

### Aula 04

## Exercícios Turma 5

### Professores:

Anarosa Alves Franco Brandão (PCS 2302)  
Marcos A. Simplicio Junior (PCS 2302/2024)  
Ricardo L. A. Rocha (PCS 2024)

**Monitores:** Felipe Leno, Michel Bieleveld e Diego Queiroz



# Exercício 01

- Considere o domínio da geometria analítica, onde polígonos representam figuras geométricas convexas com pelo menos 3 vértices. A verificação de que um polígono é triângulo ou quadrilátero pressupõe avaliar o número de vértices e uma propriedade geométrica que garanta convexidade. Polígonos podem ser categorizados através de propriedades algébricas. Por exemplo, três vetores no plano cartesiano definem um triângulo se representarem uma forma fechada e pelo menos dois deles forem linearmente independentes (ou seja, o determinante da matriz formada por pelo menos dois deles é diferente de 0).
- Este domínio é interessante para treinarmos os fundamentos de herança e polimorfismo.



# Exercício 01

- Para efeito de testes de triângulos use os seguintes vetores:
  - $(0,0), (0,1), (0,2)$  (não é triângulo)
  - $(1,1), (2,2), (-3,-3)$  (não é triângulo)
  - $(0,2), (1,0), (-1,-2)$  (escaleno)
  - $(0,1), (1,0), (-1,-1)$  (isósceles)
  - $(4,0), (-2,2\sqrt{3}), (-2,-2\sqrt{3})$  (equilátero)
- Obs.: você pode usar os métodos da classe Math para auxiliar no desenvolvimento

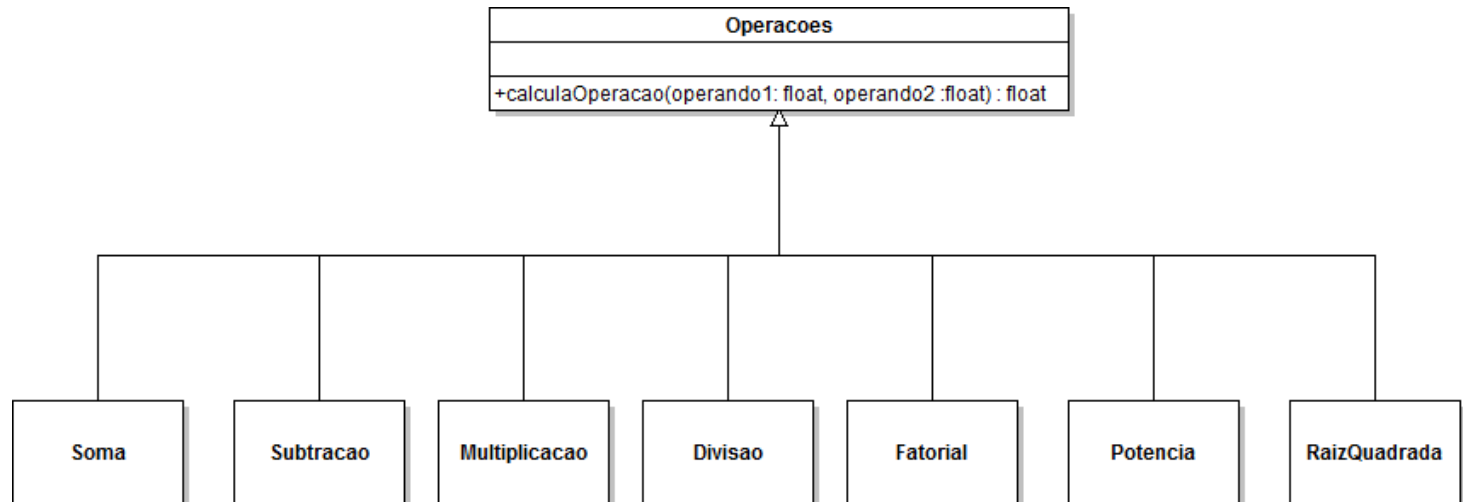
# Exercício 01

- O exercício consiste em:
  - a) Para os objetos Vector, Poligono, Triangulo, Quadrilatero, TrianguloIsosceles, TrianguloEquilatero, TrianguloEscaleno, Retangulo, Quadrado, construa um diagrama de classes em UML indicando os relacionamentos de herança. Inclua na classe Poligono os métodos  
`boolean validar()`, `float perimetro()` e `void imprime()`.
  - b) Baixe o arquivo `codigos.zip` que já terá uma parte da implementação feita. Complete o que estiver indicado com **TODO** (obs.: não é necessário criar novas classes).  
Preencha a **abstração procedimental** (preparação para JavaDoc) dos métodos a serem programados.
  - c) Submeta o exercício com o nome **TYGXXA04E01.zip**, incluindo todos os arquivos `.java` no arquivo zipado.

Use VioletUML para modelar o diagrama de classes

## Exercício 02

- Agora você está encarregado de desenvolver algumas das classes que serão utilizadas em uma calculadora simples.
  - Você terá que desenvolver as classes que realizam operações matemáticas. O seguinte modelo foi contruído:





## Exercício 02

- As classes Operacoes e Soma já foram implementadas, agora implemente as outras classes.
- Para o fatorial e a raiz quadrada, ignore o segundo operando.

Submeta o exercício com o nome  
**TYGXXA04E02.zip**, incluindo todos os arquivos  
.java no arquivo zipado.