

Programação Web II

Tratamento de Exceções em Java



Curso Superior de Tecnologia em Sistemas para Internet

Professor Substituto Me. Anderson José Lauer

Exceções em Java

Quando são desenvolvidos softwares em Java, existe a possibilidade de ocorrer erros imprevistos durante a execução. Esses erros chamamos de **exceções**.

“Indicação de um problema que ocorre durante a execução de um programa” (DEITEL, 2005).

Exemplos de Exceções

☐ Externos:

- ☐ Tentar abrir um arquivo que não existe;
- ☐ Tentar fazer uma consulta a um banco de dados que não está disponível;
- ☐ Tentar escrever algo em um arquivo que não tem permissão de escrita;
- ☐ Tentar conectar em um servidor inexistente.

Exemplos de Exceções

☐ Lógica:

- ☐ Tentar manipular um objeto que está com valor nulo;
- ☐ Tentar manipular um tipo de dado como se fosse outro;
- ☐ Tentar utilizar um método ou classe não existente;
- ☐ Dividir um número por zero.

Exceções em Java

- ❑ Em Java temos uma classe que representa essas exceções:

```
java.lang.Exception
```

- ❑ Sendo assim, todas as exceções geradas durante a execução do software são tratadas como objetos do tipo **Exception**.
- ❑ O tratamento dessas exceções vai permitir que o software continue em execução.
- ❑ Favorecendo o desenvolvimento de aplicativos tolerante a falhas.

Exceções em Java

❑ Para tratar as exceções em nosso software utilizamos os comandos **try** e **catch**.

try {...}: Neste bloco estão contidas todas as linhas de código que podem vir a apresentar uma exceção.

catch {...}: Neste bloco será codificado o que o software deverá fazer caso uma exceção seja capturada.

```
try {  
    // trecho de código que pode vir a lançar uma exceção  
} catch (Exception e) {  
    // ação a ser tomada com a exceção capturada  
}
```

Exceções em Java

finally {...}: Neste bloco estão contidas as ações que serão executadas mesmo após a captura ou não da exceção.

```
try {  
    // trecho de código que pode vir a lançar uma exceção  
} catch (Exception e) {  
    // ação a ser tomada com a exceção capturada  
} finally{  
    // ação a ser tomada  
}
```

Exceções em Java

throws: O comando throws serve para quando não queremos que a classe ou método faça o tratamento da exceção, mas sim outro que venha lhe chamar.

```
public class TesteString {
    private static void aumentarLetras() throws NullPointerException //lançando exceção
    {
        String frase = null;
        String novaFrase = null;
        novaFrase = frase.toUpperCase();
        System.out.println("Frase antiga: "+frase);
        System.out.println("Frase nova: "+novaFrase);
    }

    public static void main(String args[])
    {
        try
        {
            aumentarLetras();
        }
        catch(NullPointerException e)
        {
            System.out.println("Ocorreu um
            NullPointerException ao executar o método aumentarLetras() "+e);
        }
    }
}
```


Exceções em Java

throw: O comando throw serve para quando queremos lançar uma exceção padrão ao invés de uma específica.

```
1 public class TesteString {
2     private static void aumentarLetras() throws Exception //lançando exceção
3     {
4         String frase = null;
5         String novaFrase = null;
6         try
7         {
8             novaFrase = frase.toUpperCase();
9         }
10        catch(NullPointerException e)
11        {
12            throw new Exception(e);
13        }
14        System.out.println("Frase antiga: "+frase);
15        System.out.println("Frase nova: "+novaFrase);
16    }
17    public static void main(String args[])
18    {
19        try
20        {
21            aumentarLetras();
22        }
23        catch(Exception e)
24        {
25            System.out.println("Ocorreu uma exceção ao
26            executar o método aumentarLetras() "+e);
27        }
28    }
29 }
```

Exceções em Java

Caso nenhuma das exceções atenda a nossa necessidade, podemos criar nossas próprias exceções.

```
public class SemLetraBException extends Exception {  
    @Override  
    public String getMessage(){  
        return "Não existe letra B em sua frase";  
    }  
}
```

```
public class TesteExcecao {  
    public static void main(String args[]) throws SemLetraBException  
    {  
        String frase = "Sou um teste!";  
        if(!frase.contains("b") || !frase.contains("B"))  
            throw new SemLetraBException();  
    }  
}
```

Hierarquia

