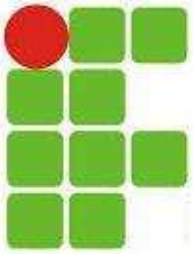
Prof. Diego Oliveira

BD



Tratamento de Exceções

Tratamento de Exceções

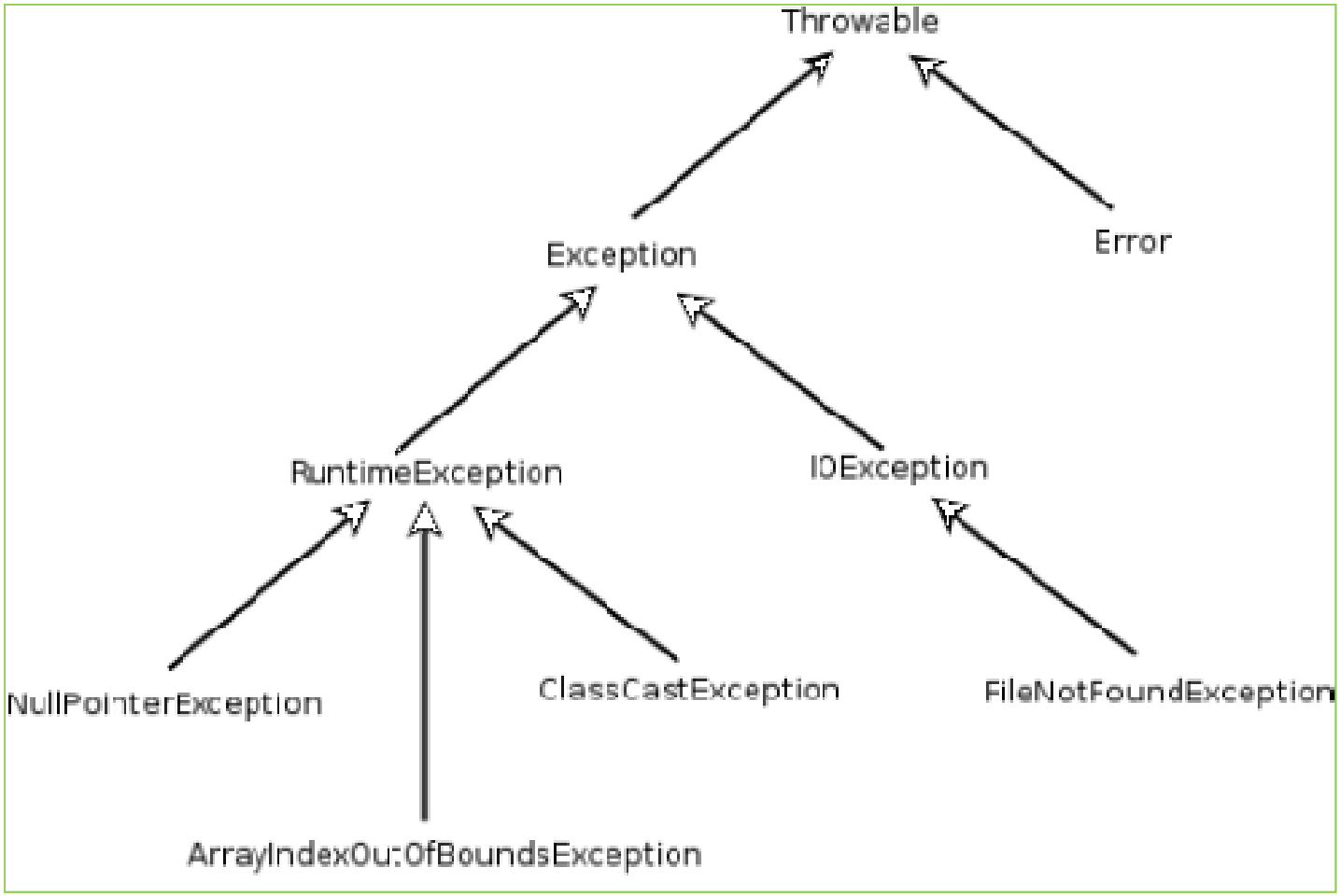
- Uma exceção é uma situação anormal que acontece durante a execução de um programa
- Exceções podem ser causadas por:
  - Erro de hardware
  - Erro de programação (lógica)
  - Erro de aritimética (divisão por zero)
  - Erro no acesso a arquivos
  - Falhas de memória
  - Valores incompatíveis com tipos de variáveis

Tratamento de Exceções

- Caso as exceções não sejam tratadas podem parar completamente o programa
- Para evitar esse travamento total fazemos o tratamento de possíveis exceções
- Quando as exceções são todas tratadas/previstas o programa se torna mais seguro e confiável
- Sistemas críticos devem obrigatoriamente tratar as exceções (hospitais, aviões, etc.)

Tratamento de Exceções

- Linha de herança das exceções:



Tratamento de Exceções

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

Tratamento de 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)
```

Tratamento de 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" execoes.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
}
```

Tratamento de Exceções

- Na parte de Banco de Dados as principais exceções são:
  - SQLException
  - RuntimeException
  - ClassNotFoundException
  - ConnectionFactoryException
  - DAOException

Atividade

- Na classe Banco.java crie os métodos para Gerenciar Aluno (insert, update, delete, remove) conforme aula 04 – Conexão
- Para cada método capture as possíveis exceções (que você criará e lançará)
- No construtor capture as possíveis exceções
- Mostre uma mensagem amigável caso ocorra alguma exceção e informe qual foi a exceção que ocorreu e como resolvê-la
- Salve tudo em um LOG no banco de dados

Perguntas?