

# Tratamento de Exceção em Java

Prof. Rafael Guterres Jeffman  
[rafael.jeffman@gmail.com](mailto:rafael.jeffman@gmail.com)

# O que é uma Exceção?

- É um evento que ocorre durante a execução do programa que modifica a ordem normal de execução do programa.
- Quando uma exceção ocorre, o sistema de execução da JVM retorna nos métodos executados até que um método trate a exceção.

# Os Três Tipos de Exceção

- *Checked Exceptions* (java.lang.Exception)
  - São exceções que podem ser antecipadas tratadas. A linguagem Java exige que estas exceções sejam tratadas no código.
- Erros (java.lang.Error)
  - São exceções que não podem ser antecipadas. Em geral são exceções que o programa não pode se recuperar.
- Exceções de Tempo de Execução (java.lang.RuntimeError)
  - São exceções que normalmente indicam um erro no código da aplicação e não devem ser tratadas, nesse caso, o código deve ser corrigido.

# O bloco “try”

- Um bloco “try” marca o código que será executado e que terá um mesmo tratamento de exceções caso uma ocorra.
- Com um bloco “try”, uma exceção não será enviada para outro método.

# O bloco “catch”

- O bloco “catch” define o tratamento de uma exceção para um bloco “try”.
- Podem ser tratadas quantas exceções forem pertinentes para cada bloco “try”.
- A ordem dos blocos “catch” é importante, sendo que as exceções mais específicas devem ficar no início do bloco, e as mais genéricas no final.

# O bloco “finally”

- O bloco “finally” é um bloco opcional que pode ser utilizado em conjunto com um bloco “try”.
- O bloco “finally” **sempre** será executado, independente da ocorrência de uma exceção ou do retorno do método.
- O bloco “finally” só não é executado caso a *thread* seja encerrada abruptamente.

# Java 7

- Capturando mais de uma exceção
  - `catch (IOException | SQLException e) {}`
- O “try-with-resources” (requer `AutoCloseable` ou `Closeable`)
  - `try (FileWriter fw = new FileWriter(filename)) {}`

# Métodos com Exceções

- Quando uma exceção “checked” for lançada dentro de um método, mas ela não for tratada, o método deve declarar que a exceção pode “sair” do método.
- Uma exceção deve ser tratada no local mais próximo possível de onde ocorreu, porém, apenas quando toda informação necessária ao tratamento da exceção está disponível, não antes.
- `public void readFile(String path) throws IOException`



# Lançando Exceções

- Exceções, como tudo em Java, são objetos.
- Para “lançar” um objeto em Java, este deve ser de uma classe derivada de **Throwable**.
- Ao lançar uma exceção, o código é interrompido naquele ponto e volta a ser executado quando encontrar um tratamento apropriado ao objeto lançado.
  - **throw new EmptyStackException();**

# Criando Novas Exceções

- Para criar uma nova exceção, basta criar uma nova classe que derive de **Exception**.
- Não é aconselhável derivar as classes **Error** e **RuntimeException**.
- Por convenção, todo nome de exceção deve terminar com *Exception*.