

Programação Orientada a Objetos

1

BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

PROF. LUCIO AGOSTINHO ROCHA

TRATAMENTO DE EXCEÇÕES

1º.SEMESTRE 2023

Tratamento de Exceções

Tratamento de Exceções

3

- **Exceção:**
 - É a ocorrência de um erro infrequente durante a execução do programa:
 - ✦ Ex.: conversão de tipos, erros aritméticos (divisão por zero), tentativa de acesso fora dos limites de um vetor, acesso a classes inexistentes, e outras exceções.
 - Tratamento de exceções:
 - ✦ Evita que o programa seja interrompido abruptamente em um estado inválido.
 - ✦ O programa trata a exceção e continua a executar como se nada tivesse acontecido, ou encerra o programa.

Tratamento de Exceções

4

- Quando utilizar o Tratamento de Exceções:
 - Em regiões do código onde a exceção poderá existir com maior frequência;
 - Em muitos casos, a própria linguagem Java exige a inserção do tratamento de exceção (Ex.: arquivos).
 - Padronização

Tratamento de Exceções

5

- O método que encontra um erro durante a execução dispara (throw) uma exceção.
 - O tratador da exceção processa especificamente esse erro.
- try:
 - Bloco que inclui o trecho de código que potencialmente gerará um erro.
- catch:
 - Bloco que captura o erro e faz o tratamento da exceção.
- finally:
 - Bloco que é sempre executado, com ou sem o erro.

Tratamento de Exceções

6

- Estrutura do bloco try-catch:

```
try {  
  
} catch ( TipoExceção1 exceção1 ) {  
  
} catch ( TipoExceção2 exceção2 ) {  
  
} finally {  
  
}
```

Tratamento de Exceções

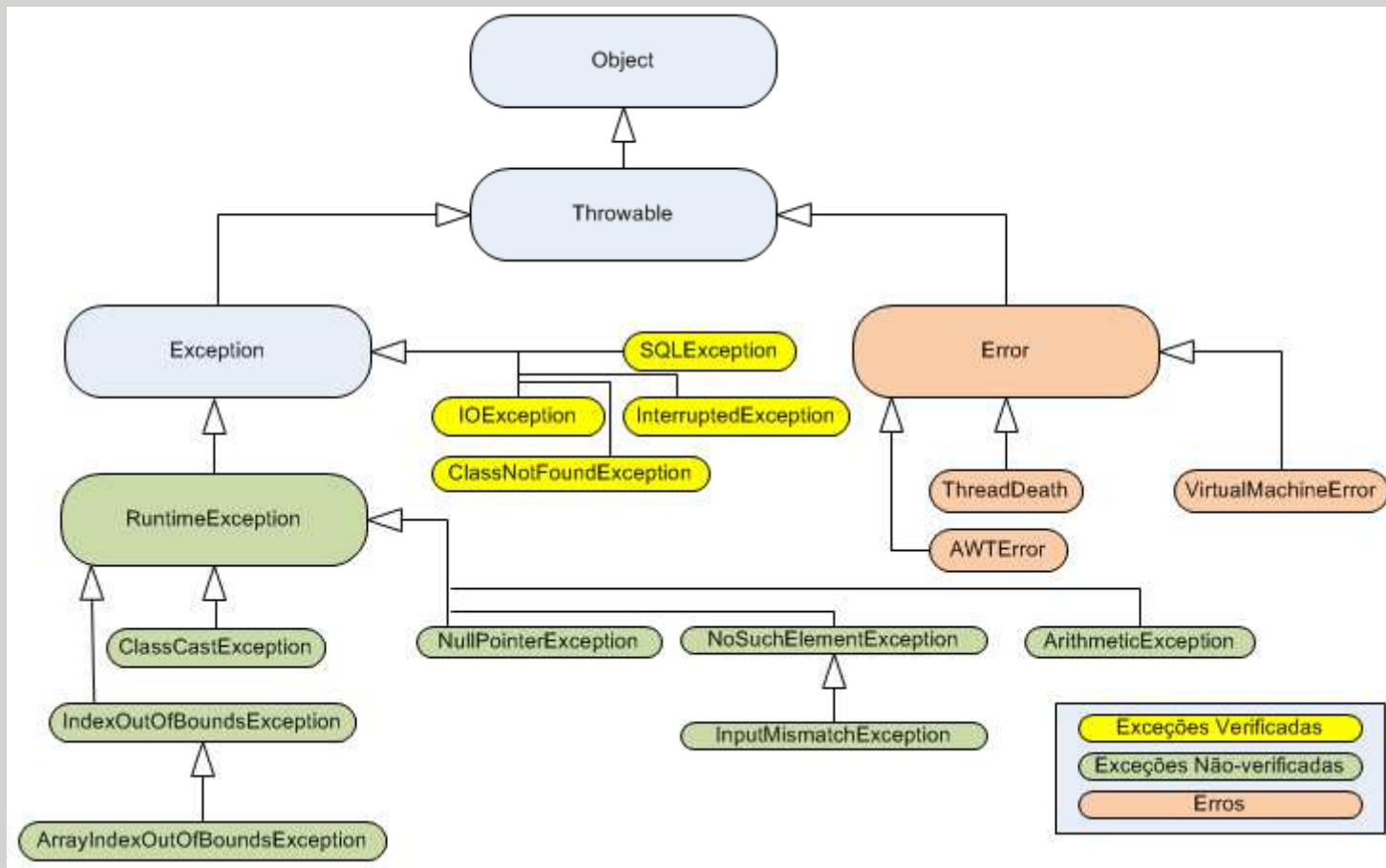
7

- **throws:**
 - É uma declaração utilizada na declaração do método para indicar que a execução do método poderá disparar uma exceção.
 - Atua em qualquer classe derivada de 'Throwable'
 - ✦ **Throwable possui as subclasses:**
 - Exception: erros que podem ser capturados e tratados.
 - Error: erros que não deveriam ser capturados.
- Programa termina se há um erro durante a execução e não houver um tratamento da exceção.
- **throw:**
 - Indica que o método dispara uma exceção.
- Um único bloco 'catch' pode capturar múltiplas exceções.
- Apenas um catch é utilizado caso uma exceção seja capturada.

Tratamento de Exceções

8

- Hierarquia da Classe Throwable (algumas das principais Classes):



Tratamento de Exceções

9

- **Exceções Verificadas:**
 - Todas as subclasses derivadas diretamente da Classe Exception são verificadas.
 - Tratáveis na compilação.
 - É exigido o tratamento na escrita do código pelo compilador.
 - Exceção deve ser capturada (catch) ou disparada (throw)
 - Exemplos: exceções personalizadas, abertura de arquivos, threads, acesso a base de dados, classes não encontradas, sockets, etc.
- **Exceções Não-verificadas:**
 - Subclasses derivadas da Classe RuntimeException.
 - Tratáveis em tempo de execução.
 - Não é exigido o tratamento na escrita do código: a exceção pode acontecer em tempo de execução.

Tratamento de Exceções

10

- Exceções personalizadas:
 - Estendem a Classe Exception ou uma Classe derivada dela.
 - Tornam o detalhamento do erro mais consistente para o usuário.
- 4 (quatro) construtores:
 - Construtor sem argumentos: mensagem de erro padrão.
 - Construtor com uma String: mensagem enviada para a superclasse.
 - Construtor com uma String e uma Throwable (para encadear exceções)
 - Construtor com uma Throwable: enviada para a superclasse.



Revisão

Revisão

12

- Tratamento de Exceções

Exercícios

13

<Ver conteúdo na plataforma de ensino>



Referências

14

- Referências bibliográficas da disciplina.