



01

ENCAPSULAMENTO

[illegible]

- É implementado através de modificadores.

ENCAPSULAMENTO – MODIFICADORES DE ACESSO

Os modificadores são palavras-chave que alteram as definições de uma classe, método, atributo ou construtor. Existem vários modificadores na linguagem Java, na qual fazem parte: static, abstract e final.

No encapsulamento usamos os **modificadores de acesso**, sendo eles:

- Public
- Protected
- **Private**

ENCAPSULAMENTO – MODIFICADORES DE ACESSO

| Palavra reservada | Descrição |
|------------------------|--|
| Private (-) | Atributos, métodos e construtores acessíveis somente dentro da própria classe. |
| Package (~) Default | Atributos, métodos e construtores acessíveis somente em classes do mesmo pacote. |
| Protected (#) | Atributos, métodos e construtores acessíveis na própria classe e também nos métodos das classes que pertencem ao pacote. |
| Public (+) | Atributos e métodos acessíveis em todos os métodos de todas as classes. |

ENCAPSULAMENTO – MODIFICADORES DE ACESSO

| | Classe | Pacote | Subclasse | Todos |
|-----------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| public | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| protected | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| default | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| private | <input checked="" type="checkbox"/> | | | |

ENCAPSULAMENTO NA PRÁTICA

Encapsulamento é um conceito de Orientação a Objetos que define **como** os métodos e atributos de uma classe podem ser visualizados ou utilizados por outras classes.

- Ao definir a interface pública de uma classe pode expor demais o funcionamento da classe.
- É prática quase que obrigatória proteger seus atributos com private.
- **Vantagens:**
 - Manutenção
 - Esconde a complexidade

ENCAPSULAMENTO – MÉTODOS GETTERS E SETTERS

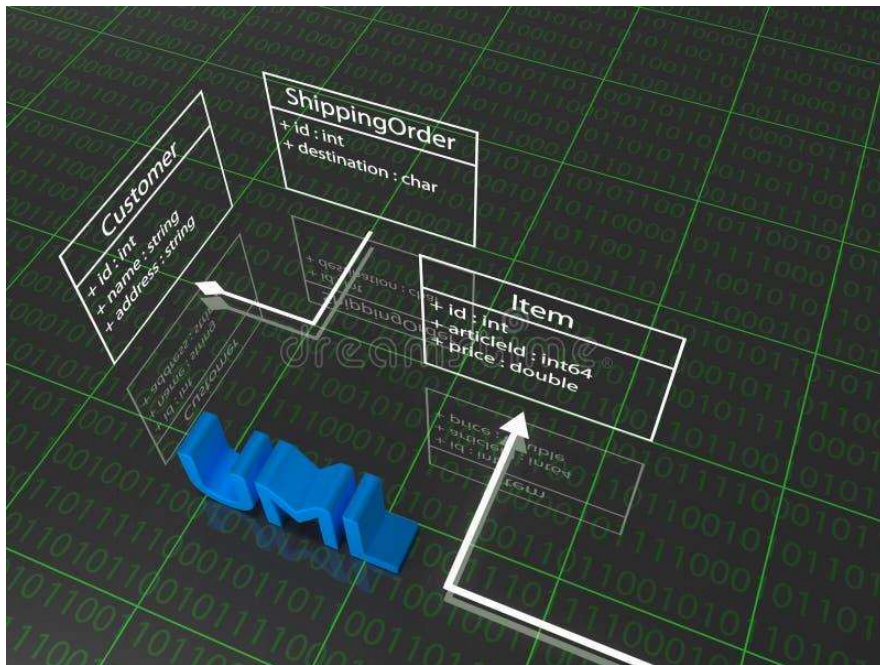
Para permitir o acesso aos atributos (quando eles forem private) de uma maneira controlada, a prática mais comum é criar dois métodos:

- Um que retorna o valor: **GET**
- Um que muda o valor: **SET**

ENCAPSULAMENTO – MÉTODOS GETTERS E SETTERS

{}

```
private String nome;  
  
public String getNome() {  
    return nome;  
}  
public void setNome(String nome) {  
    this.nome = nome;  
}
```

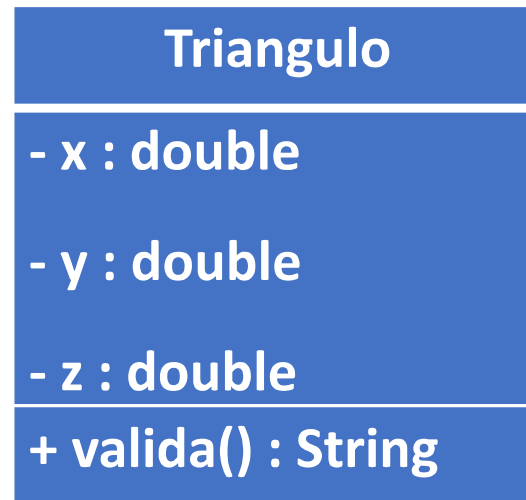



02

EXERCÍCIOS

EXERCÍCIO SOBRE ENCAPSULAMENTO

1. Crie um projeto novo chamado **pw-encapsulamento**, localize a pasta **src**.
2. Clique com o botão direito em **src**, acesse **New → Class** e coloque o nome **Triangulo**.
3. Abra a classe de modelagem Triangulo e codifique de acordo com o diagrama abaixo:

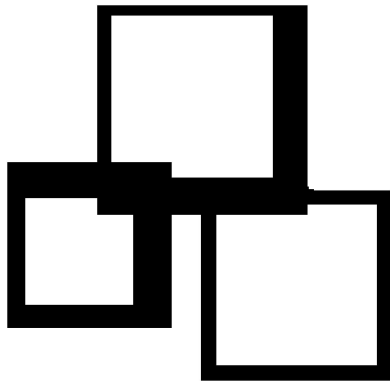


EXERCÍCIO SOBRE ENCAPSULAMENTO

4. Escreva o método `valida()` que verifica se `x`, `y` e `z` podem ser os comprimentos dos lados de um triângulo e, se forem, verificar se é um triângulo equilátero, isósceles ou escaleno. Se eles não formarem um triângulo retorna a String “Os valores não formam um triângulo”.

Considerar que:

- O comprimento de cada lado de um triângulo é menor que a soma dos outros dois lados;
- Chama-se triângulo equilátero o triângulo que tem três lados iguais;
- Chama-se triângulo isósceles o triângulo que tem o comprimento de dois lados iguais;
- Chama-se triângulo escaleno o triângulo que tem os três lados diferentes.



OBRIGADO

To be continued...

