

Hortolândia, 28 de agosto de 2013.

**Aulas 19 e 20**

**Lista de Exercícios I – Questões e Respostas**

**I. Tipos Primitivos e classe String**

**Obs.:** Acerca do uso de tipos primitivos em Java, o material das aulas 7 e 8 já contém 10 questões. Os exercícios aqui descritos são um complemento àqueles anteriormente apresentados.

**1) Elabore um programa que armazene em uma variável do tipo booleana, o estado atual de uma lâmpada. Inicialize a variável com o valor “false”. Em seguida, altere o valor da variável para “true”, e exiba no prompt de comandos o novo valor da variável.**

```
package aulas15e16;

public class Questao1 {

    public static void main(String[] args){

        boolean estadoAtualLampada = false;

        estadoAtualLampada = true;

        System.out.println("Estado atual da Lâmpada: " +
estadoAtualLampada);

    }

}
```

**2) Elabore um programa que armazene em uma variável o nome de um cliente. Atribua à variável um valor pré-definido de nome de cliente, por exemplo, “Aristeu de Souza”. Exiba a seguir no prompt de comandos o valor desta variável.**

```
package aulas15e16;

public class Questao2 {

    public static void main(String[] args){

        String nomeCliente;

        nomeCliente = "Aristeu de Souza";

        System.out.println("Nome do Cliente: " + nomeCliente);

    }

}
```

**3) Reformule o exercício da Questão 2, agora utilizando a classe JOptionPane. Esta classe possibilita a interação com o usuário, possuindo métodos de entrada e saída de dados, no formato de Strings, conforme apresentado no quadro a seguir.**

```
// Exemplos de uso da classe JOptionPane.

// Uso dos métodos showInputDialog() para entrada de dados
// e showMessageDialog() para a saída de dados.

String resposta = JOptionPane.showInputDialog("Questão para o
usuário");
```

```
JOptionPane.showMessageDialog("Mensagem para o usuário");
```

```
package aulas15e16;

import javax.swing.JOptionPane;

public class Questao3 {

    public static void main(String[] args){

        String nomeCliente;

        nomeCliente = JOptionPane.showInputDialog("Qual é o seu nome?");

        JOptionPane.showMessageDialog(null, "Nome do Cliente: " + nomeCliente);

    }

}
```

## II. Vetores e Matrizes

**Obs. 2:** Nesta Seção, apresentaremos exercícios utilizando vetores, porém ainda sem a utilização de **estruturas de controle** como os laços **for** e **while**, que são os temas da próxima Seção.

**4) Declare um array (vetor) unidimensional chamado numeros, contendo cinco elementos inteiros. Inicialize o vetor com cinco valores, atribuídos na forma {valor1, valor2 etc.}. Em seguida, exiba no prompt de comandos os valores dos itens do vetor.**

```
package aulas15e16;

public class Questao4 {

    public static void main(String[] args){

        int numeros[] = {1,2,3,4,5};

        System.out.println("Valor 1: " + numeros[0]);

        System.out.println("Valor 2: " + numeros[1]);

        System.out.println("Valor 3: " + numeros[2]);

        System.out.println("Valor 4: " + numeros[3]);

        System.out.println("Valor 5: " + numeros[4]);

    }

}
```

**5) Crie um vetor unidimensional chamado clientes, contendo dois elementos do tipo String. Leia o valor do nome de dois clientes, utilizando o método showInputDialog() da classe JOptionPane, conforme mostrado na Questão 3. Grave o valor dos nomes dos clientes no vetor. Em seguida, exiba no prompt de comandos os nomes dos clientes.**

```
package aulas15e16;

import javax.swing.JOptionPane;
```

```
public class Questao5 {  
    public static void main(String[] args){  
        String clientes[] = new String[2];  
        clientes[0] = JOptionPane.showInputDialog("Primeiro cliente: ");  
        clientes[1] = JOptionPane.showInputDialog("Segundo cliente: ");  
        System.out.println("Nomes dos Clientes");  
        System.out.println("Cliente 1: " + clientes[0]);  
        System.out.println("Cliente 2: " + clientes[1]);  
    }  
}
```

**6) Refaça o exercício da Questão 5, porém exibindo os nomes dos clientes no interior de uma caixa de mensagem da classe JOptionPane. A mensagem de saída para o usuário será no formato de uma String, e deve ser passada como parâmetro para o método showMessageDialog() da classe JOptionPane. Um exemplo de mensagem de saída está na caixa em destaque da Questão 3.**

```
package aulas15e16;  
import javax.swing.JOptionPane;  
public class Questao6 {  
    public static void main(String[] args){  
        String clientes[] = new String[2];  
        clientes[0] = JOptionPane.showInputDialog("Primeiro cliente: ");  
        clientes[1] = JOptionPane.showInputDialog("Segundo cliente: ");  
        JOptionPane.showMessageDialog(null, "Nome dos Clientes \nCliente  
1: " + clientes[0] + "\nCliente 2: " + clientes[1]);  
    }  
}
```

### III. Estruturas de Controle

**7) Elabore um programa contendo uma variável booleana de nome cadastroAtivo. Inicialize esta variável com o valor “false”. Em seguida, consulte o valor da variável cadastroAtivo, utilizando a estrutura condicional “if-then-else”, exibindo no prompt de comandos uma de duas sentenças:**

**a) Caso o valor da variável cadastroAtivo seja verdadeiro (true), escreva a mensagem “O cadastro do cliente está ativo”.**

**b) Caso o valor da variável cadastroAtivo seja falso (false), escreva a mensagem “O cadastro do cliente está inativo”.**

```
package aulas15e16;  
public class Questao7 {  
    public static void main(String[] args){  
        boolean cadastroAtivo = false;  
        if(cadastroAtivo){  
            System.out.println("O cadastro do cliente está ativo.");  
        }  
    }  
}
```

```
    }  
    else{  
        System.out.println("O cadastro do cliente está inativo.");  
    }  
}  
}
```

**8) Elabore um programa que leia o valor da idade do usuário, por meio da classe JOptionPane. Lembrar de converter o valor da entrada de String para int. Em seguida, verifique se a idade é maior ou igual a 18 anos. Exiba no prompt de comandos a mensagem “Usuário maior de idade”, caso o mesmo tenha mais de 18 anos, e “Usuário menor de idade”, em caso contrário.**

```
package aulas15e16;  
import javax.swing.JOptionPane;  
public class Questao8 {  
    public static void main(String[] args){  
        int idade;  
        String idadeString;  
        idadeString = JOptionPane.showInputDialog("Qual é o valor  
de sua idade?");  
        idade = Integer.parseInt(idadeString);  
        if(idade >= 18){  
            System.out.println("Usuário maior de idade.");  
        }  
        else{  
            System.out.println("Usuário menor de idade.");  
        }  
    }  
}
```

**9) Elabore um programa que declare um vetor de inteiros, de nome “valores”, com cinco posições. Inicialize este vetor com cinco valores inteiros aleatórios. Exiba os conteúdos do vetor, empregando a estrutura de repetição “for”, no prompt de comandos.**

```
package aulas15e16;  
public class Questao9 {  
    public static void main(String[] args){  
        int valores[] = {1, 2, 3, 4, 5};  
        System.out.println("Valores do vetor: ");  
        for(int i=0; i < valores.length; i++){  
            System.out.println("Valor["+i+"]: "+ valores[i]);  
        }  
    }  
}
```

**10) Elabore um programa que leia um valor digitado pelo usuário, compreendido entre 0 e 500, por meio do método `showInputDialog()` da classe `JOptionPane`. Converta esse valor de entrada de `String` para `int`. Crie a seguir um laço que repita, constantemente, uma comparação entre o valor lido e o valor 250. Enquanto o valor lido for inferior a 250, permaneça no laço.**

```
package aulas15e16;

import javax.swing.JOptionPane;

public class Questao10 {

    public static void main(String[] args){

        int valor;

        String valorString;

        valorString = JOptionPane.showInputDialog("Digite um valor entre
0 e 500: ");

        valor = Integer.parseInt(valorString);

        while(valor < 250){

            valorString = JOptionPane.showInputDialog("Digite um valor entre
0 e 500: ");

            valor = Integer.parseInt(valorString);

            System.out.println("Valor digitado: " + valor);

        }

        System.exit(0);

    }

}
```