FeMASS
FACULDADE PROF. MIGUEL ÂNGELO DA SILVA SANTOS

Projeto de Sistemas I

Faculdade Prof. Miguel Ângelo da Silva Santos

Material 4 – Orientação a Objetos em Java (classes, construtores e sobrecarga)

Professor: Isac Mendes Lacerda, M.Sc., PMP, CSM
e-mail: isac.curso@gmail.com

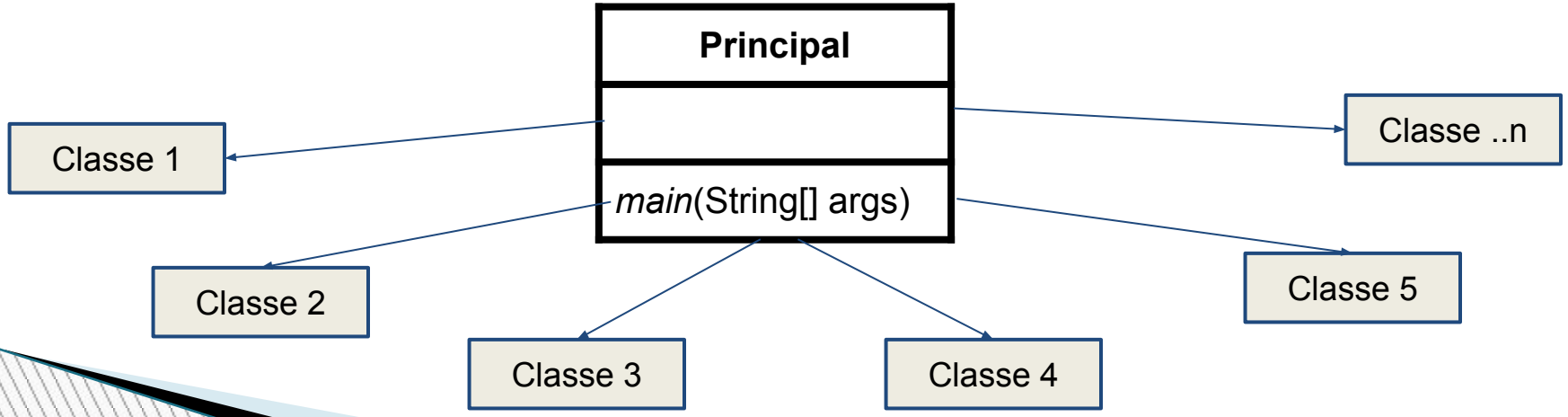
Tópicos

- Classe principal e classes comuns
- Atributos e métodos (com e sem retorno)
- Instanciação de objetos
- Sobrecarga
- Construtor

Classes (principal e comum)

- Projetos em Java normalmente têm uma classe principal e várias comuns.

- A classe principal tem um método *main* que serve como *kick off.*



Classes vazias (principal e comum)

```
10      * @author Isac
11      */
12     public class Principal {
13
14         /**
15          * @param args the command line arguments
16          */
17
18         public static void main(String[] args) {
19             // O código começa a ser executado por aqui!
20
21
22         }
23     }
```

```
6      package com.mycompany.classeprincipal;
7
8      /**
9       *
10      * @author Dell
11      */
12     public class Classe_1 {
13
14     }
```

Atributos e métodos

- Podem ser de classes ou de objetos.

- Se forem de classes têm a palavra reservada *static* na linha de definição, antes dos respectivos nomes.

Atributos e métodos

```
12  public class Aluno {
13      String nome;
14      int idade;                          //Atributos de objeto
15      double altura;
16
17      static int qtd;                     //Atributo de classe
18
19      public void estudar(){
20          System.out.println("Estudando agora...");
21      }
22      public void testar(){               //Métodos de objeto sem retorno
23          System.out.println("Realizando um teste agora...");
24      }
25      public String dormir(){
26          return("Dormingo agora...");
27      }                                   //Método de objeto com retorno
28      public static int get_qtd(){
29          return(Aluno.qtd);
30      }                                   //Método de classe com retorno
31  }
```

Instanciação de objetos

- Normalmente a instanciação de objetos em Java utiliza a palavra reservada ***new***.

```
18      public static void main(String[] args) {
19          // O código começa a ser executado por aqui!
20          ArrayList<Aluno> alunos = new ArrayList();
21
22          Aluno al = new Aluno();
23          System.out.println("Nome: " + al.nome + "Idade: " + al.idade);
24
25          for (int i=0; i<10; i++){
26              alunos.add(new Aluno());
27          }
28          for (int i=0; i<alunos.size(); i++){
29              Aluno aluno_corrente = alunos.get(i);
30              System.out.println(aluno_corrente.nome);
31          }
32
            System.out.println("Quantos objetos de criados: " + Aluno.qtd);
34
35      }
```

Instanciação de objetos

- Acesso a atributos e métodos de classe *static* podem ser feitos assim ***nome_classe.nome_atributo*** ou ***nome_classe.nome_método()***:

```
18      public static void main(String[] args) {
19          // O código começa a ser executado por aqui!
20          ArrayList<Aluno> alunos = new ArrayList();
21
22          Aluno al = new Aluno();
23          System.out.println("Nome: " + al.nome + "Idade: " + al.idade);
24
25          for (int i=0; i<10; i++){
26              alunos.add(new Aluno());
27          }
28          for (int i=0; i<alunos.size(); i++){
29              Aluno aluno_corrente = alunos.get(i);
30              System.out.println(aluno_corrente.nome);
31          }
32
            System.out.println("Quantos objetos de criados: " + Aluno.qtd);
34
35      }
```

Sobrecarga

- Métodos com mesmo nome em uma classe, mas com assinaturas diferentes. A linguagem consegue distinguir qual método deve ser executado em uma invocação, considerando o número, a ordem e tipo de parâmetros informados.

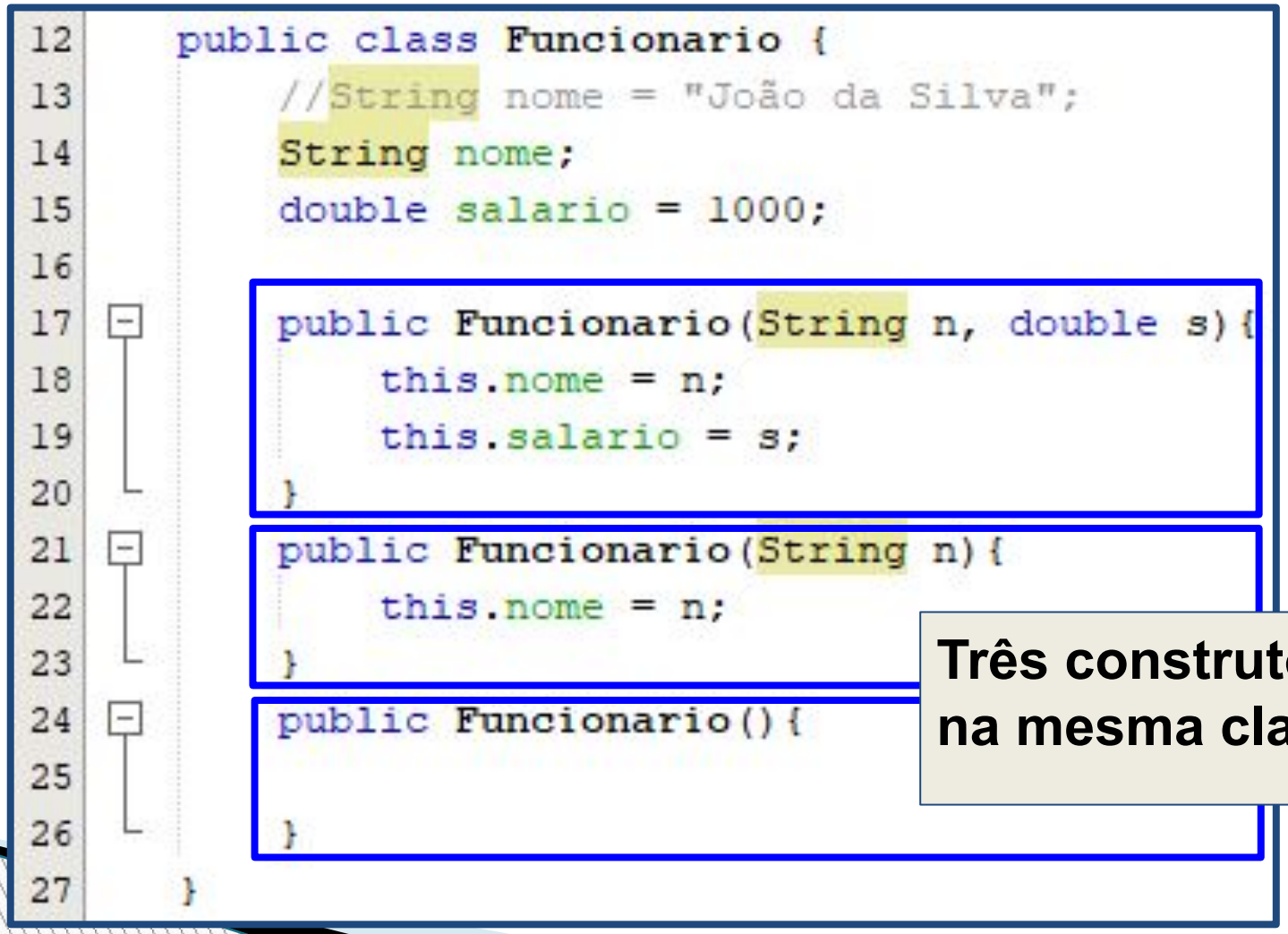
Sobrecarga

```
12  public class Sobrecarga {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          area(10, 2);
20          area(50);
21      }
22      public static void area(double lado){
23          System.out.println("Área quadrado: " + lado * lado);
24      }
25      public static void area(double base, double altura){
26          System.out.println("Área retângulo: " + base * altura);
27      }
28
29  }
```

Construtor

- estrutura que faz parte da classe e é usado para formar objetos e iniciar valores nos atributos dos objetos que nascem.
- Em Java, são definidos explicitamente ou implicitamente.
- Quando são definidos explicitamente devem ter o mesmo nome da classe.
- É possível trabalhar com vários construtores explicitamente em uma classe em Java (que é uma forma de Sobrecarga).

Construtor (com Sobrecarga)



Construtor

Por que a linha 22 não funciona?

```
12  public class Construtor {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19
20          Funcionario F1 = new Funcionario();
21          Funcionario F2 = new Funcionario("Paulo", 5000);
            Funcionario F3 = new Funcionario(5000, "Maria");
23          Funcionario F4 = new Funcionario("Isabela");
24      }
25  }
```

Exercício 1

Crie uma classe que contenha três métodos com o nome ‘média’, utilizando o conceito de sobrecarga. Os métodos devem calcular a média de dois, três e quatro valores de entrada. Utilize a classe principal para criar e testar os métodos do seu objeto.

Exercício 2

Crie uma classe chamada Lâmpada que contenha os atributos “código”, “marca”, “volts” e “estado”. Com isso, você deve instanciar pelo menos dois objetos, na classe principal. Depois disso, crie um método para ligar e desligar a lâmpada e outro para observar o estado da lâmpada.

Exercício 3

Crie uma classe chamada TV contendo os seguintes atributos: “ligado”, “canal” e “volume”. Defina dois métodos construtores (o *default* e outro para ligar a TV num canal qualquer e com volume 25. Depois disso, elabore métodos para as seguintes ações:

+ Ligar e desligar uma instância;
+ Aumentar e reduzir o volume de um em um (de 0 a 100);
+ Trocar o canal de 0 a 999;
+ Mostrar todos os atributos de uma instância.

Exercício 4

Crie uma classe chamada Caminhão contendo os atributos combustível e velocidade. Os combustíveis válidos são gasolina, diesel e gás. Ao ser criado, todo caminhão deve possuir a velocidade zero e combustível gasolina. Deve ser possível substituir o combustível, aumentar e reduzir a velocidade (de 0 a 150).

Exercício 5

Escreva uma classe que represente país. Armazene as seguintes informações dos países: nome, capital, dimensão, lista de países que faz fronteira. Represente a classe e forneça os seguintes métodos:

- Construtor que inicialize o nome, capital e a dimensão.
- Métodos de acesso (get e set) para as propriedades.
- Um método que permita verificar se dois países são iguais. Dois países são iguais se tiverem o mesmo nome e a mesma capital.
- Um método que define quais outros países fazem fronteira (note que um país não pode fazer fronteira com ele mesmo).
- Um método que retorne uma lista de países que fazem fronteira
- Um método que receba um outro país como parâmetro e retorne uma lista de vizinhos comuns aos dois países.