

Unchecked Exception e Checked Exception

```
5 public class Exemplo_1 {
6     public static void main(String[] args) {
7         Scanner scan = new Scanner(System.in);
8
9         int resultado = dividir(scan.nextInt(), scan.nextInt());
10        System.out.println(resultado);
11    }
12
13    public static int dividir(int a, int b) {
14        return a / b;
15    }
16 }
```

Errors e RuntimeExceptions são considerados *unchecked*, portanto o compilador não obriga que exista tratamento para eles.

Unchecked Exception

```
Exemplo_1 x
/home/cami/programs/jdk-17/bin/java -javaagent:/home/cami/programs/ides/ideaIC-2021.
4
0
Exception in thread "main" java.lang.ArithmeticException: / by zero
at br.com.dio.Exemplo_1.dividir(Exemplo_1.java:14)
at br.com.dio.Exemplo_1.main(Exemplo_1.java:9)
Process finished with exit code 1
```

Unchecked Exception

- Herdam da classe RuntimeException ou da classe Error.
- O compilador não verifica o código para ver se a exceção foi capturada ou declarada.
- Se uma exceção não-verificada ocorrer e não tiver sido capturada, o programa terminará ou executará com resultados inesperados.
- Em geral, podem ser evitadas com uma codificação adequada.

```
5 public class Exemplo_2 {
6     public static void main(String[] args) {
7         imprimeArquivoNoConsole("romances-blake-crouch.txt");
8     }
9
10    private static void imprimeArquivoNoConsole(String nomeDoArquivo) {
11        File file = new File(nomeDoArquivo);
12        BufferedReader br = new BufferedReader(new FileReader(file.getName()));
13        String line = br.readLine();
14
15        BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(System.out));
16
17        do{
18            bw.write(line);
19            bw.newLine();
20            line=br.readLine();
21        } while(line != null);
22        bw.flush();
23        br.close();
24    }
25 }
```

Checked Exception

Costumam indicar que uma condição necessária para a execução de um programa não está presente.

Build: Build Output x
exceptions-java: build failed At 9/27/21, 3:33 PM with 7 errors 996 ms
Exemplo_2.java src/br/com/dio/exemplos 7 errors
unreported exception java.io.FileNotFoundException; must be caught or declared to be thrown :12
unreported exception java.io.IOException; must be caught or declared to be thrown :13
unreported exception java.io.IOException; must be caught or declared to be thrown :18
unreported exception java.io.IOException; must be caught or declared to be thrown :19
unreported exception java.io.IOException; must be caught or declared to be thrown :20
unreported exception java.io.IOException; must be caught or declared to be thrown :22
unreported exception java.io.IOException; must be caught or declared to be thrown :23

Checked Exception

- As exceções que são herdadas da classe Exception, mas não de RuntimeException.
- O compilador impõe um requisito do tipo 'capturar ou declarar'.
- O compilador verifica cada chamada de método e declaração de método para determinar se o método lança (throws) exceções verificadas. Se lançar, o compilador assegura que a exceção verificada é capturada ou declarada em uma cláusula throws. Caso não capturada nem declarada, ocorre um erro de compilação.