

Exception – Fazendo tratamento de exceções no Java



Fazemos o tratamento de exceções quando sabemos que um erro pode ocorrer.

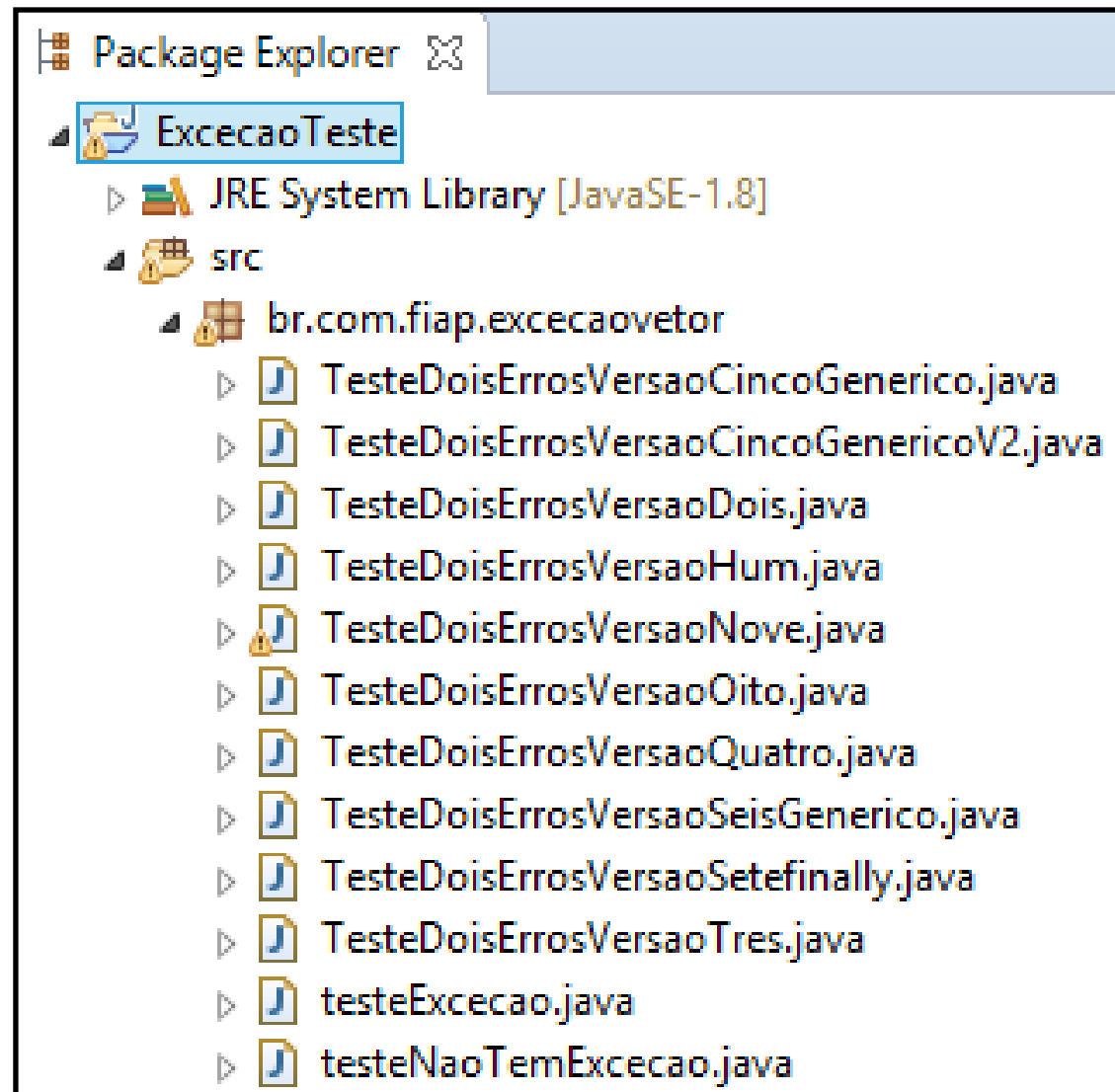
Por exemplo no próximo slides, queremos que os valores sejam atribuídos ao vetor de três posições (da posição zero até a dois) e então quando tentar alocar valor na quarta posição, que seria o índice 3, não irá conseguir, pois o vetor só tem até a dois, dessa maneira não irá continuar a execução do programa, ou seja, não irá imprimir a frase “Obrigado por utilizar esse programa”.

Porém no terceiro slide, temos como fazer para que mesmo tentando alocar valor na quarta posição o programa não vai conseguir alocar esse valor, pois essa posição não existe no vetor, mas como tem um tratamento de exceção, ou seja, para o programa já está previsto esse erro, irá continuar a execução do programa, exibindo a frase “Obrigado por utilizar esse programa”.

Resumo:

- Se não fizer tratamento de erros poderá parar o seu programa;
- Se tiver tratamento de erro para o que já é conhecido que pode ocorrer, o seu programa não para no ponto do erro, pois tem uma exceção para tratar o caso.
- Em um sistema em produção, quando analisamos e não localizamos outra saída para a correção do erro/problema, inserir a exceção e após buscamos a solução, até progredir dos ambientes de desenvolvimento para produção com a solução que remova a exceção criada.

Exception – Fazendo tratamento de exceções no Java

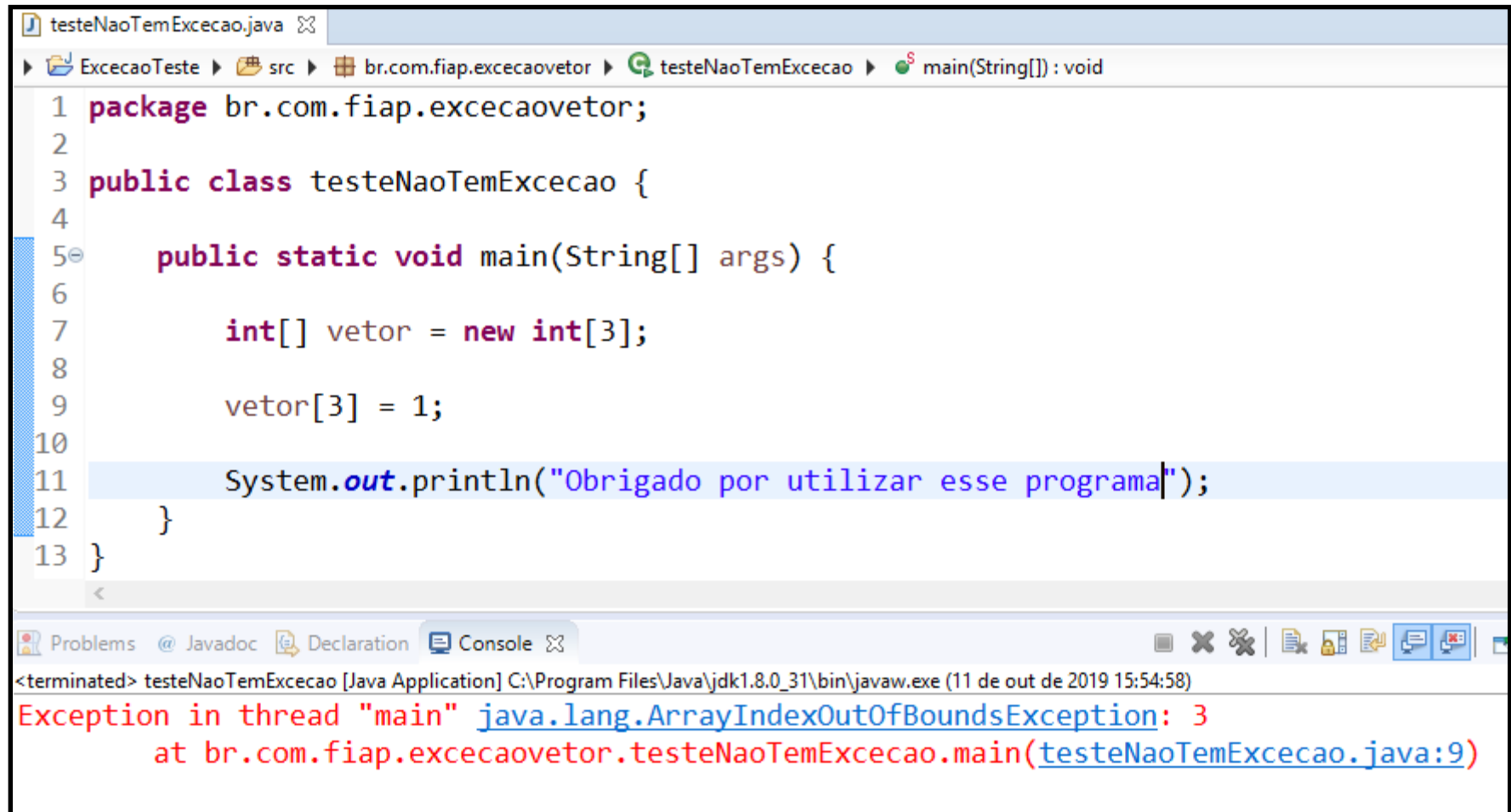


Exception – Fazendo tratamento de exceções no Java



```
package br.com.fiap.excecao vetor;  
  
public class testeNaoTemExcecao {  
  
    public static void main(String[] args) {  
  
        int[] vetor = new int[3];  
  
        vetor[3] = 1;  
  
        System.out.println("Obrigado por utilizar esse programa");  
    }  
}
```

Exception – Fazendo tratamento de exceções no Java



The screenshot shows an IDE window titled 'testeNaoTemExcecao.java'. The code is as follows:

```
1 package br.com.fiap.excecao vetor;  
2  
3 public class testeNaoTemExcecao {  
4  
5     public static void main(String[] args) {  
6  
7         int[] vetor = new int[3];  
8  
9         vetor[3] = 1;  
10  
11         System.out.println("Obrigado por utilizar esse programa");  
12     }  
13 }
```

The IDE's console at the bottom shows the following output:

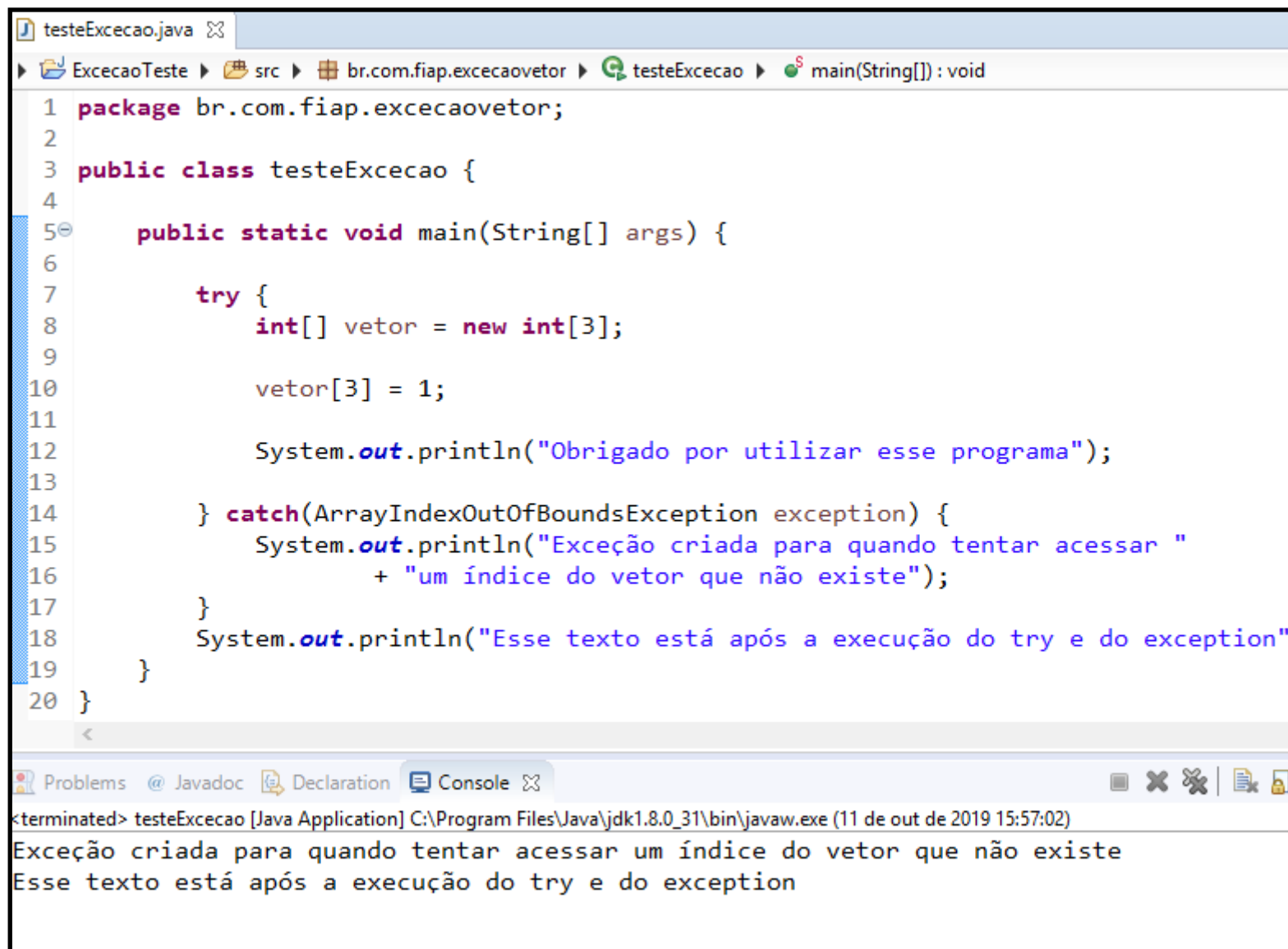
```
<terminated> testeNaoTemExcecao [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\javaw.exe (11 de out de 2019 15:54:58)  
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 3  
    at br.com.fiap.excecao vetor.testeNaoTemExcecao.main(testeNaoTemExcecao.java:9)
```

Exception – Fazendo tratamento de exceções no Java



```
package br.com.fiap.excecao vetor;  
  
public class testeExcecao {  
  
    public static void main(String[] args) {  
  
        try {  
            int[] vetor = new int[3];  
  
            vetor[3] = 1;  
  
            System.out.println("Exceção criada para quando tentar acessar "  
+ "um índice do vetor que não existe");  
        } catch (ArrayIndexOutOfBoundsException exception) {  
            System.out.println("Exceção ao acessar um índice do vetor que não existe");  
        }  
        System.out.println("Esse texto está após a execução do try e do exception");  
    }  
}
```

Exception – Fazendo tratamento de exceções no Java



```
testeExcecao.java
ExcecaoTeste ▸ src ▸ br.com.fiap.excecao vetor ▸ testeExcecao ▸ main(String[]): void

1 package br.com.fiap.excecao vetor;
2
3 public class testeExcecao {
4
5     public static void main(String[] args) {
6
7         try {
8             int[] vetor = new int[3];
9
10            vetor[3] = 1;
11
12            System.out.println("Obrigado por utilizar esse programa");
13
14        } catch (ArrayIndexOutOfBoundsException exception) {
15            System.out.println("Exceção criada para quando tentar acessar "
16                               + "um índice do vetor que não existe");
17        }
18        System.out.println("Esse texto está após a execução do try e do exception");
19    }
20 }
```

Problems @ Javadoc Declaration Console

<terminated> testeExcecao [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\javaw.exe (11 de out de 2019 15:57:02)

Exceção criada para quando tentar acessar um índice do vetor que não existe

Esse texto está após a execução do try e do exception

Segunda parte

Exception – Fazendo tratamento de exceções no Java

- Visualização do erro



```
package br.com.fiap.excecao vetor;
```

```
public class TesteDoisErros {
```

```
    public static void main(String[] args) {
```

```
        int[] numero = {2, 4, 8, 16, 32, 64};
```

```
        int[] denom = {2, 2, 0, 2, 0};
```

```
        for (int i=0; i<numero.length; i++) {
```

```
            System.out.println("Resultado: " + numero[i]/denom[i]);
```

```
        }
```

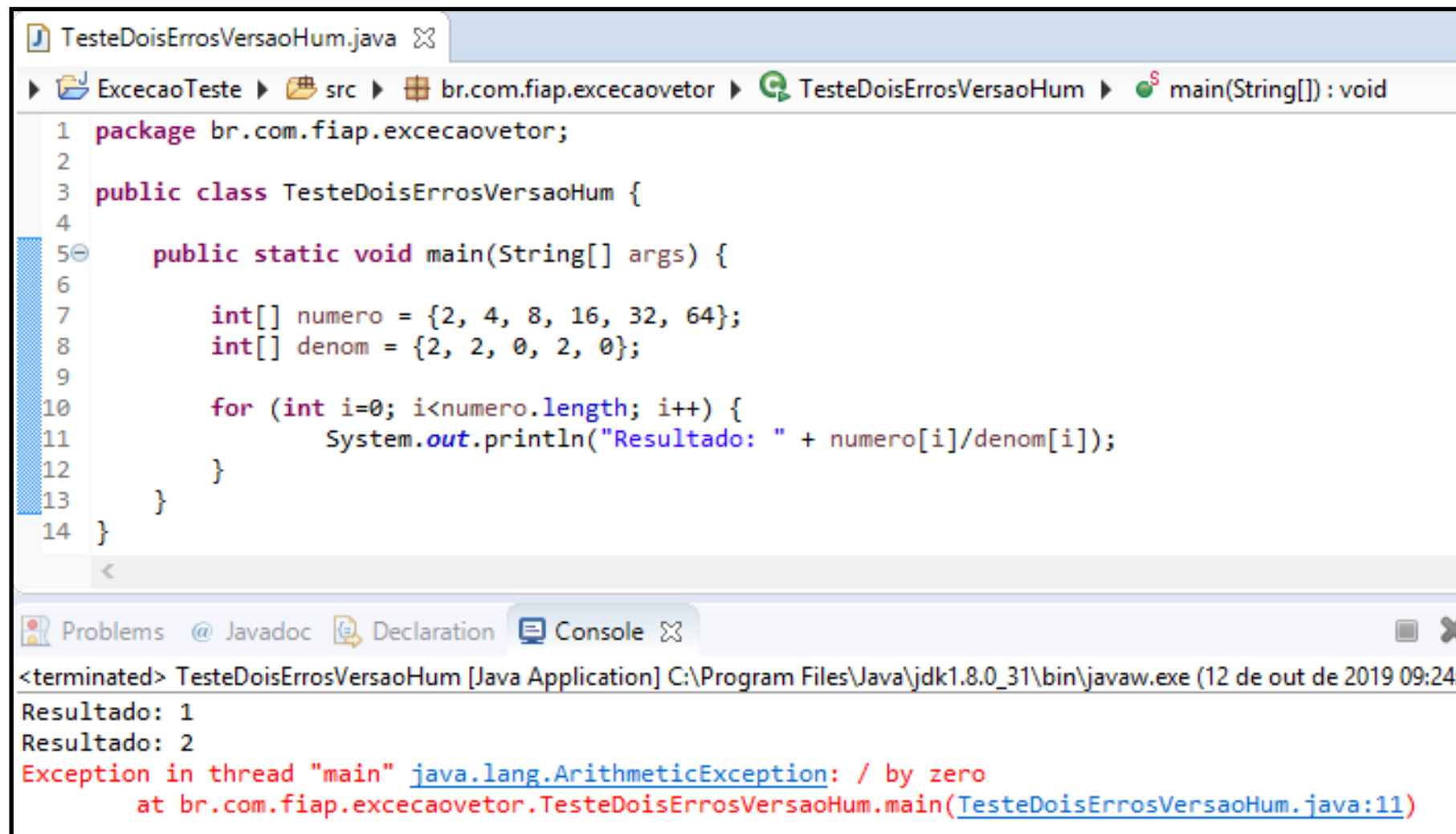
```
    }
```

```
}
```


Exception – Fazendo tratamento de exceções no Java

- Visualização do erro

FIAP



The screenshot shows an IDE window with a Java file named `TesteDoisErrosVersaoHum.java`. The code defines a class `TesteDoisErrosVersaoHum` with a `main` method. The `main` method calculates the division of two arrays of integers, `numero` and `denom`. The `denom` array contains a zero, which causes an `ArithmeticException` during execution. The IDE's console window shows the output of the program, including the exception message and the stack trace pointing to line 11 of the `TesteDoisErrosVersaoHum.java` file.

```
1 package br.com.fiap.excecaoovetor;
2
3 public class TesteDoisErrosVersaoHum {
4
5     public static void main(String[] args) {
6
7         int[] numero = {2, 4, 8, 16, 32, 64};
8         int[] denom = {2, 2, 0, 2, 0};
9
10        for (int i=0; i<numero.length; i++) {
11            System.out.println("Resultado: " + numero[i]/denom[i]);
12        }
13    }
14 }
```

<terminated> TesteDoisErrosVersaoHum [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\javaw.exe (12 de out de 2019 09:24)

Resultado: 1
Resultado: 2
Exception in thread "main" [java.lang.ArithmeticException](#): / by zero
at br.com.fiap.excecaoovetor.TesteDoisErrosVersaoHum.main([TesteDoisErrosVersaoHum.java:11](#))

Exception – Fazendo tratamento de exceções no Java

- Tratando o primeiro erro

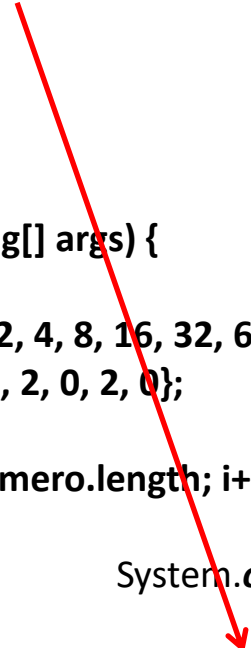
```
package br.com.fiap.excecaoovetor;

public class TesteDoisErrosVersaoDois {

    public static void main(String[] args) {

        int[] numero = {2, 4, 8, 16, 32, 64};
        int[] denom = {2, 2, 0, 2, 0};

        for (int i=0; i<numero.length; i++) {
            try {
                System.out.println("Resul " + numero[i]/denom[i]);
            }
            catch (ArithmeticException exc) {
                System.out.println("Ocorreu um erro durante a execução do for,"
                    + " pois tentou dividir um número por zero");
            }
        }
    }
}
```



Exception – Fazendo tratamento de exceções no Java

- Tratando o primeiro erro

FIAP

```
TesteDoisErrosVersaoDois.java
ExcecaoTeste > src > br.com.fiap.excecaoovetor > TesteDoisErrosVersaoDois > main(String[]) : void

1 package br.com.fiap.excecaoovetor;
2
3 public class TesteDoisErrosVersaoDois {
4
5     public static void main(String[] args) {
6
7         int[] numero = {2, 4, 8, 16, 32, 64};
8         int[] denom = {2, 2, 0, 2, 0};
9
10        for (int i=0; i<numero.length; i++) {
11            try {
12                System.out.println("Resultado: " + numero[i]/denom[i]);
13            }
14            catch (ArithmeticException exc) {
15                System.out.println("Ocorreu um erro durante a execução do for,"
16                                    + " pois tentou dividir um número por zero");
17            }
18        }
19    }
20 }
```

Problems @ Javadoc Declaration Console

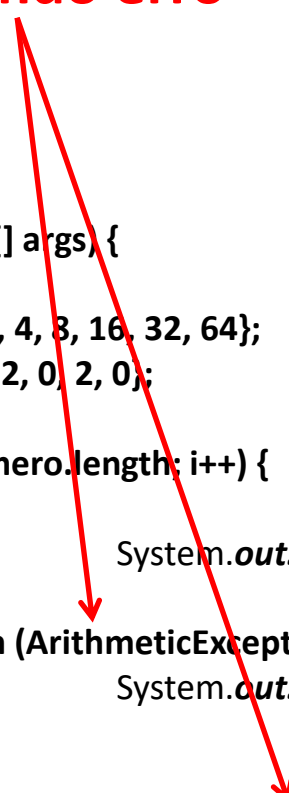
<terminated> TesteDoisErrosVersaoDois [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\javaw.exe (12 de out de 2019 09:32:23)

Resultado: 1
Resultado: 2
Ocorreu um erro durante a execução do for, pois tentou dividir um número por zero
Resultado: 8
Ocorreu um erro durante a execução do for, pois tentou dividir um número por zero
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
at br.com.fiap.excecaoovetor.TesteDoisErrosVersaoDois.main(TesteDoisErrosVersaoDois.java:12)

Exception – Fazendo tratamento de exceções no Java

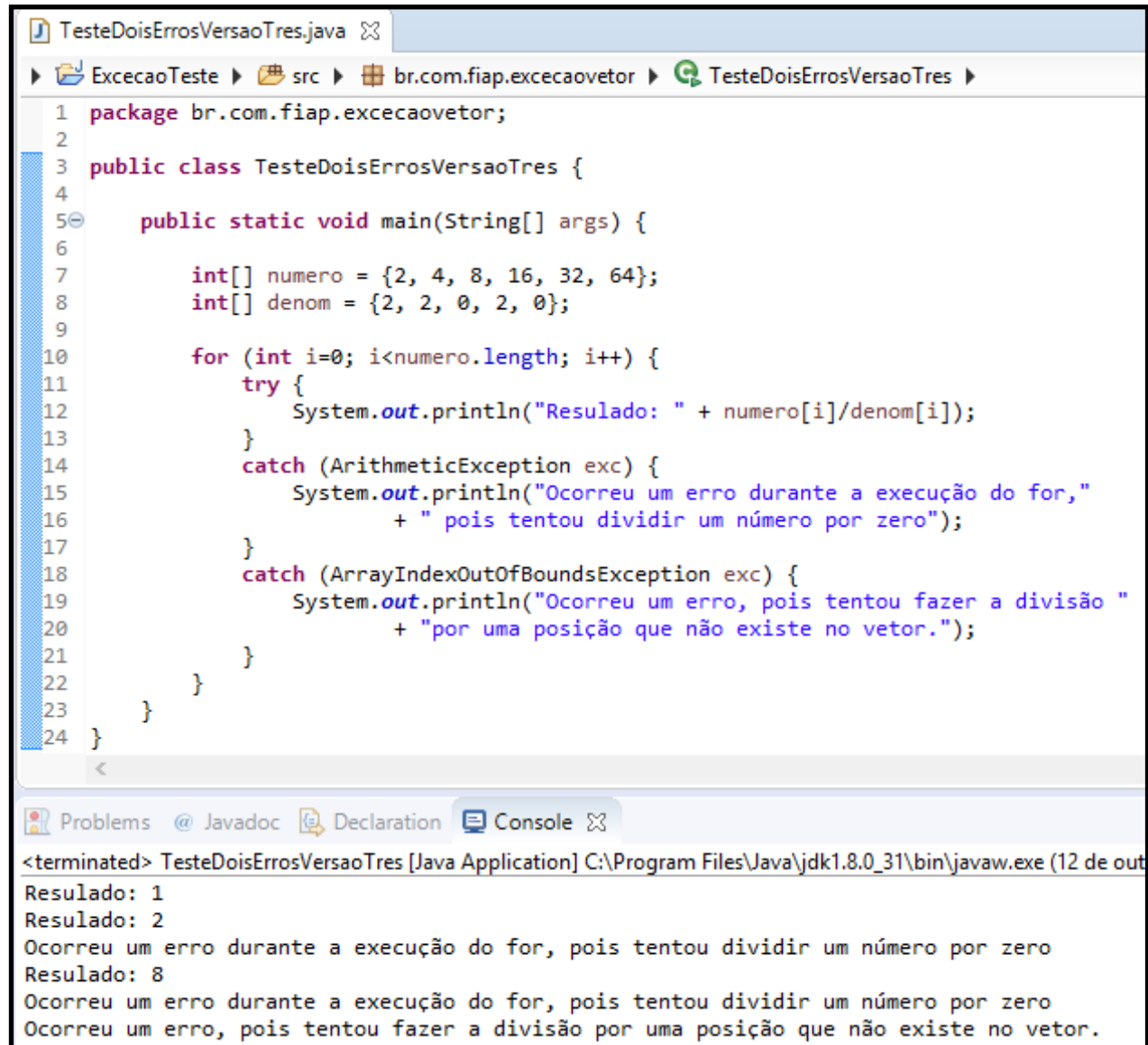
- Tratando o segundo erro

```
package br.com.fiap.excecao vetor;  
  
public class TesteDoisErrosVersaoTres {  
  
    public static void main(String[] args) {  
  
        Int[] numero = {2, 4, 8, 16, 32, 64};  
        int[] denom = {2, 2, 0, 2, 0};  
  
        for (int i=0; i<numero.length; i++) {  
            try {  
                System.out.println("Resultado: " + numero[i]/denom[i]);  
            }  
            catch (ArithmeticException exc) {  
                System.out.println("Ocorreu um erro durante a execução do for,"  
                    + " pois tentou dividir um número por zero");  
            }  
            catch (ArrayIndexOutOfBoundsException exc) {  
                System.out.println("Ocorreu um erro, pois tentou fazer a divisão "  
                    + "por uma posição que não existe no vetor.");  
            }  
        }  
    }  
}
```



Exception – Fazendo tratamento de exceções no Java

- Tratando o segundo erro



The screenshot displays an IDE window with a Java file named `TesteDoisErrosVersaoTres.java`. The code defines a `main` method that iterates over an array of numbers and divides them by an array of denominators. It uses `try-catch` blocks to handle `ArithmeticException` and `ArrayIndexOutOfBoundsException`. The output console shows the results of the execution, including division results and error messages for division by zero and out-of-bounds access.

```
1 package br.com.fiap.excecaoavetor;
2
3 public class TesteDoisErrosVersaoTres {
4
5     public static void main(String[] args) {
6
7         int[] numero = {2, 4, 8, 16, 32, 64};
8         int[] denom = {2, 2, 0, 2, 0};
9
10        for (int i=0; i<numero.length; i++) {
11            try {
12                System.out.println("Resultado: " + numero[i]/denom[i]);
13            }
14            catch (ArithmeticException exc) {
15                System.out.println("Ocorreu um erro durante a execução do for,"
16                    + " pois tentou dividir um número por zero");
17            }
18            catch (ArrayIndexOutOfBoundsException exc) {
19                System.out.println("Ocorreu um erro, pois tentou fazer a divisão "
20                    + "por uma posição que não existe no vetor.");
21            }
22        }
23    }
24 }
```

Problems @ Javadoc Declaration Console


<terminated> TesteDoisErrosVersaoTres [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\javaw.exe (12 de out
Resultado: 1
Resultado: 2
Ocorreu um erro durante a execução do for, pois tentou dividir um número por zero
Resultado: 8
Ocorreu um erro durante a execução do for, pois tentou dividir um número por zero
Ocorreu um erro, pois tentou fazer a divisão por uma posição que não existe no vetor.

Exception – Fazendo tratamento de exceções no Java

- Tratando os dois erros com apenas um catch

FIAP

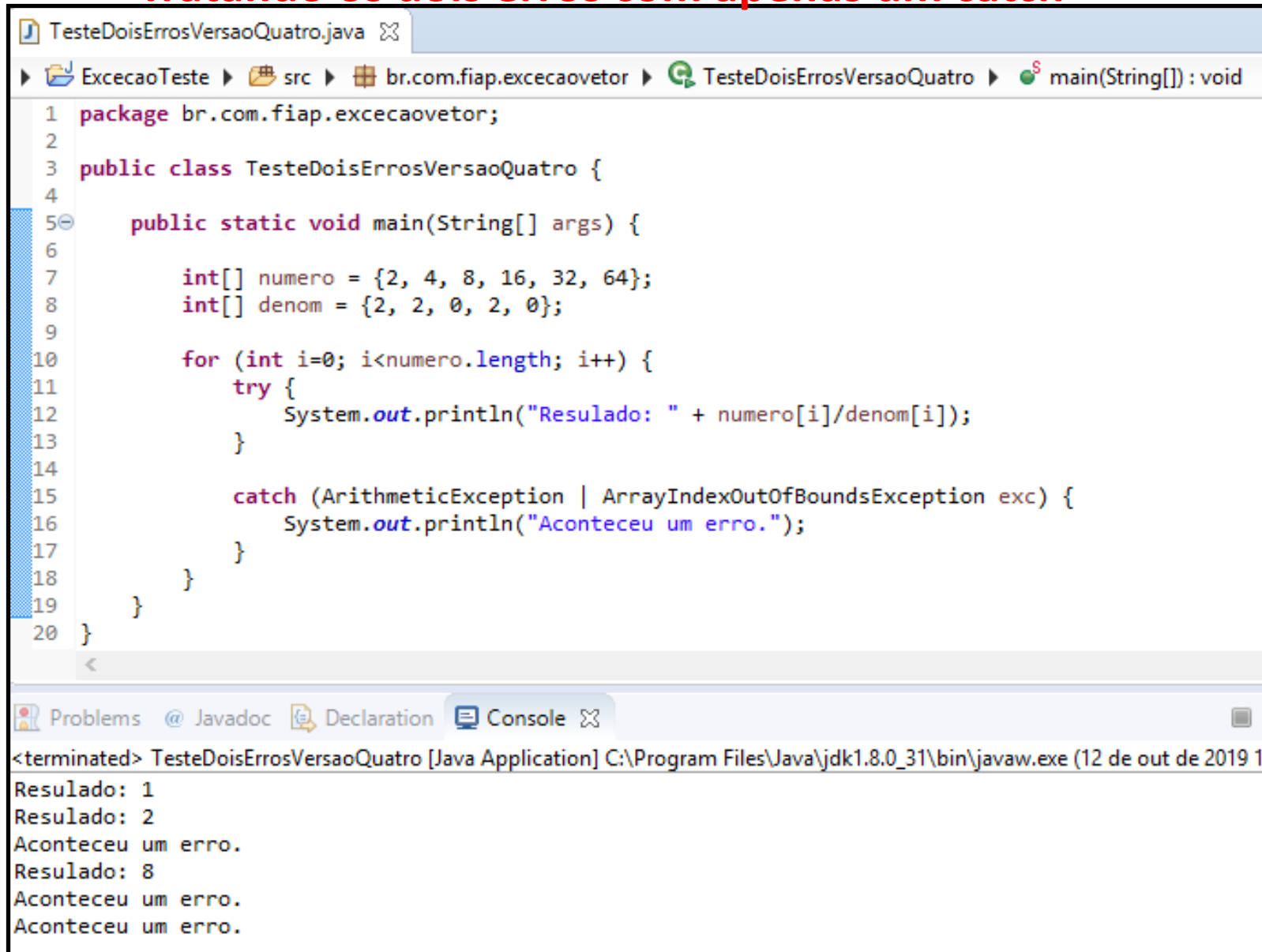
```
package br.com.fiap.excecao vetor;  
  
public class TesteDoisErrosVersaoQuatro {  
  
    public static void main(String[] args) {  
  
        int[] numero = {2, 4, 8, 16, 32, 64};  
        int[] denom = {2, 2, 0, 2, 0};  
  
        for (int i=0; i<numero.length; i++) {  
            try {  
                System.out.println("Resultado: " + numero[i]/denom[i]);  
            }  
  
            catch (ArithmeticException | ArrayIndexOutOfBoundsException exc) {  
                System.out.println("Aconteceu um erro.");  
            }  
        }  
    }  
}
```



Exception – Fazendo tratamento de exceções no Java

- Tratando os dois erros com apenas um catch

FIAP



The screenshot shows an IDE window with a Java file named `TesteDoisErrosVersaoQuatro.java`. The code defines a class `TesteDoisErrosVersaoQuatro` with a `main` method. Inside `main`, two arrays are defined: `numero` with values {2, 4, 8, 16, 32, 64} and `denom` with values {2, 2, 0, 2, 0}. A `for` loop iterates over the `numero` array. Inside the loop, a `try` block attempts to calculate `numero[i] / denom[i]` and print the result. A `catch` block catches both `ArithmeticException` and `ArrayIndexOutOfBoundsException`, printing "Aconteceu um erro." for any exception that occurs.

```
1 package br.com.fiap.excecaoovetor;
2
3 public class TesteDoisErrosVersaoQuatro {
4
5     public static void main(String[] args) {
6
7         int[] numero = {2, 4, 8, 16, 32, 64};
8         int[] denom = {2, 2, 0, 2, 0};
9
10        for (int i=0; i<numero.length; i++) {
11            try {
12                System.out.println("Resultado: " + numero[i]/denom[i]);
13            }
14
15            catch (ArithmeticException | ArrayIndexOutOfBoundsException exc) {
16                System.out.println("Aconteceu um erro.");
17            }
18        }
19    }
20 }
```

The console output shows the results of the program execution:

```
<terminated> TesteDoisErrosVersaoQuatro [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\javaw.exe (12 de out de 2019 1
Resultado: 1
Resultado: 2
Aconteceu um erro.
Resultado: 8
Aconteceu um erro.
Aconteceu um erro.
```

Exception – Fazendo tratamento de exceções no Java

- Utilizando o Throwable para erros genéricos/desconhecidos



```
package br.com.fiap.excecao vetor;
```

```
public class TesteDoisErrosVersaoCincoGenerico {
```

```
    public static void main(String[] args) {
```

```
        int[] numero = {2, 4, 8, 16, 32, 64};
```

```
        int[] denom = {2, 2, 0, 2, 0};
```

```
        for (int i=0; i<numero.length; i++) {
```

```
            try {
```

```
                System.out.println("Resultado: " + numero[i]/denom[i]);
```

```
            }
```

```
            catch (ArithmeticException exc) {
```

```
                System.out.println("Ocorreu um erro durante a execução do for,"  
                                    + " pois tentou dividir um número por zero");
```

```
            }
```

```
        /**
```

```
        * @author Ronqui
```

```
        * Throwable é a superclasse que captura todos os erros
```

```
        * e exceções no Java
```

```
        */
```

```
        catch (Throwable exc) {
```

```
            System.out.println("Ocorreu um erro.");
```

```
        }
```

```
    }
```

```
}
```

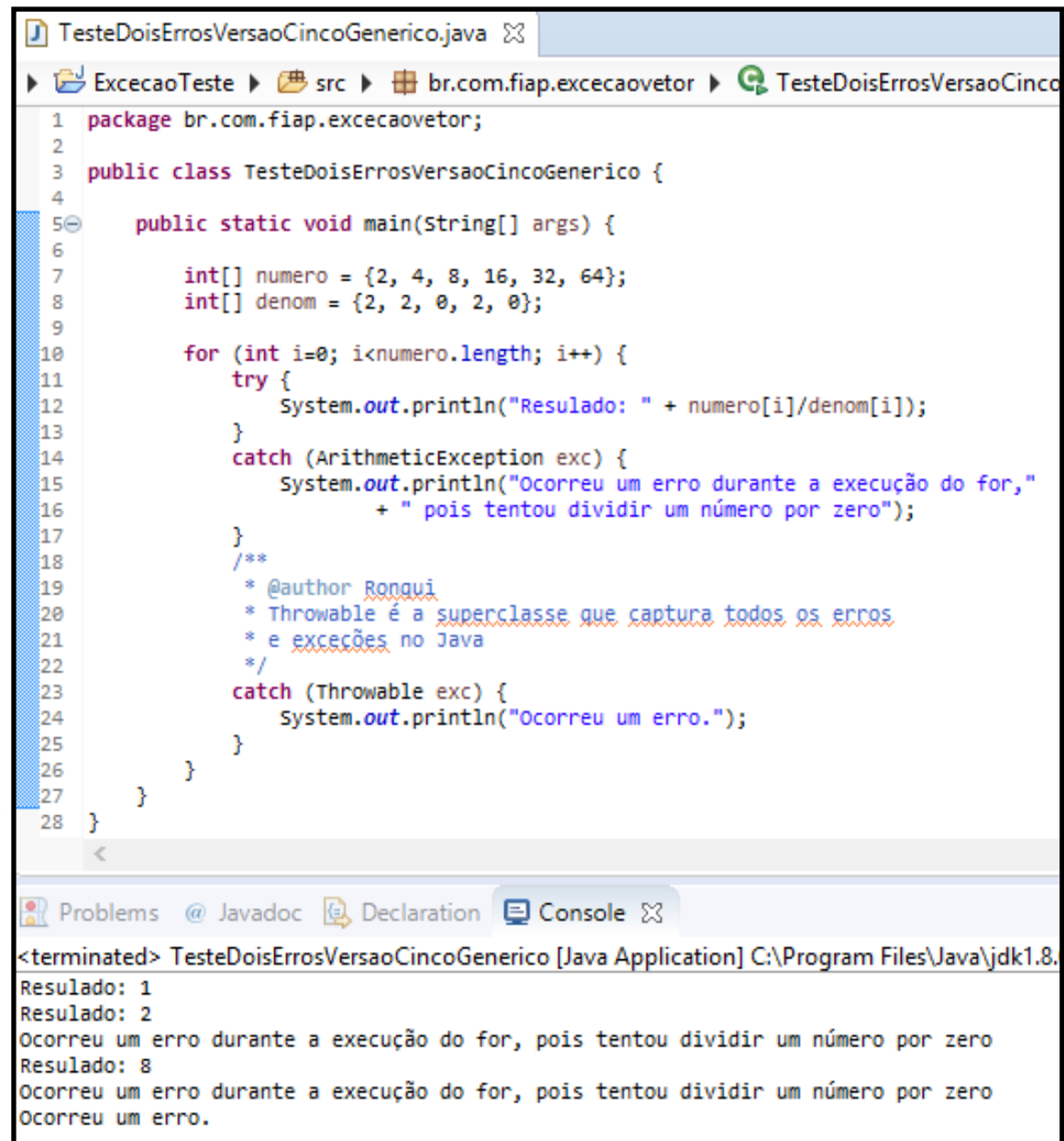
```
}
```


Exception – Fazendo tratamento de exceções no Java

- Utilizando o Throwable para erros genéricos/desconhecidos

Visualize no código ao lado que o correto é primeiro capturar os erros específicos que já são conhecidos, após. Que utilizamos o Throwable para capturar erros genéricos / desconhecidos.

Muito cuidado ao utilizar o Throwable, pois você está ignorando um erro que está ocorrendo no seu programa, ou seja, pode ocorrer um grande impacto e um sistema em funcionamento / em Produção, isso é terrível.



The screenshot shows an IDE with a Java file named `TesteDoisErrosVersaoCincoGenerico.java`. The code defines a class `TesteDoisErrosVersaoCincoGenerico` with a `main` method. It uses two arrays: `numero` (2, 4, 8, 16, 32, 64) and `denom` (2, 2, 0, 2, 0). A `for` loop iterates over these arrays. For each iteration, it attempts to calculate `numero[i]/denom[i]`. It uses a `try-catch` block to handle exceptions. The first catch block handles `ArithmeticException` (division by zero). The second catch block handles `Throwable` (generic exception). The output of the program is shown in the console window at the bottom.

```
1 package br.com.fiap.excecaoovetor;
2
3 public class TesteDoisErrosVersaoCincoGenerico {
4
5     public static void main(String[] args) {
6
7         int[] numero = {2, 4, 8, 16, 32, 64};
8         int[] denom = {2, 2, 0, 2, 0};
9
10        for (int i=0; i<numero.length; i++) {
11            try {
12                System.out.println("Resultado: " + numero[i]/denom[i]);
13            }
14            catch (ArithmeticException exc) {
15                System.out.println("Ocorreu um erro durante a execução do for,"
16                    + " pois tentou dividir um número por zero");
17            }
18            /**
19             * @author Ronqui
20             * Throwable é a superclasse que captura todos os erros
21             * e exceções no Java
22             */
23            catch (Throwable exc) {
24                System.out.println("Ocorreu um erro.");
25            }
26        }
27    }
28 }
```

Problems @ Javadoc Declaration Console

<terminated> TesteDoisErrosVersaoCincoGenerico [Java Application] C:\Program Files\Java\jdk1.8.

Resultado: 1
Resultado: 2
Ocorreu um erro durante a execução do for, pois tentou dividir um número por zero
Resultado: 8
Ocorreu um erro durante a execução do for, pois tentou dividir um número por zero
Ocorreu um erro.

Exception – Fazendo tratamento de exceções no Java

- Utilizando o Throwable para erros genéricos/desconhecidos



Está ocorrendo esse erro, pois primeiro eu coloquei o Throwable que captura todos os erros.

Para o Java o correto primeiro é colocar a busca dos erros específicos e depois os genéricos (Throwable).

O correto para o Java é como fizemos na classe “TesteDoisErrosVersaoCincoGenerico”, primeiro

Capturar os erros mais específicos e depois colocamos um catch para capturar um erro mais genérico.

```
package br.com.fiap.excecabvetor;
```

```
public class TesteDoisErrosVersaoCincoGenericoV2 {
```

```
    /**
```

```
     * Foi criado com erro, para testar o Throwable, sendo
```

```
     * colocado primeiro que a busca dos erros específicos
```

```
     *
```

```
     * @param args
```

```
    */
```

```
    public static void main(String[] args) {
```

```
        int[] numero = {2, 4, 8, 16, 32, 64};
```

```
        int[] denom = {2, 2, 0, 2, 0};
```

```
        for (int i=0; i<numero.length; i++) {
```

```
            try {
```

```
                System.out.println("Resultado: " + numero[i]/denom[i]);
```

```
            }
```

```
            catch (Throwable exc) {
```

```
                System.out.println("Ocorreu um erro.");
```

```
            }
```

```
            catch (ArithmeticException exc) {
```

```
                System.out.println("Ocorreu um erro durante a execução do for,"
```

```
                + " pois tentou dividir um número por zero");
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

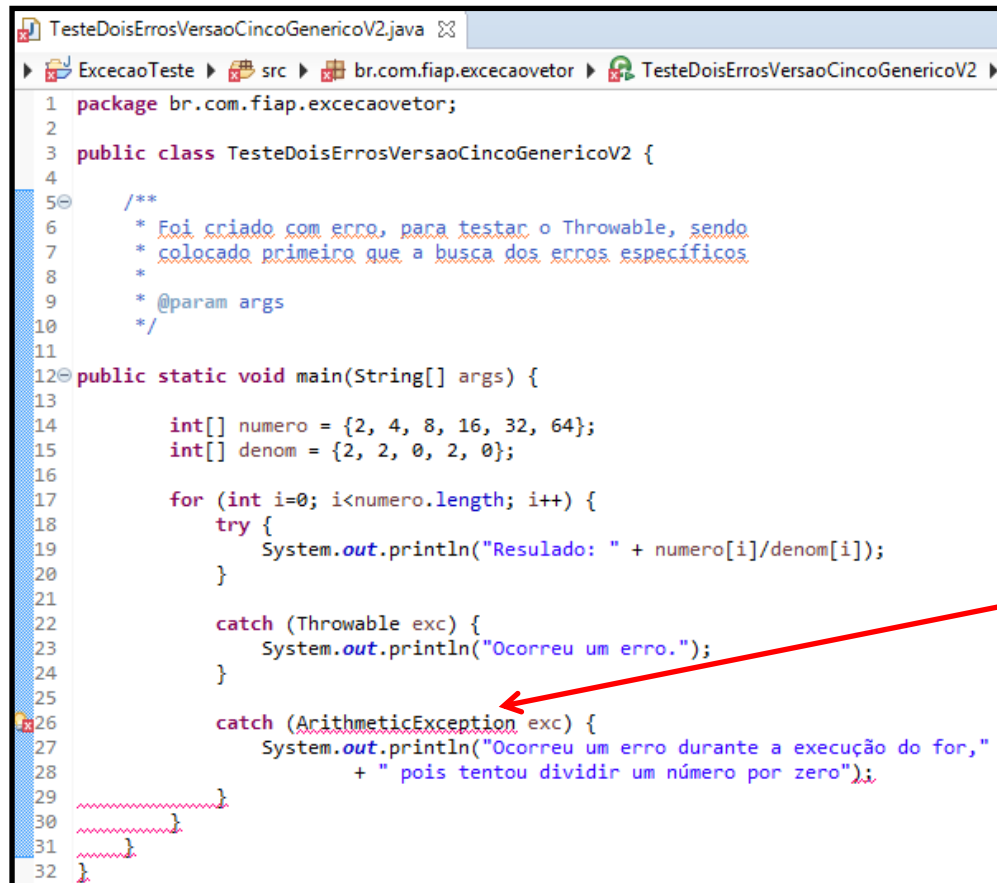
Exception – Fazendo tratamento de exceções no Java

- Utilizando o Throwable para erros genéricos/desconhecidos

FIAP

Está ocorrendo esse erro, pois primeiro eu coloquei o Throwable que captura todos os erros. Para o Java o correto primeiro é colocar a busca dos erros específicos e depois os genéricos (Throwable).

O correto para o Java é como fizemos na classe “TesteDoisErrosVersaoCincoGenerico”, primeiro Capturar os erros mais específicos e depois colocamos um catch para capturar um erro mais genérico.



```
1 package br.com.fiap.excecao vetor;
2
3 public class TesteDoisErrosVersaoCincoGenericoV2 {
4
5     /**
6      * Foi criado com erro, para testar o Throwable, sendo
7      * colocado primeiro que a busca dos erros específicos
8      *
9      * @param args
10     */
11
12     public static void main(String[] args) {
13
14         int[] numero = {2, 4, 8, 16, 32, 64};
15         int[] denom = {2, 2, 0, 2, 0};
16
17         for (int i=0; i<numero.length; i++) {
18             try {
19                 System.out.println("Resultado: " + numero[i]/denom[i]);
20             }
21
22             catch (Throwable exc) {
23                 System.out.println("Ocorreu um erro.");
24             }
25
26             catch (ArithmeticException exc) {
27                 System.out.println("Ocorreu um erro durante a execução do for,"
28                     + " pois tentou dividir um número por zero");
29             }
30         }
31     }
32 }
```

Exception – Fazendo tratamento de exceções no Java

- Utilizando o Throwable para erros genéricos/desconhecidos



```
package br.com.fiap.excecaoovetor;
```

```
public class TesteDoisErrosVersaoSeisGenerico {
```

```
    public static void main(String[] args) {
```

```
        int[] numero = {2, 4, 8, 16, 32, 64};
```

```
        int[] denom = {2, 2, 0, 2, 0};
```

```
        int[] resul = {1, 2, 3, 4, 5, 6};
```

```
        for (int i=0; i<numero.length; i++) {
```

```
            try {
```

```
                System.out.println("Resultado: " + numero[i]/denom[i]);
```

```
                resul[i] = numero[i]/denom[i];
```

```
            }
```

```
            catch (ArithmeticException exc) {
```

```
                System.out.println("Ocorreu um erro durante a execução do for,"  
                                    + " pois tentou dividir um número por zero");
```

```
            }
```

```
        /**
```

```
         * @author Ronqui
```

```
         * Throwable é a superclasse que captura todos os erros
```

```
         * e exceções no Java
```

```
         */
```

```
            catch (Throwable exc) {
```

```
                System.out.println("Ocorreu um erro.");
```

```
            }
```

```
        }
```

```
        for (int i=0; i<numero.length; i++) {
```

```
            System.out.println(resul[i]);
```

```
        }
```

```
    }
```

```
}
```

Exception – Fazendo tratamento de exceções no Java

- Utilizando o Throwable para erros genéricos/desconhecidos

FIAP

```
TesteDoisErrosVersaoSeisGenerico.java
ExcecaoTeste > src > br.com.fiap.excecaoavetor > TesteDoisErrosVersaoSeisGenerico

7      int[] numero = {2, 4, 8, 16, 32, 64};
8      int[] denom = {2, 2, 0, 2, 0};
9      int[] resul = {1, 2, 3, 4, 5, 6};
10
11     for (int i=0; i<numero.length; i++) {
12         try {
13             System.out.println("Resultado: " + numero[i]/denom[i]);
14             resul[i] = numero[i]/denom[i];
15         }
16         catch (ArithmeticException exc) {
17             System.out.println("Ocorreu um erro durante a execução do for,"
18                 + " pois tentou dividir um número por zero");
19         }
20     /**
21      * @author Ronqui
22      * Throwable é a superclasse que captura todos os erros
23      * e exceções no Java
24      */
25     catch (Throwable exc) {
26         System.out.println("Ocorreu um erro.");
27     }
28 }
29
30 for (int i=0; i<numero.length; i++) {
31     System.out.println(resul[i]);
32 }
```

Problems @ Javadoc Declaration Console

```
<terminated> TesteDoisErrosVersaoSeisGenerico [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\
Resultado: 1
Resultado: 2
Ocorreu um erro durante a execução do for, pois tentou dividir um número por zero
Resultado: 8
Ocorreu um erro durante a execução do for, pois tentou dividir um número por zero
Ocorreu um erro.
1
2
3
8
5
6
```

No código ao lado estamos fazendo a inserção dos resultados das divisões no veto resul (linha 14), após estamos mostrando os valores do vetor resul na linha 31.

Exception – Fazendo tratamento de exceções no Java

- Utilizando o finally, após o try catch



```
package br.com.fiap.excecao vetor;

public class TesteDoisErrosVersaoSetefinally {

    public static void main(String[] args) {

        int[] numero = {2, 4, 8, 16, 32, 64};
        int[] denom = {2, 2, 0, 2, 0};
        int[] resul = {1, 2, 3, 4, 5, 6};

        for (int i=0; i<numero.length; i++) {
            try {
                System.out.println("Resultado: " + numero[i]/denom[i]);
                resul[i] = numero[i]/denom[i];
            }
            catch (ArithmeticException exc) {
                System.out.println("Ocorreu um erro durante a execução do for,"
                    + " pois tentou dividir um número por zero");
            }
            /**
             * @author Ronqui
             * Throwable é a superclasse que captura todos os erros
             * e exceções no Java
             */
            catch (Throwable exc) {
                System.out.println("Ocorreu um erro.");
            }
            finally {
                System.out.println("Utilizando o finally, após o try catch.");
            }
        }

        for (int i=0; i<numero.length; i++) {
            System.out.println(resul[i]);
        }
    }
}
```

Exception – Fazendo tratamento de exceções no Java

- Utilizando o finally, após o try catch

```
TesteDoisErrosVersaoSetefinally.java
ExcecaoTeste > src > br.com.fiap.excecaoovetor > TesteDoisErrosVersaoSetefinally

13      System.out.println("Resultado: " + numero[i]/denom[i]);
14      resul[i] = numero[i]/denom[i];
15  }
16  catch (ArithmeticException exc) {
17      System.out.println("Ocorreu um erro durante a execução do for,"
18          + " pois tentou dividir um número por zero");
19  }
20  /**
21   * @author Ronqui
22   * Throwable é a superclasse que captura todos os erros
23   * e exceções no Java
24   */
25  catch (Throwable exc) {
26      System.out.println("Ocorreu um erro.");
27  }
28  finally {
29      System.out.println("Utilizando o finally, após o try catch.");
30  }
31  }
32
33  for (int i=0; i<numero.length; i++) {
34      System.out.println(resul[i]);
35  }
36  }
37  }
```

Problems @ Javadoc Declaration Console

```
<terminated> TesteDoisErrosVersaoSetefinally [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\java
Utilizando o finally, após o try catch.
Ocorreu um erro durante a execução do for, pois tentou dividir um número por zero
Utilizando o finally, após o try catch.
Ocorreu um erro.
Utilizando o finally, após o try catch.
1
2
3
8
5
6
```

O finally sempre será executado após o try / catch.

O finally só não será executado, se inserir no código o comando `System.exit(0);`. Esse comando `System.exit(0);` iremos verificar o funcionamento dele no próximo slide.

Exception – Fazendo tratamento de exceções no Java



- Utilizando o **finally**, após o **try catch** com **getMessage** e **printStackTrace**

```
package br.com.fiap.excecao vetor;
```

```
public class TesteDoisErrosVersaoOito {
```

```
    public static void main(String[] args) {
```

```
        int[] numero = {2, 4, 8, 16, 32, 64};
```

```
        int[] denom = {2, 2, 0, 2, 0};
```

```
        for (int i=0; i<numero.length; i++) {
```

```
            try {
```

```
                System.out.println("Resultado: " + numero[i]/denom[i]);
```

```
            }
```

```
            catch (Exception e) {
```

```
                System.out.println(" \n" + e.getMessage());
```

```
                e.printStackTrace();
```

```
            }
```

```
        }
```

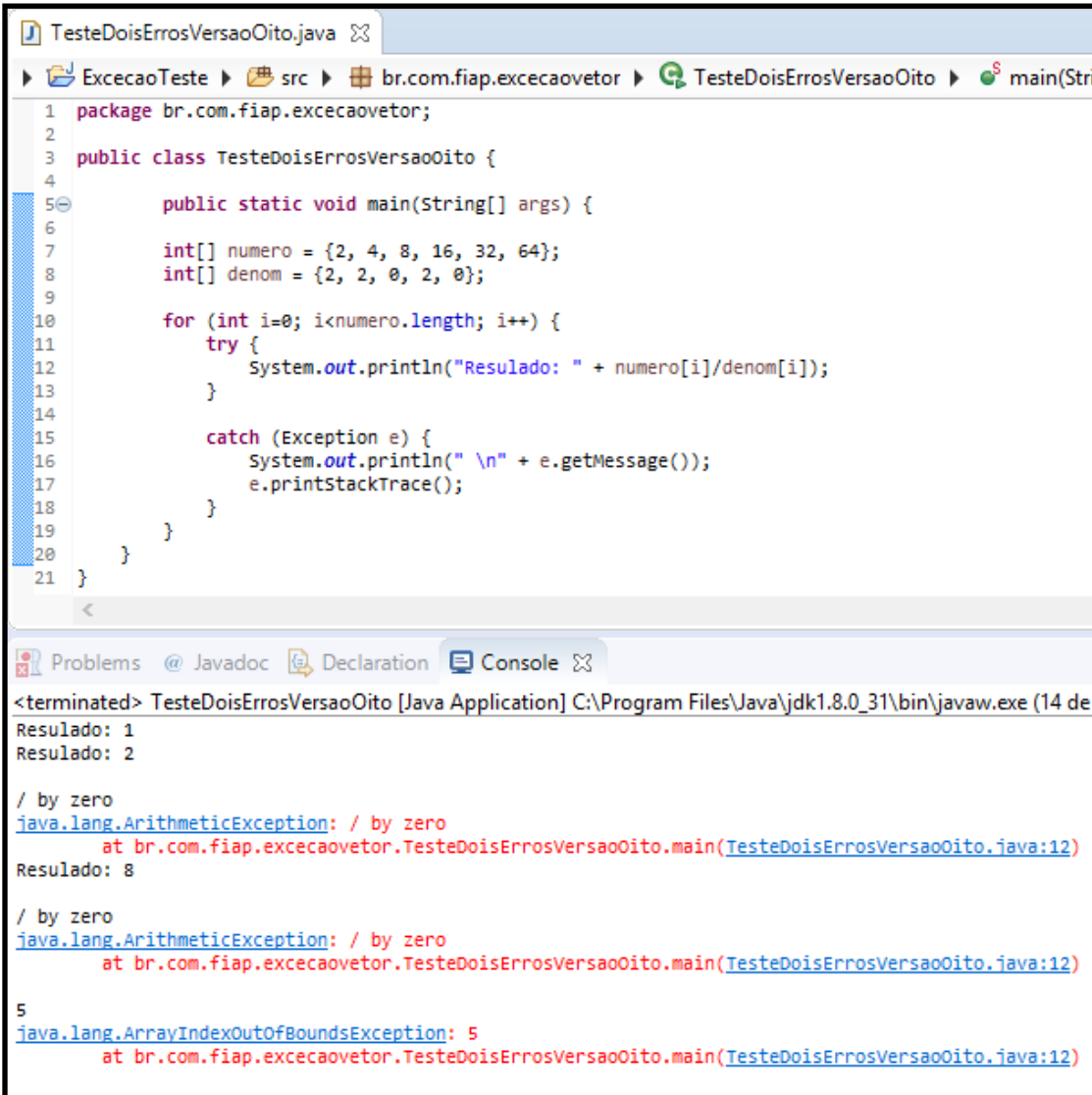
```
    }
```

```
}
```


Exception – Fazendo tratamento de exceções no Java

- Utilizando o finally, após o try catch com
getMessage e printStackTrace

O getMessage() mostra o erro que ocorreu e o printStackTrace mostra a linha que ocorreu o erro.



The screenshot shows an IDE with a Java file named `TesteDoisErrosVersaoOito.java`. The code defines a `main` method that iterates over two arrays: `numero` and `denom`. For each pair of elements, it attempts to calculate the division `numero[i] / denom[i]` within a `try` block. If an `Exception` occurs, it is caught in a `catch` block, and the error message is printed using `e.getMessage()` and the stack trace is printed using `e.printStackTrace()`.

The console output shows the results of the execution:

```
<terminated> TesteDoisErrosVersaoOito [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\javaw.exe (14 de
Resulado: 1
Resulado: 2

/ by zero
java.lang.ArithmeticException: / by zero
    at br.com.fiap.excecao vetor.TesteDoisErrosVersaoOito.main(TesteDoisErrosVersaoOito.java:12)
Resulado: 8

/ by zero
java.lang.ArithmeticException: / by zero
    at br.com.fiap.excecao vetor.TesteDoisErrosVersaoOito.main(TesteDoisErrosVersaoOito.java:12)

5
java.lang.ArrayIndexOutOfBoundsException: 5
    at br.com.fiap.excecao vetor.TesteDoisErrosVersaoOito.main(TesteDoisErrosVersaoOito.java:12)
```

Exception – Fazendo tratamento de exceções no Java

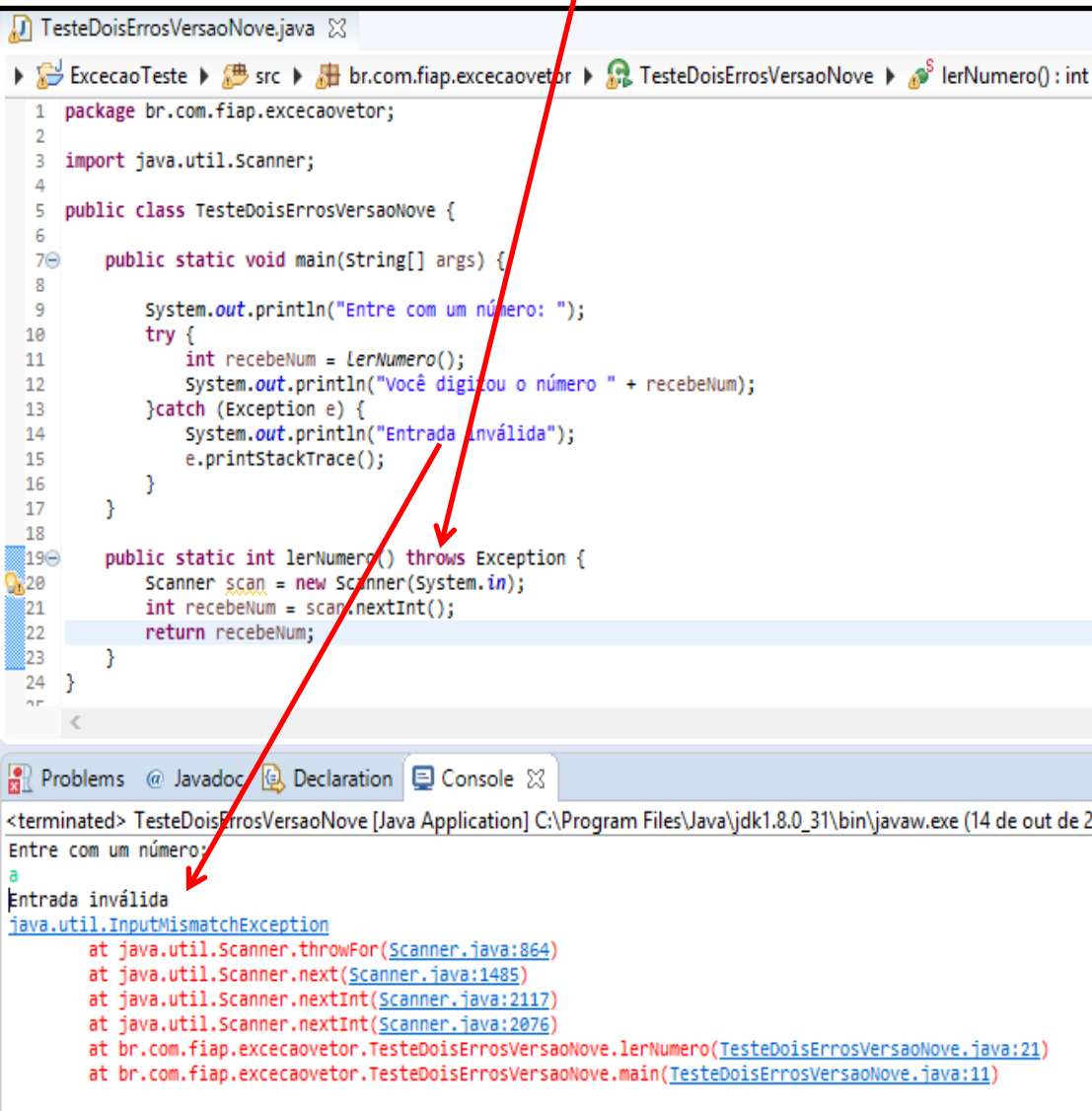
- Utilizando o throws dentro de um método



```
package br.com.fiap.excecao vetor;  
  
import java.util.Scanner;  
  
public class TesteDoisErrosVersaoNove {  
  
    public static void main(String[] args) {  
  
        System.out.println("Entre com um número: ");  
        try {  
            int recebeNum = lerNumero();  
            System.out.println("Você digitou o número " + recebeNum);  
        } catch (Exception e) {  
            System.out.println("Entrada inválida");  
            e.printStackTrace();  
        }  
    }  
  
    public static int lerNumero() throws Exception {  
        Scanner scan = new Scanner(System.in);  
        int recebeNum = scan.nextInt();  
        return recebeNum;  
    }  
}
```

Exception – Fazendo tratamento de exceções no Java

- Utilizando o throws dentro de um método



```
TesteDoisErrosVersaoNove.java
br.com.fiap.excecaoetor
TesteDoisErrosVersaoNove
lerNumero(): int
1 package br.com.fiap.excecaoetor;
2
3 import java.util.Scanner;
4
5 public class TesteDoisErrosVersaoNove {
6
7     public static void main(String[] args) {
8
9         System.out.println("Entre com um número: ");
10        try {
11            int recebeNum = lerNumero();
12            System.out.println("Você digitou o número " + recebeNum);
13        } catch (Exception e) {
14            System.out.println("Entrada inválida");
15            e.printStackTrace();
16        }
17    }
18
19    public static int lerNumero() throws Exception {
20        Scanner scan = new Scanner(System.in);
21        int recebeNum = scan.nextInt();
22        return recebeNum;
23    }
24 }
Problems @ Javadoc Declaration Console
<terminated> TesteDoisErrosVersaoNove [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\javaw.exe (14 de out de 2
Entre com um número:
a
Entrada inválida
java.util.InputMismatchException
    at java.util.Scanner.throwFor(Scanner.java:864)
    at java.util.Scanner.next(Scanner.java:1485)
    at java.util.Scanner.nextInt(Scanner.java:2117)
    at java.util.Scanner.nextInt(Scanner.java:2076)
    at br.com.fiap.excecaoetor.TesteDoisErrosVersaoNove.lerNumero(TesteDoisErrosVersaoNove.java:21)
    at br.com.fiap.excecaoetor.TesteDoisErrosVersaoNove.main(TesteDoisErrosVersaoNove.java:11)
```

Nessa classe utilizamos o throws dentro do método, pois dessa maneira a responsabilidade do tratamento do erro, fica para quem chamou o método.

EXCEPTIONS MAIS COMUNS

Exception: é a super-classe de todas as classes que tratam exceções.

NullPointerException: quando você aponta para algo que está nulo, como por exemplo, um atributo de um objeto nulo.

ArrayIndexOutOfBoundsException: dispara quando você aponta para uma posição que não existe dentro de uma matriz. Exemplo, apontar para o elemento 11 sendo que a matriz possui 10 elementos.

NumberFormatException: lança a exceção quando o formato do dado passado não condiz com o tipo de dado esperado, exemplo, quando digita-se uma letra para um tipo de dado inteiro.

FileNotFoundException: quando não encontra o arquivo especificado no código.

IllegalArgumentException: dispara quando é fornecido um parâmetro fora dos padrões estabelecidos, por exemplo, uma referência nula.

FIM da explicação

I Dúvidas...

