

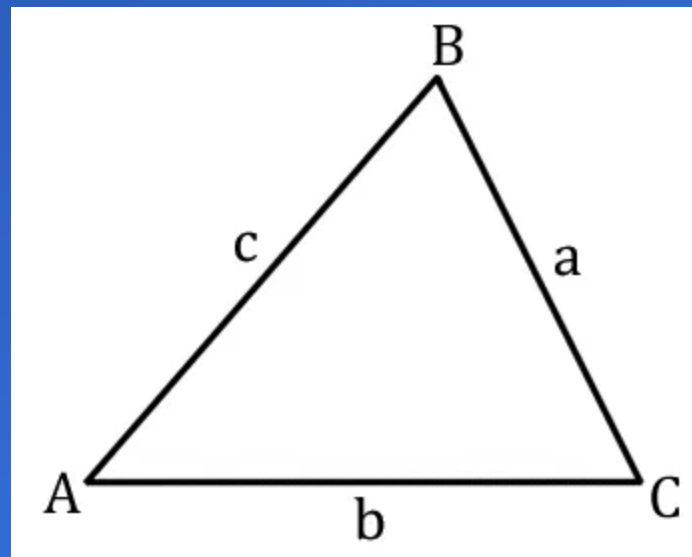
# Resolução da Questão 1

## Classe Triangulo

### Enunciado:

Crie uma classe chamada **Triangulo** contendo os três lados como atributos (a, b e c) e os seguintes métodos:

- Método construtor para inicializar o triângulo, desde que seja válido.
- Método **ehValido** que retorna verdadeiro se o triângulo é válido, senão retorna falso.
- Método **tipoTriangulo** que retorna o tipo do triângulo (equilátero, isósceles ou escaleno).
- Método **calculaPerimetro** que retorna o perímetro do triângulo.
- Método **calculaArea** que retorna a área do triângulo usando a fórmula de Heron.
- Crie uma classe de teste para testar a sua classe com os diferentes métodos e situações.



# Estrutura da Classe Triangulo

Atributos, construtor e validação

## Atributos e Construtor:

```
public class Triangulo {  
    private double a, b, c;  
  
    public Triangulo(double a, double b, double c) {  
        if (ehValido(a, b, c)) {  
            this.a = a;  
            this.b = b;  
            this.c = c;  
        } else {  
            System.out.println("Triângulo inválido. Os lados não formam um triângulo.");  
        }  
    }  
}
```

## Características:

- Três atributos privados (a, b, c) representando os lados
- Construtor que valida se os lados formam um triângulo válido
- Mensagem de erro caso os lados não formem um triângulo

Triangulo
- a: double
- b: double
+ Triangulo(a: double, b: b: c)
+ ehValido(a: double, b: beeaan
+ tipoTrianglo::): String
+ calculaPerimeto): doube
+ calculaArea): doube

# Método ehValido

Verificação da condição de existência de um triângulo

## Implementação:

```
public boolean ehValido(double a, double b, double c) {  
    return (a + b > c) && (a + c > b) && (b + c > a);  
}
```

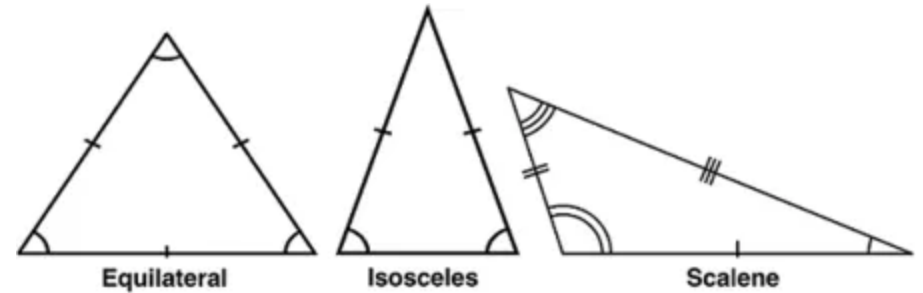
### Condição de existência:

Para que três segmentos de reta formem um triângulo, é necessário que:

- A soma de dois lados seja maior que o terceiro lado
- Esta condição deve ser verificada para todas as combinações de lados

### Casos inválidos:

- Lados negativos ou zero
- Um lado muito grande em relação aos outros
- Exemplo: Triângulo(1, 2, 5) é inválido pois  $1 + 2 \leq 5$



# Método tipoTriangulo

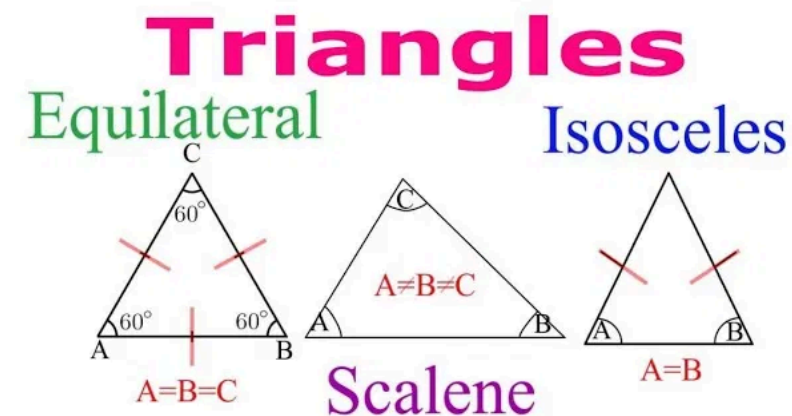
Classificação do triângulo quanto aos lados

## Implementação:

```
public String tipoTriangulo() {  
    if (!ehValido(a, b, c)) {  
        return "Inválido";  
    }  
    if (a == b && b == c) {  
        return "Equilátero";  
    } else if (a == b || a == c || b == c) {  
        return "Isósceles";  
    } else {  
        return "Escalaeno";  
    }  
}
```

## Classificação dos triângulos:

- **Equilátero:** Três lados iguais ( $a = b = c$ )
- **Isósceles:** Dois lados iguais ( $a = b$  ou  $a = c$  ou  $b = c$ )
- **Escalaeno:** Três lados diferentes ( $a \neq b \neq c$ )
- **Inválido:** Não forma um triângulo



# Métodos calculaPerimetro e calculaArea

Cálculo do perímetro e da área usando a fórmula de Heron

## Implementação:

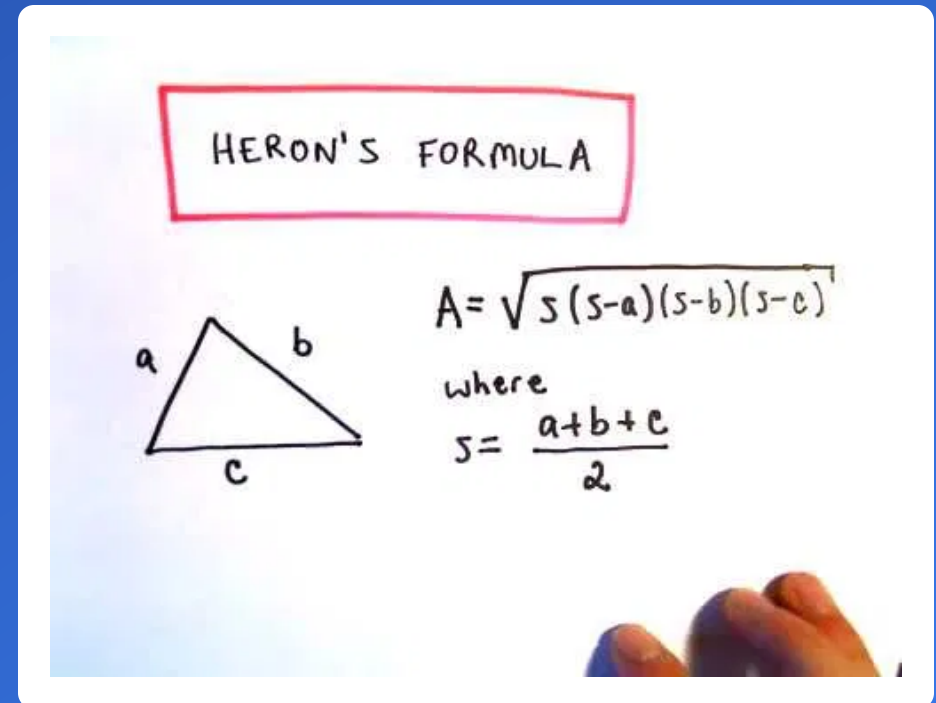
```
public double calculaPerimetro() {  
    if (!ehValido(a, b, c)) {  
        return 0;  
    }  
    return a + b + c;  
}  
  
public double calculaArea() {  
    if (!ehValido(a, b, c)) {  
        return 0;  
    }  
    double p = calculaPerimetro() / 2; // Semiperímetro  
    return Math.sqrt(p * (p - a) * (p - b) * (p - c));  
}
```

## Fórmula de Heron:

A fórmula de Heron permite calcular a área de um triângulo conhecendo apenas o comprimento dos seus três lados:

$$\text{Área} = \sqrt{p \times (p - a) \times (p - b) \times (p - c)}$$

onde  $p = (a + b + c) / 2$  (semiperímetro)



# Classe de Teste

Verificação das funcionalidades da classe Triangulo

## Casos de Teste:

```
public class TesteTriangulo {
    public static void main(String[] args) {
        // Teste 1: Triângulo Equilátero válido
        Triangulo t1 = new Triangulo(5, 5, 5);
        System.out.println("\n--- Teste 1 (Equilátero) ---");
        System.out.println("É válido? " + t1.ehValido(5, 5, 5));
        System.out.println("Tipo: " + t1.tipoTriangulo());
        System.out.println("Perímetro: " + t1.calculaPerimetro());
        System.out.println("Área: " + t1.calculaArea());

        // Teste 2: Triângulo Isósceles válido
        Triangulo t2 = new Triangulo(5, 5, 8);

        // Teste 3: Triângulo Escaleno válido
        Triangulo t3 = new Triangulo(3, 4, 5);

        // Teste 4: Triângulo inválido
        Triangulo t4 = new Triangulo(1, 2, 5);

        // Teste 5: Triângulo inválido (lados negativos)
        Triangulo t5 = new Triangulo(-1, 2, 2);
    }
}
```

## Estratégia de Teste:

- Teste de triângulos válidos (equilátero, isósceles, escaleno)
- Teste de triângulos inválidos (desigualdade triangular)
- Teste com valores negativos
- Verificação de todos os métodos implementados

## Resultados dos Testes:

```
--- Teste 1 (Equilátero) ---
É válido? true
Tipo: Equilátero
Perímetro: 15.0
Área: 10.825317547305483

--- Teste 2 (Isósceles) ---
É válido? true
Tipo: Isósceles
Perímetro: 18.0
Área: 12.0

--- Teste 3 (Escaleno) ---
É válido? true
Tipo: Escaleno
Perímetro: 12.0
Área: 6.0

--- Teste 4 (Inválido) ---
Triângulo inválido. Os lados fornecidos não formam um triângulo.
É válido? false
Tipo: Inválido
Perímetro: 0.0
```

## Análise dos Resultados:

- Todos os tipos de triângulos válidos foram corretamente identificados
- Triângulos inválidos foram detectados e tratados adequadamente
- Cálculos de perímetro e área estão corretos
- Valores de retorno para triângulos inválidos são consistentes (0)

# Resultados e Conclusões

Análise dos testes e funcionamento da classe Triangulo

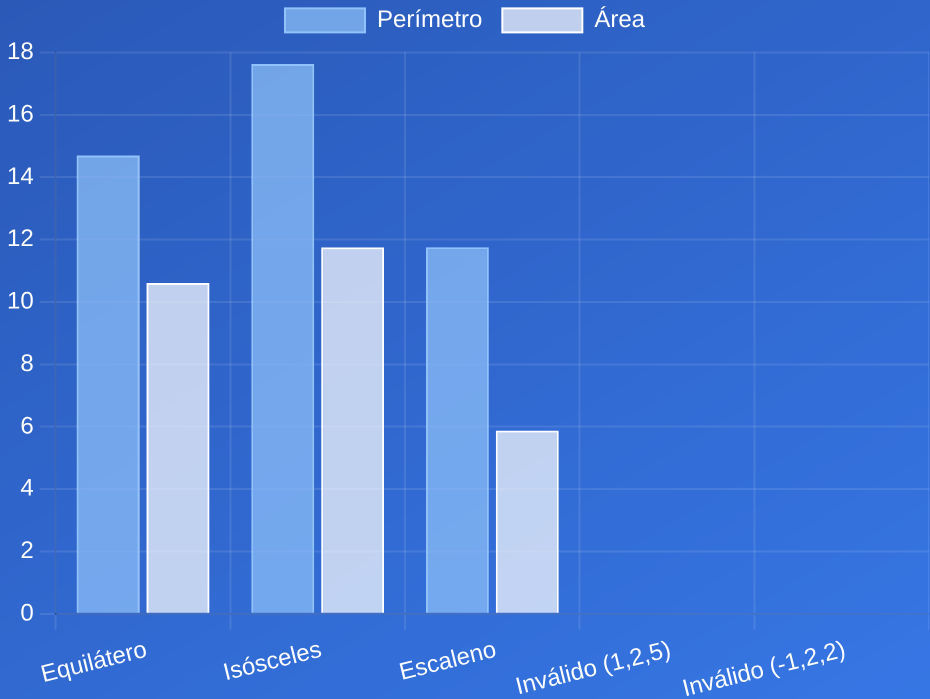
## Resultados dos Testes:

Tipo de Triângulo	Lados	Validação	Perímetro	Área
Equilátero	(5, 5, 5)	Válido	15.0	10.83
Isósceles	(5, 5, 8)	Válido	18.0	12.0
Escaleno	(3, 4, 5)	Válido	12.0	6.0
Inválido	(1, 2, 5)	Inválido	0.0	0.0
Inválido	(-1, 2, 2)	Inválido	0.0	0.0

### Conclusões:

- A classe implementa corretamente a validação de triângulos
- Identifica adequadamente os três tipos de triângulos
- Calcula perímetro e área com precisão usando a fórmula de Heron
- Trata casos inválidos retornando valores zero ou mensagens de erro
- Implementação robusta para uso em aplicações geométricas

Comparação de Perímetro e Área por Tipo de Triângulo



### Observações:

O gráfico mostra a relação entre perímetro e área para os diferentes tipos de triângulos testados. Note que triângulos inválidos têm valores zero para ambas as métricas.