



# Teste de Conhecimento

avalie sua aprendizagem

PROGRAMAÇÃO I  
CCT0827\_A9\_20210110137\_V1Lupa Calc.  
🔍 🧮Aluno: DOUGLAS MATOS DA SILVA  
Disc.: PROGRAMAÇÃO IMatr.: 20210110137  
2022.1 EAD (GT) / EX

Prezado (s) Aluno(a),

Você fará agora seu **TESTE DE CONHECIMENTO!** Lembre-se que este exercício é opcional, mas não valerá ponto para sua avaliação. O mesmo será composto de questões de múltipla escolha.

Após responder cada questão, você terá acesso ao gabarito comentado e/ou à explicação da mesma. Aproveite para se familiarizar com este modelo de questões que será usado na sua AV e AVS.

**1.** Qual a classe representa situação anormais que podem acontecer na JVM?

- ☐ Throws.Error  
☐ Throw.Error  
☐ Erros  
☐ Erro  
☒ Error

**2.** Considere o trecho abaixo:

```
try {  
    // Código a ser tratado  
} catch (Exception e) {  
    System.out.println("Erro: " + e.getMessage());  
} finally {  
    // código  
}
```

Julgue as afirmativas a seguir:

- I-A variável **e** referencia a exceção que ocorreu.  
II- finally não é obrigatório. Deve ser usado para instruções de limpeza  
III-A variável **e** possibilita acessar informações sobre a exceção, como por exemplo, com o uso do método `getMessage`, evidenciado acima.

Estão corretas:

- ☐ Somente I  
☒ Todas as afirmações  
☐ Somente II e III  
☐ Somente I e III  
☐ Somente I e II

**3.** Na linguagem Java, a sintaxe correta de um bloco de controle de exceção é:

- ☐ try { // código a ser executado } catch (nomeExcecao: TipoExcecao) { // tratamento da exceção }  
☐ try { // código a ser executado } except { // tratamento da exceção }  
☒ try { // código a ser executado } catch (TipoExcecao nomeExcecao) { // tratamento da exceção }  
☐ switch (nomeExcecao) { default: // código a ser executado break; case TipoExcecao: // tratamento da exceção break; }  
☐ if (TipoExcecao) { // código a ser executado } else { // tratamento da exceção }

**4.** A classe \_\_\_\_\_ é a classe mãe de todas as exceções que nos interessam. Ela possui alguns métodos úteis e comuns a todas as exceções.

A classe acima mencionada é:

- ☐ Throws  
☐ Throw  
☒ Exception  
☐ Error  
☐ getMessage

**5.** Considere o trecho a seguir no Java:

```
try {  
    // Código a ser tratado  
} catch (Exception e) {  
    // Código que será executado quando ocorrer a exceção  
}
```

Julgue as afirmativas a seguir:

- I- Se ocorrer uma exceção no bloco do try, então a execução é automaticamente desviada para o bloco catch.  
II- No catch devemos definir a exceção a ser tratada. Quando definimos uma exceção estamos tratando também todas as suas subclasses.

III- O **e**, mostrado na linha do catch, referencia a exceção que ocorreu. Com ela é possível acessar informações sobre essa exceção.

Estão corretas:

- ☐ Somente I, III  
☐ Somente I, II  
☐ Somente II, III  
☒ Todas as afirmações  
☐ Somente I

**6.** Se escrevermos um programa para ler e imprimir arquivo texto ocorrerá exceção do tipo `FileNotFoundException` ou `IOException`. Note que, neste caso, o Java não compile esse código! Para conseguirmos compilar nesse caso, o que devemos usar?

- ☐ case  
☐ while  
☐ for  
☐ if  
☒ throws

**7.** As exceções em Java estão organizadas em uma hierarquia de classes: No topo da hierarquia temos a classe `Throwable`, qual são suas classes filhas?

- ☐ Erro e Except  
☒ Error e Exception  
☐ Except  
☐ Erro  
☐ Error e Except

**8.** Como se sabe, em algumas situações, é necessário tratar casos especiais no âmbito da programação. Alguns desses casos podem ser a divisão por zero, erros de aberturas de arquivos, erros de conversão de tipos, entre outros.

No Java, isso é chamado de:

- ☒ Tratamento de exceções (exception)  
☐ Tratamento orientados a casos especiais (case-oriented)  
☐ Tratamento de compilação e execução (compile-and-run)  
☐ Tratamento de erros matemáticos (math-error)

Tratamento de código (code-care)

Col@bore

Sugira! Sinalize! Construa!  
Antes de finalizar, clique aqui para dar a sua opinião sobre as questões deste exercício.

■ Não Respondida ■ Não Gravada ■ Gravada

Exercício iniciado em 20/04/2022 08:46:19.