



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação III

Gabarito da AP2 1º semestre de 2010.

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 é composto das seguintes informações a respeito de figuras geométricas concretas, separadas por “/”:

- nome da figura (pode ser círculo, quadrado, retângulo e triângulo); e
- suas dimensões, dependendo da figura em questão:
 - se a figura for um círculo, a próxima informação é o valor do raio;
 - se a figura for um quadrado, a próxima informação é o valor do lado;
 - se a figura for um retângulo, a próxima informação é o valor da base e da altura; e
 - se a figura for um triângulo, a próxima informação é o valor da base e da altura.

Um exemplo de arquivo de entrada seria:

```
circulo/1.0
quadrado/2.0
retangulo/1.0/2.0
triangulo/1.0/2.0
```

Escreva um programa que, leia este arquivo, ordene e imprima, de maneira decrescente, as áreas das figuras geométricas concretas pertencentes ao arquivo. Sabendo-se como se calcula a área de cada uma destas figuras:

- a) círculo (dado de entrada: raio (r)) $\rightarrow \text{área} = (\pi * r^2) / 2$;
- b) quadrado (dado de entrada: lado (l)) $\rightarrow \text{área} = l^2$;
- c) retângulo (dado de entrada: base (b) e altura (h)) $\rightarrow \text{área} = b * h$; e
- d) triângulo (dado de entrada: base (b) e altura (h)) $\rightarrow \text{área} = (b * h) / 2$.

Um exemplo de uso desse programa seria `java ordena figuras.txt`, onde `figuras.txt` é o nome do arquivo de entrada.

A resposta para o exemplo de arquivo de entrada seria:

```
Quadrado    area = 4.0
Circulo      area = 3.141516
Retangulo    area = 2.0
Triangulo     área = 1.0
```

DICA PARA RESOLVER ESTA QUESTÃO: Considere uma classe abstrata figura que generalize as figuras geométricas concretas círculo, quadrado, retângulo e triângulo.

RESPOSTA:

```
import java.io.*;
```

```
import java.util.Scanner;
```

```
abstract class forma{
```

```
    abstract double calcArea();
```

```
}
```

```
class Circulo extends forma{
```

```
    double r;
```

```
    Circulo(double r0){
```

```
        r = r0;
```

```
    }
```

```
    double calcArea(){ return (Math.PI * r * r);
}
```

```
    public String toString(){
```

```
        return "Circulo\tarea = " + calcArea();
```

```
    }
```

```
}
```

```
class Quadrado extends forma{
```

```

    double l;

    Quadrado(double l0){
    l = l0;
    }

    double calcArea(){
    return (l * l);
    }

    public String toString(){
    return "Quadrado\tarea = " + calcArea();
    }
}

```

```

class Triangulo extends forma{

    double b, h;

    Triangulo(double b0, double h0){
    b = b0;
    h = h0;
    }

    double calcArea(){
    return (b * h) / 2.0;
    }

    public String toString(){
    return "Triangulo\tarea = " + calcArea();
    }
}

```

```

class Retangulo extends forma{

    double b, h;

    Retangulo(double b0, double h0){
    b = b0;
    h = h0;
    }

    double calcArea(){
    return (b * h);
    }

    public String toString(){
    return "Retangulo\tarea = " + calcArea();
    }
}

```

```

public class ordena{

    public static void ordena_vet (forma[] vet){

        int i, j, maior;

        forma temp;

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

            maior = i;

            for (j = i + 1; j < vet.length; j++)

                if(vet[j].calcArea()>vet[maior].calcArea()) maior = j;

            temp = vet[maior];

            vet[maior] = vet[i];

            vet[i] = temp;

        }

    }

    public static void main(String[] args) throws IOException{

        BufferedReader sc = new BufferedReader(new FileReader(args[0]));

        String s;

        int n = 0;

        try{

            while ((s = sc.readLine()) != null) n++;

            sc.close();

        }

        catch (Exception e) {

            System.out.println("Excecao1\n");

        }

    }

}

```

```

}

try{

    sc = new BufferedReader(new FileReader(args[0]));

    if(n >= 0){

        forma vet[] = new forma[n];

        int i;

        String vs[], p;

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

            s = sc.readLine();

            vs = s.split("/");

            p = vs[0];

            if(p.equals("circulo")){

                double r = Double.parseDouble(vs[1]);

                vet[i] = new Circulo(r);

            }

            else if(p.equals("quadrado")){

                double l = Double.parseDouble(vs[1]);

                vet[i] = new Quadrado(l);

            }

            else if(p.equals("retangulo")){

                double b = Double.parseDouble(vs[1]);

                double h = Double.parseDouble(vs[2]);

                vet[i] = new Retangulo(b,h);

            }

            else{

                double b = Double.parseDouble(vs[1]);

```

```

        double h = Double.parseDouble(vs[2]);

        vet[i] = new Triangulo(b,h);

    }

}

for(i = 0; i < vet.length; i++) System.out.println(vet[i]);

ordena_vet(vet);

for(i = 0; i < vet.length; i++) System.out.println(vet[i]);

sc.close();

}

} catch (Exception e){

    System.out.println("Excecao2\n");

}

}

}

```

Questão 2) (3.0 pontos)

Suponha a estrutura do código abaixo, o qual fez parte da questão 2 pedida na AP3 (2009.2).

```

enum TipoOperacao {
    saque, deposito, transferencia
};

class Transacao {
    ContaCorrente conta;
    TipoOperacao operacao;
    float valor;
    GregorianCalendar hora;
    public Transacao(ContaCorrente c, TipoOperacao op, float v) {
        ...
    }
    public void exibeTransacao () {
        ...
    }
}

class Transferencia extends Transacao {

```

```

        ContaCorrente conta;
        public Transferencia(ContaCorrente c, float v, ContaCorrente c2) {
            ...
        }
        public void exibeTransacao () {
            ...
        }
    }

class ContaCorrente {
    int numConta;
    float saldo;
    List<Transacao> transacoes;
    public ContaCorrente(int pConta, float pSaldo) {
        ...
    }
    public float consultaSaldo() {
        ...
    }
    public void realizaDeposito(float valor) {
        ...
    }
    public void realizaSaque(float valor) {
        ...
    }
    public void realizaTransferencia(float valor, ContaCorrente
pBenef) {
        ...
    }
    public void exibeExtrato() {
        for (Transacao trans : transacoes) {
            trans.exibeTransacao();
        }
    }
}

```

- Crie uma classe Pagamento, a qual será utilizada para armazenar um conjunto de pagamentos de boletos a serem pagos. Cada pagamento deve conter a data de pagamento, o valor, um código de barra e uma breve descrição. O conjunto de pagamentos deve estar associado a uma conta corrente.
- Crie um método chamado volumeTotal(), o qual exibe, para um dado cliente, a soma total movimentada por este. Neste caso, saques, transferências e pagamentos subtraem uma quantidade do total, enquanto que os depósitos acrescem uma quantia ao total.
- Altere o método exibeExtrato() para que liste também os pagamentos a serem efetuados.

RESPOSTA:

a) Neste item, observe que poderíamos criar a classe sem atentarmos para o fato que esta seria uma transação. Desta maneira, na classe ainda teríamos que declarar o valor do pagamento e a conta a ser debitada. Já colocando desta forma, ganhamos de graça o item c), pois basta a classe Pagamento herdar da classe Transacao para que os objetos da primeira sejam exibidos. Naturalmente, de forma a se ter um tratamento mais adequado na exibição, podemos redefinir o método `exibeExtrato()` para a classe Pagamento.

Obs.: Nos gabaritos, costumo abstrair de questões importantes, como encapsulamento (por ex., tornar os campos privados e definir métodos públicos `get/set` para estes campos), para destacar o que realmente é importante.

```
class Pagamento extends Transacao {
    int codigo;
    String descricao;
    GregorianCalendar dataPagamento;
    public Pagamento(ContaCorrente c, float v, int codigo, String
descricao, GregorianCalendar data) {
        super(c, v);
        this.codigo = codigo;
        this.descricao = descricao;
        this.dataPagamento = data;
    }
}
```

b) Neste item, o método criado pode percorrer a lista de transações e obter o valor envolvido em cada transação. Para tal, cada classe que implementa uma transação teria que definir o método `obtemValor()` (por ex., para o código que forneci). Para ter esta garantia, poderíamos definir este método como abstrato na classe Transação (isto não era pedido na questão, mas mostraria domínio na questão).

```
class ContaCorrente {
    int numConta;
    float saldo;
    List<Transacao> transacoes;

    ...

    public float volumeTotal() {
        float total = 0f;
        for (Transacao trans : transacoes) {
            total = total + trans.obtemValor();
        }
        return total;
    }
}

class Pagamento extends Transacao {
    int codigo;
```



```

String descricao;
GregorianCalendar dataPagamento;
...

public float obterValor() {
    return -valor;    // Indica que a transação debita um valor
}
}

```

c) Se já fizemos com que a classe Pagamento herdasse de Transação, bastaria redefinir o método `exibeExtrato()`.

```

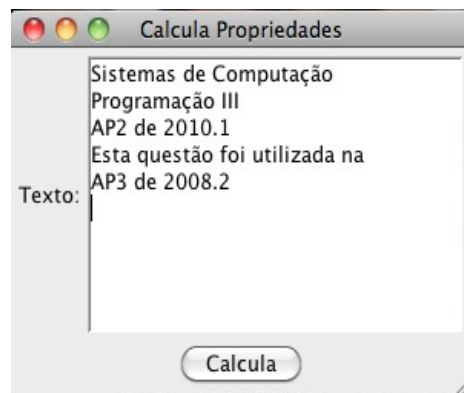
class Pagamento extends Transacao {
    ...

    public void exibeTransacao () {
        System.out.println("****");
        System.out.println("Cliente: " + conta.numConta);
        System.out.println("Operacao: Transação");
        System.out.println("Valor: " + valor);
        SimpleDateFormat sdf = new SimpleDateFormat("hh:mm:ss -
dd.MM.yyyy");
        System.out.println("Hora: " + sdf.format(hora.getTime()));
    }
}

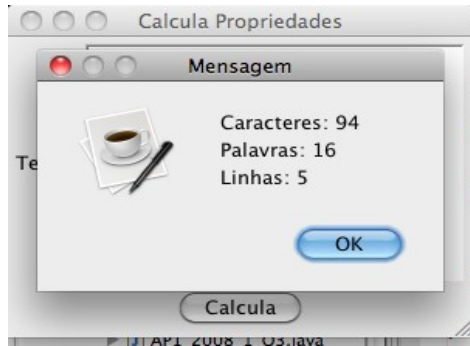
```

Questão 3) (3.0 pontos)

Escreva um programa que crie a janela abaixo:



Após digitarmos algum texto na caixa de texto apresentada e clicarmos no botão “Calcula”, a seguinte caixa de diálogo deve ser apresentada com algumas propriedades do texto:



RESPOSTA:

Comentários da possível solução apresentada ao longo do código.

```
/*
Classe principal que inicia a janela da aplicação
*/
public class AP2_2010_1_Q3 {
    public static void main(String[] args) {
        new JTexto();
    }
}

/*
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 JTexto implements ActionListener {
    JFrame frame = new JFrame("Calcula Propriedades");
    JLabel texto = new JLabel("Texto:");
    JTextArea caixaTexto = new JTextArea(10, 20);
    JButton bt = new JButton("Calcula");

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

        caixaTexto.setBorder(BorderFactory.createLoweredBevelBorder());
        c.add(texto); c.add(caixaTexto); c.add(bt);
        frame.setSize(300, 230);
        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 {
            int tamanho =
caixaTexto.getDocument().getLength();
            texto = caixaTexto.getDocument().getText(0,
tamanho);
            // Gera sequência de palavras delimitadas por
            espaço e \n

            String [] palavras = texto.split("[ \n]");
            int numPalavrasVazias = 0;
            for (int i=0; i<palavras.length; i++)
                if (palavras[i].trim().length() == 0)
                    numPalavrasVazias++;
            int numPalavras = palavras.length -
numPalavrasVazias;

            int numLinhas = texto.split("\n").length;
            JOptionPane.showMessageDialog(frame,
                "Caracteres: " + (tamanho-numLinhas+1) +
                "\nPalavras: " + numPalavras +
                "\nLinhas: " + numLinhas);
        } catch (BadLocationException e1) {
            e1.printStackTrace();
        }
    }
}
}

```