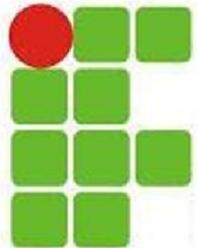


---

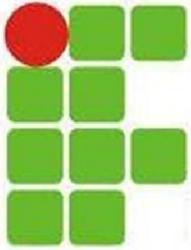
# Programação Orientada a Objetos

**Professor: Diego Oliveira**



**Conteúdo 13:  
Tratamento de Exceções**

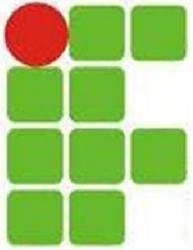




# Exceções

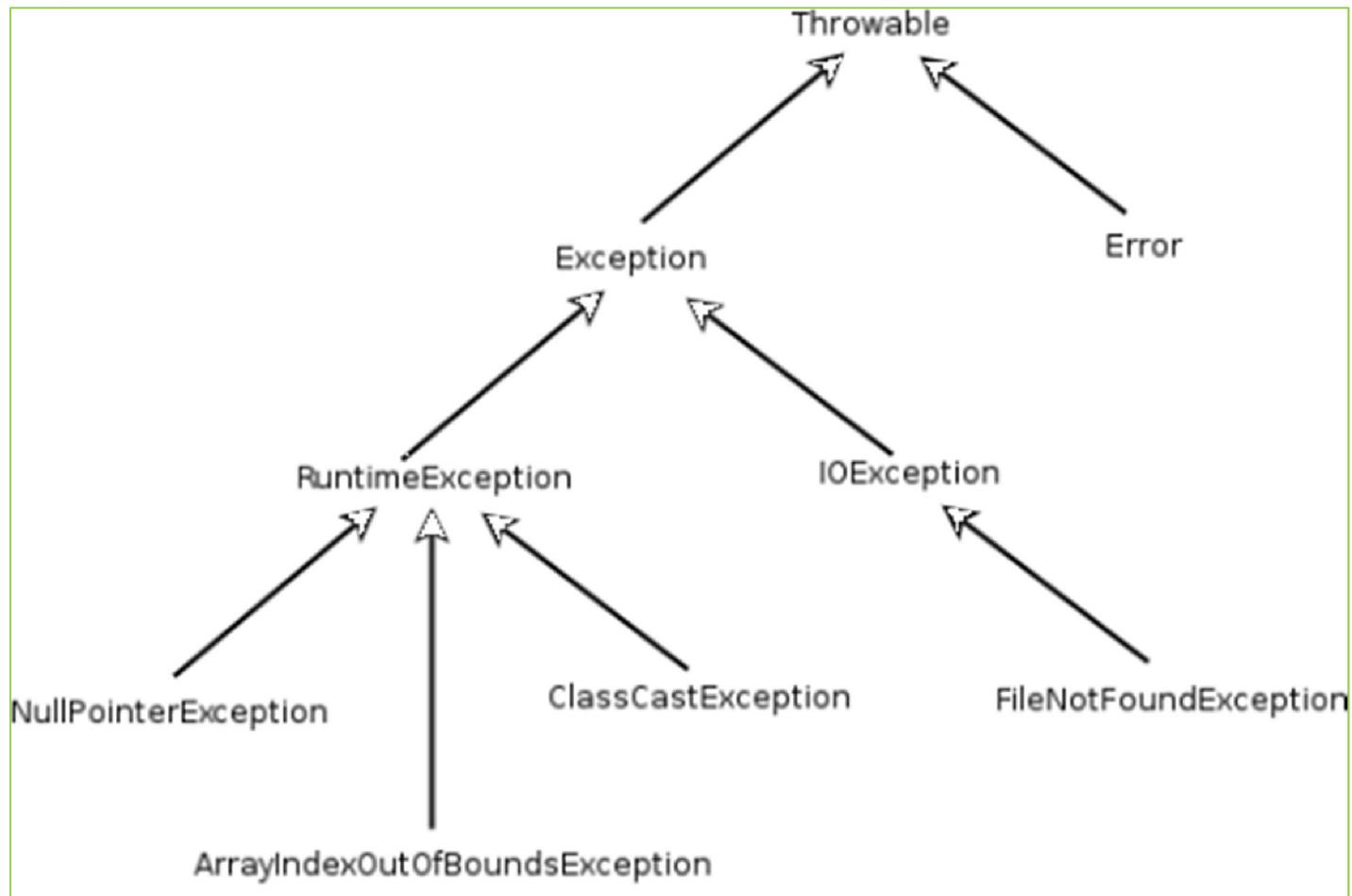
- Uma exceção é uma indicação de um problema que ocorre durante a execução de um programa
- O nome 'exceção' dá a entender que a execução do código fugiu da sua 'regra'
- O Java possui vários tipos de exceções
- Os dois principais tipos de exceções são:
  - RuntimeException
  - IOException

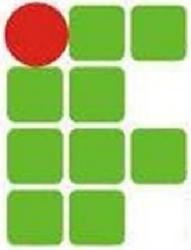




# Exceções

- Vejamos a linha de herança:

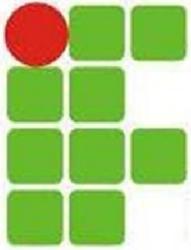




# Exceções

- Principais exceções do Java:
  - NullPointerException
  - ArrayIndexOutOfBoundsException
  - ClassCastException
  - FileNotFoundException
  - ArithmeticException
  - InputMismatchException
  - SQLException





# Tratando Exceções

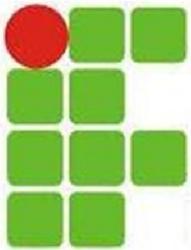
- Quando um trecho de código lança uma exceção, a sua execução é interrompida pelo Java:

```
6 public class TesteExcecoes {
7     public static void main(String[] args) {
8         int x = 1/0;
```

Output - POO-Exercicio13 (run)

```
run:
Exception in thread "main" java.lang.ArithmeticException: / by zero
|       at teste.TesteExcecoes.main(TesteExcecoes.java:8)
Java Result: 1
BUILD SUCCESSFUL (total time: 0 seconds)
```





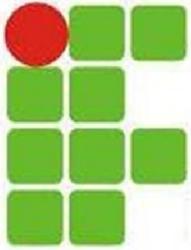
# Tratando Exceções

- Para que isto não aconteça colocamos este trecho de código dentro de um bloco 'try-catch' e tratamos a exceção:

```
7 public static void main(String[] args) {  
8     try{  
9         int x = 1/0;  
10    }catch(ArithmeticException ae){  
11        System.out.println(ae.getMessage());  
12    }  
13    System.out.println("Continuou a execução...");  
}
```

```
Output - POO-Exercicio13 (run)  
run:  
/ by zero  
Continuou a execução...  
BUILD SUCCESSFUL (total time: 4 seconds)
```

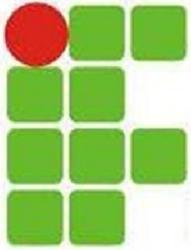




# Lançando Exceções

- Quando, dentro de um método, uma exceção pode ocorrer, podemos lançar a exceção para ser tratada onde este método é chamado, para isto utilizamos a palavra-chave 'throws':

```
5 public class ExemploExcecoes {
6     public void lancaExcecao(int x, int y) throws DivisaoPorUmException{
7         if(y == 1){
8             throw new DivisaoPorUmException("Divisão por um!!!");
9         }else{
10            //continua o código
11        }
12    }
13 }
```



# Lançando Exceções

- Caso este método seja chamado e não tratado, o código irá parar sua execução:

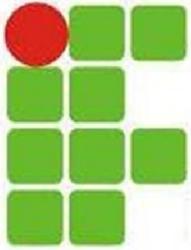
```
7 public class TesteExcecoes {
8     public static void main(String[] args) {
9         ExemploExcecoes ee = new ExemploExcecoes();
10        ee.lancaExcecao(1, 1);

```

```
Output - POO-Exercicio13 (run)
run:
Exception in thread "main" excecoes.DivisaoPorUmException: Divisão por um!!!
    at classes.ExemploExcecoes.lancaExcecao(ExemploExcecoes.java:8)
    at teste.TesteExcecoes.main(TesteExcecoes.java:10)

```





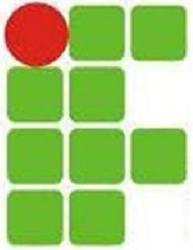
# Criando Exceções

- Nó código abaixo, um novo tipo de exceção é criado, a `DivisaoPorUmException` (que não é um erro, porém é inútil na prática):

```
1 package execoes;  
2  
3 public class DivisaoPorUmException extends RuntimeException{  
4     public DivisaoPorUmException(String s) {  
5         super(s);  
6     }  
7 }
```

Output - POO-Exercicio13 (run)

```
run:  
Exception in thread "main" execoes.DivisaoPorUmException: Você tentou dividir por 1, isto é inútil!  
    at teste.TesteExcecoes.main(TesteExcecoes.java:13)  
Java Result: 1  
BUILD SUCCESSFUL (total time: 3 seconds)
```

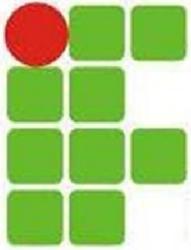


# Utilizando Novas Exceções

- Nó código abaixo, uma variável é verificada, caso seja igual a 1, a exceção criada anteriormente é lançada:

```
String denominador = JOptionPane.showInputDialog(null,  
        "Digite um valor para o denominador", "Denominador?",  
        JOptionPane.QUESTION_MESSAGE);  
  
if (denominador.equals("1")) {  
    throw new DivisaoPorUmException("Você tentou dividir por 1, isto "  
        + "é inútil!");  
} else {  
    //continua o código normalmente|  
}
```

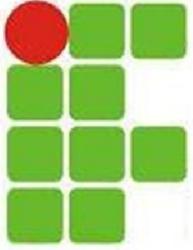




# Exercício

- Crie dois tipos novos de exceção
- Crie duas classes, cada uma com um método que lance uma de suas exceções
- Crie uma classe de teste, na qual serão inicializados dois objetos, um de cada classe criada
- Chame os métodos que lançam exceções através destes objetos
- Trate as exceções com try-catch





# Perguntas?

---

