

**PROGRAMAÇÃO ORIENTADA A OBJETOS**

**Prof. Milton Palmeira Santana**



## Tratamento de exceções

- » No modelo procedural é comum que o tratamento de erros em tempo de execução de um programa fique restrito a uma entrada incorreta de usuário ou tentativa de gravação em arquivo de leitura apenas.
- » No mundo da orientação a objetos, no entanto, a quantidade e diversidade dos erros que podem ocorrer são muito maiores.
- » O tratamento da exceção serve justamente para que o programa possa continuar sendo executado em vez de ser encerrado repentinamente, o que confere confiabilidade e robustez às aplicações.

## Tratamento de exceções

---

```
import java.util.Scanner;

public class DivisaoPorZeroSemTratamento {
    public static int quociente(int numerador, int denominador) {
        return numerador / denominador;
    }

    public static void main(String args[]) {
        Scanner entrada = new Scanner(System.in);
        System.out.print("Informe o numerador: ");
        int numerador = entrada.nextInt();
        System.out.print("Informe o denominador: ");
        int denominador = entrada.nextInt();
        int resultado = quociente(numerador, denominador);
        System.out.printf("\nResultado: %d / %d = %d", numerador, denominador, resultado);
    }
}
```

## Tratamento de exceções

### » Testes:

- Adicionar zero no denominador
- Adicionar caracteres no denominador

### » Em Java, as exceções são divididas em duas categorias:

- **Unchecked Exception:** significa “exceção não verificada”. Neste tipo de exceção o Java não verifica o código-fonte para determinar se a exceção está sendo capturada. Por isso, o tratamento aqui é opcional. Fazem parte destas exceções de tratamento opcional, por exemplo, a verificação de acesso a um índice inexistente num vetor, a tentativa de se usar um método de um objeto ainda não instanciado e a conversão de um String em inteiro.

## Tratamento de exceções

- » Em Java, as exceções são divididas em duas categorias:
  - **Checked Exception:** significa “exceção verificada”. Neste tipo de exceção o compilador Java obriga o programador a tratá-la. O Java verifica o código-fonte com a finalidade de determinar se a exceção está sendo capturada.

## Tratamento de exceções

- » Quando queremos verificar se houve algum erro durante um determinado código, temos que criar um bloco try-catch para evitar maiores problemas.

```
try {  
    // Código  
}  
catch(Exceção) {  
    // Tratamento do erro  
}  
finally {  
}
```

## Tratamento de exceções

- » A linguagem Java oferece controle para geração e tratamento de muitas exceções.
- » A instrução `throw` serve para forçar a ocorrência de uma determinada exceção. Exceções são objetos e todos os seus tipos devem estender à classe `Throwable` ou a uma de suas subclasses. A classe `Throwable` possui uma cadeia de caracteres que pode servir como descritor da exceção e recuperada com o método `getMessage`.

## Tratamento de exceções

### » Exemplo 1:

```
String frase = null;  
String novaFrase = null;  
novaFrase = frase.toUpperCase();  
System.out.println("Frase antiga: "+frase);  
System.out.println("Frase nova: "+novaFrase);
```



## Tratamento de exceções

### » Exemplo 1:

```
String frase = null;  
String novaFrase = null;  
try  
{  
    novaFrase = frase.toUpperCase();  
}  
catch(NullPointerException e) //CAPTURA DA POSSÍVEL exceção.  
{  
    System.out.println("O frase inicial está nula,  
    para solucionar o problema, foi atribuido um valor default.");  
    frase = "Frase vazia";  
    novaFrase = frase.toUpperCase();  
}  
System.out.println("Frase antiga: "+frase);  
System.out.println("Frase nova: "+novaFrase);
```

## Tratamento de exceções

### » Exemplo 2:

```
try {  
    Scanner s = new Scanner(System.in);  
    System.out.print("Digite a senha: ");  
    String senha = s.nextLine();  
    if(!senha.equals(SENHASECRETA)) {  
        throw new Exception("Senha inválida!!!");  
    }  
    System.out.println("Senha correta!!!\nBem vindo(a)!!!");  
} catch (Exception ex) {  
    System.out.println(ex.getMessage());  
}
```

## Exercícios

- » **1)** Num programa que lê um valor inteiro, faça a sua validação através do tratamento de uma exceção. Caso não seja inteiro continuar pedindo a introdução do valor inteiro. Adicione uma exceção genérica também caso aconteça outros erros.
- » **2)** Liste os tipos de Exceções que podem ser utilizados em JAVA.



Anhanguera