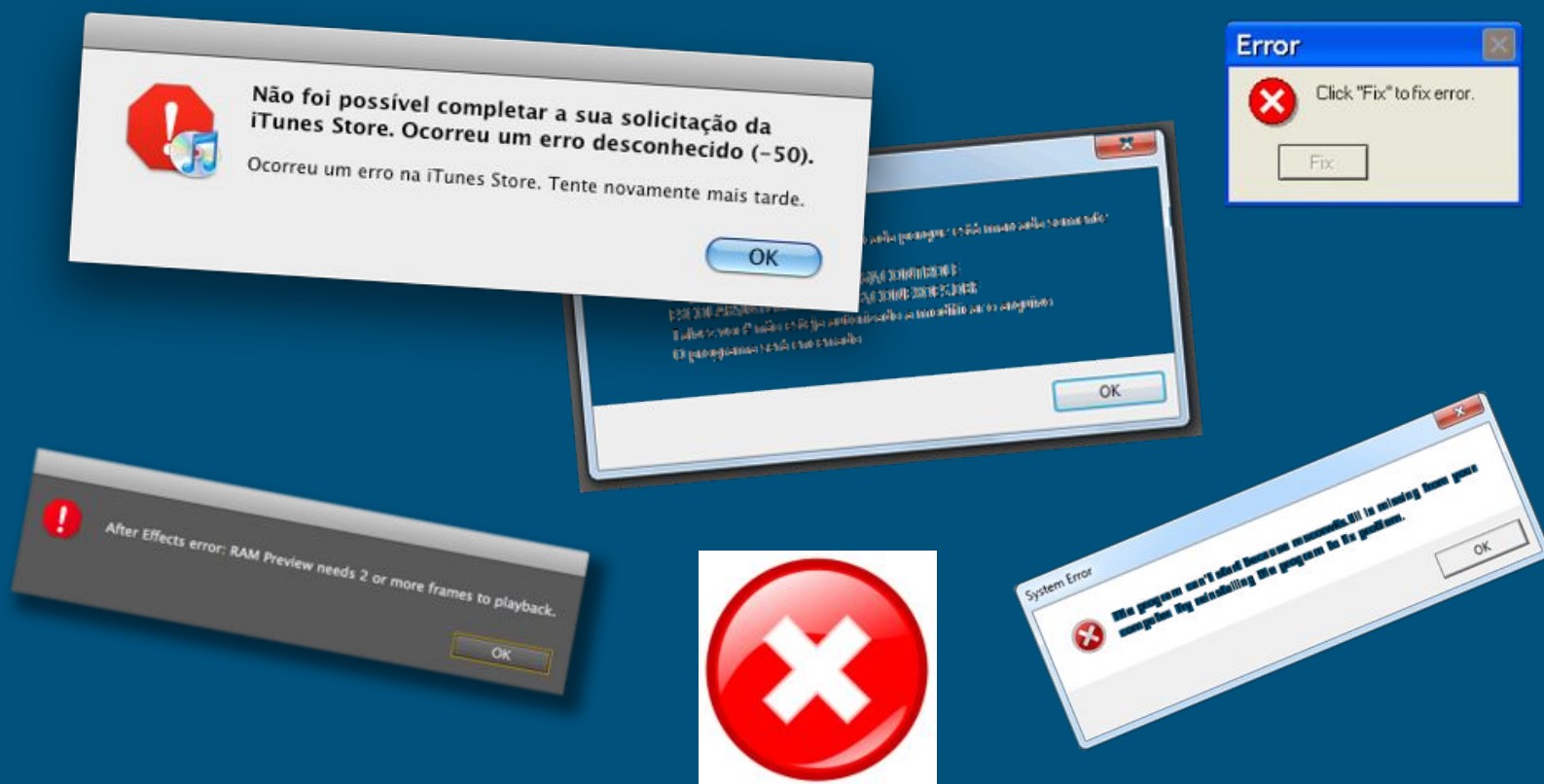


Aula xx

Tratamento de Exceção

emerson@paduan.pro.br



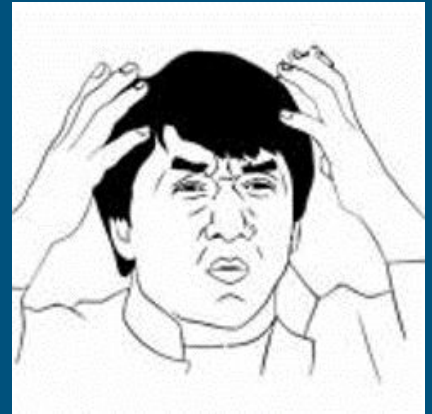
emerson@paduan.pro.br

Exceção

Para toda regra há exceção.

*Essa afirmação se torna
a regra da exceção!*

*Sendo uma regra, deve haver uma
exceção para a regra da exceção ou
a regra da exceção é uma exceção da
regra da exceção?*



emerson@paduan.pro.br

Exceção



Uma exceção é um evento não esperado que ocorre no sistema em tempo de execução (Runtime).

emerson@paduan.pro.br

Exceção



São os mecanismos utilizados pelo Java para o tratamento de erros ou situações indesejadas.

- **Erros de programação:** Acesso a uma posição inválida de um vetor, divisão por zero, invocação de método em uma referência nula.
- **Situações indesejadas:** Uma conexão de rede indisponível durante uma comunicação remota, a ausência de um arquivo procurado localmente.

emerson@paduan.pro.br

Modelo Geral

```
try {  
    //fazer o processamento  
} catch ( <Tipo de Exceção> nome) {  
    //tratar o erro  
}  
finally {  
    //final do processamento (ex. fechar conexões)  
}
```

emerson@paduan.pro.br

Modelo Geral

O bloco **try** tenta processar o código e, se ocorrer uma exceção, a execução do código pula para bloco catch.

O bloco **catch** trata a exceção lançada.

O bloco **finally** finaliza a sequência de comandos do try/catch, independente de ocorrer um erro ou não. Esse bloco é *opcional*.

emerson@paduan.pro.br

Múltiplas Exceções

Para cada bloco try, pode-se ter vários blocos catch para capturar diferentes tipos de exceções.

```
try {  
    // código que pode lançar uma exceção  
} catch(DivideByZeroException e1) {  
    System.out.println(" Divisão por zero");  
} catch(Exception e2) {  
    System.out.println(" Alguma outra exceção");  
    System.out.println(e2.getMessage());  
}
```

emerson@paduan.pro.br

Exemplo - Let's code
