



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Programação III**

**AP3 2º semestre de 2013.**

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) (5.0 pontos)**

Suponha que num arquivo texto estejam armazenados os resultados de todos os jogos de futebol já disputados de algumas copas do mundo, e que, em cada uma das copas guardadas nesse arquivo, tenham sido jogadas 64 partidas. Os dados salvos no arquivo em questão seguem o padrão abaixo:

**ANO DA COPA: LOCAL**

**Time A/Time B/Quantidade Gols Time A/Quantidade Gols Time B**

Isto é, para cada copa, a linha “**Time A/Time B/Quantidade Gols Time A/Quantidade Gols Time B**” é repetida 64 vezes. Suponha, também, que o jogo final seja sempre a última linha de cada copa. Escreva um programa que leia esse arquivo de dados de copas do mundo (passado como parâmetro de entrada) **UMA ÚNICA VEZ** e que seja capaz de:

- (a) imprimir todas as seleções campeãs, informando também o local onde foi realizada a copa e o ano da disputa;
- (b) informar, para cada copa, qual(is) foi(foram) a(s) seleção(seleções) que mais perdeu(perderam). Esse dado é obtido dividindo-se o número de jogos em que uma seleção perdeu pelo total de jogos que ela disputou;
- (c) semelhantemente, apresentar, para cada copa, qual(is) foi(foram) a(s) seleção(seleções) que mais empatou(empataram); e
- (d) de modo análogo, mostrar, para cada copa, qual(is) foi(foram) a(s) seleção(seleções) que mais ganhou(ganharam).

## RESPOSTA:

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

class Selecao{
    String nome;
    int num_vit, num_emp, num_der;
    float med_vit, med_emp, med_der;

    Selecao(String n){
        nome = n;
        num_vit = num_emp = num_der = 0;
    }

    void incEmp() { num_emp++; }
    void incVit() { num_vit++; }
    void incDer() { num_der++; }

    void calcMedias(){
        float total = num_vit + num_emp + num_der;
        med_vit = num_vit / total;
        med_emp = num_emp / total;
        med_der = num_der / total;
    }

    public String toString(){
        return nome + "\tVit: " + num_vit + "\tEmp: " + num_emp + "\tDer: "
+ num_der;
    }
}

public class Copa{
    public static void main(String[] args) throws Exception{
        BufferedReader in = new BufferedReader(new FileReader(args[0]));
        String campea = "", local, linha;
        int ano;
        ArrayList<Selecao> l = new ArrayList<Selecao>();
        try{
            linha = in.readLine();
            String partes[];
            while(linha != null){
                partes = linha.split(":");
                ano = Integer.parseInt(partes[0]);
                local = partes[1];
                for(int i = 0; i < 64; i++){
                    linha = in.readLine();
                    partes = linha.split("/");
                    int gols1 = Integer.parseInt(partes[2]);
                    int gols2 = Integer.parseInt(partes[3]);

                    if(i == 63)
                        if(gols1 > gols2) campea = partes[0];
                        else campea = partes[1];
                }
            }
        }
    }
}
```

```

        boolean existe = false;
        for(Selecao atual: l){
            if(atual.nome.equals(partes[0])){
                if(gols1 > gols2) atual.incVit();
                else if(gols1 < gols2) atual.incDer();
                else atual.incEmp();
                existe = true;
                break;
            }
        }
        if(!existe){
            Selecao atual = new Selecao(partes[0]);
            if(gols1 > gols2) atual.incVit();
            else if(gols1 < gols2) atual.incDer();
            else atual.incEmp();
            l.add(atual);
        }

        existe = false;
        for(Selecao atual: l){
            if(atual.nome.equals(partes[1])){
                if(gols1 < gols2) atual.incVit();
                else if(gols1 > gols2) atual.incDer();
                else atual.incEmp();
                existe = true;
                break;
            }
        }
        if(!existe){
            Selecao atual = new Selecao(partes[1]);
            if(gols1 < gols2) atual.incVit();
            else if(gols1 > gols2) atual.incDer();
            else atual.incEmp();
            l.add(atual);
        }
    }
    //letra (a)
    System.out.println("\n" + ano + " " + local + " " + campea);

    for(Selecao atual : l) atual.calcMedias();

    //letra (d)
    System.out.println("Maior vencedor:");
    imprimeMaiorVencedor(l);

    //letra (b)
    System.out.println("Maior perdedor:");
    imprimeMaiorPerdedor(l);

    //letra (c)
    System.out.println("Maior empatador:");
    imprimeMaiorEmpatador(l);

    //proxima copa
    l = new ArrayList<Selecao>();
    linha = in.readLine();
}

```

```

        } catch (Exception e) {
            System.out.println("Erro");
        } finally {
            in.close();
        }
    }

    static void imprimeMaiorVencedor(ArrayList<Selecao> l){
        Selecao maior = null;
        for(Selecao atual : l)
            if((maior == null) || (atual.med_vit > maior.med_vit))
                maior = atual;
        for(Selecao atual : l)
            if(atual.med_vit == maior.med_vit)
                System.out.println(atual);
    }

    static void imprimeMaiorPerdedor(ArrayList<Selecao> l){
        Selecao maior = null;
        for(Selecao atual : l)
            if((maior == null) || (atual.med_der > maior.med_der))
                maior = atual;
        for(Selecao atual : l)
            if(atual.med_der == maior.med_der)
                System.out.println(atual);
    }

    static void imprimeMaiorEmpatador(ArrayList<Selecao> l){
        Selecao maior = null;
        for(Selecao atual : l)
            if((maior == null) || (atual.med_emp > maior.med_emp))
                maior = atual;
        for(Selecao atual : l)
            if(atual.med_emp == maior.med_emp)
                System.out.println(atual);
    }
}

```

## Questão 2) (5.0 pontos)

Polinômios são sequências de termos, chamados de monômios, da forma  $ax^n$ , onde  $a$  é um coeficiente,  $x$  é sua base e  $n$  é um expoente. Um exemplo de polinômio seria “ $3x^3 + 2x^1$ ”. Para manipular polinômios, foi criada uma classe, a qual segue abaixo. Em linhas gerais, o construtor desta classe recebe um polinômio como uma string no formato “ $3x^3 + 2x^1$ ” e armazena estes numa lista de termos.

```

import java.util.*;
import java.util.regex.*;

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

    public Polinomio(String polinomio) { // P. ex.: "3x3 + 2x1"
        termos = new ArrayList<Termo>();
    }
}

```

```

        String regexp = "(\\d)*x(\\d)*"; // expressão regular para pegar padrão axn no
        parâmetro polinomio dado no construtor

        Pattern pattern = Pattern.compile(regexp);
        Matcher matcher = pattern.matcher(polinomio);
        while (matcher.find()) {
            String t = matcher.group(); // Strings da forma axn, coeficiente e
            expoente, respectivamente
            Termo termo = new Termo(t);
            termos.add(termo);
        }
    }

    public void imprimePolinomio() {
        for (Termo t: termos) {
            System.out.print(t + " ");
        }
    }
}

public class AP3_2013_2_Q2 {
    public static void main(String[] args) {
        Polinomio p1 = new Polinomio("3x3 + 2x1");
        p1.imprimePolinomio();
        System.out.println("\nValor do polinomio para x igual a 2: " +
        p1.calculaPolinomio(2));
        System.out.println("Polinomio completo: " + p1.completo());
    }
}

```

- α) Analise o código e defina a classe Termo para que esta classe Polinomio funcione.
- β) No método main() é chamado o método calculaPolinomio, o qual atribui um valor ao x do polinômio e retorna o resultado. Implemente esta função.
- γ) Ainda no método main() é chamada um método completo(), o qual verifica se um polinômio é completo, ou seja, se possui todos os expoentes, de 0 até um dado valor. O exemplo dado ("3x3 + 2x1") não é completo, pois faltam os termos de expoente 2 e 0.

## GABARITO:

```

/* !!! Letra a) !!! */
class Termo {
    int coeficiente;
    int expoente;

    public Termo(int c, int e) {
        coeficiente = c;
        expoente = e;
    }

    public Termo(String t) {
        String valores[] = t.trim().split("x");
        coeficiente = Integer.valueOf(valores[0]);
        expoente = Integer.valueOf(valores[1]);
    }
}

```

```

    public int getCoeficiente() {
        return coeficiente;
    }

    public int getExpoente() {
        return expoente;
    }

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

class Polinomio {
    private List<Termo> termos;

    /* !!! Restante da classe definida no enunciado !!! */

    /* !!! Letra b) !!! */
    public double calculaPolinomio (double x) {
        double soma = 0;
        for (Termo t: termos) {
            soma = soma + Math.pow(x, t.getExpoente())*t.getCoeficiente();
        }
        return soma;
    }

    public boolean contemExpoente (int e) {
        for (Termo t: termos) {
            if (t.getExpoente() == e)
                return true;
        }
        return false;
    }

    /* !!! Letra c) !!! */
    public boolean completo() {
        int tamanho = termos.size();
        for (int i=0; i<tamanho; i++)
            if (! this.contemExpoente(i))
                return false;
        return true;
    }
}

```