

Tratamento de Exceções

Leonardo Gresta Paulino Murta
leomurta@ic.uff.br

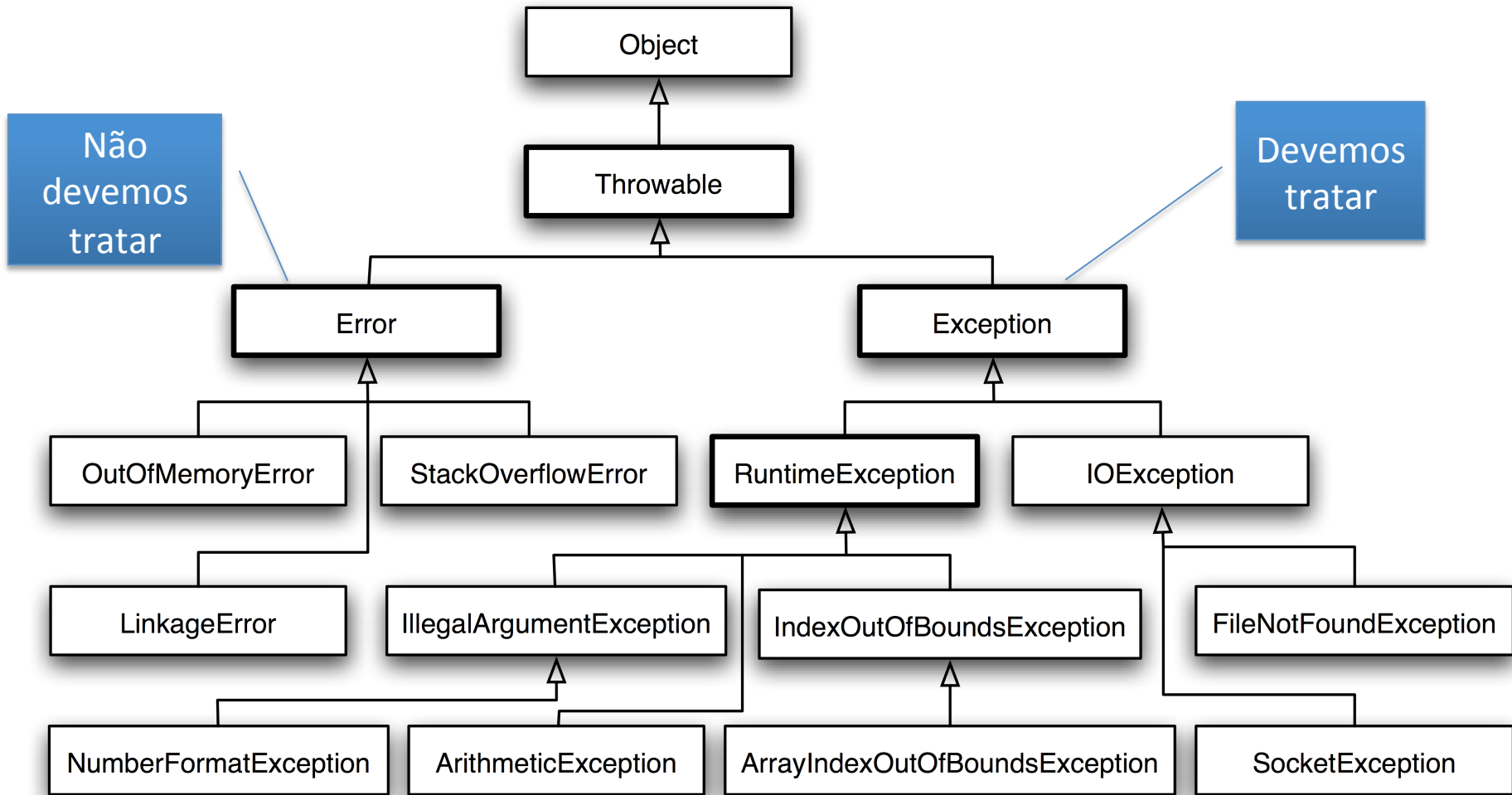
Aula de hoje

- Estudaremos formas de tratamento de exceções
 - try...catch...finally

Exceções

- Conceito
 - Exceções representam situações de erro tratáveis, ocorridas durante a execução de um programa
 - Exemplos de exceções são divisão por zero ou incapacidade de ler dados de um arquivo
- Geradores de exceções
 - Interpretador Java: quando percebe uma situação de erro padrão (divisão por zero, falha de segurança, ...)
 - Métodos do programa: quando percebe uma situação de erro interna do programa (informação inválida, ...)

Hierarquia de Exceção



Ver mais em: <https://docs.oracle.com/javase/8/docs/api/java/lang/Exception.html>

Tipos de Exceção

- Descrição de alguns tipos de exceções:
 - Error: erro genérico (não devemos tratar)
 - Exception: exceção genérica (devemos tratar suas subclasses)
 - RuntimeException: exceção detectada em tempo de execução – não é indicada pela IDE durante a compilação
 - IOException: erros de entrada e saída
 - ArithmeticException: erro de cálculo algébrico (ex.: divisão por zero)
 - NullPointerException: erro de acesso a variável nula
 - SQLException: erro de acesso a banco de dados
- Diversos métodos das bibliotecas Java geram exceções
 - Estas exceções devem ser tratadas pelas classes que utilizem as classes das bibliotecas

Tratamento de Exceções

- A palavra reservada *throws*, seguida pela classe de exceção gerada, deve ser indicada no cabeçalho de um método que gere uma exceção
- Os comandos *try-catch-finally* executam um código que pode gerar exceções de maneira segura, realizando o tratamento das exceções

```
public int gravaRegistro () throws IOException {
    ... // Código que gera a exceção
}
```

Tratamento de Exceções

```

try {
    // Código que pode disparar exceções
} catch (Excecao1 e) {
    // Código executado caso o código no bloco try
    // dispare uma exceção tipo Excecao1
}
...
catch (ExcecaoN e) {
    // Código executado caso o código no bloco try
    // dispare uma exceção tipo ExcecaoN
} finally {
    // Código executado sempre, mesmo que tenha ocorrido
    // uma exceção no bloco try
}
  
```

Tratamento de Exceções

- A ocorrência de uma exceção transfere o fluxo de execução para o primeiro catch que trate a exceção
- Após o tratamento da exceção no catch, o fluxo é transferido para o finally do mesmo grupo try-catch-finally
- O finally é executado sempre, tendo ou não exceção, sendo útil para fechar arquivo, transação, etc.

Propagação de Exceções

```

getContent() {
  try {
    openConnection();
    readData();
  }
  catch (IOException e) {
    // Trata erro de E/S
  }
  ...
}

```

```

openConnection() throws IOException{
  openSocket();
  sendRequest();
  receiveResponse();
}

```

```

sendRequest() throws IOException{
  write(header);
  write(body); // Write Error!
}

```

Exercício

- Em quais situações o programa da calculadora pode lançar exceções?
 - Caso o usuário entre com valor não numérico
 - Caso o usuário entre com zero no caso da divisão
- Teste as duas situações
- Reescreva o programa fazendo uso de tratamento de exceção para cada uma dessas situações

Tratamento de Exceções

Leonardo Gresta Paulino Murta
leomurta@ic.uff.br