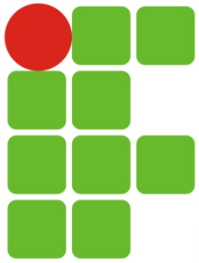
INSTITUTO FEDERAL
Farroupilha
Campus Avançado Uruguaiana

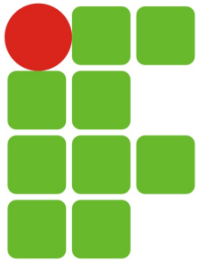
[POO]
Visibilidade em Objetos

Prof. Leandro Martins Dallanora
leandro.dallanora@iffarroupilha.edu.br



Modificadores de acesso

- Os modificadores de acesso são elementos de fundamental importância na Programação Orientada a Objetos.
- Tratam-se de palavras reservadas da linguagem de programação que determinam a forma de acesso a atributos e métodos de uma classe.
- Com os modificadores de acesso determinamos a visibilidade de um método ou atributo pertencente a uma classe. Ou seja, definimos se ele pode ou não ser acessado fora da classe em que foi declarado.



Modificadores de acesso

- Possuímos 3 modificadores de acesso:
  - **Público**, na UML simbolizamos com um **+**
  - **Privado**, na UML simbolizamos com um **–**
  - **Protegido**, na UML simbolizamos com um **#**
- Vamos entender o que são cada um deles.

Modificadores de acesso

- Vamos usar como exemplo o **telefone**.

[+] **Público** - qualquer um pode usar.

[-] **Privado** - somente você pode usar.

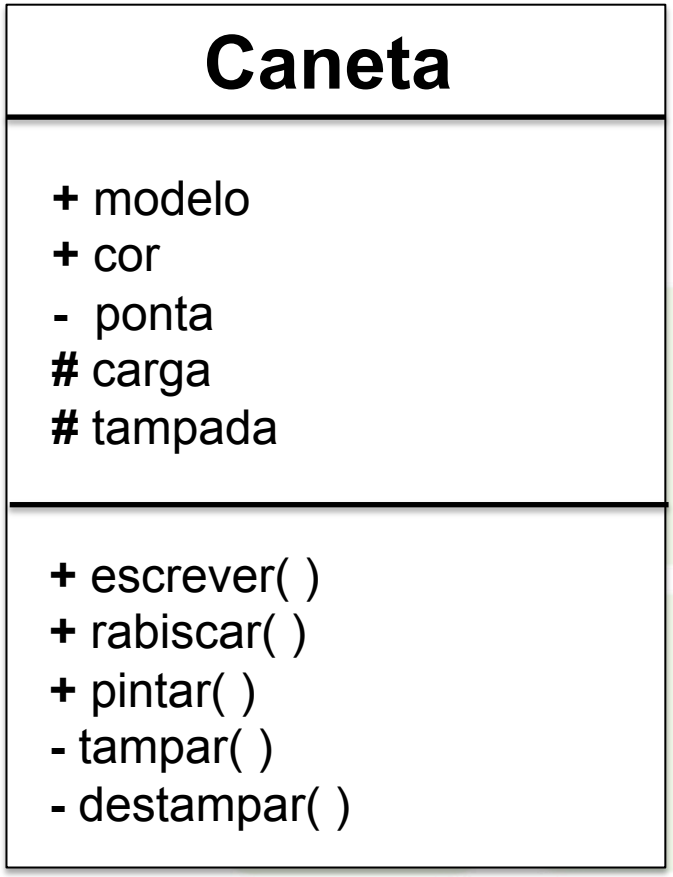
[#] **Protegido** - somente você e a sua família pode usar.

Modificadores de acesso

- Para a POO temos que:
  - **Público**: a classe atual e todas as outras classes tem acesso a ela, seus atributos e métodos.
  - **Privado**: somente a classe atual pode ter acesso a seus atributos e métodos.
  - **Protegido**: a classe atual e todas as suas sub-classes podem ter acesso a seus atributos e métodos.

Modificadores de acesso

- Se criarmos um diagrama de classe para o nosso exemplo da caneta, definindo suas visibilidades, poderia ficar assim:



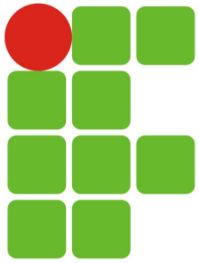
Modificadores de acesso

- No PHP, para modificar a visibilidade de um atributo ou método devemos preceder sua declaração com uma das palavras reservadas que representam o modificador, da seguinte forma:

```
modificador $atributo;

modificador function metodo() { }
```

- Os valores possíveis para o modificador são **public**, **private** e **protected**.



Public

- No código abaixo podemos ver um exemplo de uso do modificador de acesso **public**:

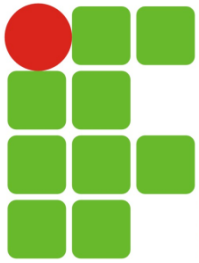
```
01 class Exemplo {
02      public $publico = 'Public';
03      public function metodoPublico() { }
04 }
```

- Foi declarada uma **classe** denominada **Exemplo** que possui um **atributo** público chamado **$publico** e um **método** público intitulado **metodoPublico()**.
- Neste caso ambos podem ser acessados por qualquer parte do código, incluindo outras classes.

Public

- Por exemplo, se criarmos uma instância dessa classe podemos acessar esses dois elementos da seguinte forma:

```
$obj = new Exemplo();
$obj->publico = 'Teste';
$obj->metodoPublico();
```



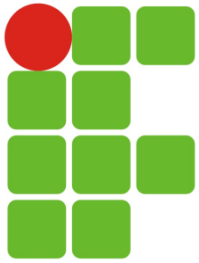
Private

- No código abaixo podemos ver um exemplo de uso do modificador de acesso **private**:

```
01 class Exemplo{
02  private  $privado = 'Privado';
03  private function metodoPrivado() {}
04 }
```

- Agora, se instanciarmos essa classe e tentarmos acessar esses dois elementos, como no código abaixo, um erro será gerado.

```
$obj = new Exemplo();
$obj->privado = 'Teste';//Erro
$obj->metodoPrivado();//Erro
```



Private

- Esse erro se deve ao fato de que somente a própria classe em que um atributo ou método foi declarado pode acessá-lo.
- Ou seja, nenhuma outra parte do código, nem mesmo as sub-classes, podem acessar esse atributo ou método.

Protected

- Esse modificador indica que somente a própria classe e as classes que herdam dela podem acessar o atributo ou método.
- Dessa forma, ao instanciar a classe os elementos protegidos (**protected**) não podem ser acessados diretamente, como ocorre com o public.
- A seguir podemos ver um exemplo de uso do modificador de acesso **protected**:

```
01 class Exemplo {
02      protected $protegido = 'Protegido';
03      protected function metodoProtegido() { }
04 }
```

Protected

- Neste caso tanto o atributo quanto o método protegidos, só podem ser acessados pela própria classe e pelas suas sub-classes.
- Por exemplo, se criarmos uma instância dessa classe não poderemos acessar esses dois elementos desta forma, pois um erro fatal será gerado:

```php
$obj = new Exemplo();
$obj->protegido = 'Teste'; //Erro
$obj->metodoProtegido(); //Erro
```

Exemplo Prático

- A seguir temos um exemplo prático de como declarar atributos e métodos usando os modificadores de acesso em PHP:

```
01  class Funcionario{
02   public $nome = 'Alex';
03   protected  $salario = 200;
04   private $rg ='00.000.000-0';
05  }
```

- Nesse código pode-se observar a declaração de três atributos, cada um com seu respectivo modificador de acesso.

Exemplo Prático

- Agora, se instanciarmos a classe **Funcionario** e tentarmos acessar seus atributos, como consta no código abaixo, podemos analisar o comportamento da linguagem em cada situação:

```
01 $funcionario = new Funcionario;
02 echo $funcionario->nome;
03 echo $funcionario->salario;
04 echo $funcionario->rg;
```

Exemplo Prático

- Linha 1: Criamos uma instância da classe **Funcionario**;
- Linha 2: Acessamos o atributo público **$nome** sem problemas;
- Linha 3: Tentamos acessar o atributo protegido **$salario**, o que gerará um erro;
- Linha 4: Novamente será gerado um erro ao tentarmos acessar o atributo privado **$rg**.