

Hortolândia, 02 de abril de 2014.

Lista de Exercícios 1 - Programação Orientada a Objetos

Introdução

A presente lista de exercícios contempla os conceitos de programação orientada a objetos e da linguagem Java, vistos até o momento na disciplina de LP2 I2.

Os exercícios desta lista serão corrigidos na aula do dia 08/04, como parte da revisão para a Prova 1 (do dia 15/04).

A fim de ilustrar a entrada de dados via a classe **JOptionPane**, reapresentamos o exemplo de cálculo da idade (também presente no material das aulas 19, 20, 21 e 22).



Exemplo 1

```
import javax.swing.JOptionPane;

public class CalculaIdade {
    public static void main(String args[ ]){
        int idade;
        String idadeS;
        idadeS = JOptionPane.showInputDialog("Entre com sua idade:");
    };

    idade = Integer.parseInt(idadeS);
    idade = idade + 10;

    System.out.println("Daqui a dez anos você terá: " + idade +
" anos de idade.");

    JOptionPane.showMessageDialog(null, "Daqui a dez anos você
terá: " + idade + " anos de idade.");
    }
}
```

Listagem 1: código fonte da classe CalculaIdade.

O exemplo acima (classe **CalculaIdade**) ilustra o uso da classe **JOptionPane**, pertencente ao pacote gráfico **javax.swing** (também conhecido como Swing).

A classe **JOptionPane** apresenta uma janela, que pode conter uma figura, rótulo, e/ou caixa de texto, conforme o método chamado. No caso do método **showInputDialog()**, uma janela contendo uma caixa de texto será apresentada, o que permite a coleta de informações por parte do usuário.

O texto que é lido por meio deste método, é do tipo **String**. Por essa razão, após a leitura do valor "idadeS", o mesmo é convertido para o formato inteiro, a fim de realizar a operação de adição do exemplo de cálculo de idade apresentado.

Por fim, após calcular a nova idade, faz-se uso novamente da classe **JOptionPane**, agora chamando-se o método **showMessageDialog()**, que apresenta uma janela contendo uma mensagem ao usuário.

Questões

1) Um programa deverá possuir um **contador**, que será responsável pelo armazenamento da quantidade de pessoas que se inscrevem em um concurso.

Indique:

- o tipo de dados mais apropriado para o contador.
- A instrução de declaração do contador.
- A instrução de atribuição do valor “25” ao contador.

2) Um programa deverá possuir uma variável que armazene o salário de um funcionário. Esta variável salário deverá ser inicializada com o valor 3500.00. Ao término do programa, o mesmo deverá exibir no *prompt* de comandos o valor do salário do funcionário.

3) Crie um programa que armazene em uma variável um valor booleano, o qual corresponderá ao estado atual de um dispositivo elétrico externo ao computador (exemplo: um motor). Considere que o motor deverá iniciar no estado desligado (*false*). Ao término do programa, o mesmo deverá exibir no *prompt* de comandos o valor da variável booleana.

4) Crie um programa que possua uma variável do tipo **int**, e outra variável do tipo **float**. Inicialize a variável do tipo **float** com o valor **15.0**. Armazene a seguir o valor da variável do tipo **float** na variável do tipo **int**. Ao término do programa, apresente no *prompt* de comandos o valor da variável do tipo **int**.

5) Crie um programa que possua uma variável do tipo **int**, e outra variável do tipo **float**. Inicialize a variável do tipo **int** com o valor **15**. Armazene a seguir o valor da variável do tipo **int** na variável do tipo **float**. Ao término do programa, apresente no *prompt* de comandos o valor da variável do tipo **float**.

6) Apresente um programa que converta um valor do tipo **double** em um valor do tipo **int**. Inicialize a variável do tipo **double** com o valor 17.8. Armazene a seguir o valor da variável **double** na variável do tipo **int**. Ao término do programa, apresente no *prompt* de comandos o valor da variável do tipo **int**, e interprete o resultado.

7) Crie um programa que declare uma variável do tipo vetor com três posições do tipo **int**. Em seguida, inicialize cada uma das posições do vetor com os valores 5, 7 e 11. Imprima as posições do vetor.

8) Crie um programa que declare uma variável do tipo vetor com três posições do tipo **String**. Em seguida, inicialize cada uma das posições do vetor com os valores “Antônio Carlos”, “Maria Constantina” e “José Francisco”. Imprima as posições do vetor.

9) Criar um novo projeto, denominado **Concessionaria**.

A classe principal também será denominada **Concessionaria**.

Dentro do pacote “concessionaria”, criar uma nova classe denominada **Veiculo**.

A classe **Veiculo** deverá possuir os seguintes atributos: **ano**, **marca**, **modelo**, **preço**. Definir os tipos de dados de cada atributo conforme o que for mais indicado.



Criar, para a classe **Veiculo**, os métodos de acesso aos atributos da classe (*getters* e *setters*).

Em seguida, na classe **Concessionaria**, criar um novo objeto do tipo Veiculo.

Os valores iniciais para os atributos devem ser obtidos por meio da utilização da classe **JOptionPane**, interagindo com o usuário (conforme mostrado no Exemplo 1, da classe Calculadade).

Ao final, exibir na linha de comandos os valores dos atributos da classe Veiculo.

10) Modificar a classe **Concessionaria**, do Exercício 9, de forma que sejam criados três Veiculos.

A classe **Concessionaria** deve possuir, em seu método principal, um vetor de objetos do tipo **Veiculo**.

Obs.: O assunto de Vetores e Matrizes foi visto nas aulas 13 e 14.

Exemplo de instrução de declaração do vetor:

```
Veiculo frota[ ] = new Veiculo[3];
```

Após a declaração do vetor de veículos, criar cada novo objeto **Veiculo** dentro de um laço for, dentro do qual também estarão as instruções de interação com o usuário para coleta dos valores dos atributos de cada Veiculo.

Ao final, exibir no prompt de comandos a descrição (os valores dos atributos) de cada Veiculo cadastrado.