

Programação Orientada a Objetos – Aula 09

Prof. Dr. Eduardo Takeo Ueda
eduardo.tueda@sp.senac.br

Tratamento de Exceções

- **Robutez** é a capacidade de um software continuar seu processamento mesmo sob condições anormais
- Exemplos de situações anormais: divisão por zero, utilização de objeto não instanciado, acesso a um índice inválido de um array, etc

Tratamento de Exceções

- **Exceção** é um evento não usual, causado por erro ou não, que é detectável por hardware, software ou ambos e que pode necessitar um tratamento especial
- Exceção é uma condição provocada por uma situação excepcional, que requer uma ação específica imediata

Tratamento de Exceções

- Existem linguagens de programação que não possuem mecanismos para tratar exceções, como por exemplo, linguagem C

```
if ( x!=0 )  
    resto = y/x;  
else  
    printf("Erro de divisão por zero!");
```

Tratamento de Exceções

- Você **não deve** fazer isso em Java:

```
public double porcao (double valor, double porCento) {  
    if ((porCento > 100) || (porCento < 0))  
        return -1;  
    return valor * porCento / 100;  
}
```

- Você também não deve usar variáveis globais (estáticas) para representar erros

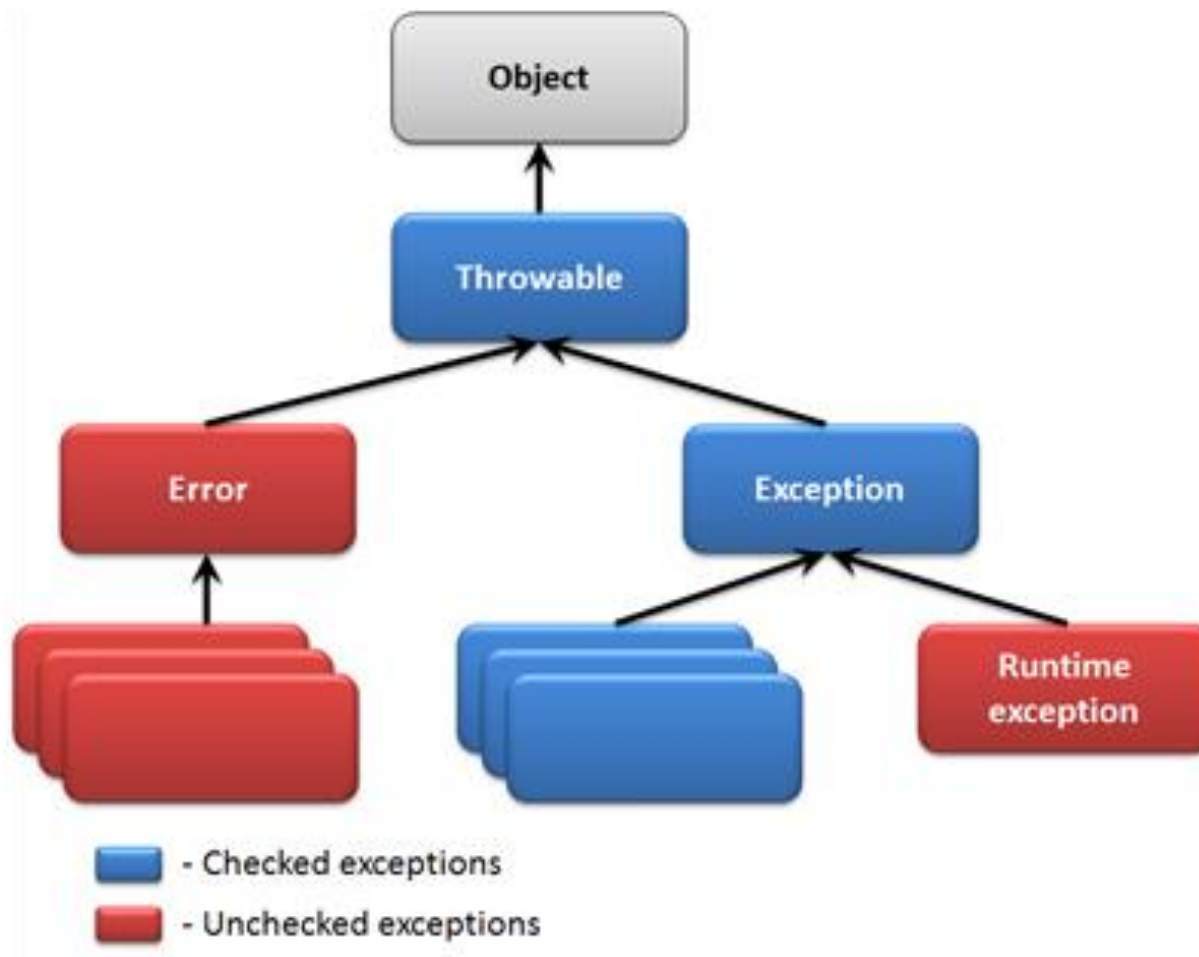
Tratamento de Exceções

- As linguagens orientadas a objetos tipicamente fornecem suporte para o tratamento de exceções
- Para o tratamento de exceções é preciso de:
 - **representar** uma exceção
 - **lançar** uma exceção
 - **tratar** a exceção

Exceções em Java

- A linguagem Java tem suporte para exceções:
 - representação por classes (**Exception, ...**)
 - lançamento pelo comando **throw** (com/sem **throws**)
 - tratamento pela estrutura **try-catch-finally**
- Para uma classe representar uma exceção, ela deve pertencer a uma certa hierarquia de classes

Exceções em Java



Exceções em Java

- ***Checked Exceptions*** (*precisa usar **throws***) são exceções checadas em tempo de compilação que devem ser **obrigatoriamente** tratadas (a menos que explicitamente passadas adiante)
 - **FileNotFoundException**
 - **SQLException**
- ***Unchecked Exceptions*** (*não precisa usar **throws***) são exceções não checadas em tempo de compilação que “**não necessitam**” ser declaradas e nem tratadas
 - **ArithmeticException**
 - **IllegalArgumentException**

throws e throw

```
public class MinhaException extends Exception {  
    // ou seja, posso criar uma exceção  
}
```

```
public void metodo() throws MinhaException {  
    // ...  
    if (erro)  
        throw new MinhaException("erro");  
}
```

- **throws**: o método declara explicitamente a exceção que pode lançar
- **throw**: uma instância da exceção é criada e lançada

try-catch ou try-catch-finally

```
try {  
    // código a ser executado  
} catch (ClasseDeExceção instânciaDaExceção) {  
    // tratamento da exceção  
}
```

```
try {  
    // código a ser executado  
} catch (ClasseDeExceção instânciaDaExceção) {  
    // tratamento da exceção  
} finally {  
    // código a ser executado sempre  
}
```

Descascar ou não o “abacaxi”?

```
public void quemChama() {  
    try {  
        instancia.metodo();  
    } catch (MinhaException e) {  
        // trata a exceção  
    }  
}
```

```
public void quemChama() throws MinhaException {  
    //...  
    instancia.metodo();  
    // passou a exceção para outro método que o invocar  
}
```

Cuidado com o “salto” para o catch

```
antesDoTry();  
try {  
    antesDaExcecao();  
    instancia.metodo();  
    depoisDaExcecao();  
} catch (Exception e) {  
    // trata a exceção  
}  
depoisDoTry();
```



- Se ocorrer uma exceção **depoisDaExcecao()** não é executado, pois não existe volta depois do tratamento no catch

Cuidado com vários blocos catch

```
try {  
    instancia.metodo();  
} catch (IOException e) {  
    // tratamento 1...  
} catch (Exception e) {  
    // tratamento 2...  
}
```

- Apenas um dos catch será executado
- Subclasses devem aparecer primeiro, senão vai ter “*código não alcançável*”

Exemplo 1 de Exceção em Java

```
Programa.java ✖
1 package senac.poo.aula9;
2
3 public class Programa {
4
5     public static void main(String[] args) throws Exception {
6         a(100);
7     }
8
9     public static void a(int i) throws Exception {
10        System.out.println("Executando o método a() com " + i);
11        b(200);
12    }
13
14    public static void b(int i) throws Exception {
15        System.out.println("Executando o método b() com " + i);
16        throw new Exception("Aviso de erro!");
17    }
18 }
```

Exemplo 1 de Exceção em Java

Stack Trace

```
Executando o método a() com 100  
Exception in thread "main" Executando o método b() com 200  
java.lang.Exception: Aviso de erro!  
    at senac.poo.aula9.Programa.b(Programa.java:16)  
    at senac.poo.aula9.Programa.a(Programa.java:11)  
    at senac.poo.aula9.Programa.main(Programa.java:6)
```


Exemplo 1 de Exceção em Java

```
Programa.java ✕
1 package senac.poo.aula9;
2
3 public class Programa {
4
5     public static void main(String[] args) throws Exception {
6         a(100);
7     }
8
9     public static void a(int i) throws Exception {
10        System.out.println("Executando o método a() com " + i);
11        try {
12            b(200);
13        } catch (Exception e) {
14            System.out.println("Tratamento da exceção: " + e.getMessage());
15        }
16    }
17
18    public static void b(int i) throws Exception {
19        System.out.println("Executando o método b() com " + i);
20        throw new Exception("Aviso de erro!");
21    }
22 }
```

Exemplo 1 de Exceção em Java

```
Executando o método a() com 100  
Executando o método b() com 200  
Tratamento da exceção: Aviso de erro!
```

Exemplo 2 de Exceção em Java

ExemploDeExcecao.java

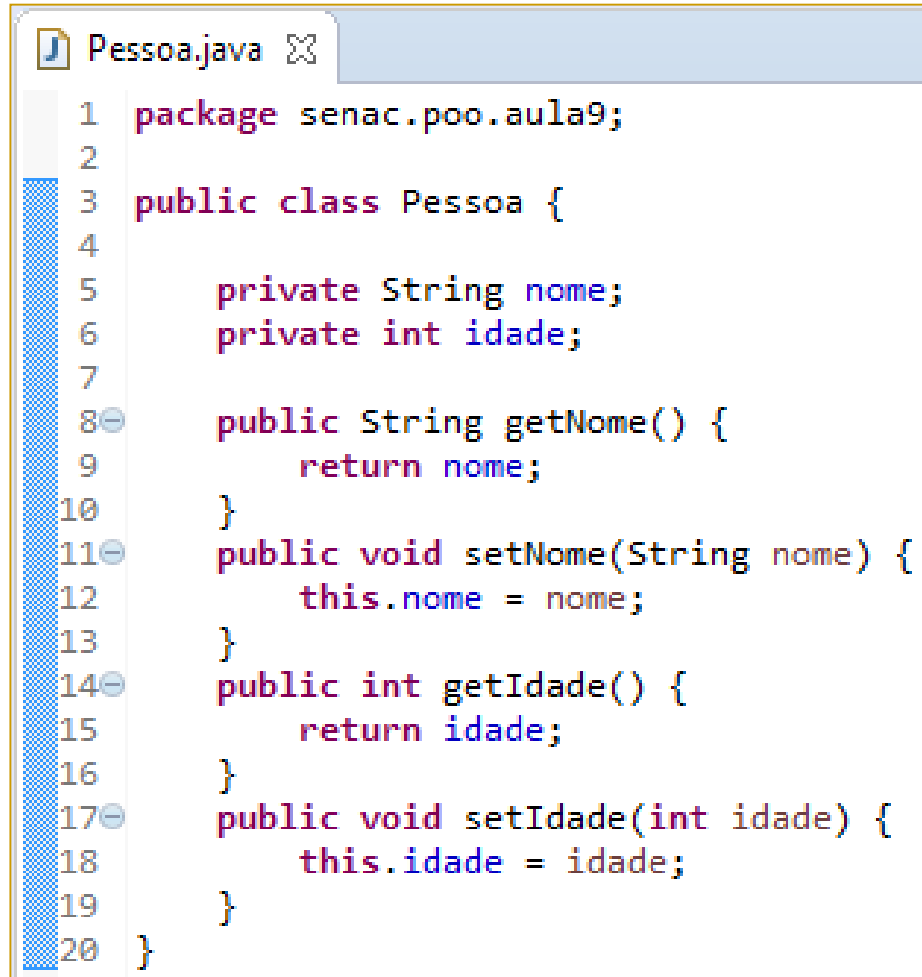
```
1 package senac.poo.aula9;
2
3 public class ExemploDeExcecao {
4
5     public static void main(String[] args) {
6
7         String var = "ABC";
8         try {
9             Integer i = new Integer(var);
10            System.out.println("A variável i vale " + i);
11        } catch (NumberFormatException e) {
12            System.out.println("Não é possível atribuir a string " + var + " a um objeto inteiro.\n"
13                               + "A seguinte mensagem foi retornada: \n\n" + e.getMessage());
14        }
15    }
16 }
```

Exemplo 2 de Exceção em Java

ExemploDeExcecao.java

```
1 package senac.poo.aula9;
2
3 public class ExemploDeExcecao {
4
5     public static void main(String[] args) {
6
7         String var = "ABC";
8         try {
9             Integer i = new Integer(var);
10            System.out.println("A variável i vale " + i);
11        } catch (NumberFormatException e) {
12            System.out.println("Não é possível atribuir a string " + var + " a um objeto inteiro.\n"
13                               + "A seguinte mensagem foi retornada: \n\n" + e.getMessage());
14        } finally {
15            System.out.println("\nExemplo finalizou com sucesso!");
16        }
17    }
18 }
```

Exemplo 3 de Exceção em Java



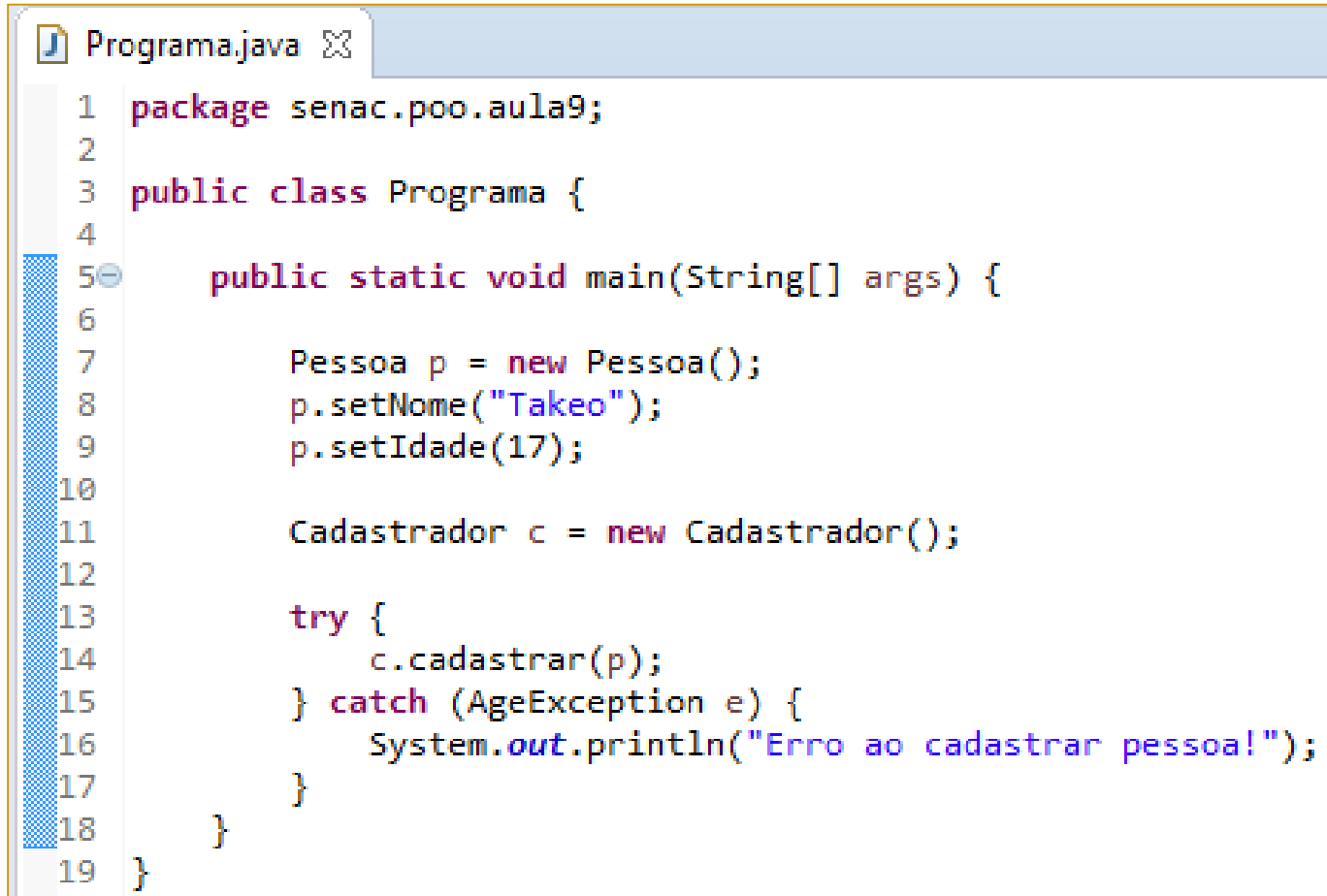
```
Pessoa.java ✕
1 package senac.poo.aula9;
2
3 public class Pessoa {
4
5     private String nome;
6     private int idade;
7
8     public String getNome() {
9         return nome;
10    }
11    public void setNome(String nome) {
12        this.nome = nome;
13    }
14    public int getIdade() {
15        return idade;
16    }
17    public void setIdade(int idade) {
18        this.idade = idade;
19    }
20 }
```

Exemplo 3 de Exceção em Java

```
AgeException.java ✕  
1 package senac.poo.aula9;  
2  
3 public class AgeException extends Exception {  
4  
5     // Criação de Exceção  
6 }
```

```
Cadastrador.java ✕  
1 package senac.poo.aula9;  
2  
3 public class Cadastrador {  
4  
5     public void cadastrar(Pessoa p) throws AgeException {  
6  
7         if(p.getIdade() < 18)  
8             throw new AgeException();  
9         else  
10             System.out.println("Cadastro realizado!");  
11     }  
12 }
```

Exemplo 3 de Exceção em Java



```
Programa.java ✖
1 package senac.poo.aula9;
2
3 public class Programa {
4
5     public static void main(String[] args) {
6
7         Pessoa p = new Pessoa();
8         p.setNome("Takeo");
9         p.setIdade(17);
10
11         Cadastrador c = new Cadastrador();
12
13         try {
14             c.cadastrar(p);
15         } catch (AgeException e) {
16             System.out.println("Erro ao cadastrar pessoa!");
17         }
18     }
19 }
```

Exemplo 4 de Exceção em Java

```
DivisaoPorZero.java ✖
1 package senac.poo.aula9;
2
3 public class DivisaoPorZero {
4
5     public static void main(String[] args) {
6
7         int a = 5;
8         int b = 0;
9
10        System.out.println("Exemplo de divisão por zero:");
11
12        try {
13            System.out.println("a/b = " + a / b);
14            System.out.println("Esta linha não será impressa!");
15        } catch (ArithmeticException e) {
16            System.out.println("Exceção: " + e.getMessage());
17        }
18        System.out.println("Mensagem depois do try-catch");
19    }
20 }
```


Exemplo 5 de Exceção em Java

```
Programa.java ✕
1 package senac.poo.aula9;
2
3 import javax.swing.JOptionPane;
4
5 public class Programa {
6
7     public static void main(String[] args) {
8
9         int numero;
10
11         try {
12             numero = Integer.parseInt(JOptionPane.showInputDialog("Digite um número:"));
13             JOptionPane.showMessageDialog(null, "O número digitado foi " + numero);
14         } catch (Exception erro) {
15             JOptionPane.showMessageDialog(null, "Valor inválido! Erro: " + erro);
16         } finally {
17             JOptionPane.showMessageDialog(null, "Fim!");
18         }
19     }
20 }
```