



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação III

AP2 1º semestre de 2012.

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) (3.0 pontos)

Escreva um programa em Java que receba o nome de um arquivo texto, cujo conteúdo são valores inteiros, e um inteiro N e imprima na tela o número de vezes que N aparece e em quais linhas.

RESPOSTA:

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

```
class no{  
    int linha;  
    no prox;
```

```
    no(int l){  
        linha = l;  
        prox = null;  
    }
```

```
public String toString(){  
    String s = "linha: " + linha + "\n";  
    return s;
```

```
    }  
}
```

```
class lista{
```

```
    int qtde;
```

```
    no prim;
```

```
    lista(){
```

```
        qtde = 0;
```

```
        prim = null;
```

```
    }
```

```
    void insere(int linha){
```

```
        qtde++;
```

```
        no novo = new no(linha);
```

```
        if(prim == null) prim = novo;
```

```
        else{
```

```
            no p = prim;
```

```
            while(p.prox != null) p = p.prox;
```

```
            p.prox = novo;
```

```
        }
```

```
    }
```

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

```
        no p = prim;
```

```
        String s = "Quantidade: " + qtde + "\n";
```

```
        while(p != null){
```

```
            s += p.toString();
```

```
            p = p.prox;
```

```
        }
```

```
        return s;
```

```
    }
```

```
}
```

```
public class Q1_AP2_2012_1{
```

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

```
        BufferedReader in;
```

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

```
        int N = Integer.parseInt(args[1]), aux, linha = 0;
```

```
        String s;
```

```
        try{
```

```
            lista l = new lista();
```

```

while((s = in.readLine()) != null){
    linha++;
    aux = Integer.parseInt(s);
    if(aux == N) l.insere(linha);
}
in.close();
System.out.println(l);
} catch(Exception e){ System.out.println("Excecao\n"); }
}
}

```

Questão 2) (3.0 pontos)

Considere que seu programa tem como objetivo retornar um relatório de cidades, com as distâncias de cada cidade para as demais. Neste relatório, para cada cidade CID, você deve mostrar todas as distâncias para CID, a cidade mais perto e a mais distante. Este programa recebe um arquivo texto contendo, em cada linha, duas cidades e distância entre elas (em km), separadas por “/”. Um exemplo de um arquivo de entrada (DICA: este programa tem que funcionar para QUAISQUER arquivos no mesmo formato) é mostrado a seguir:

```

Rio de Janeiro/Guarapari/500
Rio de Janeiro/Sao Paulo/600
Guarapari/Vitoria/50
Rio de Janeiro/Vitoria/550
Vitoria/Sao Paulo/750

```

Escreva um programa que leia o arquivo de entrada, gere um arquivo de saída, cujo nome é saída-<nome do arquivo texto de entrada>, com as informações agrupadas por cidade. No caso do exemplo acima, o arquivo de saída seria:

```

Rio de Janeiro:
Guarapari      500km
Sao Paulo      600km
Vitoria        550km
MAIS PERTO     Guarapari
MAIS LONGE     Sao Paulo

Guarapari:
Rio de Janeiro 500km
Vitoria        50km
MAIS PERTO     Vitoria
MAIS LONGE     Rio de Janeiro

Sao Paulo:
Rio de Janeiro 600km
Vitoria        750km
MAIS PERTO     Rio de Janeiro
MAIS LONGE     Vitoria

```

Vitoria:
Rio de Janeiro 550km
Guarapari 50km
Sao Paulo 750km
MAIS PERTO Guarapari
MAIS LONGE Sao Paulo

Um exemplo de uso desse programa seria java Q2_AP2_2012_1 distancias.txt, onde distancias.txt é o nome do arquivo de entrada.

RESPOSTA:

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

```
class cidade{  
    String viz;  
    int dist;  
    cidade prox;  
  
    cidade(String v, int d){  
        viz = v;  
        dist = d;  
        prox = null;  
    }  
  
    public String toString(){  
        String s = viz + "\t" + dist + "km\n";  
        return s;  
    }  
}
```

```
class no{  
    String nome, longe, perto;  
    int dist_longe, dist_perto;  
    cidade prim;  
    no prox;  
  
    no(String n){  
        nome = n;  
        longe = perto = "";  
        dist_longe = dist_perto = 0;  
        prox = null;  
        prim = null;  
    }
```

```

void ins_cid(String viz, int dist){
    cidade novo = new cidade(viz, dist);
    if(prim == null) prim = novo;
    else{
        cidade p = prim;
        while(p.prox != null) p = p.prox;
        p.prox = novo;
    }

    if(dist_longe == 0){
        longe = perto = viz;
        dist_longe = dist_perto = dist;
    }

    else{
        if(dist_longe < dist){
            longe = viz;
            dist_longe = dist;
        }

        else if (dist_perto > dist){
            perto = viz;
            dist_perto = dist;
        }
    }
}

public String toString(){
    String s = "\n";
    s = s + nome + ": \n";
    cidade p = prim;
    while(p != null){
        s += p.toString();
        p = p.prox;
    }
    s = s + "MAIS PERTO\t" + perto + "\n";
    s = s + "MAIS LONGE\t" + longe + "\n";
    return s;
}
}

class lista{
    no prim;

```

```

lista(){
    prim = null;
}

no busca(String nome){
    no p = prim;
    while((p != null) && (!(nome.equals(p.nome)))) p = p.prox;
    return p;
}

void insere(String nome, String viz, int dist){
    no p = busca(nome);
    if(p == null){
        p = new no(nome);
        if(prim == null) prim = p;
    else{
        no q = prim;
        while(q.prox != null) q = q.prox;
        q.prox = p;
    }
}
    p.ins_cid(viz, dist);
}

public String toString(){
    String s = "";
    no p = prim;
    while(p != null){
        s += p.toString();
        p = p.prox;
    }
    return s;
}
}

public class Q2_AP2_2012_1{
    public static void main(String[] args) throws IOException{
        BufferedReader in;
        in = new BufferedReader(new FileReader(args[0]));
        String s;
        try{
            lista l = new lista();
            String vs[];

```

```

while((s = in.readLine()) != null){
    vs = s.split("/");
    l.inserere(vs[0], vs[1], Integer.parseInt(vs[2]));
    l.inserere(vs[1], vs[0], Integer.parseInt(vs[2]));
}
in.close();
BufferedWriter out;
out = new BufferedWriter(new FileWriter("saida-" + args[0]));
out.write(l.toString());
out.close();
} catch (Exception e) { System.out.println("Excecao\n"); }
}
}

```

Questão 3) (4.0 pontos)

Suponha as definições de interface e classe dadas à seguir, as quais servem para representar figuras geométricas (no código abaixo, retângulos) no plano cartesiano (x, y):

```

import java.awt.Point;

interface Figura {
    void moveCima ();
    void moveBaixo ();
    void moveEsq ();
    void moveDir ();
    void desloca (Point pos);
    void desenha ();
}

class Retangulo implements Figura {
    Point lados[];

    public Retangulo(int x1, int y1, int x2, int y2, int x3, int y3, int x4,
int y4) {
        lados = new Point[4];
        lados[0] = new Point(x1, y1);
        lados[1] = new Point(x2, y2);
        lados[2] = new Point(x3, y3);
        lados[3] = new Point(x4, y4);
    }

    public void moveCima() {
        for (Point p : lados) {
            p.y = p.y - 1;
        }
    }

    public void moveBaixo() {
        for (Point p : lados) {
            p.y = p.y + 1;
        }
    }
}

```

```

    public void moveEsq() {
        for (Point p : lados) {
            p.x = p.x - 1;
        }
    }
    public void moveDir() {
        for (Point p : lados) {
            p.x = p.x + 1;
        }
    }
    public void desloca (Point pos) {
        for (Point p : lados) {
            p.x = p.x + pos.x;
            p.y = p.y + pos.y;
        }
    }
    public void desenha() {
        System.out.println("Retangulo: ");
        for (Point p : lados) {
            System.out.println("(" + p.x + "," + p.y + ")");
        }
    }
}

```

- Defina uma classe Circulo, a qual deverá conter um ponto (dois inteiros, representando coordenadas x e y), o qual representa o centro do círculo e seu raio (um valor decimal). Leve em consideração que objetos do tipo círculo, assim como objetos do tipo retângulo, poderão ser adicionados numa mesma lista.
- Defina uma classe Main, com um método main(), e crie neste método um objeto retângulo nas posições (1,1), (1,2), (3,1) e (3,2). Crie também um objeto do tipo círculo com posição central (4,4) e raio 2.5.
- Defina uma lista no método main() e adicione os objetos criados. Desenhe (exiba os atributos) de cada objeto na lista deslocado 10 unidades em x e 10 unidades em y. Ou seja, para cada figura na lista, adicione 10 unidades na componente x e 10 na componente y e desenhe (exiba os valores resultantes dos seus atributos).

RESPOSTA:

```

class Circulo implements Figura {
    double raio;
    Point centro;

    public Circulo(double raio, int x, int y) {
        this.raio = raio;
        this.centro = new Point(x, y);
    }

    public void moveCima() {
        centro.y = centro.y - 1;
    }

    public void moveBaixo() {
        centro.y = centro.y + 1;
    }

    public void moveEsq() {

```



```

        centro.x = centro.x - 1;
    }

    public void moveDir() {
        centro.x = centro.x + 1;
    }

    public void desloca(Point pos) {
        centro.x = centro.x + pos.x;
        centro.y = centro.y + pos.y;
    }

    public void desenha() {
        System.out.println("Circulo:");
        System.out.println("(" + centro.x + "," + centro.y + ") " + " Raio: " +
raio);
    }
}

public class AP2_2012_1_Q3 {
    public static void main(String[] args) {
        List<Figura> objetos = new ArrayList<Figura>();
        Retangulo r = new Retangulo(1, 1, 1, 2, 3, 1, 3, 2);
        Circulo c = new Circulo(2.5, 4, 4);
        objetos.add(r);
        objetos.add(c);
        Point pos10 = new Point(10,10);
        for (Figura f : objetos) {
            f.desloca(pos10);
            f.desenha();
        }
    }
}

```