



Disciplina: Java Profa Kátia Lívia 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



UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO" Campus de Bauru



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



UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO" Campus de Bauru



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.

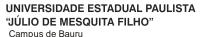
6. Escreva exatamente a saída da execução do método main da classe UsaRadio:

```
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());
}

String exibeVolume() {
    return "Vol: " + this.volume;
}
```

- 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();





口

+

114 🚍

119 -

java.awt.EventQueue.invokeLater(new Runnable() {

new Frame principal().setVisible(true);

@Override

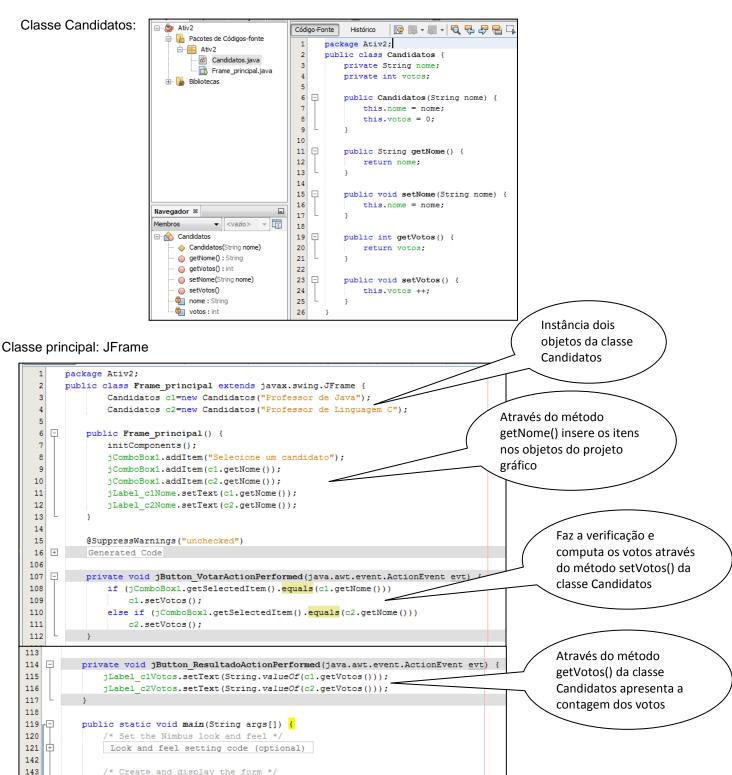
});

public void run() {



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:





UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO" Campus de Bauru



Execução:



