



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação III

AP2 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)

Escreva um programa que receba como parâmetro de entrada, o nome de um arquivo texto, cujo conteúdo são o nome do aluno e as três notas dos alunos do curso, uma em cada linha, e que ordene o arquivo de saída em ordem crescente pela média do aluno. Isto é, se eu tiver como entrada o arquivo:

Antônio	10,0	9,5	7,5
Carlos	3,0	4,0	5,0
João	7,0	7,0	4,0

A saída será:

4,0
6,0
9,0

Um exemplo de uso desse programa seria `java calculaMedia notas.txt`.

Resposta:

```
import java.io.*;
public class calculaMedia {
    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 {
            float vet[] = new float[n];
            int cont = 0;
            String vs[];
            float x, media;

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

            while((s = in.readLine()) != null) {
                media = 0f;
                vs = s.split("\t");
                x = Float.parseFloat(vs[1]);
                media += x;
                x = Float.parseFloat(vs[2]);
                media += x;
                x = Float.parseFloat(vs[3]);
                media += x;
                media /= 3;
                vet[cont++] = media;
            }
            BufferedWriter out = new BufferedWriter(new FileWriter("saida-
"+args[0]));
            Ordena(vet);
            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();
        }
    }

    public static void Ordena (float[] vet){
        int i, j, menor;
        float temp;

        for (i = 0; i < vet.length; i++){
```

```

        menor = i;

        for (j = i + 1; j < vet.length; j++)
            if (vet[j] < vet [menor]) menor = j;

        temp = vet[menor];
        vet[menor] = vet[i];
        vet[i] = temp;
    }
}
}

```

Questão 2) (2.5 pontos)

O que será impresso no código a seguir. Sua resposta só será válida se você explicar o porquê.

```

class Estado{ Estado(String s){ System.out.println(s); } }

class Pessoa{
    Estado p = new Estado ("Ativo");
    Pessoa(){ System.out.println("Pessoa"); }
}

class Idoso extends Pessoa{
    Estado i = new Estado ("Sabio");
    Idoso(){ System.out.println("Idoso"); }
}

class Avo extends Idoso{
    Estado a1 = new Estado ("Alegre");
    Estado a2 = new Estado ("Amigo");
    Estado a3 = new Estado ("Satisfeito");

    Avo(){
        System.out.println("Avo");
        a3 = new Estado ("Orgulhoso");
    }

    void Fim(){System.out.println("Fim"); }
}

public class Teste{
    public static void main (String[] args){
        Avo a = new Avo();
        a.fim();
    }
}

```

Resposta:

Ativo
Pessoa
Sabio
Idoso
Alegre

Amigo
Satisfeito
Avo
Orgulhoso
Fim

Questão 3) (2.5 pontos)

Escreva um programa para criar uma janela contendo um campo de texto e um botão, como mostrado na figura abaixo, e que tenha o seguinte comportamento:

(a) O evento de pressionar a tecla *Return*, quando o cursor está posicionado sobre o campo de texto, deve ocasionar a criação de uma janela de diálogo na qual é impresso o texto contido no campo de texto, precedido por “O campo de texto contém: ”.

(b) O evento de clicar no botão deve “limpar” o campo de texto, isto é, deve apagar o conteúdo do campo de texto, fazendo com que ele fique em branco.

Resposta:

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

public class AP2_2007_2_Q3 {
    public static void main(String[] args) {
        new JTextto();
    }
}

class JTextto implements ActionListener {
    JFrame frame = new JFrame("Apaga Texto");
    JTextField tf = new JTextField(20);
    JButton bt = new JButton("Limpa");

    public JTextto() {
        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(bt);
        frame.setSize(300, 120);
        frame.setVisible(true);
    }

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

```
    }  
}
```

Questão 4) (2.5 pontos)

Podemos representar um polinômio como uma lista ordenada de termos, onde os termos são ordenados por seus expoentes. Para adição de 2 (dois) polinômios, suas respectivas listas são percorridas e os termos nas posições correntes dos iteradores são examinadas. Se o expoente de um é menor que o de outro, insere o primeiro na lista resultante e avança o iterador do primeiro. Se os expoentes são iguais, então criamos um novo termo com este expoente, somamos os coeficientes e avançamos os 2 (dois) iteradores. Por exemplo: $3x^4 + 2x^2 + 3x + 7$ somado com $2x^3 + 4x + 5$ resulta em $3x^4 + 2x^3 + 2x^2 + 7x + 12$.

Escreva um programa para ler e adicionar polinômios. Você deve definir uma classe Termo que contém atributos expoente e coeficiente.

Resposta:

```
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.util.ArrayList;  
import java.util.Iterator;  
import java.util.List;  
  
public class AP2_2007_2_Q4 {  
    public static void main(String[] args) {  
        Polinomio polinomio1 = new Polinomio();  
        Polinomio polinomio2 = new Polinomio();  
  
        polinomio1.lePolinomio();  
        polinomio2.lePolinomio();  
  
        Polinomio polinomio3 = polinomio1.adicao(polinomio2);  
  
        polinomio3.imprimePolinomio();  
    }  
}  
  
/* Classe que modela 1 único termo */  
class Term {  
    int coeficiente;  
    int expoente;  
  
    public Term(int c, int e) {  
        coeficiente = c;  
        expoente = e;  
    }  
  
    public int getCoeficiente() {  
        return coeficiente;  
    }  
  
    public int getExpoente() {
```

```

        return expoente;
    }

    public void setCoeficiente(int coeficiente) {
        this.coeficiente = coeficiente;
    }

    public void setExpoente(int expoente) {
        this.expoente = expoente;
    }

    public String toString() {
        return coeficiente + "x" + expoente;
    }
}

/* Classe que modela um polinômio */
class Polinomio {
    private List termos;

    public Polinomio() {
        termos = new ArrayList();
    }

    public List obterTermos() {
        return termos;
    }

    public void imprimePolinomio() {
        Iterator it = termos.iterator();
        while (it.hasNext()) {
            System.out.print(((Term)it.next()).toString() + " ");
        }
    }

    public void lePolinomio() {
        try {
            BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
            String str = "";
            int exp = -1, coef = -1;
            System.out.println("Digite os expoentes e coeficientes do
polinômio ");
            while (exp != 0 && coef != 0) {
                System.out.print("Expoente: ");
                str = in.readLine();
                exp = Integer.parseInt(str);
                System.out.print("Coeficiente: ");
                str = in.readLine();
                coef = Integer.parseInt(str);
                if (exp != 0 && coef != 0) {
                    Term t = new Term(coef, exp);
                    termos.add(t);
                }
            }
        } catch (IOException e) {
        }
    }
}

```

```

    }

    public Polinomio adicao(Polinomio p) {
        Term t1 = null;
        Term t2 = null;

        Iterator it1 = termos.iterator();
        Iterator it2 = p.obtemTermos().iterator();

        Polinomio pRes = new Polinomio();
        List res = pRes.obtemTermos();

        while (it1.hasNext() && it2.hasNext()) {
            if (t1 == null)
                t1 = (Term) (it1.next());
            if (t2 == null)
                t2 = (Term) (it2.next());
            if (t1.getExpoente() == t2.getExpoente())
                res.add(new
Term(t1.getCoeficiente()+t2.getCoeficiente(), t1.getExpoente()));
            else
                if (t1.getExpoente() < t2.getExpoente())
                    res.add(new Term(t1.getCoeficiente(),
t1.getExpoente()));
                else
                    res.add(new Term(t2.getCoeficiente(),
t2.getExpoente()));

                if (it1.hasNext())
                    t1 = (Term) (it1.next());
                else
                {
                    res.add(new Term(t2.getCoeficiente(),
t2.getExpoente()));

                    while (it2.hasNext()) {
                        t2 = (Term) (it2.next());
                        res.add(new Term(t2.getCoeficiente(),
t2.getExpoente()));
                    }
                    break;
                }
                if (it2.hasNext())
                    t2 = (Term) (it2.next());
                else
                {
                    res.add(new Term(t1.getCoeficiente(),
t1.getExpoente()));

                    while (it1.hasNext()) {
                        t1 = (Term) (it1.next());
                        res.add(new Term(t1.getCoeficiente(),
t1.getExpoente()));
                    }
                    break;
                }
            }

        while (it1.hasNext()) {

```

```
        t1 = (Term) (it1.next());
        res.add(new Term(t1.getCoeficiente(),
t1.getExpoente()));
    }

    while (it2.hasNext()) {
        t2 = (Term) (it2.next());
        res.add(new Term(t2.getCoeficiente(),
t2.getExpoente()));
    }

    return pRes;
}
}
```