System.out.println("Executou");

} catch (Exception e) {

System.out.println("Deu erro!");

} finally {

System.out.println("Finalizou!");

}COPIAR CÓDIGO

Repare que agora a instrução aparece apenas uma vez, dentro do bloco finally, evitando com isso uma duplicação de código desnecessária.

O finally é muito utilizado em situações onde é necessário limpar recursos, fechar conexões de banco de dados ou fechar arquivos que foram abertos no bloco try.