



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação III

AP3 2º semestre de 2011.

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) (5 pontos)

Supondo que você tenha uma empresa de software, e que o Comitê Olímpico contrate sua empresa para escrever um programa que ordene, de maneira decrescente, o quadro de medalhas de Olimpíadas, além de listar todos os esportes ganhadores de medalhas.

A ordem do quadro de medalhas é determinada a partir do maior número de medalhas de ouro, prata e bronze, seguindo esta ordem. Caso dois países tenha o mesmo número de medalhas de ouro, o critério de desempate é o número de medalhas de prata. Já se dois países tiverem o mesmo número de medalhas de ouro e de prata, o critério de desempate é o número de medalhas de bronze.

Os dados de entrada são as modalidades de esportes, juntamente com os países que conseguiram as medalhas, separadas por “/”. Um exemplo de arquivo de entrada seria:

```
//ESPORTE/OURO/PRATA/BRONZE
Futebol feminino/Canada/Brasil/Mexico
Futebol masculino/Mexico/Argentina/Uruguai
Volei masculino/Brasil/Cuba/Argentina
Volei feminino/Brasil/Cuba/USA
```

A saída será:

```
1) Brasil: 3
Ouro: 2
Volei masculino
Volei feminino
Prata: 1
Futebol feminino
```

Bronze: 0

2) Mexico: 2

Ouro: 1

Futebol masculino

Prata: 0

Bronze: 1

Futebol feminino

3) Canada: 1

Ouro: 1

Futebol feminino

Prata:0

Bronze:0

4) Cuba: 2

Ouro: 0

Prata: 2

Volei masculino

Volei feminino

Bronze: 0

5) Argentina: 2

Ouro: 0

Prata: 1

Futebol masculino

Bronze: 1

Volei masculino

6) Uruguai: 1

Ouro: 0

Prata: 0

Bronze: 1

Futebol masculino

6) USA: 1

Ouro: 0

Prata: 0

Bronze: 1

Volei feminino

RESPOSTA:

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

```
class Medalha{  
    String modalidade;  
    Medalha prox;
```

```
    Medalha(String nome){  
        prox = null;  
        modalidade = nome;
```

```
    }
```

```
    public String toString(){ return modalidade + "\n"; }  
}
```

```

class Lista_Medalha{
    int total;
    String nome;
    Medalha prim;

    Lista_Medalha(String nome){
        this.nome = nome;
        total = 0;
        prim = null;
    }

    public String toString(){
        String s = nome + ": " + total + "\n";
        Medalha p = prim;
        while(p != null){
            s += p.toString();
            p = p.prox;
        }
        return s;
    }

    void insere(String mod){
        Medalha nova = new Medalha(mod);
        total++;
        nova.prox = prim;
        prim = nova;
    }
}

class Pais{
    String nome;
    int total;
    int ordem;
    Lista_Medalha ouro;
    Lista_Medalha prata;
    Lista_Medalha bronze;

    Pais(String nome){
        this.nome = nome;
        total = ordem = 0;
        ouro = new Lista_Medalha("ouro");
        prata = new Lista_Medalha("prata");
        bronze = new Lista_Medalha("bronze");
    }

    void insere_medalha(String tipo, String mod){
        total++;
        if(tipo.equals(ouro.nome)) //Ouro
            ouro.insere(mod);
        else if(tipo.equals(prata.nome)) //Prata
            prata.insere(mod);
        else //Bronze
            bronze.insere(mod);
    }
}

```

```

    public String toString(){
        String s = ordem + ")" + nome + ": " + total + "\n";
        s += ouro.toString();
        s += prata.toString();
        s += bronze.toString() + "\n";
        return s;
    }
}

class Lista_Paises{
    Pais vet[];
    int pos;

    Lista_Paises(int n){
        vet = new Pais[n];
        pos = 0;
    }

    Pais Busca(String nome){
        int i;
        for(i = 0; i < pos; i++) if(vet[i].nome.equals(nome)) return
vet[i];
        return null;
    }

    void insere(String mod, String nome_o, String nome_p, String nome_b){
        Pais p = Busca(nome_o);
        if(p == null){
            vet[pos] = new Pais(nome_o);
            vet[pos].insere_medalha("ouro", mod);
            pos++;
        }
        else p.insere_medalha("ouro", mod);
        p = Busca(nome_p);
        if(p == null){
            vet[pos] = new Pais(nome_p);
            vet[pos].insere_medalha("prata", mod);
            pos++;
        }
        else p.insere_medalha("prata", mod);
        p = Busca(nome_b);
        if(p == null){
            vet[pos] = new Pais(nome_b);
            vet[pos].insere_medalha("bronze", mod);
            pos++;
        }
        else p.insere_medalha("bronze", mod);
    }

    public String toString(){
        int i = 0;
        String s = "";
        while(i < pos){
            s = s + vet[i].toString(); i++;
        }
        return s;
    }
}

```

```

    }}
    public class Q1_AP3_2011_2{
        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 += 3;
                in.close();
            } catch (Exception e) { System.out.println("Excecao 1\n"); }
            try {
                Lista_Paises lp = new Lista_Paises(n);
                String vs[];
                in = new BufferedReader(new FileReader(args[0]));
                while((s = in.readLine()) != null) {
                    vs = s.split("/");
                    lp.insere(vs[0], vs[1], vs[2], vs[3]);
                }
                in.close();
                Ordena(lp);
                System.out.println(lp);
            } catch (Exception e) { System.out.println("Excecao 2\n"); }
        }

        static void Ordena(Lista_Paises l){
            int j,k;
            for(j = 0; j < l.pos; j++){
                for(k = j + 1; k < l.pos; k++){
                    if((l.vet[j].ouro.total < l.vet[k].ouro.total) ||
                       ((l.vet[j].ouro.total == l.vet[k].ouro.total) &&
                        (l.vet[j].prata.total < l.vet[k].prata.total)) ||
                       ((l.vet[j].ouro.total == l.vet[k].ouro.total) &&
                        (l.vet[j].prata.total == l.vet[k].prata.total) &&
                        (l.vet[j].bronze.total < l.vet[k].bronze.total)))){
                        Pais temp = l.vet[j];
                        l.vet[j] = l.vet[k];
                        l.vet[k] = temp;
                    }
                }
            }
            for(j = 0; j < l.pos; j++) l.vet[j].ordem = j + 1;
            for(j = 1; j < l.pos; j++){
                if(l.vet[j].ordem != l.vet[j - 1].ordem) {
                    if((l.vet[j].ouro.total == l.vet[j - 1].ouro.total) &&
                       (l.vet[j].prata.total == l.vet[j - 1].prata.total) &&
                       (l.vet[j].bronze.total == l.vet[j - 1].bronze.total))
                        l.vet[j].ordem = l.vet[j - 1].ordem;
                }
            }
        }
    }
}

```

Questão 2) (5 pontos)

Suponha o código do programa abaixo:

```

public class AP3_2011_2_Q2 {
    public static void main(String[] args) {
        Agenda ag = new Agenda();
    }
}

```

```

        Contato c = new Contato("Fulano", "Petrópolis");
        c.adicionaTelefone(new Tel(55, 24, 99999999));
        c.adicionaTelefone(new Tel(55, 21, 33333333));
        c.adicionaTelefone(new Ramal(55, 24, 22222222, 3500));
        ag.adicionaContato(c);
        Contato c2 = new Contato("Ciclano", "Rio das Ostras");
        c2.adicionaTelefone(new Tel(55, 22, 99999999));
        ag.adicionaContato(c2);
        ag.imprimeContatos();
    }
}

```

Defina as classes, seus respectivos construtores e métodos, de forma que o código fornecido funcione.

RESPOSTA:

```

import java.util.ArrayList;
import java.util.List;

class Tel {
    int pais;
    int cidade;
    int numero;

    public Tel(int pais, int cidade, int numero) {
        this.pais = pais;
        this.cidade = cidade;
        this.numero = numero;
    }

    public String toString() {
        return "+" + pais + " " + cidade + " " + numero;
    }
}

class Ramal extends Tel {
    int ramal;

    public Ramal(int pais, int cidade, int numero, int ramal) {
        super(pais, cidade, numero);
        this.ramal = ramal;
    }

    public String toString() {
        return super.toString() + " #" + this.ramal;
    }
}

class Contato {
    String nome;
    String endereco;
    List<Tel> telefones;

    public Contato(String nome, String endereco) {
        this.nome = nome;
        this.endereco = endereco;
        this.telefones = new ArrayList<Tel>();
    }
}

```

```

    }

    public void adicionaTelefone (Tel tel) {
        this.telefones.add(tel);
    }

    public String toString () {
        String tels = "";
        for (Tel t : this.telefones)
            tels = tels + t.toString() + " ";
        return this.nome + "\n" + this.endereco + "\n" + tels + "\n";
    }
}

class Agenda {
    List<Contato> contatos;
    public Agenda() {
        contatos = new ArrayList<Contato>();
    }

    public void adicionaContato (Contato c) {
        contatos.add(c);
    }

    public void imprimeContatos() {
        for (Contato c: contatos) {
            System.out.println(c);
        }
    }
}

```