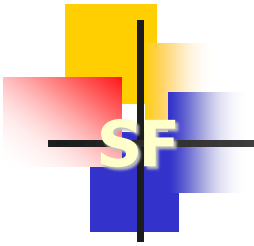


Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Programação Orientada a Objetos

Semana 05
Classes, Objetos e Métodos

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Reflexão

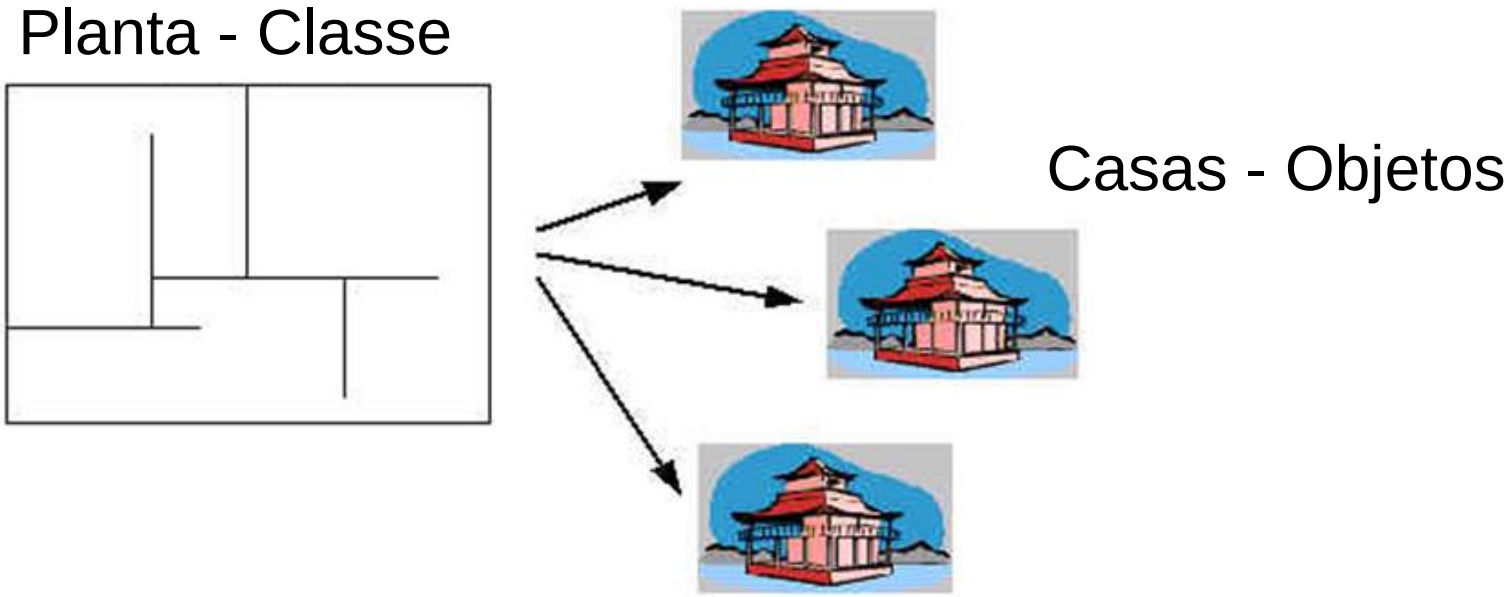
"As escolas existem não para ensinar as respostas, mas para ensinar as perguntas. As respostas nos permitem andar sobre a terra firme. Mas somente as perguntas nos permitem entrar pelo mar desconhecido."

Rubem Alves

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Classes e Objetos

Na orientação a objetos tudo se inicia a partir da classe. A classe é uma espécie de molde, a partir do qual são criados objetos do mesmo tipo.



http://www.dsc.ufcg.edu.br/~jacques/cursos/p2/html/oo/usando.htm

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Classes

Objeto do mundo real

http://animais.colorir.com/caes/cachorro.html

É representado

Classe

Cachorro

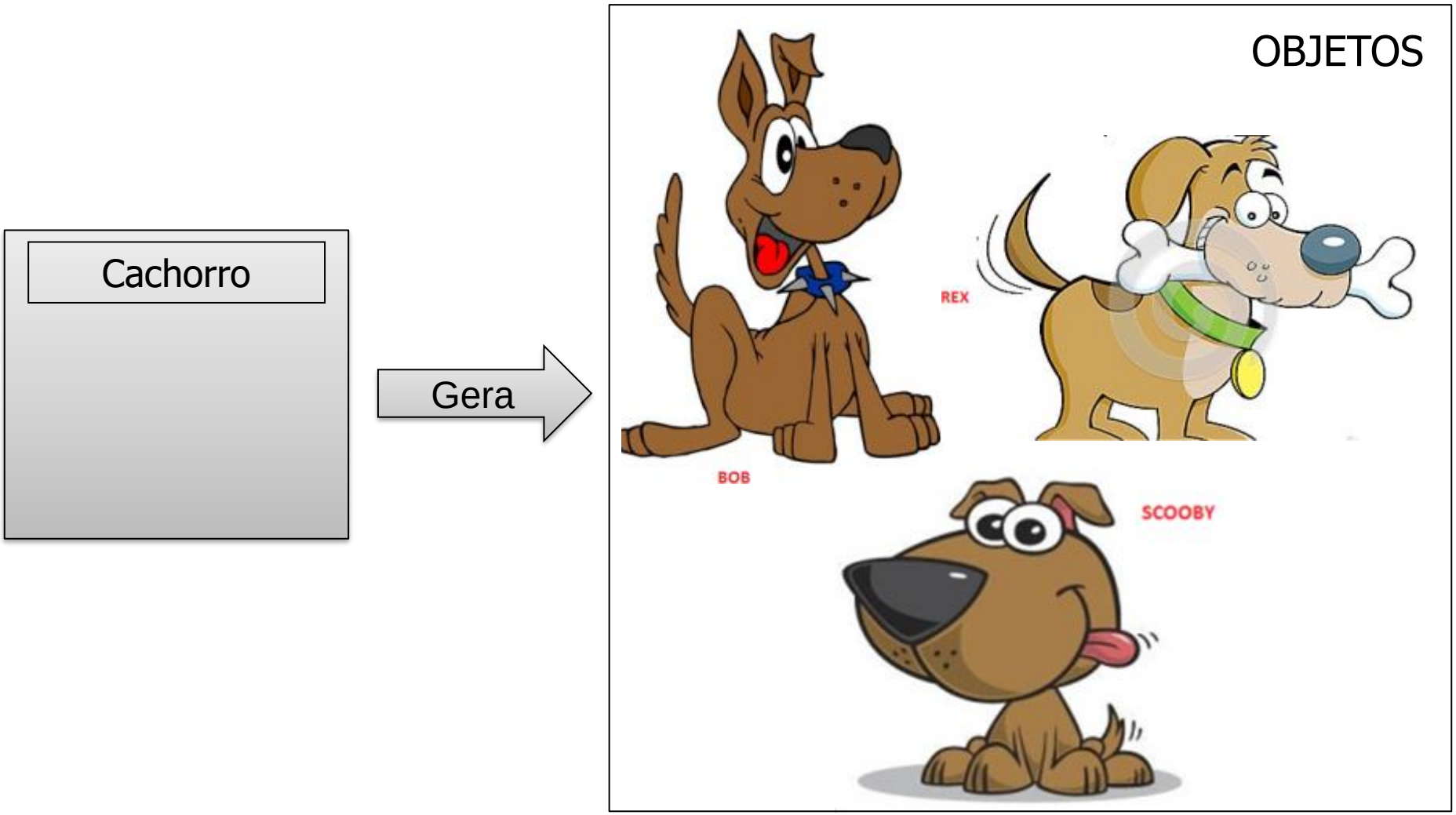
Contém

Atributos

Métodos

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Uma classe gera muitos objetos



Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Estrutura

Uma classe é declarada contendo um qualificador (public, abstract, final), a palavra-reservada ***class*** seguida do nome da classe e de seu corpo entre chaves.

Sintaxe:

qualificador class NomeDaClasse { ... }

Regras para nomes de classes

- ✓ Não podem conter espaços
- ✓ Deve ser iniciado por uma letra (maiúscula) ou '_' ou '$'
- ✓ Recomenda-se não utilizar acentos
- ✓ Pode conter números
- ✓ Não pode ser uma palavra reservada da linguagem Java

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Estrutura

```
/*
* A estrutura básica de uma classe
* em Java pode ser observada nesse
* pequeno exemplo
*/

public class NomeDaClasse {

      /* Corpo da classe
      definição de atributos e métodos */

}//Fim da declaração da classe
```

Início do bloco de comentário

Fim do bloco de comentário

Início do comentário de linha única

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Estrutura

Exemplo de classe executável diretamente

```
/**
 * A classe 'OlaTurma' implementa uma aplicação Java que simplesmente
imprime na
 * saída padrão (console) o
 */
class OlaTurma {

    public static void main(String[] args) {

        System.out.println("Olá turma!!!"); // Mostra o texto entre aspas

    }
}
```

Cabeçalho do método especial `main`

Início do corpo do método

Fim do corpo do método

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Qualificadores da classe

public: indica que a classe é visível (e por isso pode ser usada) por outras classes;

final: indica que a classe não pode ser herdada (ou redefinida) por outras classes;

abstract: indica que a classe não admite a geração de instâncias.

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Métodos

São também denominados de operações em Java.
Nomes de métodos também seguem regras como os nomes de classes:

- ✓ Não podem conter espaços
- ✓ Deve ser iniciado por uma letra ou '_' ou '$'
- ✓ Recomenda-se não utilizar acentos
- ✓ Pode conter números
- ✓ Não pode ser uma palavra reservada da linguagem Java
- ✓ Não podem ser criados dentro de outras operações, nem fora do corpo da classe à que pertencem

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Métodos

```
public class Pessoa
{
        private String nome;
        private int idade;
        private char sexo;

        public void imprimeNome() {
                System.out.println(nome);
        }

        public void imprimeIdade() {
                System.out.println(idade);
        }

        public void imprimeSexo() {
                System.out.println(sexo);
        }
} // fim da classe
```

Método para imprimir o nome

Método para imprimir a idade

Método para imprimir o sexo

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Método Construtor

Um construtor é um método especial responsável por criar o objeto em memória, e segue alguns critérios para sua criação:

- ✓ A assinatura de um construtor diferencia-se das assinaturas dos outros métodos pois não tem nenhum tipo de retorno (nem mesmo void).
- ✓ O nome do construtor deve ser o próprio nome da classe.
- ✓ O construtor pode receber argumentos, como qualquer método.
- ✓ Mais de um construtor pode ser definido para uma classe.

Sintaxe:

qualificador(mesmo da classe) + nome da classe + parâmetros.

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Método Construtor

Exemplo

```
public class Automovel {
    private String cor;
    private double preco;
    private String modelo;

    public Automovel() {
        System.out.println("Consrutor padrão da classe Automóvel.");
    }

    public Automovel(String modelo, double preco) {
        this.cor = "PRETA";
        this.modelo = modelo;
        this.preco = preco;
    }

    public Automovel(String cor, String modelo, double preco) {
        this.cor = cor;
        this.modelo = modelo;
        this.preco = preco;
    }
}
```

Construtor padrão

Construtor com 2 parâmetros

Construtor com 3 parâmetros

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Objetos

Quando se cria um objeto, esse objeto adquire:

- ✓ Um espaço em memória para armazenar seu estado (os valores de seu conjunto de atributos, definidos pela classe)
- ✓ Um conjunto de operações que podem ser aplicadas ao objeto (o conjunto de métodos definidos pela classe).

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Referências e criação de objetos

Para instanciar um objeto em Java, utiliza-se o operador **new**

Exemplo:
criação de dois objetos a partir da classe Automóvel

Criação do método construtor

```
public class Automovel { ...
        public Automovel(String c, String m, double p) {
                cor = c; modelo = m; preco = p;
}
```

Objetos da classe Automovel

```
        Automovel carro, moto;

        carro = new Automovel("Cinza", "Toyota Corolla",
        70.000);
        moto  = new Automovel("Vermelha", "NXR Bros", 10.000);
```

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Referências e criação de objetos

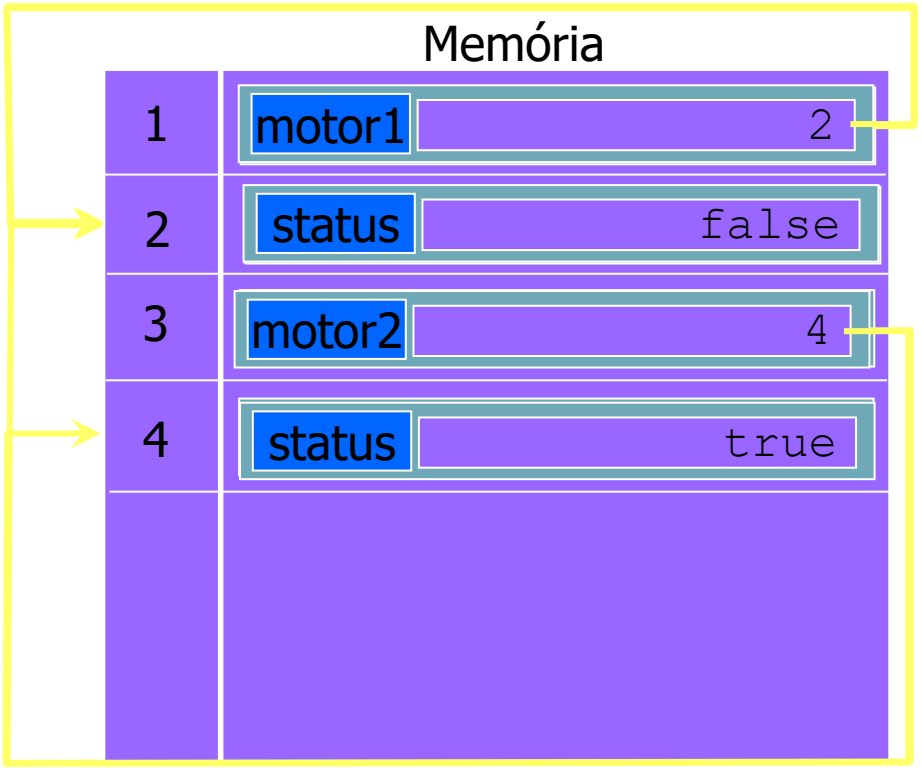
```
3  public class Motor {
4      boolean status;
5
6      public void ligarMotor() {
7          status = true;
8      }
9
10     public void desligarMotor() {
11         status = false;
12     }
13 }
```

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Referências e criação de objetos

- Agora, tanto `motor1`, quanto `motor2` estão apontando (referenciando) para um mesmo objeto, na posição 2 de memória
- O objeto na posição 2 também pode ser manipulado através da referência `motor2`

```
Motor motor1 = null;
motor1 = new Motor();
motor1.ligarMotor();
Motor motor2 = motor1;
motor2.desligarMotor();
motor2 = new Motor();
motor2.ligarMotor();
```



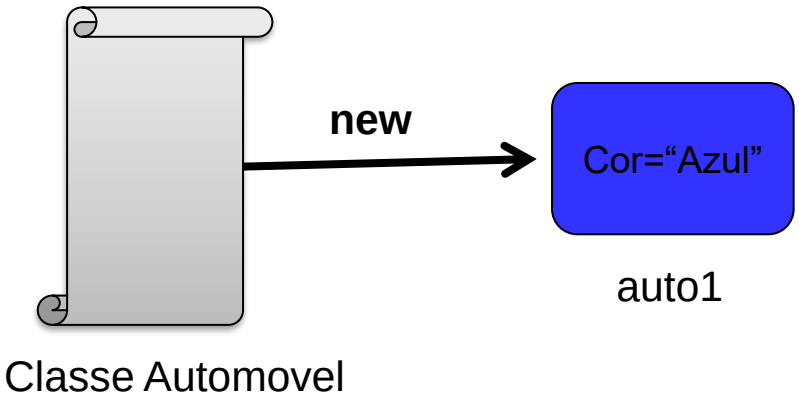
Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Variáveis de instância

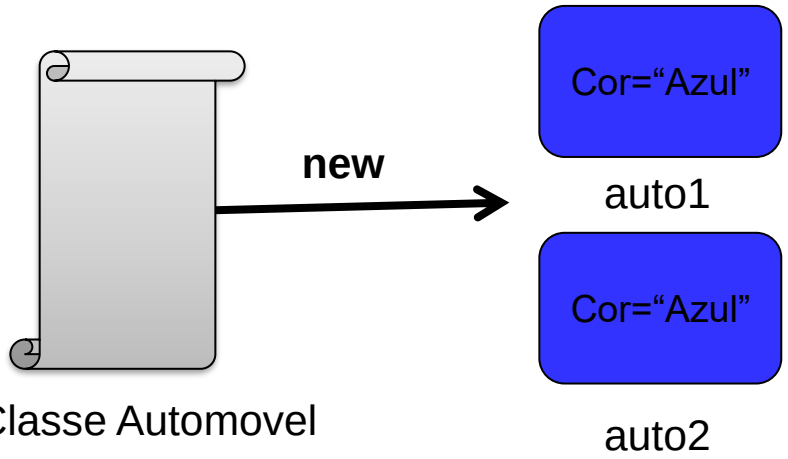
```
public class Automovel {
   public String cor = "Azul";
}
```

Todo objeto do tipo Automovel instanciado terá a sua variável cor inicializada com o valor “Azul”

```
Automovel auto1 = new Automovel();
```




```
Automovel auto2 = new Automovel();
```



Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Variáveis de instância

Cada um dos objetos do tipo Automovel tem sua própria variável de instância.

Uma alteração em uma variável do objeto `auto1` não afeta a variável do objeto `auto2`.

```
auto1.cor = "Cinza";
```

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Variáveis de instância x Variáveis locais

Variáveis de instância:

✓ Variáveis declaradas na declaração de classe;

✓ Cada objeto (instância) da classe tem uma instância separada da variável;

✓ Existe enquanto o objeto existir: **antes** e **depois** de chamadas aos métodos;

Variáveis locais:

✓ Declaradas no corpo do método;

✓ Só podem ser utilizadas nesse método;

✓ Só existem **durante** a execução do método;

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Exercícios

1. Faça uma aplicação que implemente uma classe chamada Aluno para definir os objetos que representarão os alunos de uma escola. Os atributos da classe são nome, RG e data de nascimento.

   a) Faça uma classe chamada UsaAluno.

   b) Crie dois objetos da classe Aluno.

   c) Altere os valores dos atributos desses objetos e exiba no Console os valores armazenados nesses atributos.

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Exercícios

2. Faça uma aplicação que implemente uma classe chamada Pedido, considerando que esta deva possuir:

Construtor
Deve ter um construtor, cujos parâmetros são: número do Pedido, Nome do Cliente e Descrição.

Atributos
Número do Pedido
Nome do Cliente
Descrição do Pedido

Métodos
Modifica o nome do Cliente
Modifica a descrição do pedido
Retorna a descrição do pedido
Retorna o nome do cliente.

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Exercícios

3. Crie um programa que instancie dois objetos da classe Pedido e que modifique os valores dos atributos: Nome do Cliente e Descrição do Pedido. Antes e após as modificações, deverá imprimir os valores dos atributos, utilizando os métodos da classe.

4. Faça uma aplicação identificando as classes, atributos e métodos necessários para modelar e implementar uma garrafa contendo um status para saber se já foi aberta e um valor em ml para saber seu conteúdo.

5. Faça uma aplicação identificando as classes, atributos e métodos necessários para modelar e implementar uma TV contendo um status para saber se está ligada ou desligada e valores do nível de volume e do número do canal. Métodos para mudar o canal (de um em um) e aumentar/reduzir o volume. A troca de volume e canal apenas devem funcionar com a TV ligada.

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Exercícios

6. Faça uma aplicação identificando as classes, atributos e métodos necessários para modelar e implementar um forno micro-ondas contendo uma método para definir o tempo de aquecimento (em segundos) e outro para acionar o forno (que mostra a contagem regressiva do tempo).

7. Faça uma aplicação identificando as classes, atributos e métodos necessários para modelar e implementar uma lâmpada que possa ter três estados: apagada, acesa e meia-luz. Inclua, no modelo “Lâmpada”, uma operação “estáLigada” que retorne verdadeiro se a lâmpada estiver ligada e falso, caso contrário.

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Exercícios

8. Observe o código abaixo:

```
3  public class Funcionario {
4      private String nome;
5      private double salario;
6
7      private Funcionario(String nome, double salario) {
8          this.nome = nome;
9          this.salario = salario;
10     }
11     public void mostrar() {
12         System.out.println(nome + ", ganha " + salario + " reais");
13     }
14     public void aumentarSalario(double aumento) {
15         salario *= 1 + aumento / 100;
16     }
17 }
```

a) Crie uma nova classe chamada UsaFuncionario com um método main

b) Declare um array destes objetos, e o preencha com alguns Empregados.

c) Imprima o conteúdo do array, com base no comportamento já definido para os Funcionarios.

d) Dê um aumento a todos os funcionários e imprima o array novamente.

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Exercícios

9. Faça uma aplicação que implemente uma classe chamada Produto com os atributos nome, marca e preco.

   a) Um construtor padrão
   b) Um construtor com 2 parâmetros
   c) Um construtor com 3 parâmetros
   d) Crie uma classe UsaProduto para testar os construtores da classe.

10. Faça uma aplicação que implemente uma classe chamada Gerente para definir os objetos que representarão os gerentes de um determinado banco. Defina dois métodos de aumento salarial nessa classe. O primeiro deve aumentar o salário com uma taxa fixa de 10%. O segundo deve aumentar o salário com uma taxa variável.