



Programação Orientada a Objetos – Aula 02

IFPE – Campus Igarassu
2016.1

Ranieri Valença
07/07/2016



Relembrando...

Programar Orientado a objetos

Uma forma de escrever programas,
aproximando o mundo real (já que vivemos
num outro mundo...) do mundo da
programação



Objeto

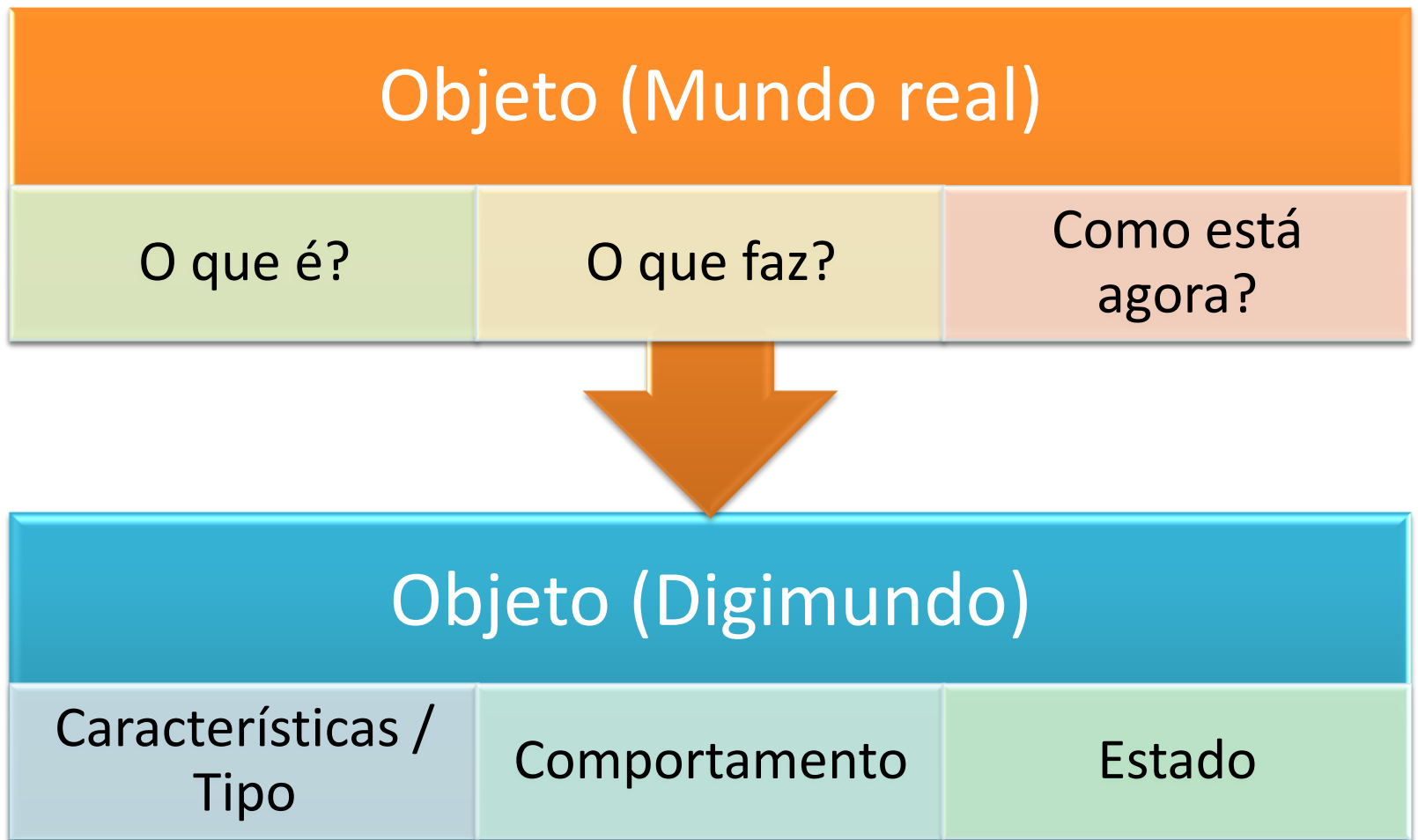
São modelos de objetos do mundo real, que guardam suas **características, o estado atual** e os **comportamentos**

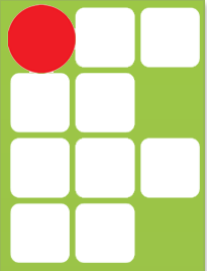


Objeto - Exemplos

Objeto	Estado / características	Comportamento(s)
Carro	Marca, modelo, cor, velocidade máxima, velocidade atual, marcha atual, ...	Ligar, trocar de marcha, acelerar, frear, desligar, ...
Gato	Nome, raça, com fome, onde está dormindo, ...	Miar, dormir, comer, comer, dormir, dormir, comer, ronronar, entrar na caixa, ...
Usuário	Nome, sobrenome, sexo, idade, endereço, email, senha, ativo, ...	Fazer login, trocar email, trocar senha, mudar endereço, habilitar, ...

Objeto



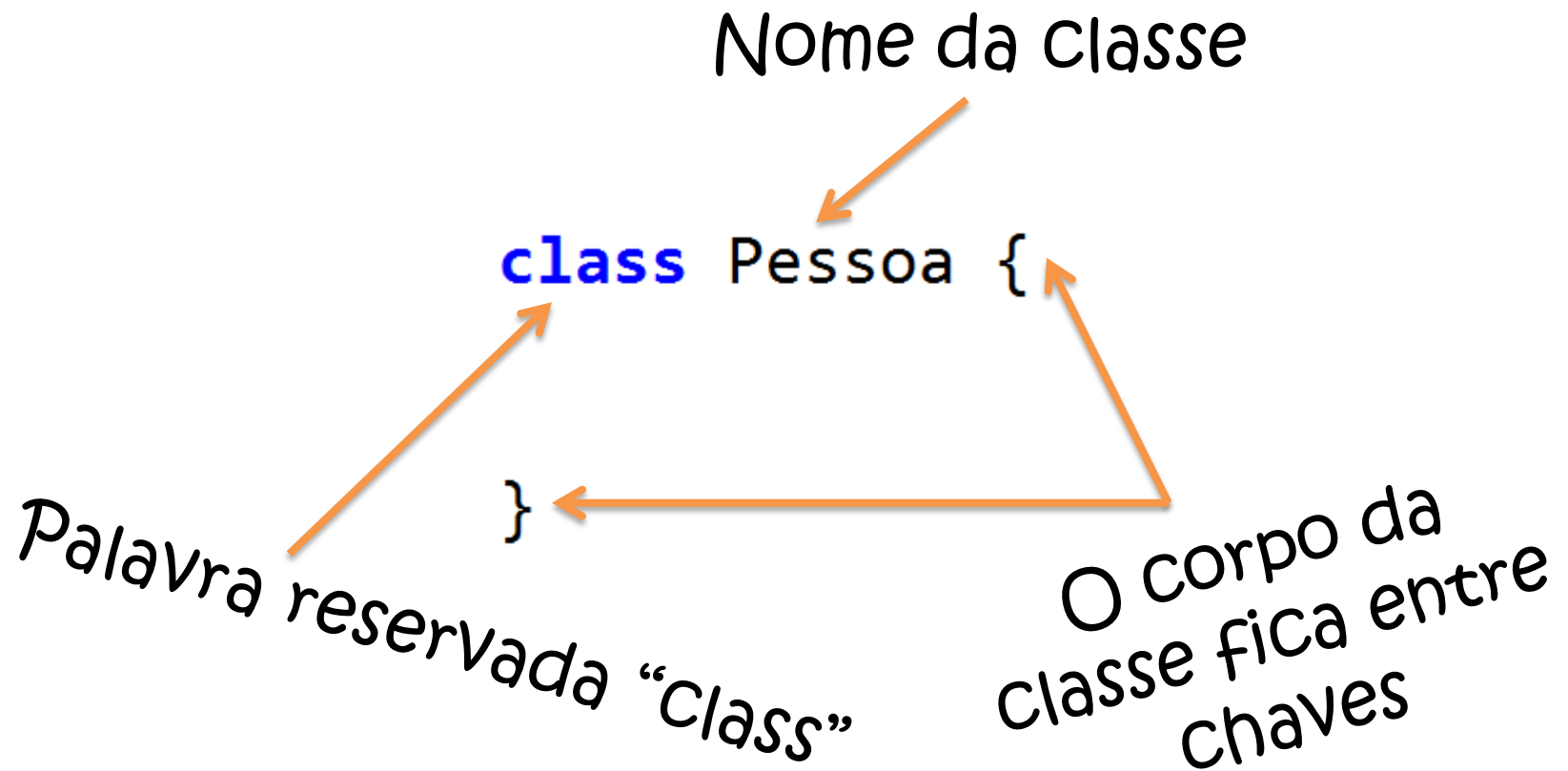


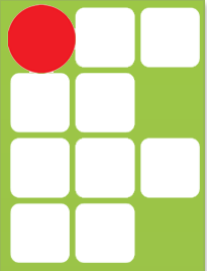
Classe

Protótipos utilizados para
construir objetos

Os **comportamentos** são **comuns**
aos objetos de uma classe

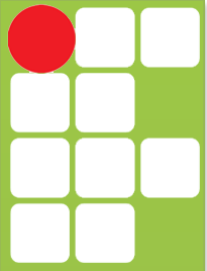
Criando classes em Java





Instância

Dizemos que um objeto é uma
Instância
de uma classe



Atributos

Propriedades dos objetos que armazenam suas **características e estados**

Declarando Atributos

`int idade;`

tipo *nome* *ponto e vírgula*



Revisando os tipos de dados em Java

Java possui os seguintes tipos primitivos

- boolean
- byte
- char
- double
- float
- int
- long
- short



Revisando os tipos de dados em Java

Java possui os seguintes tipos primitivos

- boolean
- byte
- char
- double
- float
- int
- long
- short

Ué, cadê
String???

Revisando os tipos de dados em Java

Java possui os seguintes tipos primitivos

- boolean
- byte
- char
- double
- float
- int
- long
- short

Os atributos podem ter esses tipos...

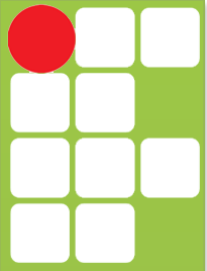
E se o atributo não for nenhum desses tipos? Então o tipo vai ser uma classe pae!

Tipos de dados em Java – Exemplo

```
class Pessoa {  
    int idade;  
    float pesoEmQuilogramas;  
    float alturaEmMetros;  
    char sexo;  
    String nome;  
    String sobrenome;  
    String[] telefone;  
}
```

Classe

Array



Ok, temos as classes. Mas como **instanciamos** objetos?

Usando o operador **new**

Instanciando objetos

Método `main` – o ponto de entrada para um programa em Java

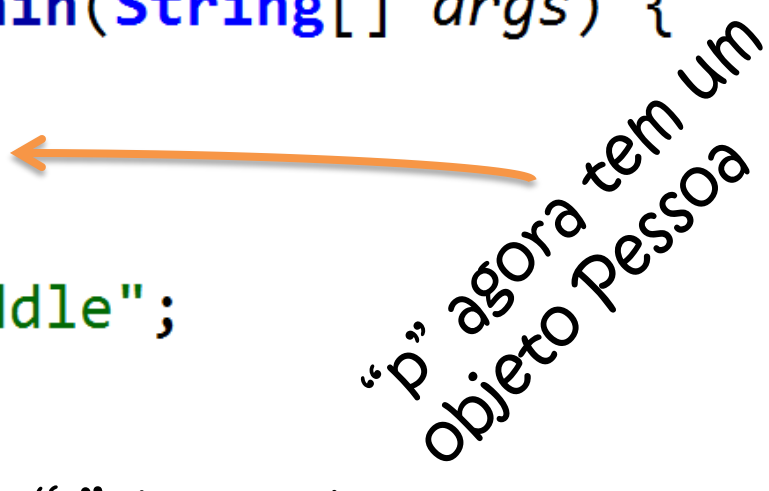
```
class App {  
    public static void main(String[] args) {  
        Pessoa p;  
        p = new Pessoa();  
        p.idade = 30;  
        p.nome = "Tom Riddle";  
    }  
}
```

The diagram consists of two orange arrows. The first arrow starts at the `main` method signature in the code and points to the `Pessoa p;` line. The second arrow starts at the `Pessoa p;` line and points to the closing curly brace of the `main` method.

} Declarando
variável "p" do
tipo `Pessoa`

Instanciando objetos

```
class App {  
    public static void main(String[] args) {  
        Pessoa p;  
        p = new Pessoa();  
        p.idade = 30;  
        p.nome = "Tom Riddle";  
    }  
}
```



Usamos “.” (ponto) para
acessar e alterar os
atributos



Falando em Java

Para instanciarmos uma classe num programa Java, ela precisa estar **disponível** para o programa

Da mesma forma que para o caso de utilizarmos **uma classe como parte de outra classe**

Veremos isso nos próximos capítulos, quando falarmos sobre pacotes – por enquanto, vamos deixar tudo na mesma pasta



Prática

1. Represente o objeto “**Veiculo**” com as seguintes propriedades
 - # de passageiros
 - # de rodas
 - Marca
 - Modelo
 - Capacidade do tanque
 - Consumo médio em km por litro



Prática – parte 2

1. Faça uma aplicação que instancia três veículos: “van”, “carro esportivo” e “moto”
 - Preencha seus atributos com valores viáveis
 - Imprima marca, modelo e **autonomia** do veículo
 - Autonomia é a quantidade de km que o veículo pode andar com um tanque de combustível