



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação III

AP3 2º semestre de 2007.

Nome –

Assinatura –

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

Questão 1) (2.5 pontos)

Indique quais erros que o código de JAVA abaixo apresenta? Suas respostas só serão aceitas se você explicar o porquê.

```
interface X{}
class UC{}
class USC extends UC implements X{}
class OSC extends UC{}
class USSC extends USC{}
class Teste{
    public static void main(String[] args){
        UC uc1 = new OSC();
        UC uc2 = new USSC();
        UC uc3 = new USC();
        OSC osc1 = uc1;
        OSC osc2 = new UC();
        OSC osc3 = (OSC) uc3;
        X x1 = new USC();
        X x2 = new X();
        USSC ussc1;
        if(uc2 instanceof USSC) ussc1 = uc2;
    }
}
```

Resposta:

```
(1) OSC osc1 = uc1;           // falta cast
(2) OSC osc2 = new UC();      // inicialização só OSC ou filhos dela
(3) OSC osc3 = (OSC) uc3;     // cast errado
(4) X x2 = new X();           // não se instancia interface
(5) ussc1 = uc2;              // falta cast
```

Questão 2) (2.5 pontos)

Escreva um programa que receba como parâmetro de entrada, o nome de um arquivo texto, cujo conteúdo são números inteiros, um em cada linha, e que inverta a ordem destes mesmos números num arquivo de saída, cujo nome será a palavra “saída-” concatenada com o nome do arquivo de entrada.

Um exemplo de uso desse programa seria “java Inverte dados.txt” (neste caso, o nome do arquivo de saída será saída-dados.txt).

Resposta:

```
import java.io.*;
public class Inverte {
    public static void main(String[] args) throws IOException {
        BufferedReader in = new BufferedReader(new
        FileReader(args[0]));
        int n = 0;
        String s;

        try{
            while ((s = in.readLine()) != null) n++;
        }
        catch (Exception e) {
            System.out.println("Excecao1\n");
        }

        try {
            int vet[] = new int[n];
            int cont = n - 1, aux;

            in.close();
            in = new BufferedReader(new FileReader(args[0]));

            while((s = in.readLine()) != null) {
                aux = Integer.parseInt(s);
                vet[cont--] = aux;
            }
            BufferedWriter out = new BufferedWriter(new
            FileWriter("saida-"+args[0]));
            for (int i = 0; i < n; i++) out.write(vet[i]+"\\n");
            out.close();
        }
    }
}
```

```

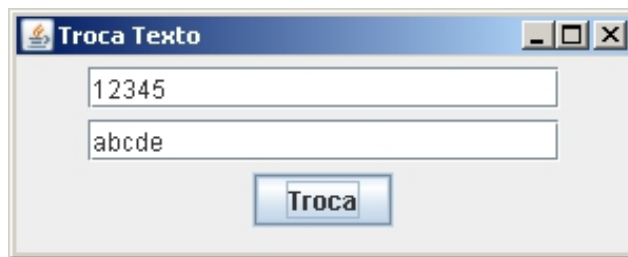
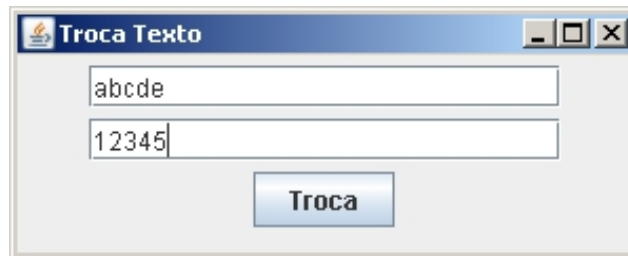
    catch (Exception e){
        System.out.println("Excecao2\n");
    }
    finally{
        in.close();
    }
}
}

```

Questão 3) (2.5 pontos)

Escreva um programa para criar uma janela contendo 2 campos de texto e um botão, como mostrado nas figuras abaixo, e que tenha o seguinte comportamento:

- O evento de clicar no botão deve trocar os conteúdos dos campos de texto, desde que ambos estejam preenchidos. As telas abaixo indicam um possível estado para a janela antes e depois de pressionarmos o botão “Troca”.



Resposta:

Nesta resposta, optamos por tratar o evento de click no botão na própria classe. Poderíamos também ter criado uma classe interna ou uma externa para fazer este mesmo tratamento.

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

```

```

public class AP3_2007_2_Q3 {
    public static void main(String[] args) {
        new JText();
    }
}

class JText implements ActionListener {
    JFrame frame = new JFrame("Troca Texto");
    JTextField tf = new JTextField(20);
    JTextField tf2 = new JTextField(20);
    JButton bt = new JButton("Troca");

    public JText() {
        tf.setEditable(true);
        tf.addActionListener(this);
        bt.addActionListener(this);
        Container c = frame.getContentPane();
        c.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));
        c.add(tf); c.add(tf2); c.add(bt);
        frame.setSize(300, 120);
        frame.setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        Object o = e.getSource();
        if (o == bt) {
            if (tf.getText().length() > 0 &&
                tf2.getText().length() > 0)
            {
                String temp = tf.getText();
                tf.setText(tf2.getText());
                tf2.setText(temp);
            }
        }
        else if (o == tf)
            JOptionPane.showMessageDialog(null, "O campo texto
contém: " + tf.getText());
    }
}

```

Questão 4) (2.5 pontos)

Suponha que num programa a representação de números é feita em binário. Os bits, com representação sem sinal, são armazenados numa string.

Ex.: 12 é armazenado como “1100”.

Declare uma classe a ser utilizada por este programa. Esta classe deve ter construtores que permitem a criação de um número à partir de um inteiro, ou diretamente à partir de uma string que contenha uma sequência de bits. Além disso, deve ter métodos para retornar a sequência de bits armazenada (uma string), retornar o valor inteiro decimal correspondente (percorrer a string e calcular o valor inteiro) e somar 2 números em binário (percorrendo as 2 strings). Observe que, para o usuário da classe, deve ser transparente a forma como a classe trata os valores internamente.

Resposta:

Nesta questão, objetivo era criar uma classe que modelava números inteiros. Vários construtores foram listados. Destes, o que recebe um número como parâmetro *public Numero(int valor)*; poderia ter a implementação como uma subdivisão sucessiva do parâmetro por 2 para obtenção da representação binária. Ao invés, utilizei o método de classe *Integer.toBinaryString()*; O método de adição poderia ser feito convertendo o parâmetro para o inteiro correspondente e realizando a soma. Entretanto, pensando-se em termos da estrutura do código, a implementação que trata os números internamente como seqüências de bits estaria dependente das operações com inteiros.

```
public class AP3_2007_2_Q4 {
    public static void main(String[] args) {
        Numero n = new Numero(5);
        Numero n2 = new Numero(3);
        System.out.println(n.toString());
        System.out.println(n2.toString());
        System.out.println(n.add(n2));
    }
}

class Numero {
    String sbits;

    public Numero() {
        sbits = "";
    }

    public Numero(int valor) {
        sbits = Integer.toBinaryString(valor);
    }

    public Numero(int valor, String bit) {
        sbits = bit;
        for (int i = 1; i < valor; i++)
            sbits = sbits + bit;
    }

    public Numero(String valor) {
        sbits = valor;
    }

    public String toString() {
        return sbits;
    }

    public String add(Numero n) {
        String res = "";
        int vail = 0;

        if (n.sbits.length() > this.sbits.length()) {
            int d;
            d = n.sbits.length() - this.sbits.length();
            this.sbits = (new Numero(d, "0")).toString() +
this.sbits;
        }
    }
}
```

```

else
    if (n.sbits.length() < this.sbits.length()) {
        int d;
        d = this.sbits.length() - n.sbits.length();
        n.sbits = (new Numero(d, "0")).toString() +
n.sbits;
    }

    for (int x = this.sbits.length()-1; x >= 0; x--) {
        if (sbits.charAt(x) == '0' && n.sbits.charAt(x) ==
'0') {
            if (vail == 0) {
                res = "0" + res;
                vail = 0;
            }
            else
            {
                res = "1" + res;
                vail = 0;
            }
        }
        else
            if (sbits.charAt(x) == '1' && n.sbits.charAt(x)
== '1') {
                if (vail == 0) {
                    res = "0" + res;
                    vail = 1;
                }
                else
                {
                    res = "1" + res;
                    vail = 1;
                }
            }
            else {
                if (vail == 0) {
                    res = "1" + res;
                    vail = 0;
                }
                else
                {
                    res = "0" + res;
                    vail = 1;
                }
            }
        }

        if (vail == 1)
            res = "1" + res;

        return res;
    }
}

```