



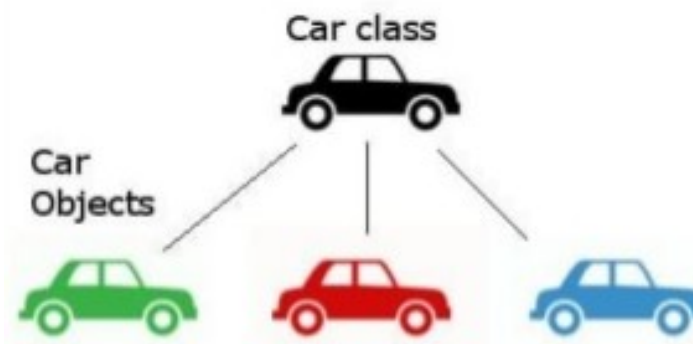
**INSTITUTO  
FEDERAL**  
Paraná

Campus  
Foz do Iguaçu

# Orientação a Objetos

Prof. Daniel Di Domenico

## Classes e objetos



# O que já sabemos?

- Linguagem PHP
  - Variáveis, tipos, operadores
  - Comandos de fluxo:
    - IF/ELSE, SWITCH, WHILE, DO-WHILE, FOR
  - Leitura de dados
  - Funções
  - Arrays (vetores)



# Objetivos da aula

- **Conhecer os conceitos fundamentais da orientação a objetos:**
  - Classes
  - Objetos
  - Atributos
  - Métodos
- **Implementar tais conceitos em PHP**



# Orientação a objetos (OO)

- Objetivo da Orientação a Objetos
  - **Aproximar** a programação do mundo **real**
    - Representar **conceitos reais** em um programa de computador
    - Programar de uma forma mais próxima de como expressamos as coisas na vida real
- OO é um paradigma muito utilizado atualmente
  - Muito difundido no mercado de trabalho
    - **Passou a ser uma exigência**
  - Suportado por diversas linguagens de programação
    - Ranking das linguagens de programação mais utilizadas:  
<https://www.tiobe.com/tiobe-index/>
    - Das 5 primeiras, apenas C não suporta orientação a objetos

# Orientação a objetos (OO)

- **Diferença:** como programávamos até o momento?
  - Utilizávamos o paradigma estruturado
    - Foca no uso de funções e procedimentos
  - Agora, passaremos a utilizar o paradigma de orientação a objetos
    - Foca no uso de classes/objetos e seus conceitos (serão vistos hoje e nas próximas aulas)



# Orientação a objetos (OO)

- Algumas **vantagens** da OO:
  - Organização do código da aplicação
    - Implementação em camadas
    - Facilidade de manutenção (alterações)
    - Distribuição das responsabilidades para classes/objetos
  - Facilidade para incluir novas funcionalidades
    - Partes já implementadas em classes/objetos podem ser reutilizadas
  - Naturalidade
    - Aproximação do software com o mundo real



# Orientação a objetos (OO)

- Como a OO funciona:
  - Consiste em **mapear entidades** do mundo real no código da aplicação
    - A este processo dá-se o nome de **abstração**
  - O que é **Abstração**?
    - Habilidade de concentrar-se nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes ou acidentais
  - Através da **abstração**, modelam-se conceitos/entidades do mundo real como **classes**

# Orientação a objetos (OO)

- Exemplos da abstração de **entidades do mundo real em classes**

- Cliente
- Funcionário
- Agência
- Conta

Todas essas entidades fazem parte de um **domínio**, pois estão relacionadas



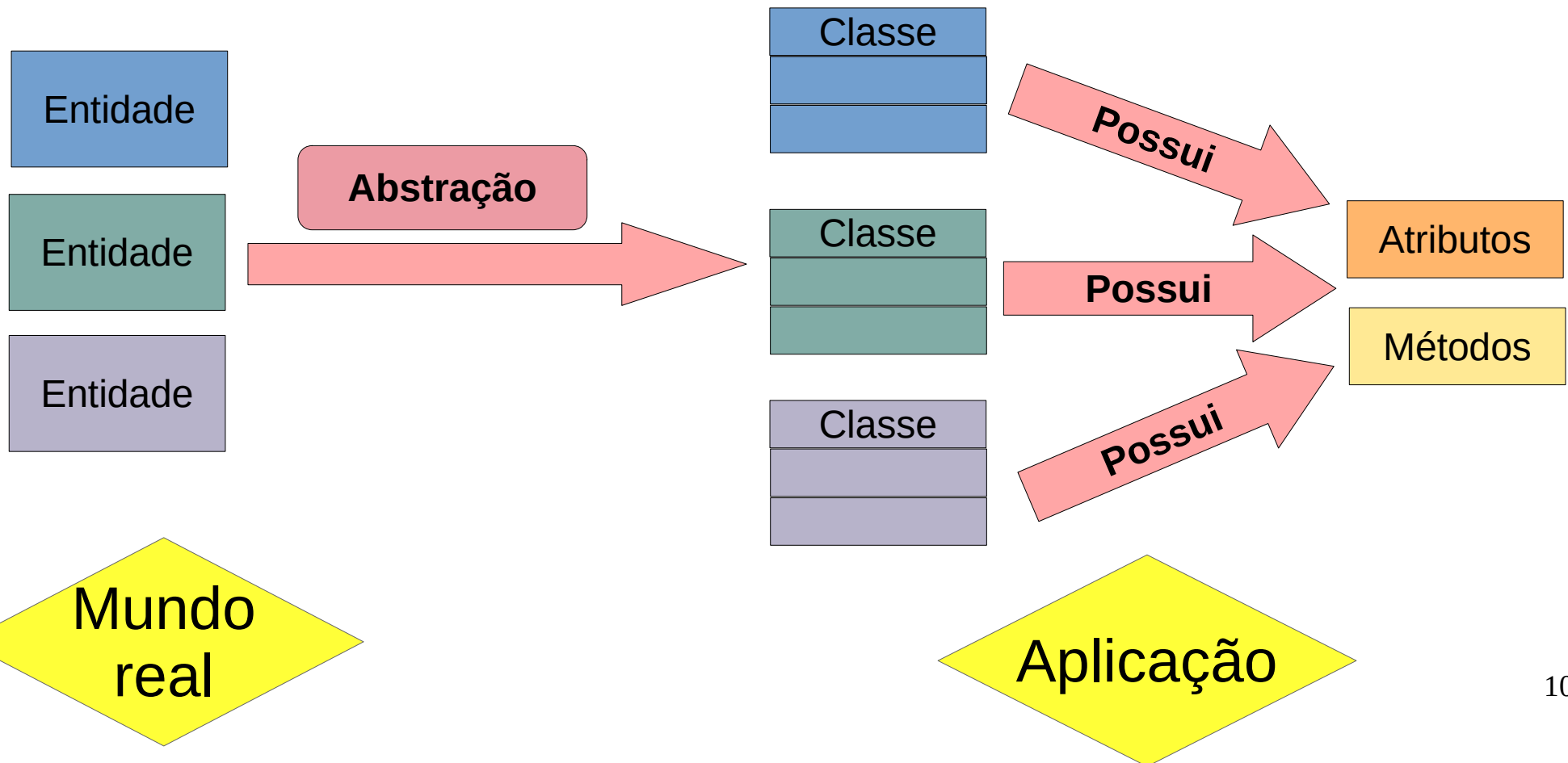


# Orientação a objetos (OO)

- Com a **abstração**, é possível adicionar mais detalhes das entidades às classes:
  - **Atributos**
    - Representam as **características**
      - Uma pessoa tem nome, data de nascimento, endereço...
      - Um carro tem modelo, cor, marca, ano de fabricação...
  - **Métodos**
    - Representam as **operações**
      - Uma pessoa pode caminhar, comprar, vender, estudar...
      - Um carro pode acelerar, frear, parar...

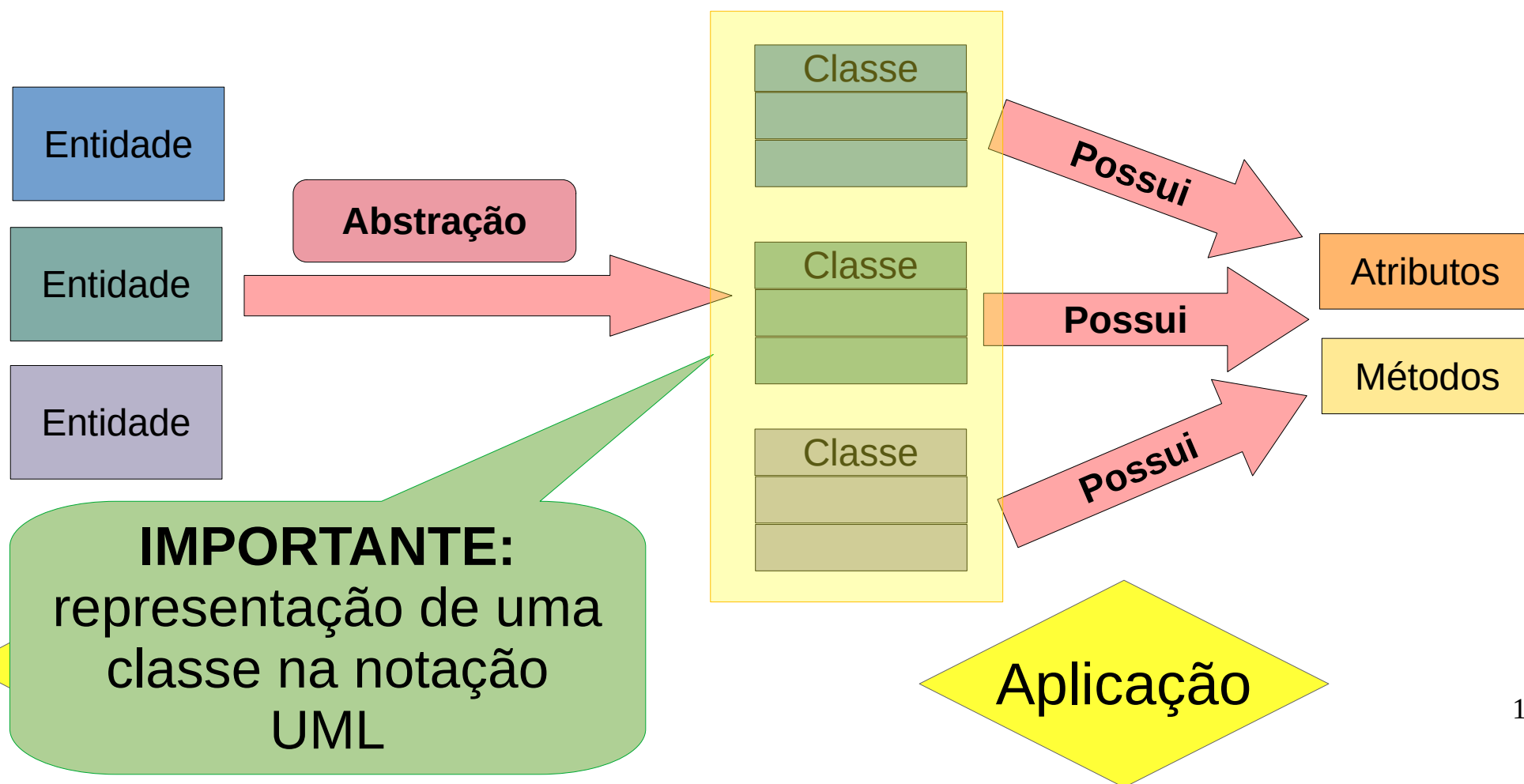
# Orientação a objetos (OO)

- Recapitulando...



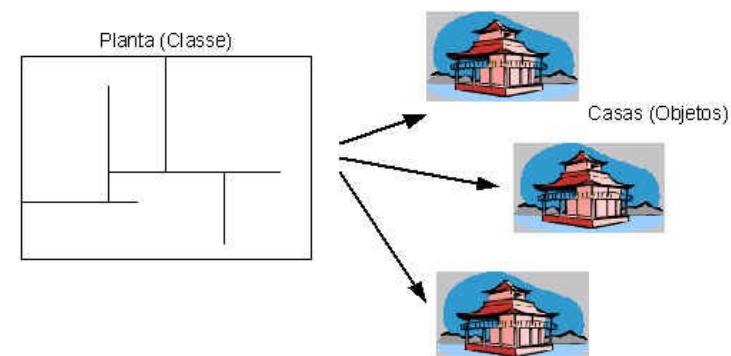
# Orientação a objetos (OO)

- Recapitulando...



# Orientação a objetos (OO)

- E os **objetos**?
  - São **instâncias** das classes
  - A partir de **uma classe**, pode-se criar **diversos objetos**
    - Cada objeto tem seus próprios valores de atributos
      - Por eles, são representadas as características próprias de cada objeto
  - Analogia:
    - Classe é como uma planta (projeto)
    - Objeto é como uma casa
    - A partir de uma planta, pode-se construir diversas casas **mudando suas características** (atributos)



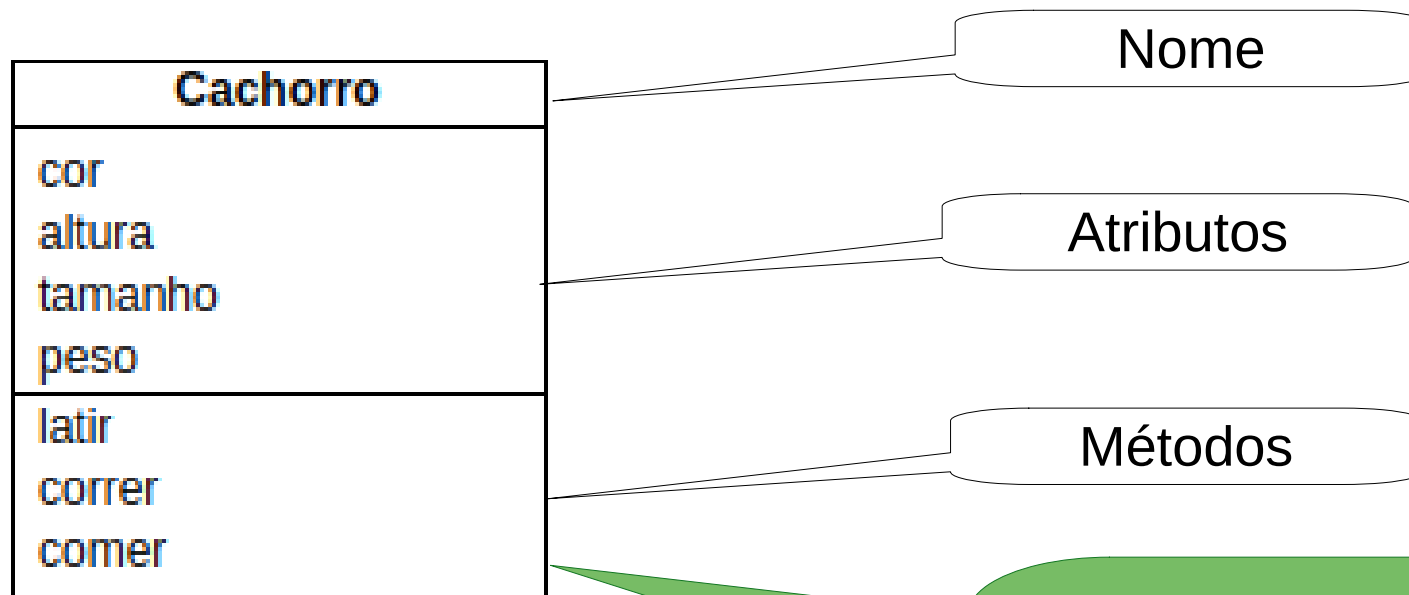
# Orientação a objetos (OO)

- Classes e Objetos



# Orientação a objetos (OO)

- Representação de uma classe (linguagem UML)



A partir da classe Cachorro, é possível criar diversos objetos

# Classes e Objetos em PHP

- Como implementar uma classe em PHP

```
class Cachorro {  
  
    public $cor;  
    public $altura;  
    public $tamanho;  
    public $peso;  
  
    function latir() {  
        echo "latindo....\n";  
    }  
    function correr() {  
        echo "correndo....\n";  
    }  
    function comer() {  
        echo "comendo....\n";  
    }  
}
```

**public** = entenderemos  
mais a frente

Atributos

Métodos

# Classes e Objetos em PHP

- Instanciar objetos em PHP

```
//Programa principal  
$cao = new Cachorro();  
  
$cao->cor = "Preto";  
$cao->altura = 25;  
$cao->tamanho = "Médio";  
$cao->peso = 9;  
  
$cao->latir();  
$cao->correr();  
$cao->comer();
```

Criar/instanciar o objeto:  
operador **new** seguido do  
nome da classe

Setar os atributos

Chamar dos métodos



# Classes e Objetos em PHP

- **Construtor**

- **Método** que cria/instancia um objeto
- Definição:
  - Método `__construct()`
    - **Não é obrigatório ser definido**
- Chamada:
  - Chamado sempre com o operador **new** seguido do nome da classe

```
class Cachorro {  
  
    public $cor;  
    public $altura;  
    public $tamanho;  
    public $peso;  
  
    function __construct($c, $a, $t, $p) {  
        $this->cor = $c;  
        $this->altura = $a;  
        $this->tamanho = $t;  
        $this->peso = $p;  
    }  
}  
  
//Programa principal  
$cao = new Cachorro("Branco", 12,  
                    "Pequeno", 3);
```

Redefinição do construtor com 4 parâmetros

**\$this**: acesso ao objeto dentro do escopo da classe

# Exercícios

- **1-** Faça um programa que crie a classe pessoa com:

- Atributos: nome, endereço, cidade, UF e altura
- Métodos:
  - falarOla: deve imprimir “Olá mundo, sou <nome>!”;
  - falarEndereco: deve imprimir “Moro em <endereço>, <cidade>/<UF>!”
  - FalarAltura: deve imprimir “Tenho <altura> metros!”;

Depois, leia os dados de uma pessoa, sete seus atributos e chame os três métodos.

- **2-** Faça um programa que crie uma classe Retangulo com os atributos base e altura e os método area() e perimetro() para calcular e **retornar** a área e o perímetro do mesmo. Depois, leia a base a altura de 3 retângulos para os atributos de 3 objetos, chamando seus métodos e imprimindo a sua área e perímetro.
- **3-** Incremente o programa do exercício 2, setando os valores dos atributos através do construtor da classe e depois chamando seus métodos.