# LEVEL 02 – POO

## Programação II

(Luis Araujo)

# Pradigma de Programação

- Existem diversos paradigmas de programação, cada um com sua especificidade. Usados em situações e plataformas específicas. Até agora trabalhamos com ideia de **programação estrutura** - que se caracteriza pela escrita do código de forma linear - usando estruturas de sequência, decisão e repetição.

# Programação orientada a objeto

- A PPO é também um paradigma. Ela presa pela reutilização de código, distribuição de responsabilidade, facilidade de manutenção e segurança, utilizando conceitos de Classes e Objeto.

- Estes possuem suas ações (métodos) e propriedades (atributos) e visam facilitar a modelagem do mundo real. Como veremos...

# Programação orientada a objeto

- Em um determinado jogo, temos um personagem cachorro. Ele é uma representação do cachorro que conhecemos na vida "real". Assim, ele **anda**, **late**, **fareja**, **corre** etc... Perceba que tudo isso são ações do nosso cachorro. Uma ótima forma de identificar as ações é perceber que elas são expressas por **verbos**. Em POO dizemos que esses são os **métodos**.

# Programação orientada a objeto

- Além disso o nosso cachorro tem algumas características como a **cor marrom**, **o nariz grande, olhos castanhos, peso, rabo curto, tamanho** e etc... Essas são, portanto, as propriedades do nosso cachorro. Em PPO dizemos que são os **atributos**!

# Programação orientada a objeto

- Se o cachorro tem métodos (ações) e propriedades (atributos) ele é um **objeto**.

Imagem disponível em <http://cdn5.colorir.com/>

# Programação orientada a objeto

- Se o cachorro é um objeto ele pertence à uma classe: **Cachorro.**

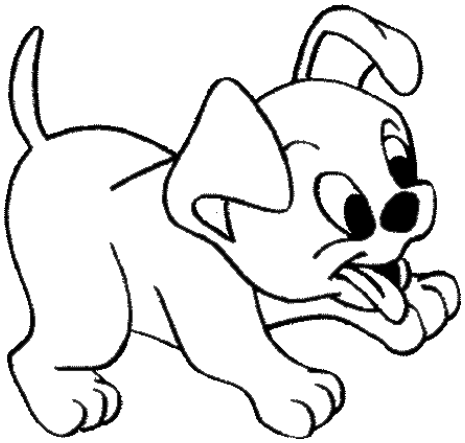
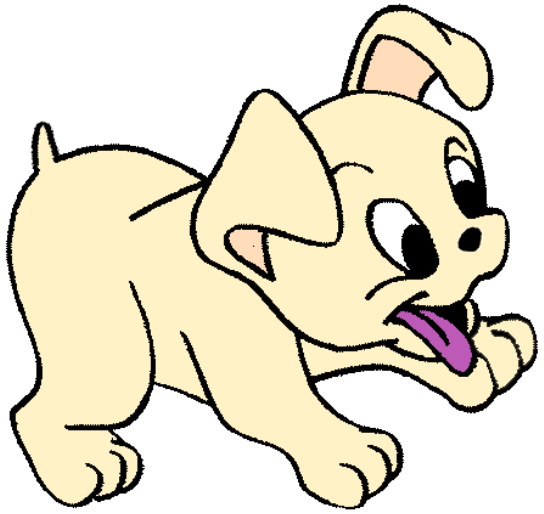


Imagem disponível em <http://cdn5.colorir.com/>

# Programação orientada a objeto

- Nesse caso podemos ter um objeto cachorro de outra cor, com língua roxa, nariz pequeno e rabo curto. Ele ainda é um objeto do tipo **Cachorro**.



# Programação orientada a objeto

- Assim, classes são abstrações ou modelos do mundo real que servem como “forma” para criar os nossos objetos. Ao criarmos os objetos, nós especificamos as suas **propriedades**.

# CLASSES

- Recapitulando... **As classes são formas para criar os objetos** (instâncias de classes), assim podemos ter muitos objetos de uma mesma classes, um mesmo tipo. Como temos várias raças de cachorro, mas todos são cachorros, pois possuem as ações e propriedades semelhantes: 4 patas, latem, farejam, são mamíferos, mordem... Etc.

# OBJETOS

- Recapitulando... Os objetos são instancias das classes, são a materialização das classes. Desse modo, podemos alterar as **propriedades dos objetos**, mas não os métodos! Ao criar um cachorro, podemos escolhe a sua cor, o tamanho do nariz e etc...

# ATRIBUTOS

- Recapitulando... Atributos são características do objeto e são estipulados pela classe. Ela expressa o atual estado dos objetos.

# MÉTODOS

- Recapitulando... Métodos são as ações de da classe e são chamadas através dos objetos (a não ser quando estáticas). Através deles podemos modificar o estado dos objetos.

# Programação orientada a objeto

- Outra analogia que podemos fazer é pensar que cada classe é uma ficha de inscrição e cada objeto é uma ficha preenchida.

- Assim, temos a classe **Ficha** (um arquivo em um editor de texto). Para criar um objeto do tipo **Ficha**, vamos Imprimir o documento. Após isso, preenchemos os dados.

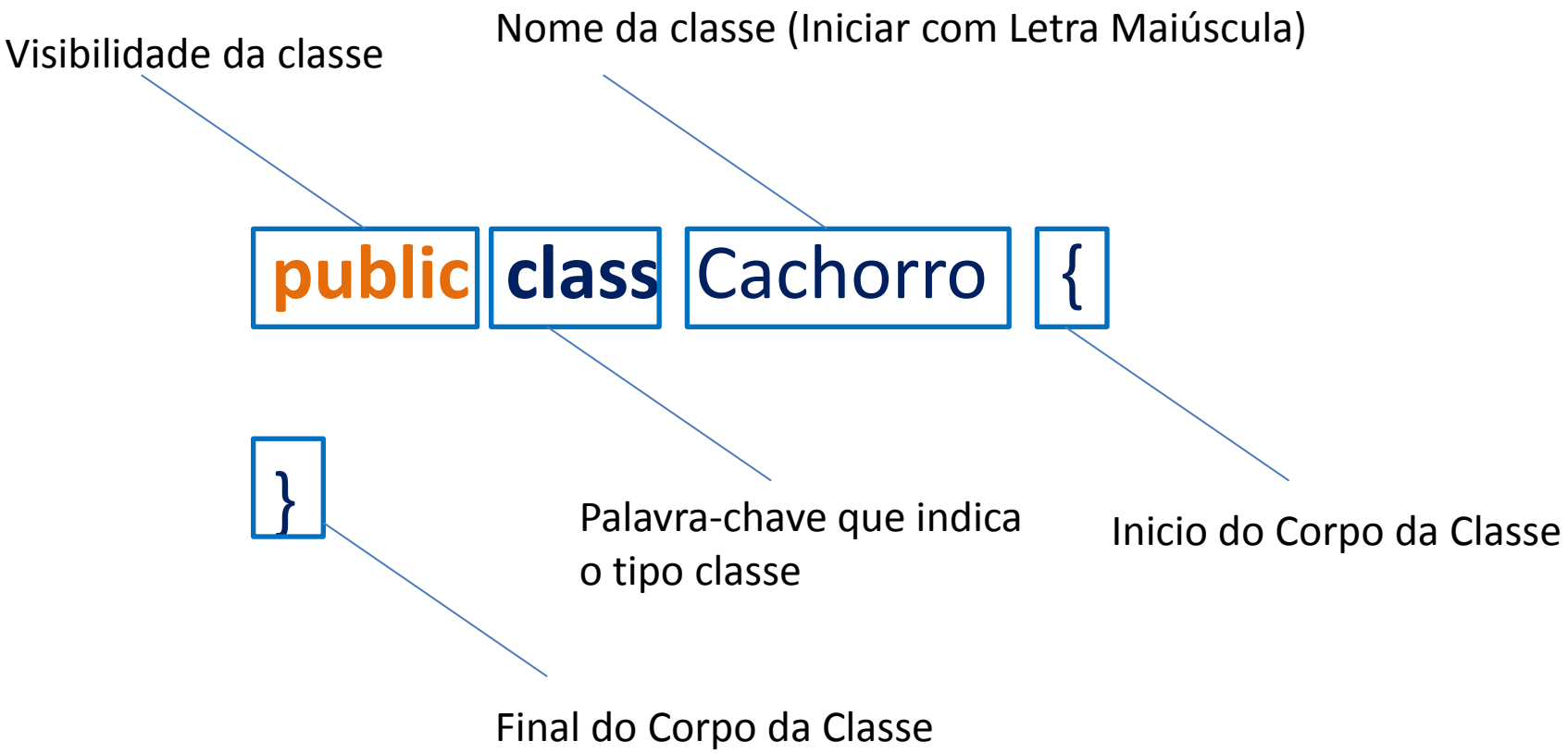
# Programação orientada a objeto

- Cada Linguagem de Programação tem uma forma particular de representar (criar) uma Classe, um objeto, chamar métodos e etc.. Na Linguagem C# (Sharp) podemos cria uma classe dessa forma:

```
class Cachorro{


}
```

# Classes em C#


Visibilidade da classe
Nome da classe (Iniciar com Letra Maiúscula)
public
class
Cachorro
{
}
Palavra-chave que indica o tipo classe
Inicio do Corpo da Classe
Final do Corpo da Classe

# Atributos em C#

```
public  class  Cachorro   {

    public string cor;

}
```

Nome do atributo seguido de virgula!!!!

Tipo do Atributo

Visibilidade do atributo

# Métodos em C#

```
public  class  Cachorro    {

    latir  ()  {

    }

}
```

Local reservado a parâmetros

Inicio do Corpo do Método

Nome do método
(iniciar com letra minúscula)

Final do Corpo do Método

# Alterando atributos em C#

```
public  class  Cachorro   {
public string cor;

private configurar Cor (string novacor)  {

        cor = novacor;
}
}
```