

---



# POO

## Tratamento básico de exceção

Prof. Alcides Calsavara

PUCPR

---



# Conceitos

---



1. Exceção
2. Prevenção de exceções por meio de testes
3. Tratamento de exceção baseado em try-catch
4. Aplicações:
  - a. Consistência na entrada de dados
  - b. Pausa na execução

---



# Exemplos simples de exceções

---



```
int x = 10;  
int y = 0;  
int z = x/y;  
System.out.println(z);
```

O que há de errado neste código?

Qual o resultado da sua execução?

```
int x = 10;  
int y = 0;  
int z = x/y;  
System.out.println(z);
```

O que há de errado neste código?

Divisão por zero.

Qual o resultado da sua execução?

```
int x = 10;  
int y = 0;  
int z = x/y;  
System.out.println(z);
```

O que há de errado neste código?

Divisão por zero.

Qual o resultado da sua execução?

Exception in thread "main" java.lang.ArithmetricException: / by zero

```
int[ ] valor = new int[5];  
valor[1] = 120;  
valor[2] = 180;  
valor[3] = 200;  
valor[4] = 250;  
valor[5] = 300;  
System.out.println("Fim");
```

O que há de errado neste código?

Qual o resultado da sua execução?

```
int[ ] valor = new int[5];  
valor[1] = 120;  
valor[2] = 180;  
valor[3] = 200;  
valor[4] = 250;  
valor[5] = 300;  
System.out.println("Fim");
```

O que há de errado neste código?

Indexação inválida.

Qual o resultado da sua execução?

```
int[ ] valor = new int[5];  
valor[1] = 120;  
valor[2] = 180;  
valor[3] = 200;  
valor[4] = 250;  
valor[5] = 300;  
System.out.println("Fim");
```

O que há de errado neste código?

Indexação inválida.

Qual o resultado da sua execução?

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5

```
String s = "Drone";
char c = s.charAt(5);
System.out.println(c);
```

O que há de errado neste código?

Qual o resultado da sua execução?

```
String s = "Drone";  
char c = s.charAt(5);  
System.out.println(c);
```

O que há de errado neste código?

Indexação inválida.

Qual o resultado da sua execução?

```
String s = "Drone";
char c = s.charAt(5);
System.out.println(c);
```

O que há de errado neste código?

Indexação inválida.

Qual o resultado da sua execução?

Exception in thread "main" java.lang.StringIndexOutOfBoundsException:  
String index out of range: 5

```
String s = null;  
char c = s.charAt(2);  
System.out.println(c);
```

O que há de errado neste código?

Qual o resultado da sua execução?

```
String s = null;  
char c = s.charAt(2);  
System.out.println(c);
```

O que há de errado neste código?

Uso de referência nula para chamada de método.

Qual o resultado da sua execução?

```
String s = null;  
char c = s.charAt(2);  
System.out.println(c);
```

O que há de errado neste código?

Uso de referência nula para chamada de método.

Qual o resultado da sua execução?

Exception in thread "main" java.lang.NullPointerException

---



# Prevenção de exceções por meio de testes

---



```
int x = 10;  
int y = 0;  
int z = x/y;  
System.out.println(z);
```

```
int x = 10;  
int y = 0;  
int z;  
if (y != 0) {  
    z = x / y;  
    System.out.println(z);  
}  
else {  
    System.out.println("Divisao por zero");  
}
```

```
String s = null;  
char c = s.charAt(2);  
System.out.println(c);
```

```
s = null;  
if (s != null) {  
    c = s.charAt(2);  
    System.out.println(c);  
}  
else {  
    System.out.println("Referência nula");  
}
```

---



Nem sempre se  
consegue evitar  
exceções

---



```
import java.util.Scanner;

public class InputMismatch {
    public static void main(String[] args)
    {
        Scanner teclado = new Scanner(System.in);

        System.out.print("Digite um número real: ");

        double x = teclado.nextDouble();

        System.out.println(x);
    }
}
```

Digite um número real: 8,3

8.3

Process finished with exit code 0

Digite um número real: inverno

Exception in thread "main" java.util.InputMismatchException  
at java.base/java.util.Scanner.throwFor(Scanner.java:947)  
at java.base/java.util.Scanner.next(Scanner.java:1602)  
at java.base/java.util.Scanner.nextDouble(Scanner.java:2573)  
at InputMismatch.main(InputMismatch.java:8)

Process finished with exit code 1

---



# Tratamento de exceção baseado em **try-catch**

-- tal que o programa execute até o final --

---



# Forma geral simples do comando try

```
try {  
    Bloco A  
}  
catch( Exception e ) {  
    Bloco B  
}
```

**Bloco A:** código que se deseja executar

**Bloco B:** código que trata qualquer exceção que venha ocorrer no Bloco A

A ocorrência de uma exceção durante a execução do **Bloco A** causa a sua imediata interrupção e desvio da execução para o **Bloco B**.

# Forma geral simples do comando try

```
try {  
    Bloco A  
}  
catch(Exception e) {  
    Bloco B  
}
```

**Exception:** classe da bilbioteca de Java

**e:** referência para o objeto da classe Exception que foi gerado durante a execução do **Bloco A**.

Tal referência pode ser usada no código do **Bloco B**.

Obs: Qualquer identificador pode ser usado no lugar de **e**.

## Exemplo de uso do comando try

```
Scanner teclado = new Scanner(System.in);
try {
    System.out.print("Digite um valor inteiro para o dividendo x: ");
    int x = teclado.nextInt();

    System.out.print("Digite um valor inteiro para o divisor y: ");
    int y = teclado.nextInt();

    int z = x / y;
    System.out.println("Resultado da divisao: " + z);      Bloco A
}
catch (Exception e) {
    System.out.println(e);                                Bloco B
}
System.out.println("Fim do programa");
```

# Comandos do Bloco A que podem gerar exceção

```
Scanner teclado = new Scanner(System.in);
try {
    System.out.print("Digite um valor inteiro para o dividendo x: ")
    int x = teclado.nextInt(); 1

    System.out.print("Digite um valor inteiro para o divisor y: ")
    int y = teclado.nextInt(); 2

    int z = x / y; 3
    System.out.println("Resultado da divisao: " + z);
}
catch (Exception e) {
    System.out.println(e) Bloco B
}
System.out.println("Fim do programa");
```

## Exemplo de execução sem ocorrência de exceção

Digite um valor inteiro para o dividendo x: 10

Digite um valor inteiro para o divisor y: 2

Resultado da divisao: 5

Fim do programa

Process finished with exit code 0

# Exemplo de execução com ocorrência de exceção na operação de leitura de um valor double para x

Digite um valor real para o dividendo x: **outono**

**java.util.InputMismatchException**

Fim do programa

Process finished with exit code 0

```
int x =  
teclado.nextInt();
```

## Exemplo de execução com ocorrência de exceção na operação de leitura de um valor double para y

Digite um valor real para o dividendo x: 10

Digite um valor real para o divisor y: inverno

**java.util.InputMismatchException**

Fim do programa

Process finished with exit code 0

```
int y =  
teclado.nextInt();
```

## Exemplo de execução com ocorrência de exceção na operação de divisão de x por y

Digite um valor real para o dividendo x: **10**

Digite um valor real para o divisor y: **0**

**java.lang.ArithmeticException: / by zero**

Fim do programa

Process finished with exit code 0

**int z = x/y;**

---



# Aplicação: consistência na entrada de dados

---



```
import java.util.Scanner;

public class RaizQuadradaInverso {
    public static void main(String[] args)
    {
        try {
            double x = leia_double("Digite um valor real para extrair raiz quadrada");
            System.out.println("Vamos calcular a raiz quadrada de " + x + " ...");
            double y = Math.sqrt(x); É possível calcular a raiz quadrada de um número negativo?
            System.out.println(String.format("Raiz quadrada: %.3f", y));
            System.out.println("Vamos calcular o inverso de " + x + " ...");
            double z = 1/x; É possível calcular o inverso de zero?
            System.out.println(String.format("Inverso: %.3f", z));
        }
        catch( Exception e ) {
            System.out.println(e);
            System.out.println("Tente novamente!");
        }
    }
}
```

Exemplo de tratamento de exceção  
em leitura de valor real

[continua]

## [continuação]

```
private static double leia_double(String prompt)
{
    Scanner teclado = new Scanner(System.in);
    double valor = 0;
    boolean lido_com_sucesso = false;
    while (!lido_com_sucesso) {
        try {
            System.out.print(prompt);
            valor = teclado.nextDouble();
            lido_com_sucesso = true;
        }
        catch( Exception e ) {
            System.out.println("Entrada inválida! Tente novamente.");
            teclado.nextLine(); // descarta a entrada inválida
        }
    }
    return valor;
}
```

Termina somente depois que um valor real for lido com sucesso.

O buffer de entrada precisa ser limpo para a próxima tentativa de leitura.

## Exemplos de execução sem a ocorrência de exceções:

Digite um valor real para extrair raiz quadrada: **16**

Vamos calcular a raiz quadrada de 16.0 ...

Raiz quadrada: 4,000

Vamos calcular o inverso de 16.0 ...

Inverso: 0,063

Digite um valor real para extrair raiz quadrada: **-2**

Vamos calcular a raiz quadrada de -2.0 ...

Raiz quadrada: **NaN**

Vamos calcular o inverso de -2.0 ...

Inverso: -0,500

**NaN:**  
Not a Number

Digite um valor real para extrair raiz quadrada: **0**

Vamos calcular a raiz quadrada de 0.0 ...

Raiz quadrada: 0,000

Vamos calcular o inverso de 0.0 ...

Inverso: **Infinity**

## Exemplo de execução com a ocorrência de exceção na leitura:

Digite um valor real para extrair raiz quadrada: **abc**  
Entrada inválida! Tente novamente.

Digite um valor real para extrair raiz quadrada: **def**  
Entrada inválida! Tente novamente.

Digite um valor real para extrair raiz quadrada: **4**

Vamos calcular a raiz quadrada de 4.0 ...

Raiz quadrada: 2,000

Vamos calcular o inverso de 4.0 ...

Inverso: 0,250

```
import java.util.Scanner;  
  
public class StackTrace {  
    public static void main(String[] args)  
    {  
        Scanner teclado = new Scanner(System.in);  
        try {  
            System.out.print("Digite a: ");  
            int a = teclado.nextInt();  
            System.out.print("Digite b: ");  
            int b = teclado.nextInt();  
            int z = calcular(a, b);  
            System.out.println("Resultado: " + z);  
        }  
        catch( Exception e ) {  
            System.out.println("Ocorreu algum problema:");  
            System.out.println(e);  
            e.printStackTrace();  
        }  
    }  
}
```

Exemplo de `printStackTrace` em  
tratamento de exceção

[continua]

## [continuação]

```
private static int calcular(int x, int y)
{
    int resultado = 0;
    if (x < y)
        resultado = adicionar(x, y);
    else
        resultado = dividir(x, y);
    return resultado;
}
private static int adicionar(int x, int y) {
    return x + y;
}
private static int dividir(int x, int y) {
    return x / y;
}
```

Decide por adicionar ou por dividir os números.

## Exemplos de execução:

Digite a: 2

Digite b: 7

Resultado: 9

Execução normal: adição

Digite a: 7

Digite b: 2

Resultado: 3

Execução normal: divisão

Digite a: 7

Digite b: 0

Ocorreu algum problema:

java.lang.ArithmetricException: / by zero

java.lang.ArithmetricException: / by zero  
at StackTrace.dividir(StackTrace.java:34)  
at StackTrace.calcular(StackTrace.java:27)  
at StackTrace.main(StackTrace.java:12)

Execução com exceção:  
divisão por zero

Saída gerada pela chamada de  
**printStackTrace**: mostra a pilha  
de execução dos métodos.

Digite a: elefante

Ocorreu algum problema:

```
java.util.InputMismatchException  
java.util.InputMismatchException
```

```
    at java.base/java.util.Scanner.throwFor(Scanner.java:947)  
    at java.base/java.util.Scanner.next(Scanner.java:1602)  
    at java.base/java.util.Scanner.nextInt(Scanner.java:2267)  
    at java.base/java.util.Scanner.nextInt(Scanner.java:2221)  
    at StackTrace.main(StackTrace.java:9)
```

Execução com exceção na leitura de a

Digite a: 7

Digite b: tigre

Ocorreu algum problema:

```
java.util.InputMismatchException  
java.util.InputMismatchException
```

```
    at java.base/java.util.Scanner.throwFor(Scanner.java:947)  
    at java.base/java.util.Scanner.next(Scanner.java:1602)  
    at java.base/java.util.Scanner.nextInt(Scanner.java:2267)  
    at java.base/java.util.Scanner.nextInt(Scanner.java:2221)  
    at StackTrace.main(StackTrace.java:11)
```

Execução com exceção na leitura de b

---



# Aplicação: pausa na execução

---



```
public class Sleep {  
    public static void main(String[] args)  
    {  
        final int ESPERA = 5;  
        try {  
            System.out.println(  
                String.format("Vamos aguardar %d segundo(s) ...",  
                    ESPERA));  
            Thread.sleep( ESPERA * 1000 );  
            System.out.println("\nPronto!");  
        }  
        catch (Exception e) {  
            System.out.println("Falha: pausa interrompida pelo sistema!");  
        }  
    }  
}
```

```
public class Pausa {  
    public static void main(String[] args)  
    {  
        final int ESPERA = 5;  
        try {  
            System.out.println(  
                String.format("Vamos aguardar %d segundo(s) ...",  
                    ESPERA));  
            for (int i = 0; i < ESPERA; i++)  
            {  
                Thread.sleep( 1000 );  
                System.out.print(String.format("%d ", i+1));  
            }  
            System.out.println("\nPronto!");  
        }  
        catch (Exception e) {  
            System.out.println("Falha: pausa interrompida pelo sistema!");  
        }  
    }  
}
```