



# Linguagem de Programação

Prof. Dr. Alan Souza

[alan.souza@unama.br](mailto:alan.souza@unama.br)

2020

## Abstração



- Um dos pontos mais importantes em qualquer linguagem de programação orientada a objetos;
- Imaginar o que a entidade (classe) a ser programada irá realizar dentro do sistema, a partir da necessidade do sistema;
- **IDENTIDADE:** deve ser única (sem conflitos)
- **PROPRIEDADES:** características da entidade, ou seja, os atributos.
- **MÉTODOS/FUNÇÕES:** ações/eventos que a entidade irá realizar.

# Abstração

A partir de um modelo geral, as especificidades são criadas.

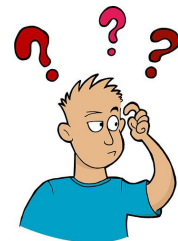


## Abstração

- Exemplo: criar um sistema para editar textos no computador.

### Abstração:

- Este sistema seria formado por quais classes?
- Quais os atributos de cada classe?
- Quais ações de cada classe?



Muitas dessas dúvidas é o usuário do sistema que precisa responder, pois o sistema será feito para ele.

-> **REUNIÕES**

# POO



## VANTAGENS DE PROGRAMAR COM ORIENTAÇÃO À OBJETOS

1. Reuso do código;
2. Aumento da produtividade;
3. Redução das linhas do projeto;
4. Separação de responsabilidades;
5. Encapsulamento;
6. Polimorfismo;
7. Componentização;
8. Facilita o teste do sistema.



# Linguagem de Programação

Prof. Dr. Alan Souza

[alan.souza@unama.br](mailto:alan.souza@unama.br)

2020

## Classes e Objetos

### DEFINIÇÃO DE CLASSE

Uma classe é a descrição de um conjunto de entidades (reais ou abstratas) do mesmo tipo e com as mesmas características e comportamentos.

**Identifique as classes nesta imagem:**



## Classes e Objetos

### DEFINIÇÃO DE CLASSE

A classe também pode ser considerada como sendo um **tipo de variável criado pelo programador**.

### COMPONENTES DA CLASSE

#### Atributos da Classe

Representam as características da classe.

#### Métodos/funções

Representam os comportamentos da classe.

## Classes e Objetos

### EXEMPLO



Qual o nome desta classe?

**Funcionario**

Quais os atributos/características?

**nome, salario, matricula,...**

Quais as funções?

**bater ponto, trabalhar, tirar férias...**

## Classes e Objetos

### CONVENÇÃO CAMEL CASE

O nome das classes devem iniciar com letra maiúscula. Se a classe for composta por duas palavras, cada início de palavra inicia com letra maiúscula.

Exemplos: **Funcionario, LinhaDeProducao.**

O nome dos atributos e das funções devem iniciar com letra minúscula. Caso seja composto por duas ou mais palavras, a primeira palavra começa com minúsculo e as outras com maiúsculo.

Exemplos: **nome, qtdFuncionarios.**

## Classes e Objetos



### Atributos - características da classe:

- Estão associados à declaração de variáveis;
- Cada atributo precisa de um tipo e de um nome (identificador);
- Exemplos:
  - String** nome;
  - int** matricula;
  - double** salario;

## Classes e Objetos



### Métodos/funções - ações da classe

- Podem receber e/ou retornar argumentos (valores).
- A variável que receber o valor retornado pelo método deve ser do mesmo tipo do retorno do método.
- O tipo de variável enviado para o método deve ser igual ou equivalente ao tipo de variável de recebimento do método.

## Classes e Objetos



### DEFINIÇÃO DE OBJETO

- Conceito semelhante ao de variável (espaço na memória principal);
- Através dele, torna-se possível **acessar atributos e/ou métodos de uma classe**, desde que estes sejam acessíveis;
- Um objeto é aquilo que se pode criar a partir da classe;
- Possui um valor inicial.

## Classes e Objetos



# PARTE PRÁTICA

## Classes e Objetos



CRIANDO A CLASSE “Funcionario” EM JAVA

```
public class Funcionario { // início da classe  
  
    // corpo da classe Funcionario  
  
} // fim da classe Funcionario
```

## Classes e Objetos



DECLARANDO OS ATRIBUTOS DA CLASSE “Funcionario” EM JAVA

```
public class Funcionario {  
    String nome;  
    int matricula;  
    double salario;  
}
```



## Classes e Objetos



DECLARANDO OS MÉTODOS/FUNÇÕES DA CLASSE “Funcionario”  
EM JAVA

```
public class Funcionario {
    //atributos omitidos
    void baterPonto(String tipo){
        System.out.println("Batendo ponto de " + tipo);
    }
    void trabalhar(){
        System.out.println("Trabalhando...");
    }
}
```

## Classes e Objetos



DECLARANDO OS MÉTODOS/FUNÇÕES DA CLASSE “Funcionario”  
EM JAVA

```
public class Funcionario {
    //códigos omitidos
    double receberSalario(){
        return salario;
    }
}
```

## Classes e Objetos



CRIANDO OBJETO DA CLASSE “Funcionario” EM JAVA

```
public class ProjetoIndustria {
    public static void main(String args[]) {
        Funcionario f1;
        f1 = new Funcionario();
        //ou...
        Funcionario f2 = new Funcionario();
    }
}
```

## Classes e Objetos

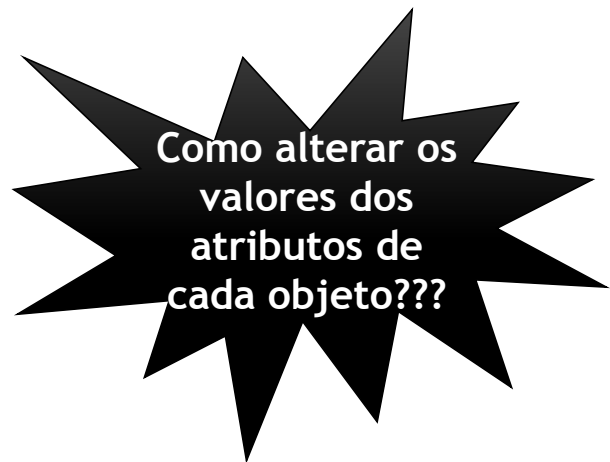
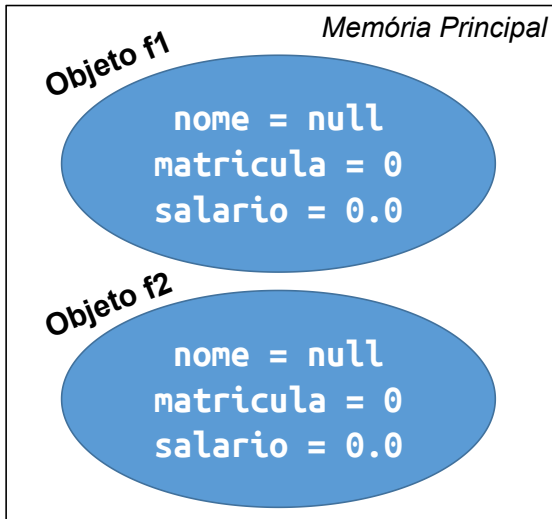


CRIANDO OBJETO DA CLASSE “Funcionario” EM JAVA

```
public class ProjetoIndustria {
    public static void main(String args[]) {
        Funcionario f1; //declaração da variável
        f1 = new Funcionario(); //instância do objeto
        //ou...
        // declaração e instanciação do objeto:
        Funcionario f2 = new Funcionario();
    }
}
```

# Classes e Objetos

## ANALISANDO OS OBJETOS CRIADOS



# Classes e Objetos

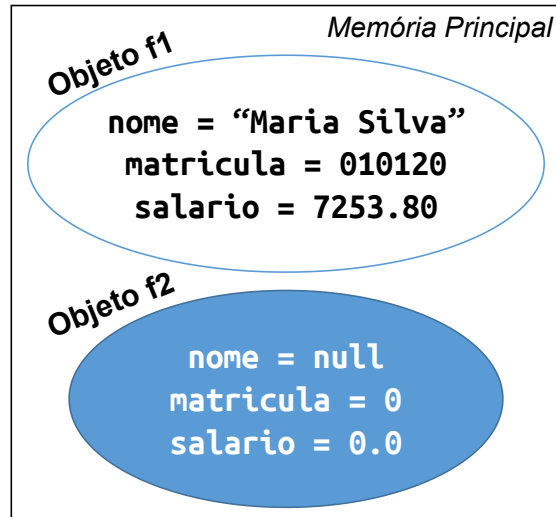
## ALTERANDO OS ATRIBUTOS DOS OBJETOS CRIADOS

```
public class ProjetoIndustria {
    public static void main(String args[]) {
        //instanciação dos objetos omitidos
        f1.nome = "Maria Silva";
        f1.matricula = 010120;
        f1.salario = 7253.80;
    }
}
```

*Operador ponto acessa atributos ou métodos da classe através do objeto*

# Classes e Objetos

## ANALISANDO OS OBJETOS CRIADOS (VALORES MUDADOS)



# Classes e Objetos

## MOSTRANDO OS VALORES DO OBJETOS f1

```

public class ProjetoIndustria {
    public static void main(String args[]) {
        //instanciação dos objetos omitidos
        System.out.println("Nome: " + f1.nome);
        System.out.println("Matrícula: " + f1.matricula);
        System.out.println("Salário: " + f1.salario);
    }
}
  
```

## Classes e Objetos



Acessando métodos pelo objeto f1

```
public class ProjetoIndustria {
    public static void main(String args[]) {
        //códigos omitidos
        //acessando métodos:
        f1.baterPonto("entrada");
        f1.trabalhar();
        double salarioF1 = f1.receberSalario();
    }
}
```

## Classes e Objetos



### EXERCÍCIOS

1) Criar um projeto de nome ProjetoIndustria no Netbeans e construir nele a classe Funcionario, de acordo com o que foi mostrado anteriormente.

Em seguida:

- crie dois objetos da classe Funcionario,
- altere os atributos através de um dos objetos,
- mostre os valores dos atributos modificados na tela,
- chame os métodos da classe através do objeto.

## Classes e Objetos

### EXERCÍCIOS

2) Analise novamente a imagem da indústria e extraia dela mais uma classe. Além disso, defina para ela:

- três atributos,
- dois métodos.



## Classes e Objetos

### EXERCÍCIOS

3) Crie a classe do exercício 2 no ProjetoIndustria no Netbeans. Depois, faça:

- Instanciação de dois objetos da classe;
- Alteração dos atributos através do objeto da classe;
- Impressão dos atributos na tela;
- Invocação dos métodos da classe.