

# PHP - Orientação a Objetos: Classes, métodos e atributos

- Programa procedural => dados estão separados do comportamento (ruim).
- Early Return: O princípio deste conceito diz que você deve se preocupar em **retornar o resultado da função o quanto antes**, ou seja, você irá estruturar o código da função de forma que retorne o mais rápido possível o valor esperado. [Torne-se um ninja das funções com Early Return \(dorianneto.com.br\)](https://dorianneto.com.br/torne-se-um-ninja-das-funcoes-com-early-return/).
- sintaxe this: \$this->saldo() (this.saldo() em java)

```
public function sacar($valorASacar):void
{
    if($valorASacar > $this->saldo) {
        echo "Saldo insuficiente";
        return;
    }
    $this->saldo -= $valorASacar;
}
```

- Não abuse de getters and setters! <https://blog.caelum.com.br/nao-aprender-oo-getters-e-setters/>
- Método construtor: \_\_construct() (2 underlines antes)
- Método destrutor: \_\_destruct() (2 underlines antes)

```
public function __construct(string $cpfTitular, string $nomeTitular)
{
    $this->cpfTitular = $cpfTitular;
    $this->nomeTitular = $nomeTitular;
    $this->saldo = 0;
}
```

- private static float \$numeroDeContas; -> 'static' define que \$numeroDeContas é um atributo da classe Contas, e não do objeto do tipo Contas. Para acessar um atributo estático, a sintaxe é a seguinte:

```
Contas::$numeroDeContas
self::$numeroDeContas
```

- 'self' é igual a 'this' mas para a classe, e não para objeto.
- Apenas escreva métodos estáticos quando eles realmente forem necessários e fizerem sentido (é raro).
- PHP tem Garbage Collector.
- [Value Object](#).

- Uma classe deve ter apenas uma responsabilidade, deve ser concisa.
  - É possível que um objeto tenha outro objeto como valor de um de seus atributos. Isto é chamado de composição.
  - A composição pode (e deve) ser utilizada ao definir uma estrutura complexa de classes.
- 

## Dúvidas e assuntos a pesquisar

- [Value Object](#).
- Não abuse de getters and setters! <https://blog.caelum.com.br/nao-aprender-oo-getters-e-setters/>.
- Early Return: O princípio deste conceito diz que você deve se preocupar em **retornar o resultado da função o quanto antes**, ou seja, você irá estruturar o código da função de forma que retorne o mais rápido possível o valor esperado. [Torne-se um ninja das funções com Early Return \(dorianneto.com.br\)](https://dorianneto.com.br/torne-se-um-ninja-das-funcoes-com-early-return/).
- É possível fazer algo do tipo:
  - `conta.titular.getNome();`
  - ou `$conta->$titular->getNome();`
  - sendo titular um objeto do tipo Titular dentro do objeto conta do tipo Conta?