



Tratamento de Erros

Lógica de Programação I

Tratamento de Erro

Quando criamos projetos Java, podem ocorrer erros durante a execução do seu projeto. Nesse momento, você já deve ter se deparado com alguns desses erros que param a execução do nosso projeto e lançam a famosa "Exception".

As **Exceptions** acontecem quando algo não desejado acontece. Mas, não queremos que esse imprevisto pare toda a execução do nosso projeto. Podemos contornar essas situações realizando o tratamento dos locais no código que podem vir a lançar possíveis exceções. Uma forma de fazer isso é utilizando o **try** e **catch**

Tratamento de Erro

```
try{  
    //codigo  
  
}catch(excecao e) {  
    //codigo  
  
}
```

try: Inserimos o trecho de código que queremos executar e que pode lançar uma exceção.

catch: Trecho de código que deve ser executado caso ocorra um erro e a exceção seja capturada.

Tratamento de Erro

Quando utilizamos o `try` e `catch` podemos adicionar o `finally` que é um bloco de código que será sempre executado, independentemente de sua conclusão ter ocorrido normalmente ou ter sido interrompida.

```
try{
    //codigo

}catch(excecao e){
    //codigo

}finally {
    // bloco de codigo
}
```

Exercícios

Exercício em código

Exercício 1: try-catch-finally e while

Implemente um programa que realiza a leitura de um arquivo. Utilize try-catch para lidar com possíveis exceções de leitura do arquivo e garanta que o arquivo seja fechado no bloco finally.

Biblioteca para leitura:

BufferedReader

FileReader

Exercício em código

Exercício 2: Scanner, try-catch-finally e método

Crie um programa que solicite ao usuário para digitar uma frase ou número, realize a leitura e utilize um bloco try-catch para verificar se a string representa um número inteiro válido.

Obrigado