Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
AD2 de Programação III
1° semestre de 2012

Nome:
Matrícula:
Pólo:

Exercício (ENTREGAR OS ARQUIVOS EM MÍDIA, PARA FINS DE TESTE, JUNTAMENTE COM A AD IMPRESSA):

Considere que sua empresa de software seja contratada pela ATP (Associação dos Tenistas Profissionais) para fornecer estatísticas sobre o ranking dos tenistas no ano corrente. Seu programa deve receber um arquivo composto de resultados, com as seguintes informações: data da final do torneio, nome da competição, país onde o torneio foi realizado, a quantidade de pontos que serão contabilizados ao vencedor da competição, a premiação (em unidades monetárias quaisquer) e, por fim, o nome do vencedor. Um EXEMPLO de arquivo neste formato seria (OBSERVE QUE SEU PROGRAMA DEVE FUNCIONAR PARA QUALQUER ARQUIVO NO FORMATO ANTERIORMENTE DEFINIDO):

```
02.01.12/Brisbane/Australia/250/434250.0/Andy Murray
02.01.12/Chennai/India/250/398250.0/Milos Raonic
02.01.12/Doha/Qatar/250/1024000.0/Jo Wilfried Tsonga
09.01.12/Heineken Open/New Zealand/250/398250.0/David Ferrer
09.01.12/Apia Int. Sydney/Australia/250/434250.0/Jarkko Nieminen
16.01.12/Australian Open/Australia/2000/11806550.0/Novak Djokovic
30.01.12/Open Sud de France/France/250/398250.0/Tomas Berdych
13.02.12/ABN AMRO World/The Netherlands/500/1207500.0/Roger Federer
13.02.12/SAP Open/U.S.A./250/531000.0/Milos Raonic
13.02.12/Brasil Open 2012/Brazil/250/475300.0/Nicolas Almagro
20.02.12/Copa Claro/Argentina/250/484100.0/David Ferrer
20.02.12/Open 13/France/250/512750.0/Juan Martin Del Potro
20.02.12/Regions Morgan Keegan Champ/U.S.A./500/1155000.0/Jurgen Melzer
27.02.12/Abierto Mexicano Telcel/Mexico/500/1155000.0/David Ferrer
27.02.12/Delray Beach Int Tennis Cham/U.S.A./250/442500.0/Kevin Anderson
27.02.12/Dubai Duty Free Tennis Champ/U.A.E./500/1700475.0/Roger Federer
08.03.12/BNP Paribas Open/U.S.A./1000/4694969.0/Roger Federer
21.03.12/Sony Ericsson Open/U.S.A./1000/3973050.0/Novak Djokovic
```

As estatísticas pedidas pela ATP são:

a) a lista de torneios vencidos, em ordem decrescente, por cada competidor, juntamente com os nomes das competições. Se existirem dois jogadores com o mesmo número de vitórias, o critério de desempate será o número de pontos obtidos. Caso os competidores tenham, também, o mesmo número de pontos, o

critério de desempate será a premiação obtida. No exemplo acima, Roger Federer seria o primeiro da lista com TRÊS torneios vencidos;

b) a lista com o número de pontos obtidos por cada jogador, em ordem decrescente (o competidor que tiver mais pontos deve ser o primeiro da listagem), juntamente com os nomes das competições. Se existirem dois jogadores com o mesmo número de pontos, o critério de desempate será o número de vitórias. Caso os competidores tenham, também, o mesmo número de vitórias, o critério de desempate será a premiação obtida. No exemplo do arquivo dado, Novak Djokovic é o líder deste ranking;

c) a lista de países onde os torneios foram sediados, em ordem decrescente, juntamente com os nomes das competições. Se existirem dois países com o mesmo número de torneios, o critério de desempate será o número de pontos fornecidos. Caso os países tenham, também, o mesmo número de pontos fornecidos aos vencedores de seus torneios, o critério de desempate será a premiação fornecida. No exemplo fornecido, U.S.A. lideram este ranking com CINCO competições; e

d) a lista com a premiação obtida pelos jogadores, em ordem decrescente, juntamente com os nomes dos torneios. Se existirem dois jogadores com a mesma premiação, o critério de desempate será o número de pontos obtidos. Caso os competidores tenham, também, o mesmo número de pontos, o critério de desempate será o número de vitórias.

Por questões de desempenho, seu programa deve ler o arquivo de entrada SOMENTE uma vez. Logo após a leitura, as estatísticas devem ser apresentadas.

RESPOSTA:

```
import java.io.*;

class Torneio{
  String nome;
  int pontos;
  float premio;
  Torneio prox;

  Torneio(String n, int ptos, float p){
    pontos = ptos;
    premio = p;
    nome = n;
    prox = null;
  }

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


class Jogador{
  String nome_jog;
  int num_vit_jog, pontos_jog;
  float premio_jog;
  Torneio prim;
  Jogador prox;
```

```
  Jogador(String n){
    nome_jog = n;
    num_vit_jog = pontos_jog = 0;
    premio_jog = 0.0F;
    prim = null;
    prox = null;
  }

  void insere_torneio(String n, int ptos, float p){
    num_vit_jog++;
    pontos_jog += ptos;
    premio_jog += p;
    Torneio q = new Torneio(n, ptos, p);
    q.prox = prim;
    prim = q;
  }

  public String toString(){
     String resp = nome_jog + " " + num_vit_jog + " " + pontos_jog + " "
+ premio_jog + ":\n";
    Torneio p = prim;
    while(p != null){
      resp += p.toString();
      p = p.prox;
    }
    return resp;
  }
}


class ListaJogador{
  Jogador prim;

  ListaJogador(){
    prim = null;
  }

  Jogador pertence(String n){
    Jogador p = prim;
    while((p != null) && (!n.equals(p.nome_jog))) p = p.prox;
    return p;
  }

  void insere(String nj, String n, int ptos, float p){
    Jogador q = pertence(nj);
    if(q == null){
      q = new Jogador(nj);
      q.prox = prim;
      prim = q;
    }
    q.insere_torneio (n, ptos, p);
  }

  public String toString(){
    Jogador p = prim;
    String resp = "";
    while(p != null){
```

```
      resp += p.toString();
      resp += "\n";
      p = p.prox;
    }
    return resp;
  }
}


class Pais{
  String nome_pais;
  int num_total, total_ptos;
  float total_premio;
  Torneio prim;
  Pais prox;

  Pais(String np){
    nome_pais = np;
    num_total = total_ptos = 0;
    total_premio = 0.0F;
    prim = null;
    prox = null;
  }

  void insere_torneio(String n, int ptos, float p){
    num_total++;
    total_ptos += ptos;
    total_premio += p;
    Torneio q = new Torneio(n, ptos, p);
    q.prox = prim;
    prim = q;
  }

  public String toString(){
     String resp = nome_pais + " " + num_total + " " + total_ptos + " "
+ total_premio + ":\n";
    Torneio p = prim;
    while(p != null){
      resp += p.toString();
      p = p.prox;
    }
    return resp;
  }
}


class ListaPais{
  Pais prim;

  ListaPais(){
    prim = null;
  }

  Pais pertence(String n){
    Pais p = prim;
    while((p != null) && (!n.equals(p.nome_pais))) p = p.prox;
    return p;
```

```
  }

  void insere(String np, String n, int ptos, float p){
    Pais q = pertence(np);
    if(q == null){
      q = new Pais(np);
      q.prox = prim;
      prim = q;
    }
    q.insere_torneio (n, ptos, p);
  }

  public String toString(){
    Pais p = prim;
    String resp = "";
    while(p != null){
      resp += p.toString();
      resp += "\n";
      p = p.prox;
    }
    return resp;
  }
}


public class AD2_2012_2{
  public static void main(String[] args) throws IOException{
    BufferedReader in = new BufferedReader(new FileReader(args[0]));
    try {
      ListaPais lp = new ListaPais();
      ListaJogador lj = new ListaJogador();
      String s, vs[];
      while((s = in.readLine()) != null) {
      vs = s.split("/");
                    lp.insere(vs[2],   vs[1],   Integer.parseInt(vs[3]),
Float.parseFloat(vs[4]));
                    lj.insere(vs[5],   vs[1],   Integer.parseInt(vs[3]),
Float.parseFloat(vs[4]));
      }
      in.close();
      System.out.println(lp);
      System.out.println("Depois da ordenacao pelo Item C:");
      Ordena_Item_C(lp);
      System.out.println(lp);
      System.out.println(lj);
      System.out.println("Depois da ordenacao pelo Item A:");
      Ordena_Item_A(lj);
      System.out.println(lj);
      System.out.println("Depois da ordenacao pelo Item B:");
      Ordena_Item_B(lj);
      System.out.println(lj);
      System.out.println("Depois da ordenacao pelo Item D:");
      Ordena_Item_D(lj);
      System.out.println(lj);
    }
    catch (Exception e){
      System.out.println("Excecao\n");
```

```
    }
  }

  public static void Ordena_Item_C(ListaPais lp){
    Pais p, q, maior;
    for(p = lp.prim; p != null; p = p.prox){
      maior = p;
      for(q = p.prox; q != null; q = q.prox){
        if(maior.num_total < q.num_total)
          maior = q;
         else if((maior.num_total == q.num_total) && (maior.total_ptos <
q.total_ptos))
          maior = q;
         else if ((maior.num_total == q.num_total) && (maior.total_ptos
== q.total_ptos) && (maior.total_premio < q.total_premio))
          maior = q;
      }

      if(maior != p){
        String n = p.nome_pais;
        p.nome_pais = maior.nome_pais;
        maior.nome_pais = n;

        int aux = p.num_total;
        p.num_total = maior.num_total;
        maior.num_total = aux;

        aux = p.total_ptos;
        p.total_ptos = maior.total_ptos;
        maior.total_ptos = aux;

        float pr = p.total_premio;
        p.total_premio = maior.total_premio;
        maior.total_premio = pr;

        Torneio t = p.prim;
        p.prim = maior.prim;
        maior.prim = t;
      }
    }
  }

  public static void Ordena_Item_A(ListaJogador lj){
    Jogador p, q, maior;
    for(p = lj.prim; p != null; p = p.prox){
      maior = p;
      for(q = p.prox; q != null; q = q.prox){
        if(maior.num_vit_jog < q.num_vit_jog)
          maior = q;
                  else  if((maior.num_vit_jog  ==  q.num_vit_jog)  &&
(maior.pontos_jog < q.pontos_jog))
          maior = q;
                 else  if  ((maior.num_vit_jog  ==  q.num_vit_jog)  &&
(maior.pontos_jog == q.pontos_jog) && (maior.premio_jog < q.premio_jog))
          maior = q;
      }
```

```
      if(maior != p){
        String n = p.nome_jog;
        p.nome_jog = maior.nome_jog;
        maior.nome_jog = n;

        int aux = p.num_vit_jog;
        p.num_vit_jog = maior.num_vit_jog;
        maior.num_vit_jog = aux;

        aux = p.pontos_jog;
        p.pontos_jog = maior.pontos_jog;
        maior.pontos_jog = aux;

        float pr = p.premio_jog;
        p.premio_jog = maior.premio_jog;
        maior.premio_jog = pr;

        Torneio t = p.prim;
        p.prim = maior.prim;
        maior.prim = t;
      }
    }
  }

  public static void Ordena_Item_B(ListaJogador lj){
    Jogador p, q, maior;
    for(p = lj.prim; p != null; p = p.prox){
      maior = p;
      for(q = p.prox; q != null; q = q.prox){
        if(maior.pontos_jog < q.pontos_jog)
          maior = q;
                    else   if((maior.pontos_jog   ==   q.pontos_jog)   &&
(maior.num_vit_jog < q.num_vit_jog))
          maior = q;
                   else   if   ((maior.pontos_jog   ==   q.pontos_jog)   &&
(maior.num_vit_jog    ==    q.num_vit_jog)    &&    (maior.premio_jog    <
q.premio_jog))
          maior = q;
      }

      if(maior != p){
        String n = p.nome_jog;
        p.nome_jog = maior.nome_jog;
        maior.nome_jog = n;

        int aux = p.num_vit_jog;
        p.num_vit_jog = maior.num_vit_jog;
        maior.num_vit_jog = aux;

        aux = p.pontos_jog;
        p.pontos_jog = maior.pontos_jog;
        maior.pontos_jog = aux;

        float pr = p.premio_jog;
        p.premio_jog = maior.premio_jog;
        maior.premio_jog = pr;
```

```
        Torneio t = p.prim;
        p.prim = maior.prim;
        maior.prim = t;
      }
    }
  }

  public static void Ordena_Item_D(ListaJogador lj){
    Jogador p, q, maior;
    for(p = lj.prim; p != null; p = p.prox){
      maior = p;
      for(q = p.prox; q != null; q = q.prox){
        if(maior.premio_jog < q.premio_jog)
          maior = q;
         else if((maior.premio_jog == q.premio_jog) && (maior.pontos_jog
< q.pontos_jog))
          maior = q;
                  else if ((maior.premio_jog == q.premio_jog) &&
(maior.pontos_jog    ==    q.pontos_jog)    &&    (maior.num_vit_jog    <
q.num_vit_jog))
          maior = q;
      }

      if(maior != p){
        String n = p.nome_jog;
        p.nome_jog = maior.nome_jog;
        maior.nome_jog = n;

        int aux = p.num_vit_jog;
        p.num_vit_jog = maior.num_vit_jog;
        maior.num_vit_jog = aux;

        aux = p.pontos_jog;
        p.pontos_jog = maior.pontos_jog;
        maior.pontos_jog = aux;

        float pr = p.premio_jog;
        p.premio_jog = maior.premio_jog;
        maior.premio_jog = pr;

        Torneio t = p.prim;
        p.prim = maior.prim;
        maior.prim = t;
      }
    }
  }
}
```