



**ROTEIRO AULA PRÁTICA**

**Unidade 3 | Seção 1**



# Linguagem orientada a objetos

# ROTEIRO DE AULA PRÁTICA

**NOME DA DISCIPLINA:** Linguagem orientada a objetos

**Desenvolvimento de classes abstratas, exceções e interfaces orientada a objetos**

**Tratamento de exceções e uso de classes abstratas**

## OBJETIVOS

Definição dos objetivos da aula prática:

Desenvolver uma aplicação com tratamento de exceção.

Criar as suas próprias exceções em Java de forma checada e não-checada.

Fazer a leitura de valores utilizando a classe Scanner.

Reforçar o conhecimento sobre a utilização da plataforma github.

## INFRAESTRUTURA

Instalações:

Laboratório de informática com computadores e com acesso à internet. Caro professor, verifique em sua instituição a necessidade de reservar o laboratório.

Materiais:

Descrição	Quantidade de materiais por procedimento/atividade
Computadores com Internet	Um para cada dois alunos

Software:

Sim ( X ) Não ( )

Em caso afirmativo, qual?	Algum IDE que dê suporte a Java (Sugestões: Netbeans, Eclipse ou IntelliJ IDEA).
---------------------------	--

Pago ( ) Não Pago ( X )

Tipo de licença:	Livre
------------------	-------

Descrição do software:

O IDE é um ambiente de programação que auxilia a vida do programador. Deverá ser utilizado algum IDE, como: Netbeans, Eclipse ou IntelliJ IDEA. Professor, fique à vontade para escolher o IDE que mais lhe agrade, ou ainda, o que tiver instalado no laboratório de informática.

Sugestões de segurança:

Não se aplica.

## PROCEDIMENTOS PRÁTICOS

Neste momento você deve ajudar o professor por meio da descrição de todas as etapas que deverão ser realizadas para a execução dos procedimentos práticos. Considerando a carga horária da aula prática, você pode replicar a caixa de procedimento/atividade quantas vezes for necessário.

### Procedimento/Atividade n.1

Atividade proposta:

Desenvolvimento de um código que encontre as raízes de equações do segundo grau. Caso não exista raízes reais uma exceção deve ser lançada. Deve-se criar métodos que lancem exceções checadas e não-cheçadas. Ao final submeter o código no github.

Procedimentos para a realização da atividade:

A ideia geral é realizar a leitura de três valores reais utilizando a classe Scanner. Esses três valores representam os coeficientes de uma equação do segundo grau. Todos sabemos que o usuário pode inserir um valor errado na leitura tipo uma string. Assim a leitura desses valores reais deve ser persistida até que os três valores reais sejam lidos. Assim, na leitura de cada um desses valores deve ter um tratamento de exceção apropriado juntamente com um loop caso a leitura dê errado. O que se deseja é calcular as raízes dessas equações. As equações do 2 grau podem possuir 2 raízes reais que são nomeadas de  $x_1$  e  $x_2$ . Caso uma equação do 2 grau não tenha raízes reais o valor de delta é negativo. Lembre-se  $\Delta = b^2 - 4 * a * c$ . A ideia é que caso o valor de delta seja negativo uma exceção deverá ser lançada. Essa exceção deverá ser tratada apenas na aplicação principal não no método. Após o término da programação crie uma conta no GitHub e envie o código para essa plataforma.

Uma equação do segundo grau é da forma:

$$ax^2 + bx + c = 0$$

As raízes dessa equação acima são dadas conforme as formulas de Bhaskara abaixo.

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ e } x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Link: [https://pt.wikipedia.org/wiki/Equa%C3%A7%C3%A3o\\_quadr%C3%A1tica](https://pt.wikipedia.org/wiki/Equa%C3%A7%C3%A3o_quadr%C3%A1tica)

#### Checklist:

Abrir a IDE de sua escolha.

- Crie uma classe chamada AulaPratica1 que conterà o método main.
  - Então, deve-se fazer a leitura de um valor real 'a' (primeiro termo da equação).
  - Aqui é importante que se trate exceções, por exemplo se o usuário digitar uma string deve-se pedir para ele digitar o valor novamente, até que um valor real seja lido. A exceção que trata valores entrados indevidamente é InputMismatchException.
  - Então, deve-se fazer a leitura de um valor real 'b' (segundo termo da equação).
  - Garantir novamente que um valor real será lido (igual explicado acima).
  - Então, deve-se fazer a leitura de um valor real 'c' (terceiro termo da equação).
  - Garantir novamente que um valor real será lido (igual explicado acima).
  - Após lidos os valores de a, b e c devemos calcular as raízes da equação do 2 grau.
- Criar uma classe chamada Equacao.
  - Criar quatro métodos estáticos dentro da classe Equacao. As assinaturas dos métodos devem ser semelhantes a essas.
    - `public static double raizEq2GrauX1(double a, double b, double c) throws NaoTemRaizesReaisException`
    - `public static double raizEq2GrauX2(double a, double b, double c) throws NaoTemRaizesReaisException`
    - `public static double raizEq2GrauX1_RT(double a, double b, double c) throws NaoTemRaizesReaisRuntimeException`
    - `public static double raizEq2GrauX2_RT(double a, double b, double c) throws NaoTemRaizesReaisRuntimeException`
    - Os métodos acima retornam as raízes da equação do 2 grau. Se tiver dúvida olhe as equações acima. O ponto principal aqui é que os dois primeiros métodos lançam exceções checadas. Já os dois últimos lançam exceções não-checadas.

- Criar uma classe que lance exceções de forma checada quando não houver raízes reais, sugestão de nome da classe `NaoTemRaizesReaisException`. Essa classe deve herdar a classe `Exception`.
- Criar uma classe que lance exceções de forma não-checada quando não houver raízes reais, sugestão de nome da classe `NaoTemRaizesReaisRuntimeException`. Essa classe deve herdar a classe `RuntimeException`.
- A classe `AulaPratica1` deverá chamar os métodos que lançam exceções checadas. Deve-se criar os blocos de try-catch adequados. E imprimir os valores das raízes. Deve-se realizar diversos testes para verificar se tudo funcionou corretamente.
- Após feito o trabalho acima, a ideia é replicar a classe `AulaPratica1` com o nome `AulaPratica2` e então chamar os métodos que lançam exceções não-cheçadas.
- Ao final da aula enviar os códigos para o GitHub.

## RESULTADOS

### Resultados da aula prática:

Ao final da aula o aluno deverá enviar para o professor o link do github do aluno no qual foi submetido o código desenvolvido em aula (enviar de preferência através de algum ambiente virtual que o professor utiliza tipo moodle ou AVA).