

Classes

Orientação a Objetos - DCC025

Prof. Edmar Welington Oliveira
edmar.oliveira@ufjf.edu.br

Universidade Federal de Juiz de Fora - UFJF
Departamento de Ciência da Computação - DCC

Classes

■ Definição

- As classes são tipos de objetos que descrevem as informações armazenadas e os serviços (métodos) providos por um objeto. Em outras palavras, **são padrões/modelos a partir dos quais os objetos são criados.**
- Uma classe é a descrição de um grupo de objetos com propriedades semelhantes (atributos), mesmo comportamentos (operações) e mesmos relacionamentos com outros objetos

Classes

Classes	Atributos	Operações
Parágrafo	página a que pertence número de linhas	ler
Pessoa	nome idade peso	mudar de emprego mudar de residência
Janela	dimensão cor posição na tela	abrir fechar ler alterar
Polígono	vértices cor do contorno cor do fundo	apresentar apagar mover rodar

Classe Notação

nome da classe
atributos (opcional)
métodos (opcional)



TocaFitas
- marca : char[30]
- modelo : char[40]
- ano : char[4]
- proprietário : char[50]
- situação : char[1]
+ ligar()
+ gravar()
+ tocar()
+ parar()
+ desligar()

nome da classe



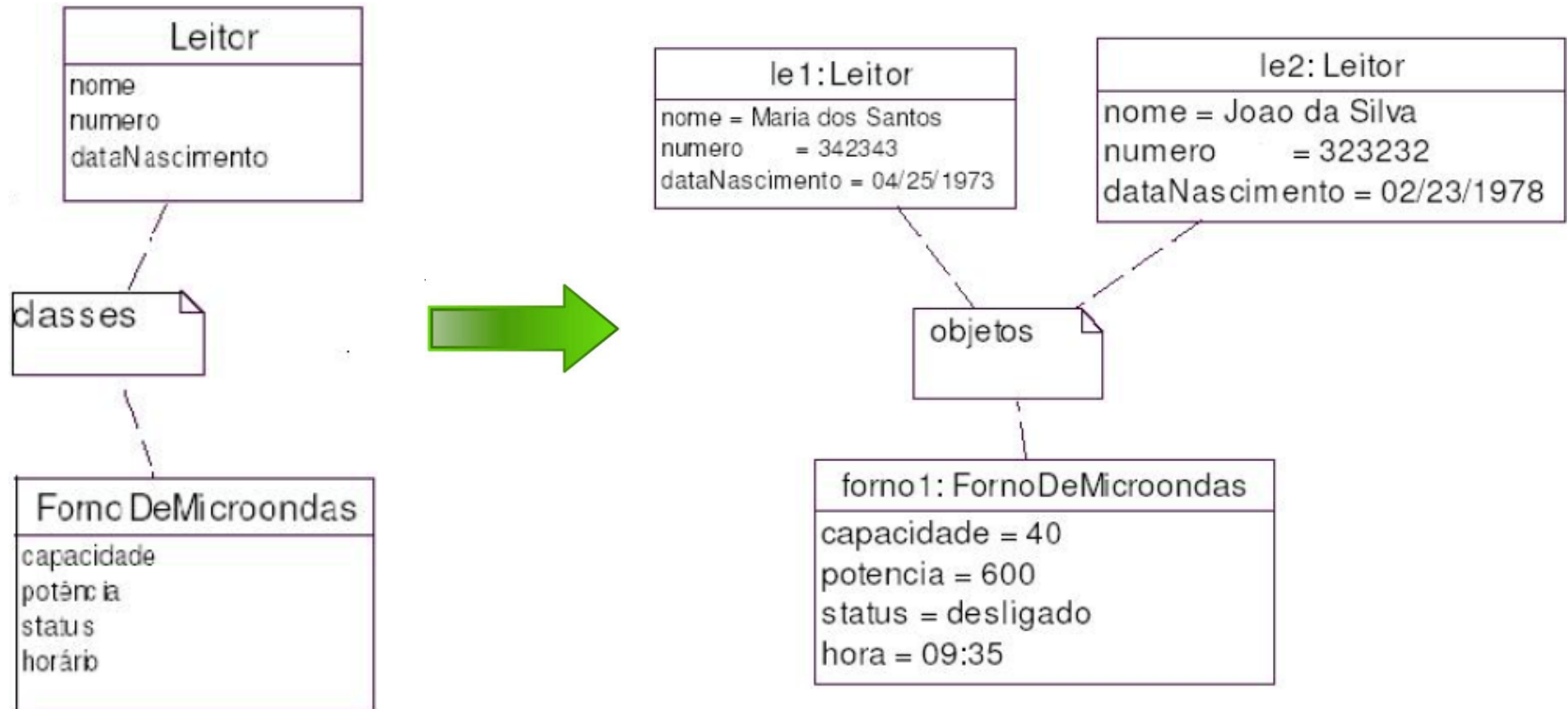
TocaFitas

Objeto e Classe

- Relação Objeto/Classe

- Objeto é uma **instância** de uma classe
- Objetos são **criados/instanciados** a partir de classes
- Objetos de uma mesma classes se diferenciam pelo valor de seus atributos

Objeto e Classe - Instanciação

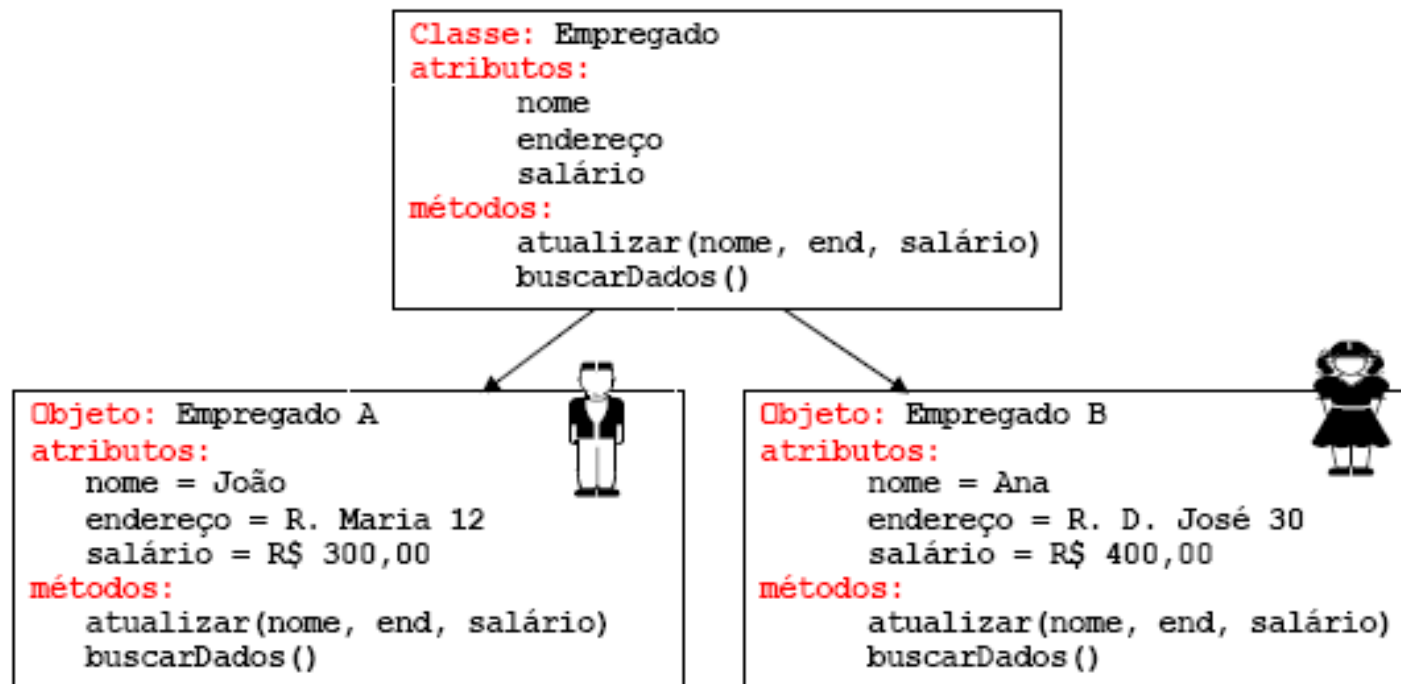


Instanciação

- Instanciação

- Para criar um objeto é necessário criar uma classe
- Criada a classe, é preciso instanciar um objeto a partir dela
- Todos os objetos criados a partir de uma classe possuem os **mesmos atributos** e os **mesmos métodos**. Quando uma instância da classe é gerada, um objeto é criado e o sistema aloca memória para seus atributos.

Instanciação



Classes

- Atributos e Métodos
 - Atributos: dados que pertencem a cada instância da classe (objetos)
 - São definidos sob a forma de variáveis
 - Métodos: definem o comportamento do objeto
 - Definidos sob a forma de módulos

Métodos - Assinatura


Visibilidade do método

Nome do atributo

```
public <tipo de retorno> <nome do método> ( <lista de argumento> ) {  
    <corpo>  
}
```

Tipo de retorno: inteiro, string, booleano, etc.

Argumentos necessários ao método para sua execução



Para um método de verificação de CPF, por exemplo, faz-se necessário informar ao método, como parâmetro, o número do CPF

```
public boolean validaCPF(String cpf){  
    return  
}
```

retorna "true" ou "false" - retorno depende da sua implementação

Implementação

Classe

- Implementação de Classe

- Criar uma classe “Aluno”

- Atributos: nome e matricula
 - Métodos: retornaNome, retornaMatricula, alteraNome, alteraMatrícula

- Objetivos

- Analisar a criação de classes em Java
 - Analisar a especificação de atributos e métodos

Classe Implementada

```
3 public class Aluno {  
4  
5     private int matricula;  
6     private String nome;  
7  
8     public int retornaMatricula() {  
9         return matricula;  
10    }  
11  
12    public String retornaNome() {  
13        return nome;  
14    }  
15  
16    public void alteraNome(String nomeAluno) {  
17        nome = nomeAluno;  
18    }  
19  
20    public void alteraMatricula(int matAluno) {  
21        matricula = matAluno;  
22    }  
23 }
```

Classe Implementada

```
3 public class Aluno {  
4  
5     private int matricula;  
6     private String nome;  
7  
8     public int retornaMatricula() {  
9         return matricula;  
10    }  
11  
12    public String retornaNome() {  
13        return nome;  
14    }  
15  
16    public void alteraNome(String nomeAluno) {  
17        nome = nomeAluno;  
18    }  
19  
20    public void alteraMatricula(int matAluno) {  
21        matricula = matAluno;  
22    }  
23 }
```

Atributos da Classe Aluno

Métodos de Retorno

Métodos para Alteração

Classe Teste

- Implementação para Teste
 - Criar uma classe de teste “Principal”
 - Criar o método main()
 - Instanciar um objeto da classe Aluno
 - Testar os métodos da classe Aluno
 - Objetivos
 - Analisar a instanciação de objetos
 - Testar chamada de métodos

Classe Implementada

```
3 public class Principal {  
4  
5     public static void main(String args[]) {  
6  
7         Aluno aluno01 = new Aluno();  
8  
9         System.out.println(aluno01.retornaMatricula());  
10        System.out.println(aluno01.retornaNome());  
11  
12        aluno01.alteraMatricula(1010);  
13        aluno01.alteraNome("NomeTeste");  
14  
15        System.out.println(aluno01.retornaMatricula());  
16        System.out.println(aluno01.retornaNome());  
17    }  
18 }
```


Classe Implementada

```
3 public class Principal {  
4  
5     public static void main(String args[]) {  
6  
7         Aluno aluno01 = new Aluno();  
8  
9         System.out.println(aluno01.retornaMatricula());  
10        System.out.println(aluno01.retornaNome());  
11  
12        aluno01.alteraMatricula(1010);  
13        aluno01.alteraNome("NomeTeste");  
14  
15        System.out.println(aluno01.retornaMatricula());  
16        System.out.println(aluno01.retornaNome());  
17    }  
18 }
```

Instanciação

Chamada de Método

Chamada de Método

Chamada de Método

Classes

Orientação a Objetos - DCC025

Prof. Edmar Welington Oliveira
edmar.oliveira@ufjf.edu.br

Universidade Federal de Juiz de Fora - UFJF
Departamento de Ciência da Computação - DCC