



PROGRAMAÇÃO I

INTRODUÇÃO À ORIENTAÇÃO A OBJETOS III

Prof. Dr. Daniel Caetano

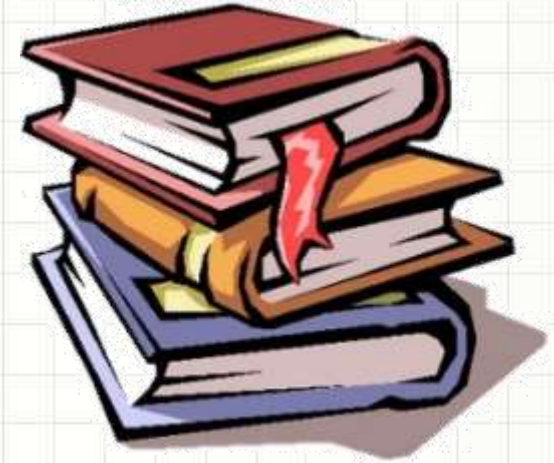
2017 - 1

Objetivos

- Conhecer o modificador *static* e seu uso em atributos e métodos
- Conhecer um pouco mais sobre a classe Math e String
- Conhecer melhor a classe JOptionPane
- Conhecer o conceito de construtor replicador: passando objetos como parâmetros



Material de Estudo



Material

Acesso ao Material

Apresentação

<http://www.caetano.eng.br/>
(Programação I – Aula 5)

Material Didático

Programação I – Espalhando nas Págs 38 a 65



RETOMANDO CLASSES EM JAVA

O que são Classes?

- Classes possuem
 - Atributos e Métodos
 - Podem ser públicos, protegidos ou privados
 - Métodos getters/setters
 - Construtores
- Base para a construção de objetos
 - Operador **new**

Sobrecarga

- Assinatura

Nome do Método + Tipos dos Parâmetros

- Possível: vários métodos com o mesmo nome
 - Desde que os tipos parâmetros sejam diferentes
 - Ou seja: nome igual, mas assinatura diferente



ATRIBUTOS ESTÁTICOS

Atributos Estáticos

- E se precisarmos que todos os objetos de uma classe compartilhem um valor?
- Definimos esse atributo como ***static***
 - Ele não é mais parte do objeto, é da classe
 - Não é preciso criar um objeto para acessá-lo

```
Classe.atributo = valor;
```

- Uso comum: contar objetos criados
 - Exemplo: número de produtos criados

Atributos Estáticos

- Contando Número de Productos

```
public Produto {  
    private static int total = 0;  
    public Produto() {  
        total = total + 1;  
    }  
    public int getTotal() {  
        return total;  
    }  
}
```

Atributos Estáticos

- Outro exemplo: Math.PI

```
public static final double PI = 3.141592;
```

- Calculando a área de um círculo?

```
double area = Math.PI * (raio*raio);
```



MÉTODOS ESTÁTICOS

Métodos Estáticos

- São parecidos com os atributos estáticos
 - Eles não são parte do objeto, mas da classe
 - Não é preciso criar um objeto para acessá-lo

`Classe.metodo();`

- Usos comuns:
 - Métodos auxiliares
 - Exemplo: consultar número de produtos criados
 - Organização de métodos básicos
 - Biblioteca de funções matemáticas

Métodos Estáticos

- Contando Número de Produtos

```
public Produto {  
    private static int total = 0;  
    public Produto() {  
        total = total + 1;  
    }  
    public static int getTotal() {  
        return total;  
    }  
}
```

Cuidado: em um método static, você só pode acessar atributos static!

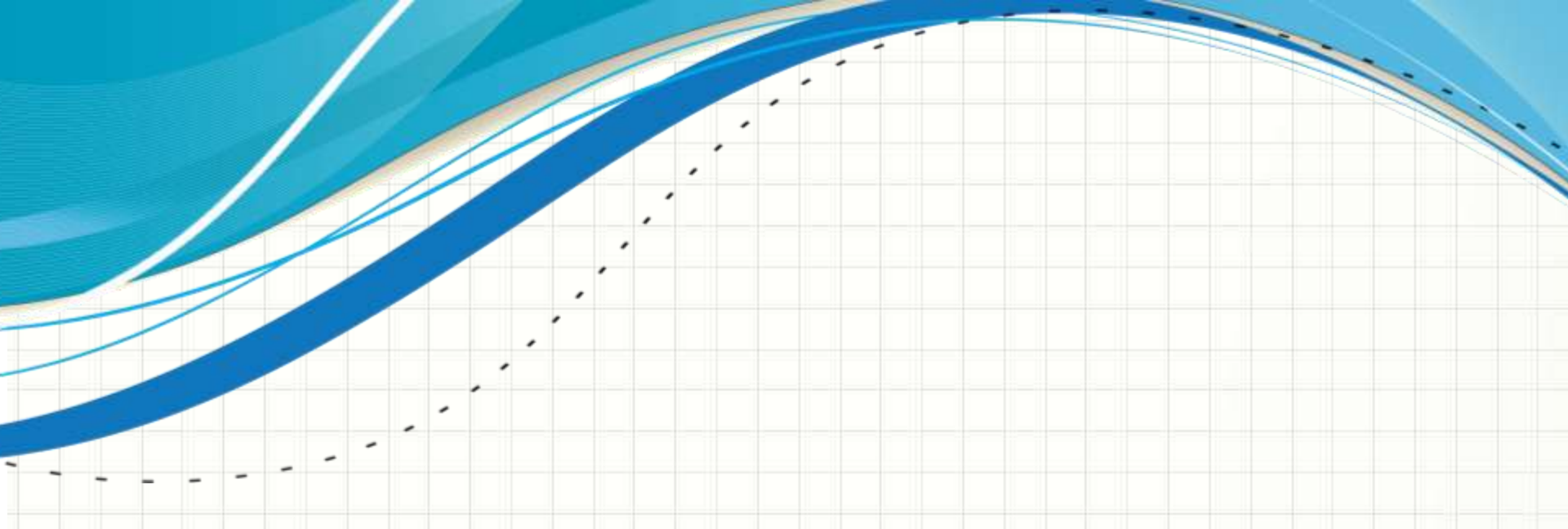
Métodos Estáticos

- Outro exemplo: `Math.pow(base,exp)`
- Calculando a área de um círculo?

```
double area = Math.PI * Math.pow(raio, 2);
```

Métodos Estáticos - Atenção

- Métodos de Instância
 - Só podem ser chamados a partir de uma instância de classe: **referência**
 - Podem acessar atributos da instância ou da classe (estáticos)
- Métodos estáticos
 - São chamados a partir da própria classe, não precisam de instância: **nome da classe**
 - Só podem acessar atributos da classe (estáticos)



ALGUNS MÉTODOS ÚTEIS: CLASSES MATH E STRING

Atributos e Métodos da Math

- `Math.PI`
- `Math.pow(base, exp)`
- `Math.sqrt(n)`
- `Math.sin(angulo)`
- `Math.cos(angulo)`
- `Math.tan(angulo)`
- `Math.abs(n);`
- `Math.min(n1,n2)`
- `Math.max(n1,n2)`
- `Math.ceil(n)`
- `Math.floor(n)`

Métodos da Classe String

- `String t1 = "aba";`
- `String t2 = "cate";`
- `t1.length()`
- `t1.charAt(pos)`
- `t1.replace(oque, comoque)`
- `t1.toLowerCase()`
- `t1.toUpperCase()`
- `t1.concat(t2)`
- `t1.compareTo(t2)`
- `t1.compareToIgnoreCase(t2)`
- `t1.equals(t2)`



A CLASSE JOPTIONPANE

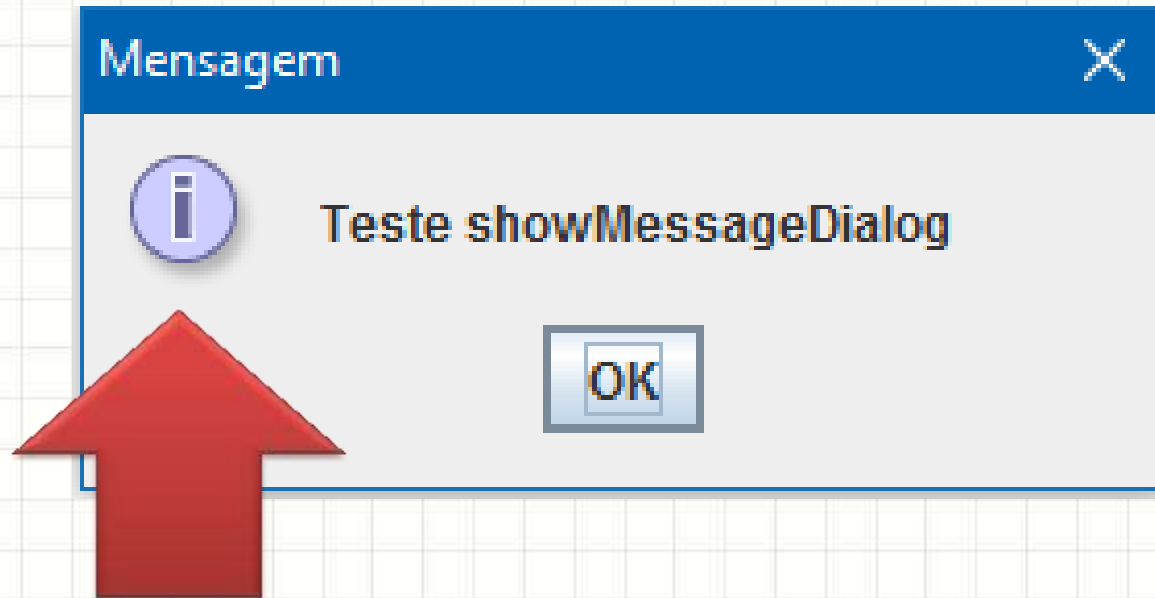
A Classe JOptionPane

- Já vimos
 - JOptionPane.showMessageDialog
 - JOptionPane.showInputDialog
- São métodos estáticos!
- Vamos conhecer um pouco mais...

JOptionPane.showMessageDialog

- Saída simples, imprime string:

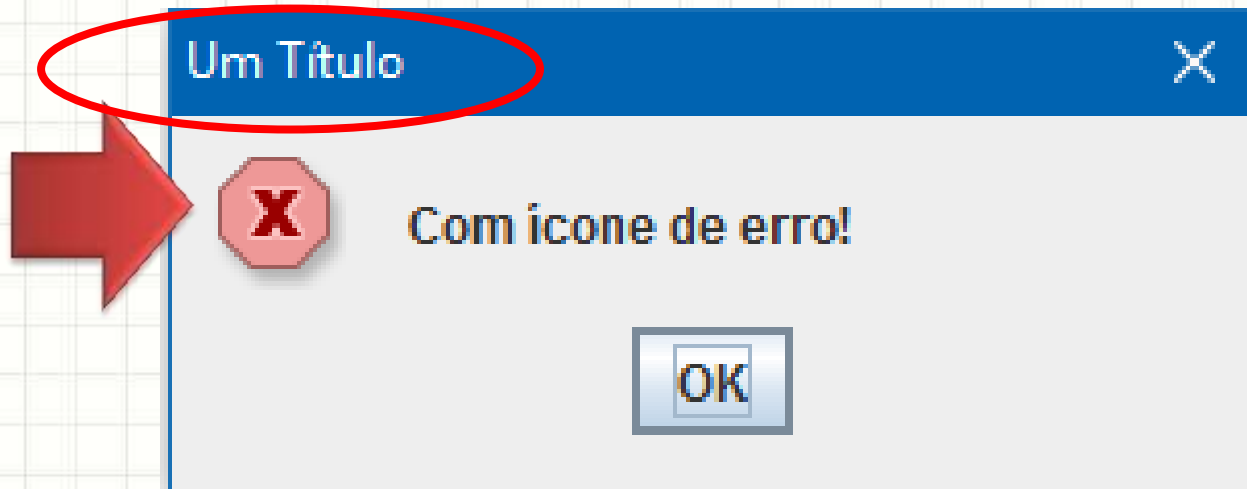
```
JOptionPane.showMessageDialog(null,  
    "Teste showMessageDialog");
```



JOptionPane.showMessageDialog

- Como mudar o título?
- Como mudar o ícone?

```
JOptionPane.showMessageDialog(null,  
    "Com ícone de erro!", "Um Título",  
    JOptionPane.ERROR_MESSAGE);
```



JOptionPane.showMessageDialog

- Formato Geral

```
JOptionPane.showMessageDialog(refJanelaPai,  
                             "Mensagem", "Título",  
                             tipoDeIcone);
```

- Quais os valores possíveis do tipo de ícone?

JOptionPane.showMessageDialog

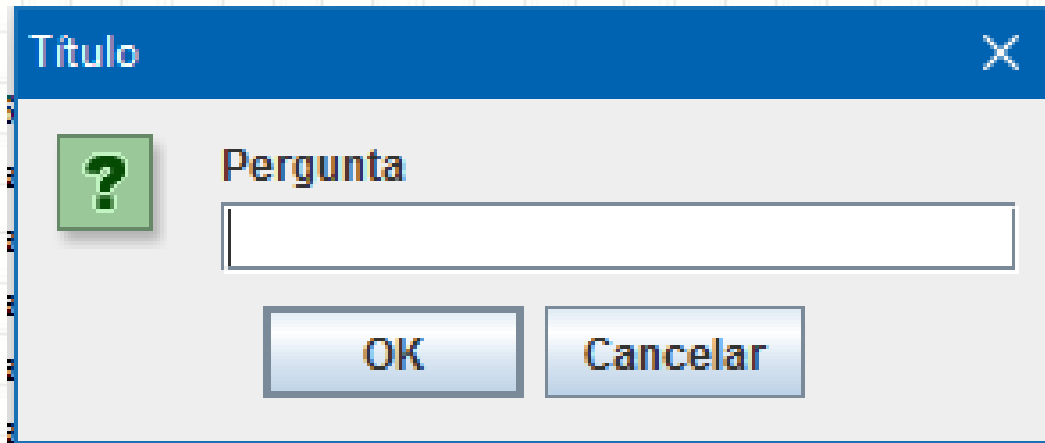
- Ícones disponíveis:
 - JOptionPane.ERROR_MESSAGE
 - JOptionPane.INFORMATION_MESSAGE
 - JOptionPane.WARNING_MESSAGE
 - JOptionPane.QUESTION_MESSAGE
 - JOptionPane.PLAIN_MESSAGE



JOptionPane.showInputDialog

- Várias versões
 - Formato geral I

```
String texto = JOptionPane.showInputDialog(  
    refJanelaPai, "Pergunta", "Título",  
    tipoDeIcone);
```

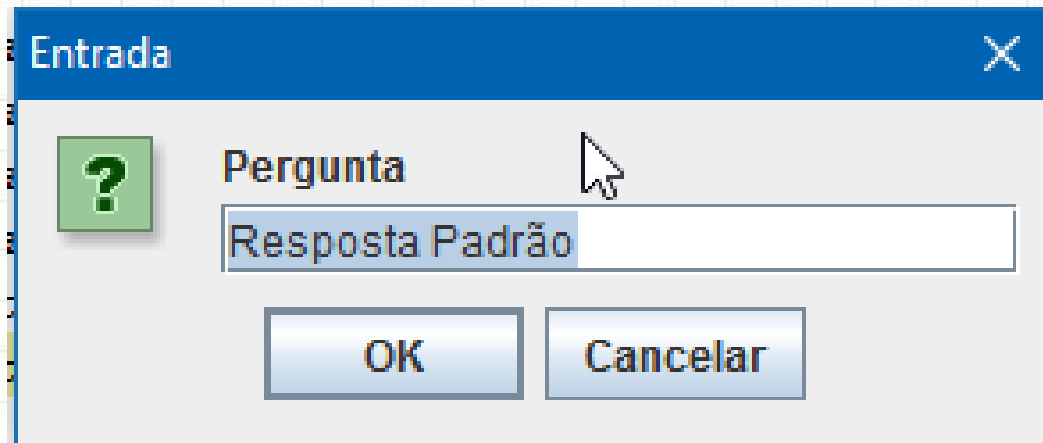


- Os ícones podem ser os mesmos do “show”

JOptionPane.showInputDialog

- Várias versões
 - Formato geral II

```
String texto = JOptionPane.showInputDialog(  
    refJanelaPai, "Pergunta",  
    "Resposta Padrão");
```

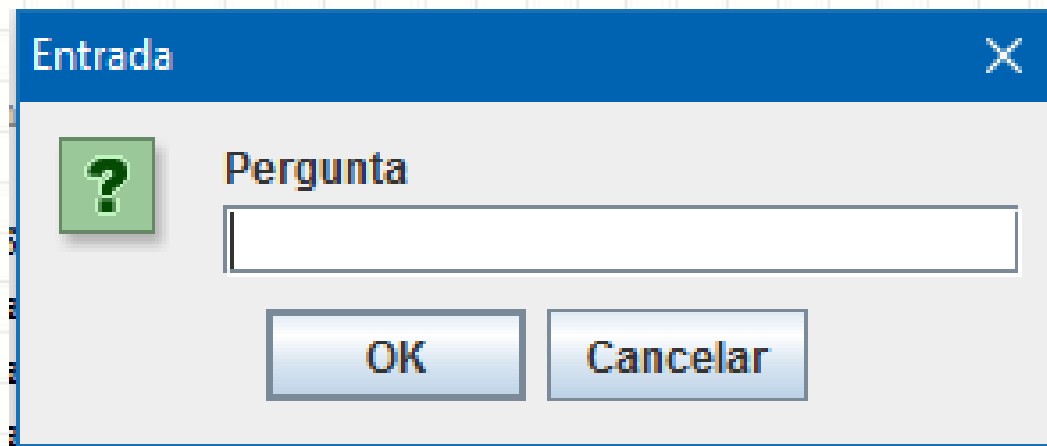


- O ícone sempre será "Question"

JOptionPane.showInputDialog

- Várias versões
 - Formato geral III

```
String texto = JOptionPane.showInputDialog(  
    "Pergunta");
```



- Ícone/título: sempre "Question"/ "Entrada"



CONSTRUTORES REPLICADORES

Construtores Replicadores

- Imagine a execução do código abaixo

```
int a, b;  
  
a = 10;  
b = a;  
a = 20;  
  
System.out.println(b);
```

- Que valor será impresso?

Construtores Replicadores

- Agora imagine a execução do código abaixo

```
Cliente c1, c2;  
  
c1 = new Cliente("012.345.678-90");  
c2 = c1;  
c1.setCpf("000.000.000-00");  
  
System.out.println(c2);
```

- Que valor será impresso?
- Por que isso ocorre?

Construtores Replicadores

- Objetos ocupam muita memória
 - Não são copiados automaticamente!
 - Criam-se referências múltiplas (nome x apelido)
- Como criar uma cópia?
 - Vários jeitos... Três mais comuns...

Construtores Replicadores

- Primeiro jeito simples:
 - Criando um objeto vazio...
 - E copiando atributo por atributo!

```
Cliente c1, c2;  
c1 = new Cliente("012.345.678-90", "Fulano");  
c2 = new Cliente();  
c2.setCpf(c1.getCpf());  
c2.setNome(c1.getNome());  
  
c1.setCpf("000.000.000-00");  
System.out.println(c2);
```


Construtores Replicadores

- Segundo jeito simples:
 - Criando um objeto com os dados de outro

```
Cliente c1, c2;  
c1 = new Cliente("012.345.678-90", "Fulano");  
c2 = new Cliente(c1.getCpf(), c1.getNome());  
  
c1.setCpf("000.000.000-00");  
System.out.println(c2);
```

- Mas se houver atributos "private" que não possuem getters/setters?

Construtores Replicadores

- Terceiro jeito, ainda mais simples:
 - Passando objeto “origem” na construção

```
Cliente c1, c2;  
c1 = new Cliente("012.345.678-90", "Fulano");  
c2 = new Cliente(c1);  
  
c1.setCpf("000.000.000-00");  
System.out.println(c2);
```

- Como se faz isso?
 - Criando um **construtor replicador** (ou cópia)!

Construtores Replicadores

- Exemplo:

```
public class Cliente {  
    private String cpf;  
    private String nome;  
  
    public Cliente(Cliente outro) {  
        cpf = outro.cpf;  
        nome = outro.nome;  
    }  
  
    [...]   
}
```

Pode?



PERGUNTAS?



PARTE PRÁTICA

Experimentando

- Vamos aprimorar a classe Produto para que ela conte o número de objetos já criados.
- Vamos aprimorar a classe e o sistema para que ele use o “contador” como “ID”
- Aprimorar a classe Produto com o construtor cópia

Experimentando

- Implemente o construtor replicador na classe Retangulo e na classe Circulo

Experimentando

- Vamos fazer um programa simples que calcule e imprima o volume de uma esfera usando JOptionPane para entrada e saída de dados.
 - Nota: $\text{volume} = (4/3) * \text{PI} * \text{raio}^3$



PROJETO / ATIVIDADE

Projeto / Atividade

- Objetivo: criar cadastro de clientes simples
- Quem: Duplas especificadas
- A primeira entrega será 17/04
 - Classe cliente (com validações)
 - Criação/Preenchimento dos dados do cliente por modo texto (Scanner)
- A segunda entrega será 05/06
 - Listagem de clientes
 - Busca de clientes
 - Edição de clientes

Projeto / Atividade

- Hoje: Continue o projeto CadCli
 - Modifique o método main da classe CadCli:
 - Imprima um menu:

```
Cadastro de Clientes
(C)adastrar Cliente
(L)istar Clientes
(B)uscar Cliente
(E)ditar Cliente
(S)air
Opção: _
```

- Leia a opção do usuário e programe a ação de todas as opções em separado, cada uma respondendo “Rotina não implementada”