

ESTRITAMENTE CONFIDENCIAL

ELABORATA
INFORMÁTICA

LINGUAGEM JAVA





Tratamento de Exceções

Tratamento de exceções

- Uma **exceção** é algo que ocorre de forma inesperada durante a execução de um programa.

Situações onde ocorrem exceções

- O programa precisa criar um arquivo em disco, mas o diretório onde o arquivo deve ser criado não tem permissão de escrita.
- O servidor do banco de dados travou devido a muitos acessos.
- A rede ficou muito lenta devido ao alto tráfego de dados e o acesso ao banco de dados ficou comprometido.

Tratamento de exceções

- As situações anteriormente citadas quando ocorrem causam uma exceção no programa e se não forem tratadas podem fazer com que o programa pare de funcionar e uma mensagem é exibida ao usuário com textos indecifráveis.

try...catch

- Para evitar que o programa trave ou pare diante de uma exceção, deve-se fazer o tratamento da exceção. O tratamento de exceções na Linguagem Java é realizado através do comando `try..catch`.

Sintaxe try...catch

```
try {  
    <comandos>  
} catch (<tipo-exceção> <nome-do-objeto>) {  
    <comandos>  
}
```

Simulando um caso de exceção

```
...  
int vetor[] = new int[2];  
  
vetor[1] = 10;  
vetor[2] = 0;  
...
```

O código acima quando executado irá gerar uma exceção do tipo `java.lang.ArrayIndexOutOfBoundsException` que significa que um índice fora dos limites do vetor foi utilizado.

Tratando múltiplas exceções

- Algumas vezes pode ocorrer mais de uma exceção num bloco de comandos do programa. Por exemplo, imagine que os dados do vetor do exemplo anterior sejam utilizados para efetuar uma operação de divisão e o valor armazenado na segunda posição do vetor seja zero.

O que acontecerá se o valor da primeira posição for dividido pelo
valor da segunda posição do vetor?

Uma exceção ocorrerá porque o programa estará
dividindo um número por zero.

Exemplo

```
...
int vetor[] = new int[2];

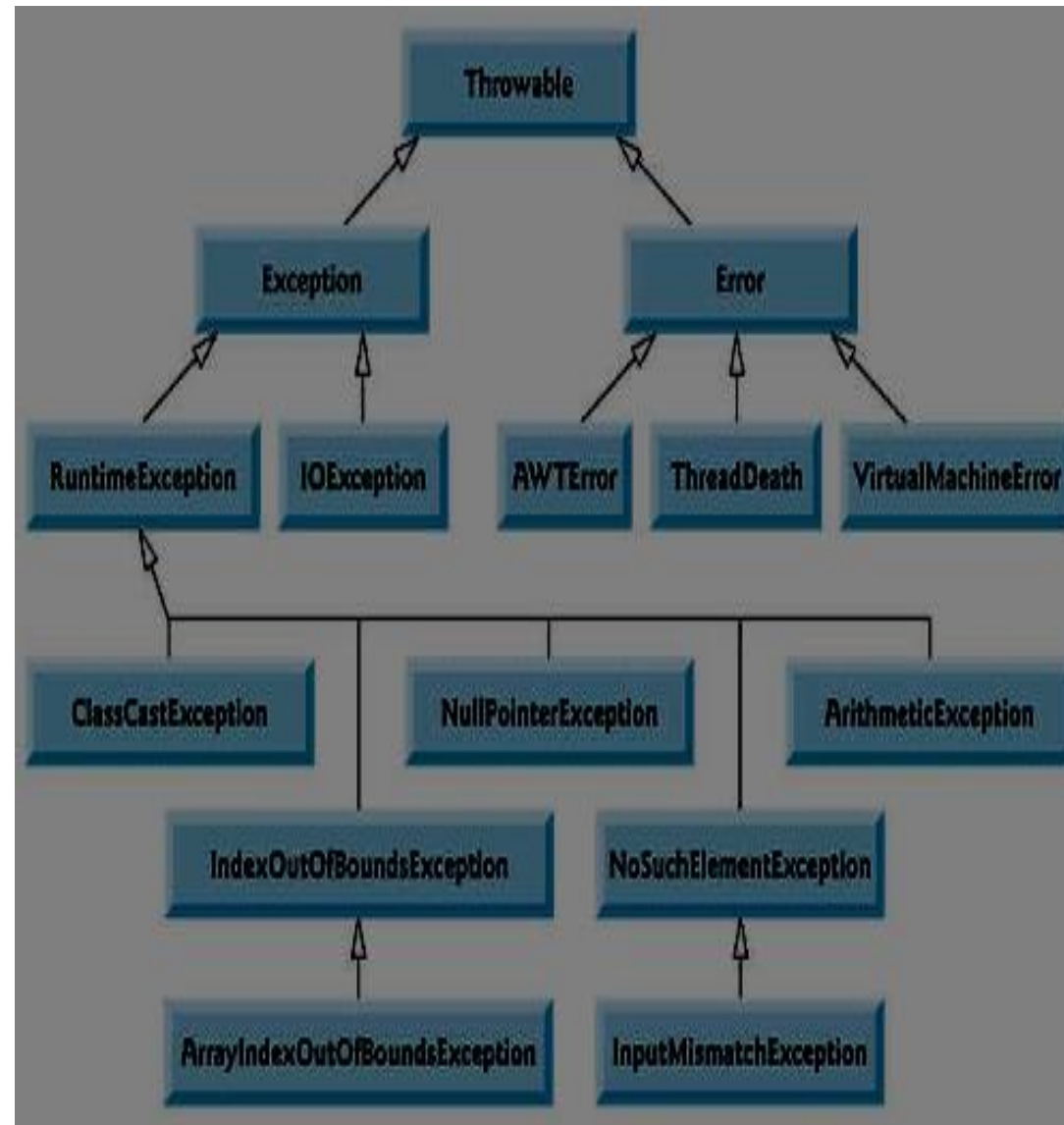
try {
    vetor[0] = 10;
    vetor[1] = 0;

    float divisao = vetor[0] / vetor[1];
} catch (ArrayIndexOutOfBoundsException e) {
    System.out.println("Uma posição inválida do vetor
                        for utilizada.");
    System.out.println("A posição " + e.getMessage() + "
                        não existe no vetor");
} catch (ArithmeticException e) {
    System.out.println("Tentativa de divisão por zero");
}
...
```

Tratando exceções de forma genérica

- Em alguns casos não é necessário tratar especificamente as exceções, podendo capturar as exceções de forma genérica. Para isso deve-se conhecer a hierarquia de classes de exceções da Linguagem Java e decidir qual a melhor classe para utilizar e generalizar o tratamento.

Hierarquia classes de exceções



Exemplo tratamento genérico

```
...
int vetor[] = new int[2];

try {
    vetor[0] = 10;
    vetor[1] = 0;

    float divisao = vetor[0] / vetor[1];
} catch (Exception e) {
    System.out.println("Uma exceção ocorreu: " + e.getMessage());
}
...
```

try...catch...finally

- O comando **finally** no bloco **try..catch** permite que alguma coisa seja executada independente de ocorrer ou não uma exceção. Isto é muito útil quando é preciso liberar algum recurso que foi alocado dentro do try..catch, por exemplo, fechar a classe Scanner, fechar a abertura de um arquivo em disco ou encerrar a conexão com o banco de dados.

Exemplo try...catch...finally

```
...  
Scanner ler = new Scanner(System.in);  
  
try {  
    System.out.println("Digite um número:");  
    int numero = ler.nextInt();  
    System.out.println("Você digitou o número: " + numero);  
} catch (InputMismatchException e) {  
    System.out.println("Você digitou um número inválido.");  
} finally {  
    ler.close();  
}  
...
```


Exercício 01

- Criar um programa que solicite um número ao usuário e depois mostre na tela se o número é par ou ímpar. O programa deve ter tratamento de exceção para impedir que uma exceção seja gerada quando o usuário informar um número inválido, por exemplo, letras no lugar de números.

OBRIGADO!

ESTRITAMENTE CONFIDENCIAL





www.elaborata.com.br

Horário de Atendimento Comercial

Segunda à sexta – das 9:00h às 19:30h e
Sábado - das 8:00h às 15:00h.

Rua Monsenhor Celso, 256 - 1º Andar
Centro - Curitiba - PR

41.3324.0015

 **41.99828.2468**

cursos@elaborata.com.br

