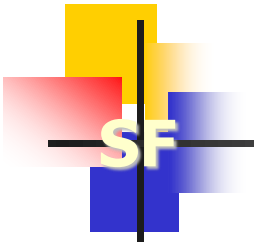


Prof. Dr. Sérgio Furgeri – sergio.furgeri@fatec.sp.gov.br

Programação Orientada a Objetos

Manipulação de métodos

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

Métodos

Termos importantes:

- ✓ Mecanismo de comunicação
- ✓ Parâmetros / Argumentos
- ✓ Visibilidade
- ✓ Assinatura
- ✓ Sem retorno
- ✓ Com retorno
- ✓ Sincronizado / não sincronizado

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

Mecanismo de comunicação

Aumentam a facilidade!

- Reduzem a complexidade
  - ✓ Abstração
  - ✓ Encapsulam a informação
  - ✓ Minimizam o tamanho do código

- Aumentam a manutenibilidade e a correção
  - ✓ Evitam duplicação do código
  - ✓ Limitam o efeito das mudanças
  - ✓ Promovem a reutilização do código

Diminuem o custo!

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

Assinatura de Métodos

A assinatura de um método permite sua identificação.

A assinatura consiste do nome do método e da lista de parâmetros

Exemplos:

public static **void main(String args[])**

public static **double sqrt(double pVal**)

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

Assinatura de Métodos

É possível sobrecarregar um método, colocando duas definições diferentes para ele.

Isto significa que, dependendo dos parâmetros que ele receber, ele vai ter um comportamento diferente.

Exemplo:

public static double **sqrt**(double pVal)

public static float **sqrt**(float pVal)

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

Assinatura de Métodos

Logo, se dois métodos têm o mesmo nome, o método correto será chamado com base nos argumentos que lhe são passados.

Por exemplo, o código abaixo chama um método definido anteriormente.

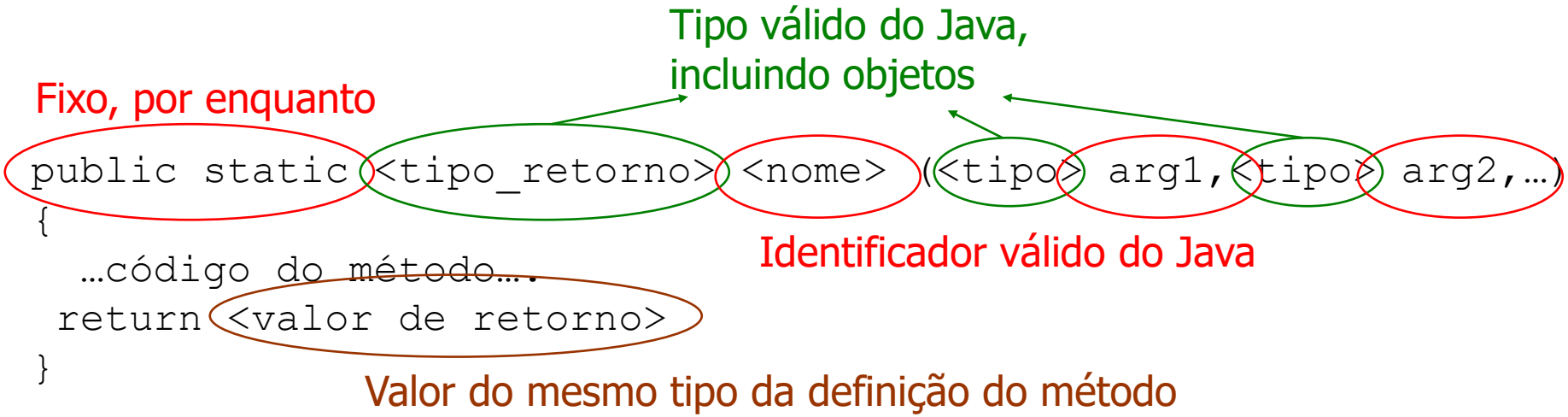
```
float  hipotenusa = 10.0f;

MyClass.sqrt(hipotenusa);
```

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

Definindo métodos

Para declarar métodos, usamos a seguinte sintaxe:



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

Métodos sem retorno

Não retornam valores. Os métodos que não retornam valores devem ser definidos como **void**.

Sintaxe:

`modificador-de-acesso` ***`void`*** `nome-do-método ([lista-de-argumentos]) { código do corpo }`

Exemplo:

O exemplo mostra a chamada de um método (imprime) que imprime na tela uma frase qualquer.

```
3  public class TesteMetodo {
4      public static void main(String args[]) {
5          semRetorno(); // invocação do método
6      }
7      private static void semRetorno() { // declaração do método
8          System.out.println("Aprendendo a Linguagem Java");
9      }
10 }
```

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

Métodos com retorno

São métodos que retornam valores, os quais podem ser comparados às funções de outras linguagens.

Sintaxe:
modificador-de-acesso **[tipo de retorno do método]** nome-do-método ([lista-de-argumentos]) { código do corpo }

Exemplo:
O exemplo mostra a chamada de um método (retornaNome) que retorna uma String.

```
public String retornaNome() { return nome; }
```

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

Métodos com passagem de parâmetros

São métodos que recebem parâmetros como valores de entrada.

Sintaxe:
modificador-de-acesso **[tipo do método]** nome-do-método ([**parâmetros**]) { código do corpo }

O exemplo seguinte mostra a chamada de um método (soma) que recebe dois valores como passagem de parâmetro.

```
3  import javax.swing.*;
4  public class PassagemParametro {
5      public static void main(String args[]) {
6          int n1 = Integer.parseInt(JOptionPane.showInputDialog
7                  (null, "forneça o 1º número inteiro"));
8          int n2 = Integer.parseInt(JOptionPane.showInputDialog
9                  (null, "forneça o 2º número inteiro"));
10         int res = soma(n1, n2);
11         JOptionPane.showMessageDialog(null, "Numeros fornecidos : "
12                 + n1 + ", " + n2 + "\nResultado = " + res);
13     }
14
15     public static int soma(int numero1, int numero2) { // declaração do método
16         return (numero1 + numero2); // retorna a soma dos números passados }
17     }
18 }
```

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

Exercícios

1. Um estudante de geometria deseja estudar mais sobre triângulos. Para ajudá-lo nesta tarefa, crie uma aplicação que armazena as medidas dos lados de um triângulo e contém os seguintes métodos:

a) validarLados: recebe três medidas (lados) e informe em tela se essas medidas podem formar (ou não) um triângulo. Para isto, a soma de dois lados quaisquer deve ser maior que o terceiro lado.

b) verificarTriangulo: este método deve retornar uma String indicando se o triangulo é equilátero (todos os lados com medidas iguais), isósceles (apenas dois lados com medidas iguais) ou escaleno (todos os lados com medidas diferentes).

c) ehTrianguloRetangulo: este é um método que avalia se os lados formam um triângulo retângulo. lembre-se que para ser retângulo, a soma do quadrado dos dois lados menores (catetos) tem que ser igual ao quadrado do lado maior (hipotenusa). Deve retornar verdadeiro ou falso.

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

Exercícios

2. Crie uma aplicação que contenha dois métodos: um denominado letraVogal que retornará true se a letra recebida for uma vogal e false caso contrário e outro denominado letraConsoante que faz o mesmo para uma consoante.

3. Um professor aplicou 3 provas durante um semestre, mas só vai levar em conta as duas notas mais altas para calcular a média. Crie uma aplicação em Java que peça o valor das 3 notas e execute os métodos seguintes:

a) **calcularMedia**: apresenta em tela a média das duas notas mais altas. Esse método deve receber a nota das três provas em tipo float.

b) **maiorNota**: apresenta em tela a maior nota entre as 3 provas. Esse método deve receber a nota das três provas em tipo float.

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

Exercícios

4. Crie uma aplicação que contenha um método sem retorno, para mostrar na tela 10 vezes a frase "Esse é um método sem retorno".

5. Crie uma aplicação contendo um método com passagem de parâmetro que realize a conversão entre as temperaturas Celsius e Farenheit. Sendo C a temperatura em Célsius e F em farenheit, as fórmulas de conversão são:

C= 5.(F-32)/9
F= (9.C/5) + 32

Solicite o tipo de temperatura a ser convertido (C ou F), o valor da temperatura e mostre a conversão.