

Disciplina: Java

Profª Kátia Livia Zambon

## **Lista 1 de exercícios – Terceiro Bimestre (2017)**

Conteúdo:

Classes, Objetos, Atributos, Métodos

Encapsulamento

Construtores

1. Crie um projeto com as especificações abaixo:

Projeto: Exerc1\_Aluno

Pacote: Exerc1\_Aluno

Classe **Aluno** com os seguintes atributos públicos:

nome - string

idade - int

altura - double

Classe **TestaAluno** (que contém o método main()) com as seguintes ações:

- Instancie um objeto de nome Aluno1 da classe Alunos
- Instancie um objeto de nome Aluno2 da classe Alunos
- Atribua valores para os atributos dos dois objetos instanciados (podem escolher)
- Mostre cada objeto e seus atributos

2. Crie um projeto com as especificações abaixo:

Projeto: Exerc2\_Carro

Pacote: Exerc2\_Carro

Classe **Carro** com os seguintes atributos do tipo private

placa - string

ano - int

modelo - string

Classe **TestaCarro** (que contém o método main()) com as seguintes ações:

- Instancie um objeto de nome c1 da classe Carro
- Instancie um objeto de nome c2 da classe Carro
- Criar os métodos getters() e setters() necessários
- Atribua valores para os atributos dos dois objetos instanciados (podem escolher)
- Mostre cada objeto e seus atributos

3. Crie um projeto com as especificações abaixo:

Projeto: Exerc3\_Calculadora

Pacote: Exerc2\_Calculadora

Classe **Calculadora** com os seguintes atributos do tipo private:

Numero1 - int

Numero2 – int

Faça um construtor que inicialize os atributos com valores maiores que zero.

Faça os métodos para a classe Calculadora:

- **retornaSomaAtributos** – este método não deve receber parâmetros, mas deve retornar a soma dos valores atuais dos seus atributos (numero1 e numero2).
- **retornaMultiplicacaoAtributos** – este método não deve receber parâmetros, mas deve retornar a multiplicação dos valores atuais dos seus atributos.
- **insereValoresAtributos** – este método deve receber como parâmetros, dois valores inteiros maiores que zero e que devem ser colocados em cada um dos atributos (obs.: este método não deve retornar valor algum).
- Métodos getters() e setters() necessários

Classe **TestaCalculadora** (que contém o método main()) com as seguintes ações:

- instancie três objetos da classe Calculadora
- coloque valores para os atributos
- mostra o valor retornado pelo Método "retornaSomaAtributos"
- mostra o valor retornado pelo Método "retornaMultiplicacaoAtributos"
- coloca outros valores nos atributos da Classe através do Método "insereValoresAtributos".
- mostra o valor dos Atributos

4. Crie um projeto com as especificações abaixo:

Projeto: Exerc4\_Cilindro

Pacote: Exerc2\_Cilindro

Classe **Cilindro** com os seguintes atributos:

raioBase - double

altura – double

Faça o encapsulamento desta classe.

Crie um construtor que receba alguma parâmetro e emita uma mensagem

Faça os métodos getters() e setters() necessários

Crie uma classe **TestaCilindro** para:

- Inclua valores
- Mostre os valores dos atributos

5. Crie um projeto com as especificações abaixo:

Projeto: Exerc5\_Funcionario

Pacote: Exerc5\_Funcionario

Classe **Funcionario** com os seguintes atributos:

número de Matrícula (pode ser do tipo Inteiro)  
nome  
idade  
cargo  
salário

Para esta classe gere três construtores:

- O 1º não deve receber parâmetros, mas deve iniciar o valor do salário com R\$ 1.500,00
- O 2º deve receber como parâmetro o nome e a idade do funcionário, e preencher estes dois campos com os valores recebidos.
- O 3º deve receber o número de matrícula, o nome, e o salário, e preencher estes três campos com os valores recebidos.

Um método para aumentar o salário de todos os funcionários que recebe a porcentagem de aumento do tipo int e não retorna valor.

Um método para aumentar o salário de pessoas com mais de 50 anos e que recebem menos que R\$ 3.000,00

Crie uma classe chamada **TestaFuncionario** para instanciar objetos e utilizar os métodos criados.

6. Escreva exatamente a saída da execução do método main da classe UsaRadio:

<pre>class UsaRadio {     public static void main(String[] args) {         Radio r1 = new Radio(3);         Radio r2 = new Radio(10);         System.out.println(r1.exibeVolume());         System.out.println(r2.exibeVolume());     } }</pre>	<pre>class Radio {     int volume;      Radio(int volume) {         this.volume = volume;     }      String exibeVolume() {         return "Vol: " + this.volume;     } }</pre>
---	---

7. O que é encapsulamento? Qual a sua principal finalidade?

8. Por que é importante **não** criarmos métodos getters() e setters() para todos os atributos?

9. Na declaração abaixo, identifique a variável, o objeto e a classe. Responda: "é correto afirmar que minhaConta é um objeto?"

Conta minhaConta = new Conta( );

**Exemplo:** A atividade 2 da aula 15 (parte 4 de OO) propõe uma classe chamada Candidatos com um contador de votos. Abaixo o código desta classe, o código do programa principal e a interface para a votação.

Classe Candidatos:

```

1 package Ativ2;
2 public class Candidatos {
3     private String nome;
4     private int votos;
5
6     public Candidatos(String nome) {
7         this.nome = nome;
8         this.votos = 0;
9     }
10
11    public String getNome() {
12        return nome;
13    }
14
15    public void setNome(String nome) {
16        this.nome = nome;
17    }
18
19    public int getVotos() {
20        return votos;
21    }
22
23    public void setVotos() {
24        this.votos ++;
25    }
26 }

```

Classe principal: JFrame

```

1 package Ativ2;
2 public class Frame_principal extends javax.swing.JFrame {
3     Candidatos c1=new Candidatos("Professor de Java");
4     Candidatos c2=new Candidatos("Professor de Linguagem C");
5
6     public Frame_principal() {
7         initComponents();
8         jComboBox1.addItem("Selecione um candidato");
9         jComboBox1.addItem(c1.getNome());
10        jComboBox1.addItem(c2.getNome());
11        jLabel_c1Nome.setText(c1.getNome());
12        jLabel_c2Nome.setText(c2.getNome());
13    }
14
15    @SuppressWarnings("unchecked")
16    Generated Code
17
18    private void jButton_VotarActionPerformed(java.awt.event.ActionEvent evt) {
19        if (jComboBox1.getSelectedItem().equals(c1.getNome()))
20            c1.setVotos();
21        else if (jComboBox1.getSelectedItem().equals(c2.getNome()))
22            c2.setVotos();
23    }
24
25    private void jButton_ResultadoActionPerformed(java.awt.event.ActionEvent evt) {
26        jLabel_c1Votos.setText(String.valueOf(c1.getVotos()));
27        jLabel_c2Votos.setText(String.valueOf(c2.getVotos()));
28    }
29
30    public static void main(String args[]) {
31        /* Set the Nimbus look and feel */
32        Look and feel setting code (optional)
33
34        /* Create and display the form */
35        java.awt.EventQueue.invokeLater(new Runnable() {
36            @Override
37            public void run() {
38                new Frame_principal().setVisible(true);
39            }
40        });
41    }
42 }

```

Instância dois  
objetos da classe  
Candidatos

Através do método  
getNome() insere os itens  
nos objetos do projeto  
gráfico

Faz a verificação e  
computa os votos através  
do método setVotos() da  
classe Candidatos

Através do método  
getVotos() da classe  
Candidatos apresenta a  
contagem dos votos

Execução:

Eleição do CTI - Atividade 2 - Aula 15

Candidatos CTI: Seleccione um candi... Votar

Seleccione um candidato

Professor de Java

Professor de Linguagem

Resultado Eleição

Professor de Java

Professor de Linguagem C

Eleição do CTI - Atividade 2 - Aula 15

Candidatos CTI: Professor de Lingu... Votar

Resultado Eleição

Professor de Java 5

Professor de Linguagem C 2