



Universidade Presbiteriana  
**Mackenzie**

**150** anos  
1870 - 2020



# Tipos de Exceções

*Programação de Sistemas II - 1º semestre de 2022*

Prof. Joaquim Pessoa Filho

Prof. Tomaz Mikio Sasaki



# Aula passada

---

*NumberFormatException*

*ArrayIndexOutOfBoundsException*

*try-catch-finally*

# Exceções do tipo *Unchecked*

---

1

São chamadas de exceções de tempo de execução (*Runtime Exception*).

2

O compilador não verifica se há um tratamento para a exceção.

3

*NumberFormatException* e *ArrayIndexOutOfBoundsException* são exemplos de exceções do tipo *Unchecked*.

# Exceções do tipo *Checked*

---

1

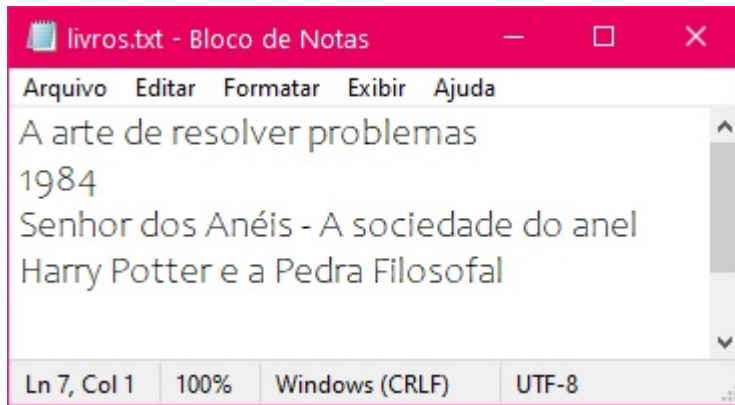
São chamadas de exceções de tempo de compilação.

2

O compilador verifica se há um tratamento para a exceção.

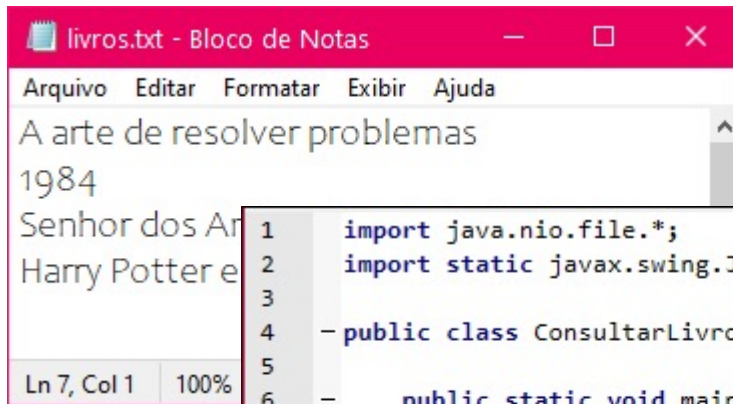
3

O programador é obrigado a tratar (ou delegar o tratamento) da exceção.



## Exemplo

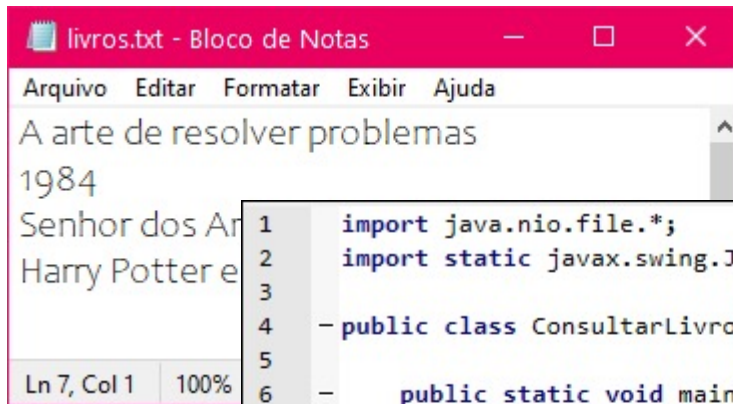
# Exemplo



```
1 import java.nio.file.*;
2 import static javax.swing.JOptionPane.*;
3
4 public class ConsultarLivro {
5
6     public static void main(String[] args) {
7         Path arquivo = Paths.get("livros.txt");
8         String[] livros = Files.readAllLines(arquivo).toArray(new String[0]);
9         String s = showInputDialog("Consultar livro da linha:");
10        int i = Integer.parseInt(s);
11        showMessageDialog(null, livros[i-1]);
12    }
13
14 }
```

programa para  
consultar livro

# Exemplo



```
1 import java.nio.file.*;
2 import static javax.swing.JOptionPane.*;
3
4 - public class ConsultarLivro {
5
6 -     public static void main(String[] args) {
7         Path arquivo = Paths.get("livros.txt");
8         String[] livros = Files.readAllLines(arquivo).toArray(new String[0]);
9         String s = showInputDialog("Consultar livro da linha:");
10        int i = Integer.parseInt(s);
11        showMessageDialog(null, livros[i-1]);
12    }
13
14 }
15
```

< >

```
>javac ConsultarLivro.java
ConsultarLivro.java:8: error: unreported exception IOException; must be caught or
declared to be thrown
        String[] livros = Files.readAllLines(arquivo).toArray(new String[0]);
                                   ^
1 error
>Exit code: 1
```

resultado da compilação

# Exemplo

```
livros.txt - Bloco de Notas
Arquivo  Editar  Formatar  Exibir  Ajuda
A arte de resolver problemas
1984
Senhor dos Anéis
Harry Potter e...
```

```
1  import java.nio.file.*;
2  import static javax.swing.JOptionPane.*;
3
4  -public class ConsultarLivro {
5
6  -    public static void main(String[] args) {
7      Path arquivo = Paths.get("livros.txt");
8      String[] livros = Files.readAllLines(arquivo).toArray(new String[0]);
9      String s = showInputDialog("Consultar livro da linha:");
10     int i = Integer.parseInt(s);
11     showMessageDialog(null, livros[i-1]);
12 }
13
14 }
15
```

```
>javac ConsultarLivro.java
ConsultarLivro.java:8: error: unreported exception IOException; must be
declared to be thrown
    String[] livros = Files.readAllLines(arquivo).toArray(new String[0]);
                              ^
1 error
>Exit code: 1
```

documentação do método readAllLines da classe Files

<https://docs.oracle.com/javase/8/docs/api/?java/nio/file/Files.html> 110%

### readAllLines

```
public static List<String> readAllLines(Path path)
    throws IOException
```

Read all lines from a file. Bytes from the file are decoded into characters using the UTF-8 charset.

This method works as if invoking it were equivalent to evaluating the expression:

```
Files.readAllLines(path, StandardCharsets.UTF_8)
```

**Parameters:**  
path - the path to the file

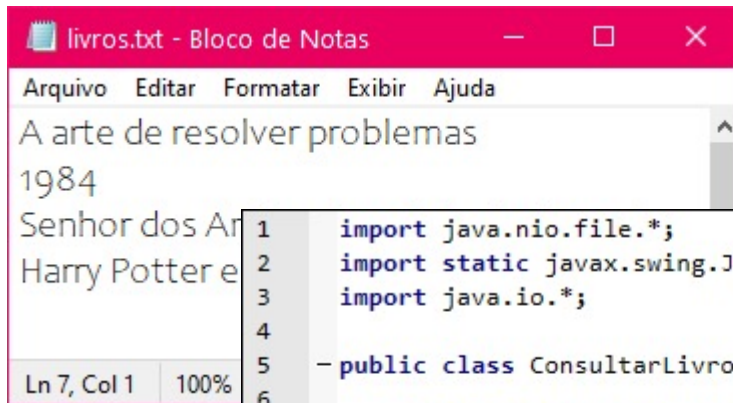
**Returns:**  
the lines from the file as a List; whether the List is modifiable or not is implementation dependent and therefore not specified

**Throws:**  
IOException - if an I/O error occurs reading from the file or a malformed or unmappable byte sequence is read  
SecurityException - In the case of the default provider, and a security manager is installed, the checkRead method is invoked to check read access to the file.

Exceção do tipo  
Checked



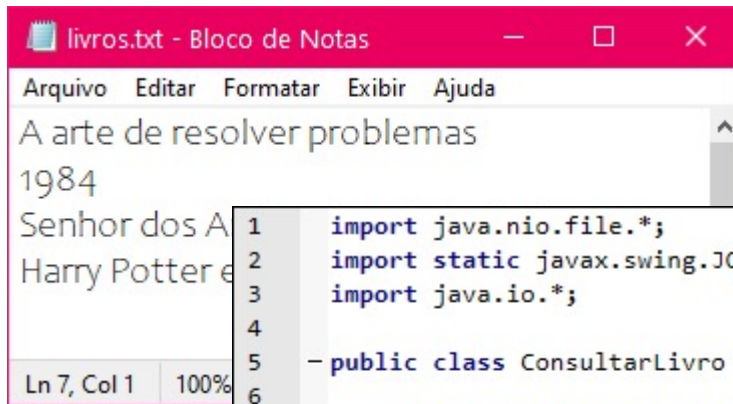
# Exemplo



versão com tratamento de exceção

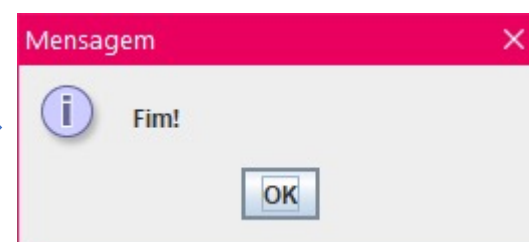
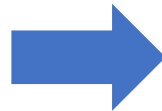
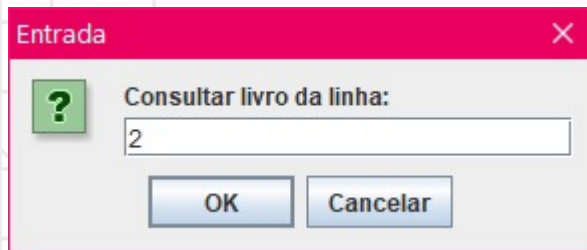
```
1  import java.nio.file.*;
2  import static javax.swing.JOptionPane.*;
3  import java.io.*;
4
5  -public class ConsultarLivro {
6
7      -    public static void main(String[] args) {
8          Path arquivo = Paths.get("livros.txt");
9          -    try {
10             String[] livros = Files.readAllLines(arquivo).toArray(new String[0]);
11             String s = showInputDialog("Consultar livro da linha:");
12             int i = Integer.parseInt(s);
13             showMessageDialog(null, livros[i-1]);
14         } catch (IOException ex) {
15             showMessageDialog(null, "Erro ao acessar o arquivo!");
16         }
17         showMessageDialog(null, "Fim!");
18     }
19
20 }
```

# Exemplo

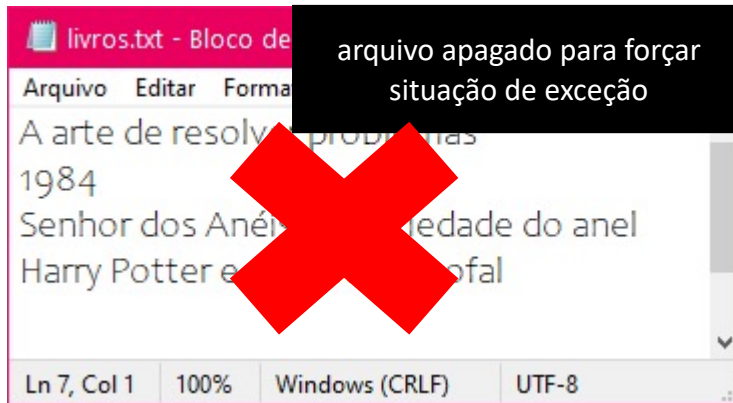


versão com tratamento de exceção

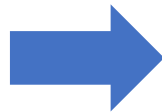
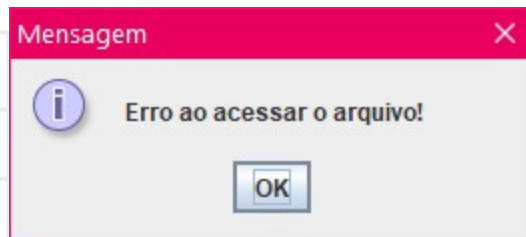
```
1  import java.nio.file.*;
2  import static javax.swing.JOptionPane.*;
3  import java.io.*;
4
5  - public class ConsultarLivro {
6
7      public static void main(String[] args) {
8          Path arquivo = Paths.get("livros.txt");
9          try {
10             String[] livros = Files.readAllLines(arquivo).toArray(new String[0]);
11             String s = showInputDialog("Consultar livro da linha:");
12             int i = Integer.parseInt(s);
13             showMessageDialog(null, livros[i-1]);
14         } catch (IOException ex) {
15             showMessageDialog(null, "Erro ao acessar o arquivo!");
16         }
17         showMessageDialog(null, "Fim!");
18     }
19
20 }
```



# Exemplo



```
7 - public static void main(String[] args) {  
8     Path arquivo = Paths.get("livros.txt");  
9 -     try {  
10         String[] livros = Files.readAllLines(arquivo).toArray(new String[0]);  
11         String s = showInputDialog("Consultar livro da linha:");  
12         int i = Integer.parseInt(s);  
13         showMessageDialog(null, livros[i-1]);  
14     } catch(IOException ex) {  
15         showMessageDialog(null, "Erro ao acessar o arquivo!");  
16     }  
17     showMessageDialog(null, "Fim!");  
18 }  
19  
20 }
```



# Exceções do tipo *Error*

---

1

Correspondem a problemas que estão fora do controle do usuário ou do programador.

2

Não são verificados no momento da compilação.

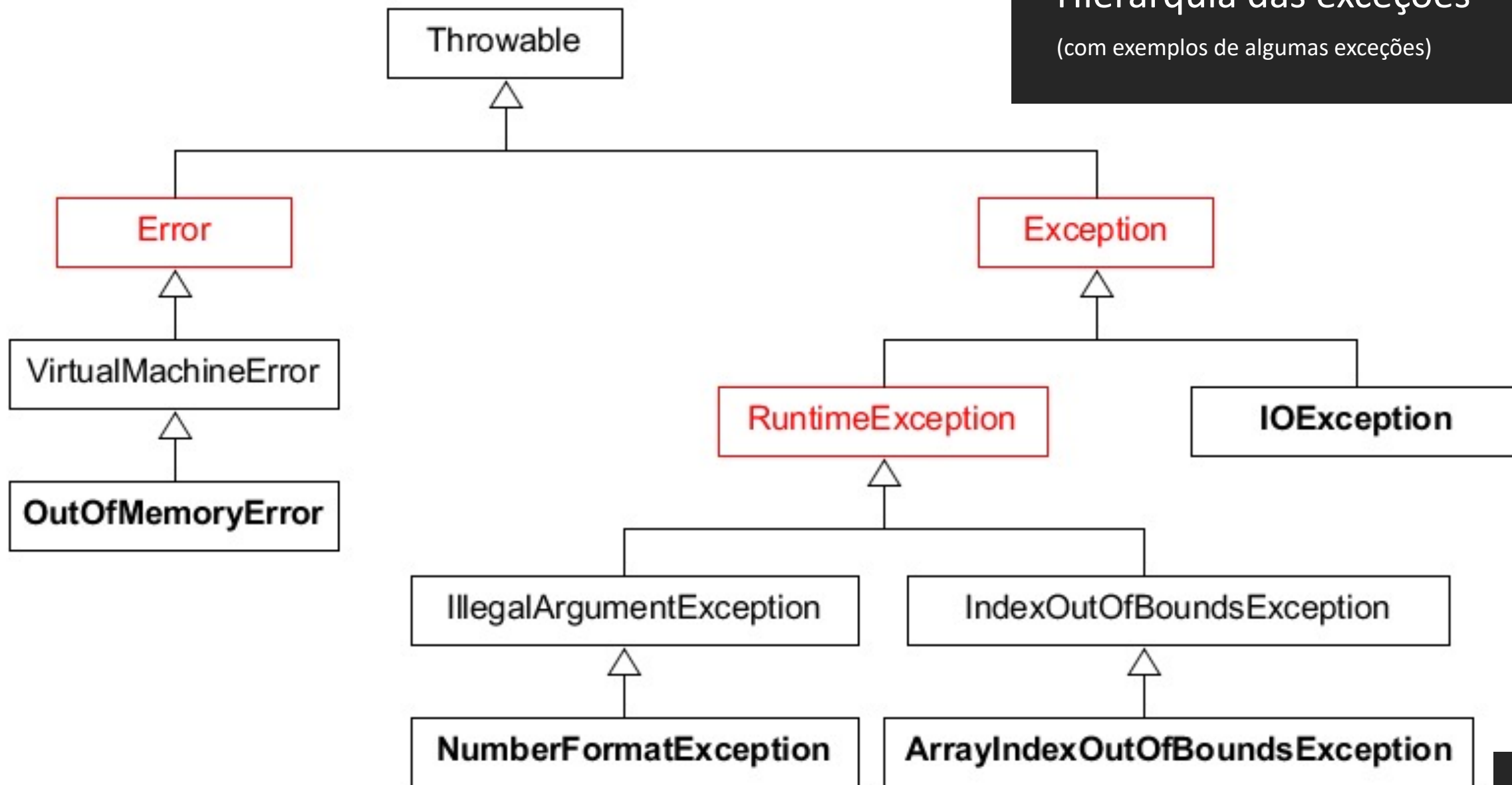
3

**Exemplo:**

*OutOfMemoryError*

# Hierarquia das exceções

(com exemplos de algumas exceções)



# Delegação do tratamento

---

1

A exceção do tipo *Checked* deve ser tratada ou **delegada**.

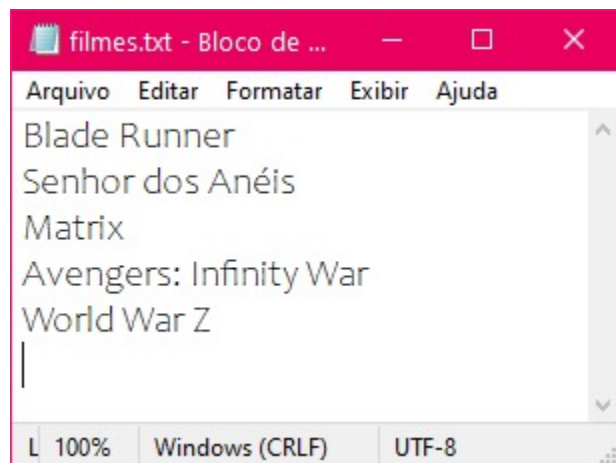
2

O tratamento é feito utilizando *try-catch-finally*

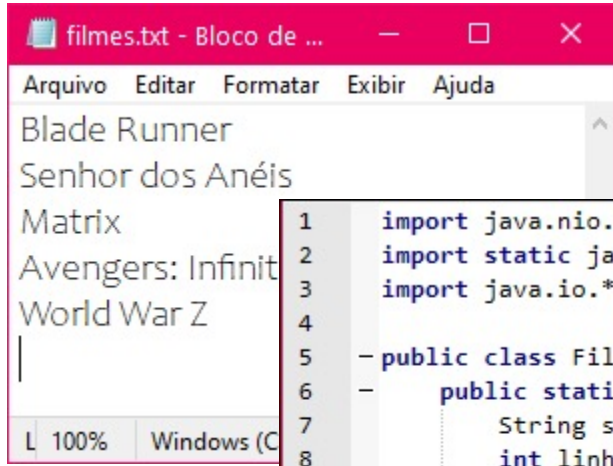
3

A delegação é feita utilizando *throws*

## Exemplo de delegação



# Exemplo de delegação

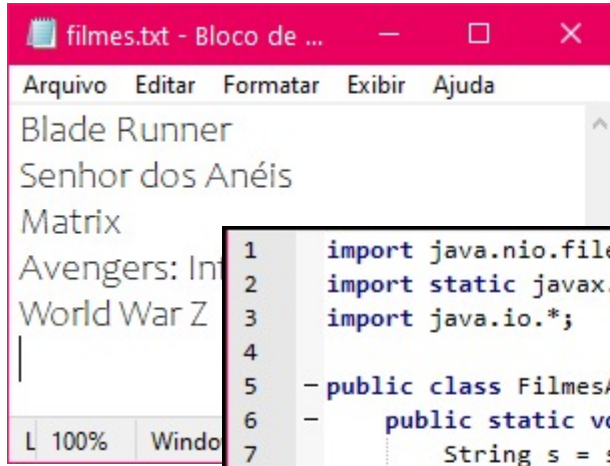


```
1  import java.nio.file.*;
2  import static javax.swing.JOptionPane.*;
3  import java.io.*;
4
5  - public class FilmesApp {
6  -      public static void main(String[] args) {
7              String s = showInputDialog("Consultar filme da linha:");
8              int linha = Integer.parseInt(s);
9              String nome = Filmes.consultar(linha-1);
10             showMessageDialog(null, nome);
11         }
12     }
13
14     - class Filmes {
15     -     public static String consultar(int indice) {
16             Path arquivo = Paths.get("filmes.txt");
17             String[] nomesFilmes = Files.readAllLines(arquivo).toArray(new String[0]);
18             return nomesFilmes[indice];
19         }
20     }
```

programa para  
consultar filme



## Exemplo de delegação

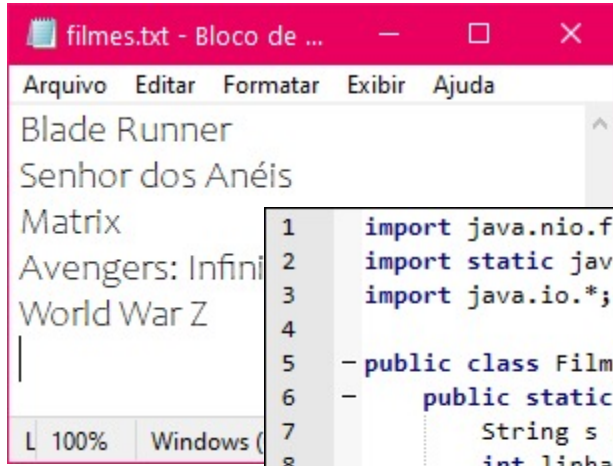


```
1  import java.nio.file.*;
2  import static javax.swing.JOptionPane.*;
3  import java.io.*;
4
5  -public class FilmesApp {
6  -    public static void main(String[] args) {
7      String s = showInputDialog("Consultar filme da linha:");
8      int linha = Integer.parseInt(s);
9      String nome = Filmes.consultar(linha-1);
10     showMessageDialog(null, nome);
11     }
12 }
13
14 -class Filmes {
15 -    public static String consultar(int indice) {
16         Path arquivo = Paths.get("filmes.txt");
17         String[] nomesFilmes = Files.readAllLines(arquivo).toArray(new String[0]);
18         return nomesFilmes[indice];
19     }
20 }
```

< >

```
>javac FilmesApp.java
FilmesApp.java:17: error: unreported exception IOException; must be caught or declared to be
thrown
    String[] nomesFilmes = Files.readAllLines(arquivo).toArray(new String[0]);
                                ^
1 error
>Exit code: 1
```

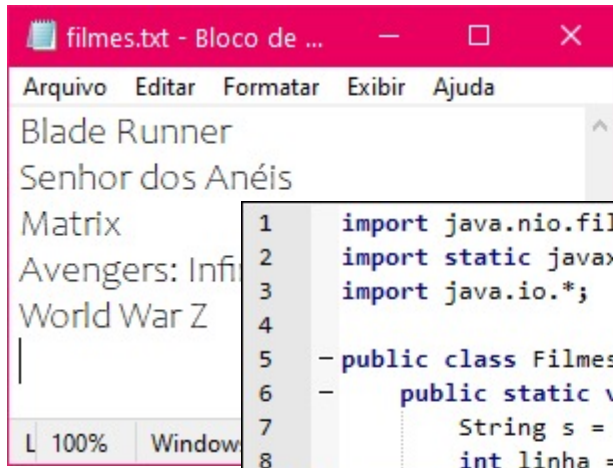
# Exemplo de delegação



```
1  import java.nio.file.*;
2  import static javax.swing.JOptionPane.*;
3  import java.io.*;
4
5  - public class FilmesApp {
6  -      public static void main(String[] args) {
7              String s = showInputDialog("Consultar filme da linha:");
8              int linha = Integer.parseInt(s);
9              String nome = Filmes.consultar(linha-1);
10             showMessageDialog(null, nome);
11         }
12     }
13
14     - class Filmes {
15     -     public static String consultar(int indice) throws IOException {
16             Path arquivo = Paths.get("filmes.txt");
17             String[] nomesFilmes = Files.readAllLines(arquivo).toArray(new String[0]);
18             return nomesFilmes[indice];
19         }
20     }
```

"delegação"

## Exemplo de delegação



```
1 import java.nio.file.*;
2 import static javax.swing.JOptionPane.*;
3 import java.io.*;
4
5 -public class FilmesApp {
6     - public static void main(String[] args) {
7         String s = showInputDialog("Consultar filme da linha:");
8         int linha = Integer.parseInt(s);
9         String nome = Filmes.consultar(linha-1);
10        showMessageDialog(null, nome);
11    }
12 }
13
14 -class Filmes {
15     - public static String consultar(int indice) throws IOException {
16         Path arquivo = Paths.get("filmes.txt");
17         String[] nomesFilmes = Files.readAllLines(arquivo).toArray(new String[0]);
18         return nomesFilmes[indice];
19     }
20 }
21
```

agora a exceção deve ser tratada aqui

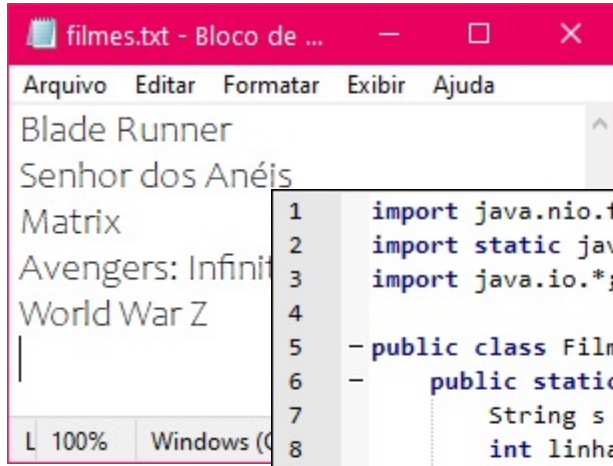
```
>javac FilmesApp.java
```

```
FilmesApp.java:9: error: unreported exception IOException; must be caught or declared to be thrown
    String nome = Filmes.consultar(linha-1);
                        ^
```

```
1 error
```

```
>Exit code: 1
```

# Exemplo de delegação



```
1  import java.nio.file.*;
2  import static javax.swing.JOptionPane.*;
3  import java.io.*;
4
5  - public class FilmesApp {
6  -      public static void main(String[] args) {
7          String s = showInputDialog("Consultar filme da linha:");
8          int linha = Integer.parseInt(s);
9          -      try {
10              String nome = Filmes.consultar(linha-1);
11              showMessageDialog(null, nome);
12          } catch (IOException ex) {
13              showMessageDialog(null, "Problema ao acessar os filmes!");
14          }
15          showMessageDialog(null, "Fim!");
16      }
17  }
18
19  - class Filmes {
20  -      public static String consultar(int indice) throws IOException {
21          Path arquivo = Paths.get("filmes.txt");
22          String[] nomesFilmes = Files.readAllLines(arquivo).toArray(new String[0]);
23          return nomesFilmes[indice];
24      }
25  }
```

versão com delegação e  
tratamento

# Atividade de Laboratório

As aplicações dadas como exemplo ainda estão sujeitas à ocorrência das exceções **NumberFormatException** e **ArrayIndexOutOfBoundsException**.

- a) Altere os dois exemplos de forma a tratar estas exceções:
  - Quando ocorrer **NumberFormatException**, apresente a mensagem "**Por favor, entre um número!**".
  - Quando ocorrer **ArrayIndexOutOfBoundsException**, apresente a mensagem "**Valor inválida!**".
- b) Além do tratamento dado no item (a), caso ocorra uma das duas exceções, o programa deve ficar dando a chance para o usuário entrar um novo valor até que ele entre um valor válido.
- c) Faça a entrega de um arquivo compactado (com os arquivos do seu projeto, no formato zip) no Moodle, na tarefa **Tipos de Exceções**.

# Referências

- HORSTMANN, C.S.; CORNELL, G. **Core Java, Volume I - Fundamentals**. Prentice Hall, 2012.
- Caelum - Excessões e controles de erros. Disponível em:  
<http://www.caelum.com.br/apostila-java-orientacao-objetos/excecoes-e-controle-de-erros>
- Tutorials Point - Java - Exceptions. Disponível em:  
[http://www.tutorialspoint.com/java/java\\_exceptions.htm](http://www.tutorialspoint.com/java/java_exceptions.htm)



Universidade Presbiteriana  
**Mackenzie**

**150** anos  
1870 - 2020



Faculdade de  
**Computação e Informática**

