unesp

Desenvolvimento de Aplicações para Internet

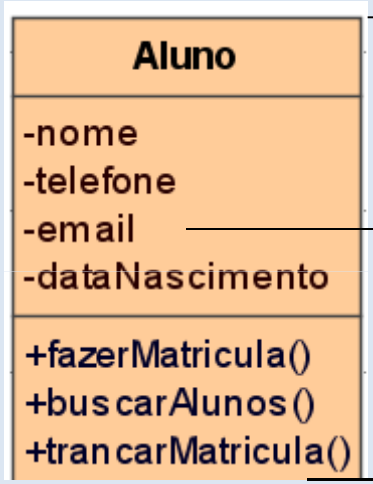
Aula 7

Celso Olivete Júnior

olivete@fct.unesp.br

PHP - Orientação a objetos

Classe e tipos de dados



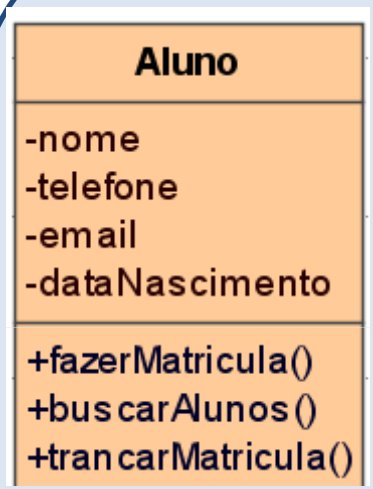
Classe – Conjunto de objetos semelhantes, isto é, com a mesma estrutura (atributos) e métodos.

Tipos de Dados:

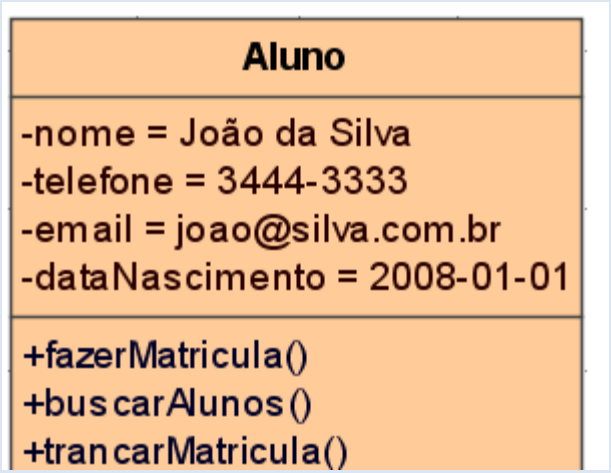
- ✓ String
- ✓ Integer
- ✓ Boolean
- ✓ Date
- ✓ Double

PHP - Orientação a objetos

Objeto e instância da classe



Objeto – "Variável" do tipo da classe, que herda a estrutura de dados (atributos) e suas operações (métodos).

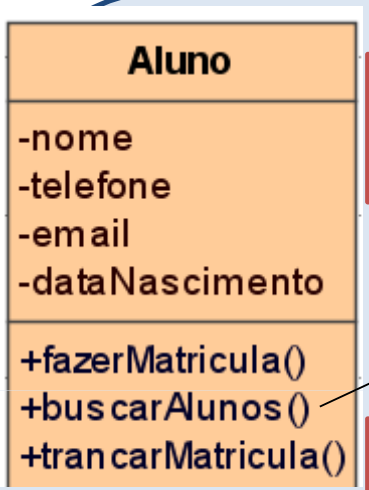


Instância da Classe – Momento no código fonte que é gerado uma "Variável" do tipo da classe, que herda a estrutura de dados (atributos) e suas operações (métodos).

```
$objAluno = new Aluno();
```

PHP - Orientação a objetos

Chamada e retorno de métodos



Chamada

```
$objAluno = new Aluno();
$objAluno->buscarAlunos();
```

Retorno

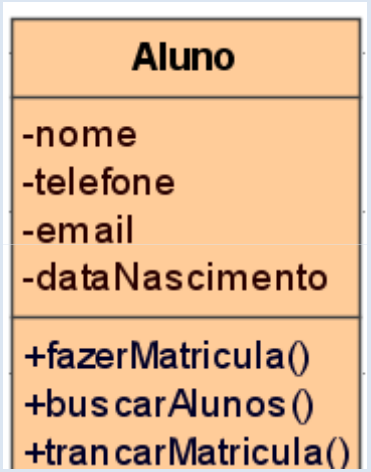
```
$objAluno = new Aluno();
$retorno = $objAluno->buscarAlunos();
```

Tipos de Retornos:

- ✓ Boolean
- ✓ Objeto
- ✓ Coleção de Objetos

PHP - Orientação a objetos

Referência aos atributos e métodos com atributos



Referência Fora da Classe

```
$objAluno = new Aluno();
echo $objAluno->getTelefone();
                    //método de retorno
```

Referência Dentro da Classe

```
echo $this->telefone;
```

Métodos com atributos

```
$objAluno = new Aluno();
$objAluno->setNome($nome_lido);
                    ou
$objAluno->setNome("php");
```

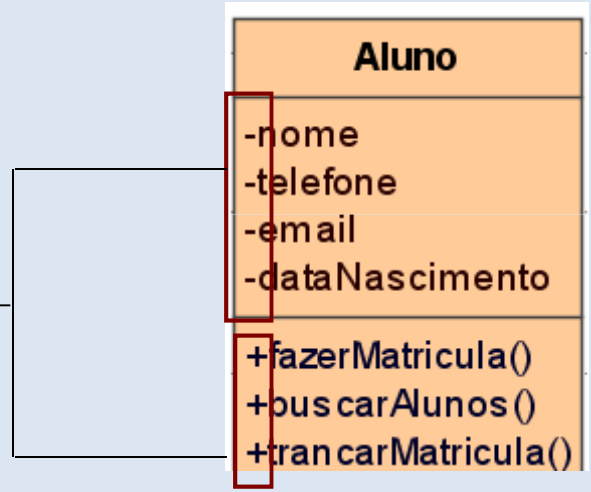
PHP - Orientação a objetos

Visibilidade dos atributos, métodos e classes

+ ou public → Acesso dentro ou fora da classe instanciada.

 # ou protected → Acesso dentro da classe ou pelas classes dependentes (herança).

- ou private → Acesso somente dentro da classe.

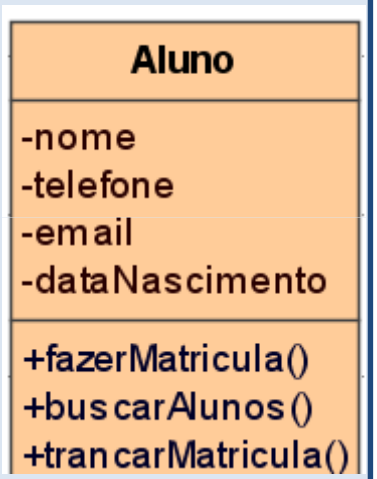


PHP - Orientação a objetos

Visibilidade dos atributos, métodos e classes

Definição da classe

```
<?php
class Aluno {
        //definição dos atributos
        private $nome;
        private $telefone;
        private $email;
        private $dataNascimento;
        //definição do método para setar o nome
        public function setNome($nome_enviado)
        {
           $this->nome = $nome_enviado;
        }
}
?>
```

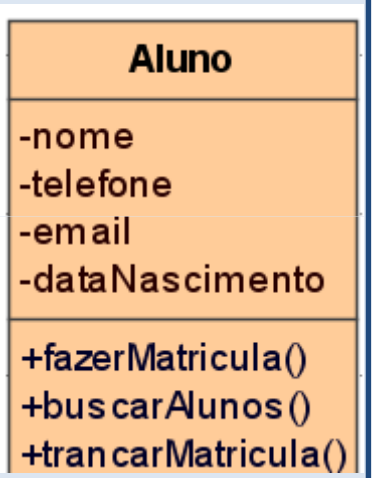


PHP - Orientação a objetos

Visibilidade dos atributos, métodos e classes

Definição da classe

```
<?php
class Aluno {
        //definição dos atributos
        private $nome;
        private $telefone;
        private $email;
        private $dataNascimento;
        //definição do método para recuperar o nome
        public function getNome()
        {
           return $this->nome;
        }
        ...
}
?>
```

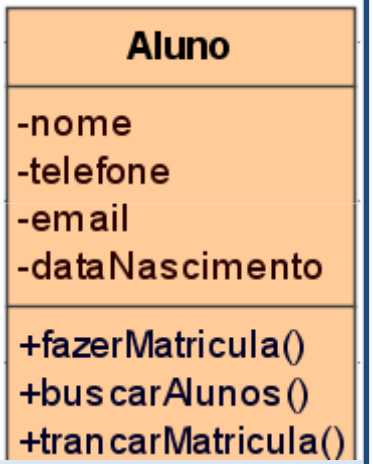


PHP - Orientação a objetos

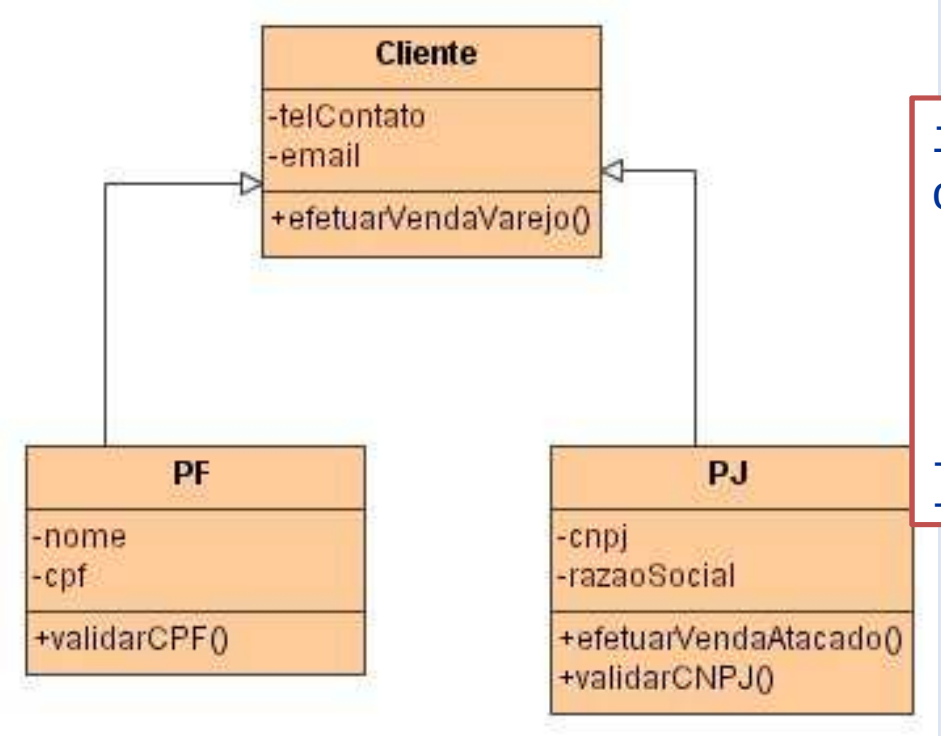
Visibilidade dos atributos, métodos e classes

Utilizando a classe

```
<?php
//inclui a classe aluno
include_once (“classe_Aluno.php”);
//cria uma instancia da classe Aluno
$objAluno = new Aluno();
//chama o método (função) setNome, que atribui
//valor para o atributo nome
$objAluno->setNome(“Abgobaldo”);
//chama o método que retorna o valor do atributo
//nome
echo “Nome:”.$objAluno->getNome();
?>
```



PHP - Orientação a objetos
Herança simples




```
include("classe_cliente.php");
class PJ extends Cliente{

  public function validarCNPJ {

      }
}
```

PHP - Orientação a objetos
Construtor

- O método construtor define os valores iniciais dos atributos de um objeto
- Construtores são funções, logo aceitam passagem de parâmetros
- O PHP aceita apenas um construtor.

PHP - Orientação a objetos
Construtor

- Sempre que você quiser fazer alguma coisa na inicialização da classe, terá que usar esse método.

- __construct()

```
class Cliente
{
 ...
function __construct()
   {

     echo “Construtor - Novo cliente";

   }
```

- Atenção para os 2 ‘underscores’.

PHP - Orientação a objetos
Construtor

- O construtor também pode ser definido utilizando o mesmo nome da classe.

- Cliente()

```
class Cliente
{
 ...
function Cliente()
   {

     echo “Construtor - Novo cliente";

   }
```

PHP - Orientação a objetos
Construtor com parâmetros

- O construtor com parâmetros.

```
class Cliente
{
 ...
function __construct($p1,$p2)
   {
     echo “Construtor - Novo cliente";
     $this->nome = $p1;
     $this->idade = $p2;

   }
```

- No momento da criação do nosso objeto podemos passar valores fixos e também variáveis

```
$objCliente=new Cliente(“joao”, 34);
                    ou
$nome = “joao”;
$idade = 34;
$objCliente=new Cliente($nome,$idade);
```

PHP - Orientação a objetos
Construtor com parâmetros

- O construtor com parâmetros.

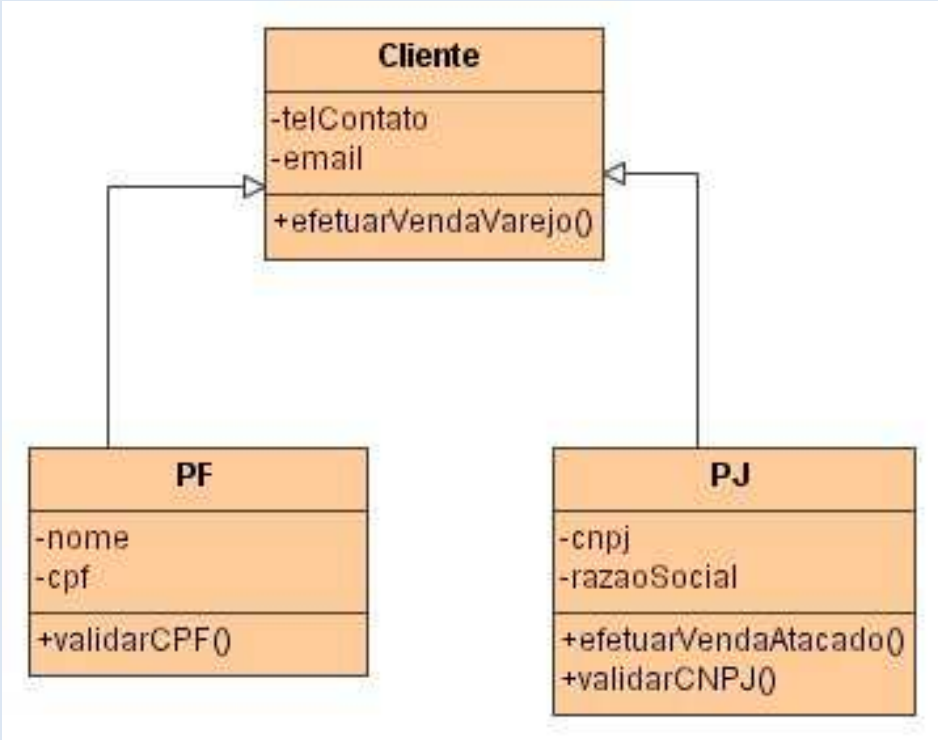
```
class Cliente
{
 ...
function Cliente($p1,$p2)
   {
     echo “Construtor - Novo cliente";
     $this->nome = $p1;
     $this->idade = $p2;

   }
```

- No momento da criação do nosso objeto podemos passar valores fixos e também variáveis

```
$objCliente=new Cliente(“joao”, 34);
                    ou
$nome = “joao”;
$idade = 34;
$objCliente=new Cliente($nome,$idade);
```

PHP - Orientação a objetos
Exercício I

- Implemente...



PHP - Orientação a objetos
Exercício II

- Implemente uma classe que contenha um método para realizar a conexão com MySql e que selecione uma determinada base de dados e, um outro método para desconectar do MySql (função mysql_close()). Esse método de conexão deverá ser invocado a cada manipulação com o BD e o método para desconectar deverá ser invocado após a manipulação.
- Implemente uma classe (ManipulaDadosBD) que contenha um método que receba um objeto e insira-o na base de dados. Exemplo:

```
<?php
...
 public function inserir($cliente) { //recebe um objeto como parâmetro
     $conexao = new Conexao(); //cria um objeto do tipo conexão
     $conexao->conectar();  // estabelece conexão
     $nome = $cliente ->getNome(); //recupera nome
     $fone = $cliente ->getFone(); //recupera fone
     $rua = $ cliente ->getRua(); //recupera rua
     //cria a instrução SQL
     $sql = "insert into pessoa(nome,fone,rua) values('$nome','$fone','$rua')";
     //executa o sql
     mysql_query($sql) or die ('Erro ao Inserir pessoa!!!');
     $conexao->desconectar(); //fecha a conexão
     echo '<h1>Pessoa inserida com sucesso</h1>';
}
...
?>
```

PHP - Orientação a objetos
Próxima aula

- Modelo MVC (Model-view-controller)
  - Banco de dados