



Programação Orientada a Objetos – Aula 03

IFPE – Campus Igarassu
2016.1

Ranieri Valença
14/07/2016



Relembrando...

Programar Orientado a objetos

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



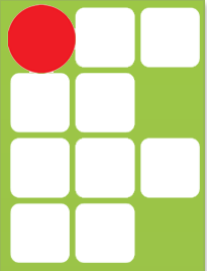
Objeto

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, ...

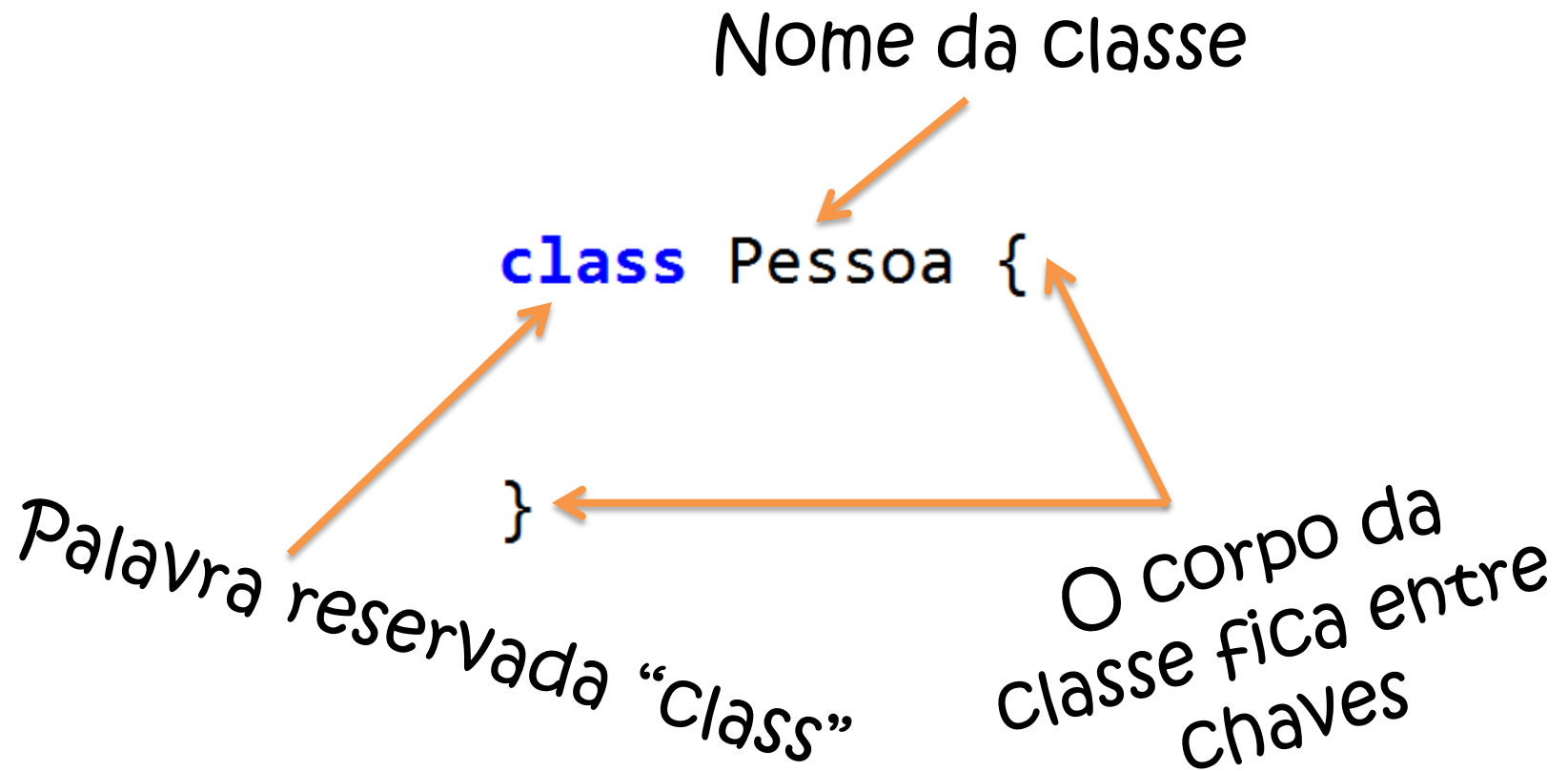


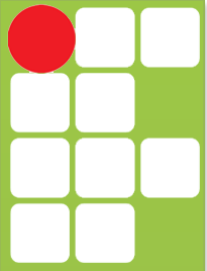
Classe

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

Declarando Atributos

`int idade;`

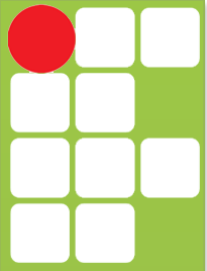
tipo *nome* *ponto e vírgula*

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

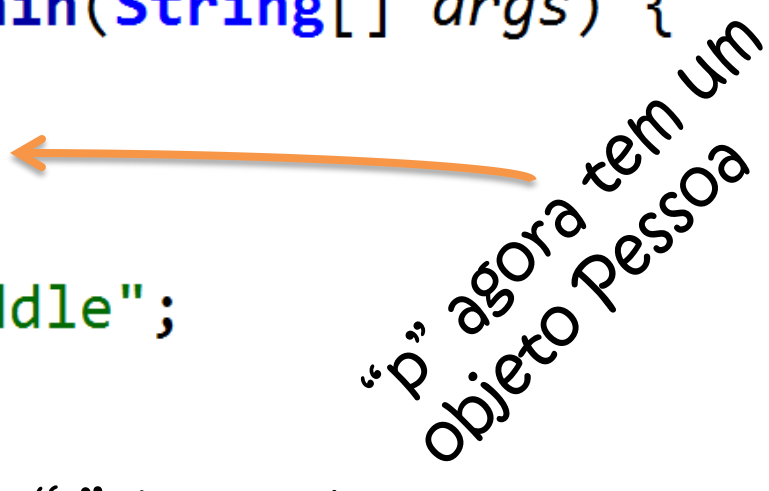
```
class App {  
    public static void main(String[] args) {  
        Pessoa p;  
        p = new Pessoa();  
        p.idade = 30;  
        p.nome = "Tom Riddle";  
    }  
}
```

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

Instanciando objetos

```
class App {
```

```
    private String middle;
```

```
}
```

Todo objeto em java precisa ser Instanciado (ou Inicializado)!!!

“p” agora tem
objeto Pessoa

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



Prática 1 (aula passada)

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 2 (aula passada)

1. Crie uma class “App” que cria três veículos (variáveis do tipo Veiculo): “van”, “carro esportivo” e “moto”
 - Preencha os valores de seus atributos
2. Imprima marca, modelo e **autonomia** do veículo
 - Autonomia é a quantidade de km que o veículo pode andar com um tanque de combustível



Prática 3

1. Dentro do App, crie uma outra variável do tipo Veiculo e tente acessar algum de seus atributos sem ter inicializado a variável



Opa...

Exceção

NullPointerException

Ponteiro nulo



Opa...

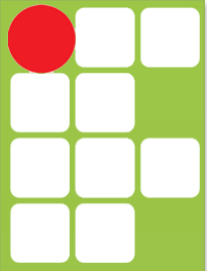
**Tenha certeza que seus
objetos estão
inicializados!**



Comportamentos dos objetos

No mundo real, os objetos têm
ações e comportamentos

Essas ações e comportamentos são
comuns a todos os **objetos** similares –
ou seja, comuns à **classe**



Traduzindo isso para nosso mundo...

Comportamentos de objetos no mundo real são traduzidos em **Métodos** no paradigma orientado a objetos



E o que é um método?

São **subprogramas** que contém
instruções de execução

A ideia é que cada método execute
uma única tarefa

Esqueleto básico de um método (por enquanto)

Tipo de retorno

void significa “sem retorno”

Parênteses

```
void imprimeNome() {  
    System.out.println("O indivíduo se chama "  
        + nome);  
}
```

Nome do método

O corpo do método
fica entre chaves

Onde ficam os métodos?? Dentro da classe!

```
class Pessoa {
```

```
    int idade;  
    float pesoEmQuilogramas;  
    float alturaEmMetros;  
    char sexo;  
    String nome;  
    String sobrenome;  
    String[] telefone;
```

Atributos

```
    void imprimeNome() {  
        System.out.println("O indivíduo se chama "  
            + nome);  
    }
```

```
}
```

Métodos



Executando métodos

Em orientação a objetos, chamamos a
execução de métodos de

invocação ~~de~~ ~~mal~~ ou **chamada**

Invocando métodos em Java

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

Operador “.” (ponto)



Prática 4

1. Crie um método chamado “**autonomia**” na classe “Veiculo”
2. Na hora de imprimir a autonomia, invoque o método “autonomia()”



Métodos que retornam algo

Algumas ações de certos objetos têm um retorno

“Por exemplo, se você pergunta a idade de uma pessoa, ela **retorna** a idade para você, que armazena aquela informação”



Métodos que retornam algo

Utilizamos a palavra especial “return” para que um método retorne algo

Obviamente quando um método retorna algum valor, seu **tipo de retorno** precisa ser especificado



Métodos que retornam algo

```
int alturaEmCentimetros() {  
    return alturaEmMetros * 100;  
}
```

```
int altura = p.alturaEmCentimetros();  
System.out.println("A criatura tem " + altura + "cm");
```



Prática 5

1. Faça o método “autonomia” da classe “Veiculo” **retornar** a autonomia do veículo ao invés de imprimi-la
2. Altere a chamada ao método e imprima a autonomia dentro do método **main** da classe “App”