



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação III

AP3 1º semestre de 2008.

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)

Descreva o que ocorre em cada trecho comentado que culmina com impressões no seguinte programa em Java, justificando suas afirmações.

```
public class Teste{
    public static void main (String[] args) {
        int a, b, c;
        b = c = 10;
        a = b++ + b++;           //(1)
        System.out.println(a);
        System.out.println(b);
        a= ++c + ++c;           //(2)
        System.out.println(a);
        System.out.println(c);

        b = 10;
        a = b++ + b;             //(3)
        System.out.println(a);
        System.out.println(b);

        a = 10;
        b = 5;
        if (a>b || ++b>5)        //(4)
            System.out.println(b);
    }
}
```

```

    a = 1;
    b = 5;
    if (a>b || ++b>5)          //(5)
        System.out.println(b);
    }
}

```

RESPOSTA:

(1) b assume o valor 12 e a assume o valor 21. Embora C não especifique a ordem de avaliação dos operandos de uma expressão, os operandos da operação + são os mesmos. Portanto, uma das expressões sempre retornará o valor 10 para b e a outra sempre retornará o valor 11.

(2) c assume o valor 12 e a assume o valor 23. Tal como na situação anterior, os operandos da operação + são os mesmos. Portanto, uma das expressões sempre retornará o valor 11 para c e a outra sempre retornará o valor 12.

(3) b assume o valor 11 e a pode assumir o valor 20 ou 21, dependendo da implementação do compilador. Como C não especifica a ordem de avaliação dos operandos de uma expressão, b pode ser avaliado antes de b++ ou vice-versa. No primeiro caso, b e b++ retornarão o valor 10, atribuindo 20 a a. No segundo caso, b++ retornará o valor 10 e b retornará 11, atribuindo 21 a a.

(4) Como a>b é verdadeiro, a outra expressão do ou lógico não será avaliada por causa da avaliação em curto-circuito. Assim, b não é incrementado, resultando em b = 5.

(5) Como a>b é falso, a outra expressão do ou lógico terá de ser avaliada. Assim, b é incrementado, resultando em b = 6.

Questão 2) (2.5 pontos)

Considere que você tenha um arquivo com valores inteiros, um em cada linha. Escreva um algoritmo em Java que calcule a média destes valores e que coloque num arquivo de saída, cujo nome é saída-<nome do arquivo de entrada>, todos os valores inteiros do arquivo de entrada que são maiores que a média, guardando a ordem do arquivo original. Por exemplo, se você tem o seguinte arquivo de entrada xxx.txt:

```

1
2
3
4

```

A média destes números é 2,5 e o conteúdo do arquivo de saída (saída-xxx.txt) será:

```

3
4

```

RESPOSTA:

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

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

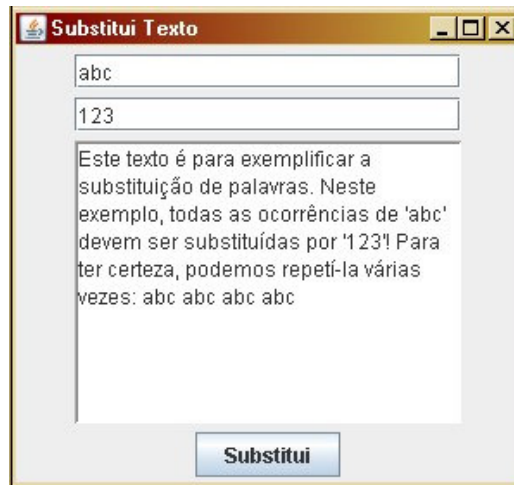
            while((s = in.readLine()) != null) {
                aux = Integer.parseInt(s);
                vet[cont++] = aux;
                soma += aux;
            }
            System.out.println("Media: " + (1.0 * soma)/n);

            BufferedWriter out = new BufferedWriter(new
FileWriter("saida-"+args[0]));
            for (int i = 0; i < n; i++)
                if (vet[i] > (soma/n)) 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 que crie a janela abaixo:



Ao clicar no botão Substitui, **todas** as ocorrências da primeira palavra (no exemplo da figura, “abc”) devem ser substituídas pela segunda palavra (no exemplo, “123”) na caixa de texto.

Dica: A caixa de texto pode ser uma instância da classe JTextArea da interface Swing.

Resposta:

```
import java.awt.*;
import java.awt.event.*;

import javax.swing.*;
import javax.swing.text.BadLocationException;

/*
 * Classe principal que inicia a janela da aplicação
 */
public class AP3_2008_1_Q3 {
    public static void main(String[] args) {
        new JTexto2();
    }
}

/*
 * Classe que modela a janela principal e seus componentes
 * Esta classe implementa a interface ActionListener, a qual
 * permite que esta classe trate as ações disparadas pelo
 * usuário na janela criada.
 */
class JTexto2 implements ActionListener {
    JFrame frame = new JFrame("Substitui Texto");
    JTextField tf = new JTextField(20);
    JTextField tf2 = new JTextField(20);
    JTextArea caixaTexto = new JTextArea(10, 20);
    JButton bt = new JButton("Substitui");
```

```

/*
    Construtor da classe da janela principal, a qual cria
    os componentes visuais e os inicializa.
*/
public JTexto2() {
    tf.setEditable(true);
    tf.addActionListener(this);
    bt.addActionListener(this);
    Container c = frame.getContentPane();
    c.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));
    caixaTexto.setWrapStyleWord(true);
    caixaTexto.setLineWrap(true);

    caixaTexto.setBorder(BorderFactory.createLoweredBevelBorder());
    c.add(tf); c.add(tf2); c.add(caixaTexto); c.add(bt);
    frame.setSize(300, 280);
    frame.setVisible(true);
}

/*
    Método que trata as ações disparadas pelo usuário. Neste
    caso, apenas o clique no botão para substituição das
    palavras no texto.
*/
public void actionPerformed(ActionEvent e) {
    Object o = e.getSource();
    if (o == bt) {
        String texto = null;
        try {
            texto = caixaTexto.getDocument().getText(0,
caixaTexto.getDocument().getLength());
        } catch (BadLocationException el) {
            el.printStackTrace();
        }
        // Se as caixas de texto não estão vazias
        if (tf.getText().length() > 0 &&
tf2.getText().length() > 0)
        {
            String textoNovo =
texto.replaceAll(tf.getText(), tf2.getText());
            caixaTexto.replaceRange(textoNovo, 0,
caixaTexto.getDocument().getLength());
        }
    }
}
}

```

Questão 4) (2.5 pontos)

Suponha que você precise implementar um programa para criação e edição de e-mails. Este programa deverá permitir a adaptação a diferentes idiomas. Veja o exemplo abaixo para os idiomas Português e Inglês.

De: Professor Para: Aluno Assunto: Internacionalização	From: Professor To: Student Subject: Internationalization
Querido(a) Aluno,	Dear Student,
Escrevo esta mensagem para ...	I write this message in order to ...
Sinceramente, Professor	Sincerely, Professor

Esta mensagem pode ser subdividida em partes: cabeçalho (até a saudação inicial Querido(a)/Dear), corpo da mensagem e rodapé (saudação final). Crie uma hierarquia para representar classes de e-mails em diferentes idiomas (basta definir para o Português e Inglês, mas deve permitir a extensão para outros idiomas). Como atributos a serem inseridos nesta hierarquia temos: remetente, destinatário, assunto e corpo do e-mail. É necessária também a criação de construtores e de um método para obter a mensagem completa. Para cada classe de idioma a ser criada, obrigue, através da definição de uma interface ou classe abstrata, que sejam fornecidas as traduções da estrutura de um e-mail: de, para, assunto, querido(a) e sinceramente (em Português neste caso). Reuse seus métodos e atributos sempre que possível.

Nota: A adequação de programas a diferentes idiomas é chamada de “Internacionalização” no desenvolvimento de sistemas.

Resposta:

```
/* Classe abstrata que modela a estrutura de um e-mail
*/
abstract class EmailAbstrato {
    // Atributos de um e-mail
    private String remetente;
    private String destinatario;
    private String assunto;
    private String mensagem;

    // Métodos que obrigam a sobrecarga por parte das
    //classes que especializam esta
    public abstract String de();
    public abstract String para();
    public abstract String assunto();
    public abstract String saudacaoInicial();
    public abstract String saudacaoFinal();

    // Construtor default de um e-mail
    public EmailAbstrato(String remet,
                        String dest,
                        String ass,
```

```

        String msg) {
            remetente = remet;
            destinatario = dest;
            assunto = ass;
            mensagem = msg;
        }

        // Métodos get/set para os atributos do e-mail
        public String getRemetente() {
            return remetente;
        }
        public void setRemetente(String remetente) {
            this.remetente = remetente;
        }
        public String getDestinatario() {
            return destinatario;
        }
        public void setDestinatario(String destinatario) {
            this.destinatario = destinatario;
        }
        public String getAssunto() {
            return assunto;
        }
        public void setAssunto(String assunto) {
            this.assunto = assunto;
        }
        public String getMensagem() {
            return mensagem;
        }
        public void setMensagem(String mensagem) {
            this.mensagem = mensagem;
        }

        // Este método obtém a mensagem completa,
        // combinando os dados presentes no e-mail
        public String obterMensagemCompleta () {
            String cabecalho = this.de() + " " + this.getRemetente() +
"\n" +
            this.getDestinatario() + "\n" +
            this.getAssunto() + "\n" +
            this.getDestinatario() +
            this.assunto() + " " +
            "\n" +
            this.saudacaoInicial() + " " +
            "\n";

            String rodape = "\n" +
            this.saudacaoFinal() + "\n" +
            this.getRemetente();

            return cabecalho + this.getMensagem() + rodape;
        }
    }

    /* Classe concreta para o Português */

```

```

class EmailPortugues extends EmailAbstrato {
    public EmailPortugues(String remet, String dest, String ass,
String msg) {
        super(remet, dest, ass, msg);
    }

    public String de() {
        return "De:";
    }

    public String para() {
        return "Para:";
    }

    public String assunto() {
        return "Assunto:";
    }

    public String saudacaoInicial() {
        return "Querido(a) ";
    }

    public String saudacaoFinal() {
        return "Sinceramente";
    }
}

/* Classe concreta para o Inglês */
class EmailIngles extends EmailAbstrato {
    public EmailIngles(String remet, String dest, String ass, String
msg) {
        super(remet, dest, ass, msg);
    }

    public String de() {
        return "From:";
    }

    public String para() {
        return "To:";
    }

    public String assunto() {
        return "Subject:";
    }

    public String saudacaoInicial() {
        return "Dear(a) ";
    }

    public String saudacaoFinal() {
        return "Sincerely";
    }
}

/* Classe Principal com exemplos de instanciação
* de objetos das classes criadas. */

```



```
public class AP3_2008_1_Q4 {  
    public static void main(String[] args) {  
        EmailAbstrato email = new EmailPortugues("Bazilio", "Aluno",  
"boa prova",  
            "Este e-mail é para desejar-lhe boa sorte!");  
        System.out.println(email.obtemMensagemCompleta());  
        EmailAbstrato emailIng = new EmailIngles("Bazilio", "Aluno",  
"Boa prova",  
            "Este e-mail é para desejar-lhe boa sorte!");  
        System.out.println(emailIng.obtemMensagemCompleta());  
    }  
}
```