



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação III

AP2 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) (3 pontos)

Escreva um programa que leia um número N , dois arquivos contendo números inteiros, um por linha, e teste se N pertence aos dois arquivos. Se N não pertencer a um dos dois arquivos, os arquivos permanecerão inalterados. Se N pertencer aos dois, os arquivos originais serão modificados da seguinte forma:

A) Se o primeiro arquivo é composto de:

1
2
3
4

B) O segundo arquivo é formado por:

10
2
5

C) E $N = 2$, então o primeiro arquivo será composto de:

1
2
5

D) E o segundo arquivo será:

10
2
3
4

Isto é, Se os dois arquivos possuem N , o primeiro arquivo ficará com todos os elementos do segundo que aparecem depois de N , e o segundo arquivo ficará com todos os elemen-

tos do primeiro que aparecem após N. Um exemplo de uso desse programa seria `java arq1.txt arq2.txt <N>`.

RESPOSTA:

```
import java.io.*;

class No{
    int info;
    No prox;

    No (int info){ this.info = info; prox = null; }

    public String toString(){ return String.valueOf(info) + "\n";}
}

class Lista{
    No prim;

    Lista(){ prim = null; }

    void ins_fim (int info){
        No novo = new No(info);
        if(prim == null) prim = novo;
        else{
            No p = prim;
            while(p.prox != null) p = p.prox;
            p.prox = novo;
        }
    }

    No Busca(int info){
        No p = prim;
        while((p != null) && (p.info != info)) p = p.prox;
        return p;
    }

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

public class Q1{
    public static void main(String[] args) throws IOException {
        BufferedReader in1 = new BufferedReader(new FileReader(args[0]));
        String s;
        Lista l1 = new Lista();
```

```

    try{
        while ((s = in1.readLine()) != null)
            l1.ins_fim(Integer.parseInt(s));
    }catch (Exception e){
        System.out.println("Excecao leitura 1\n");
    }

    BufferedReader in2 = new BufferedReader(new FileReader(args[1]));
    Lista l2 = new Lista();
    try{
        while ((s = in2.readLine()) != null)
            l2.ins_fim(Integer.parseInt(s));
    }catch (Exception e){
        System.out.println("Excecao leitura 2\n");
    }

    int elem = Integer.parseInt(args[2]);

    if((l1.Busca(elem) != null) && (l2.Busca(elem) != null)){
        Troca(l1, l2, elem);
        BufferedWriter out1 = new BufferedWriter(new
FileWriter(args[0]));
        BufferedWriter out2 = new BufferedWriter(new
FileWriter(args[1]));
        try{
            out1.write(l1.toString());
            out2.write(l2.toString());
        }catch (Exception e){
            System.out.println("Excecao de escrita\n");
        }finally{
            out1.close();
            out2.close();
        }
    }
}

public static void Troca(Lista l1, Lista l2, int N){
    No p = l1.prim;
    while(p.info != N) p = p.prox;
    No q = l2.prim;
    while(q.info != N) q = q.prox;

    No temp = p.prox;
    p.prox = q.prox;
    q.prox = temp;
}
}

```

Questão 2) (3 pontos)

Escreva uma classe Conjunto para ser usada na seguinte classe de teste:

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

```
public class Teste{
    public static void main(String[] args){
```

```

int i, n1, n2;

Scanner sc = new Scanner(System.in);
n1 = Integer.parseInt(sc.nextLine());
Conj p = new Conj(n1);
for(i = 0; i < n1; i++)
    p.modElem(i, Float.parseFloat(sc.nextLine()));
System.out.println(p);

n2 = Integer.parseInt(sc.nextLine());
Conj q = new Conj(n2);
for(i = 0; i < n2; i++)
    q.modElem(i, Float.parseFloat(sc.nextLine()));
System.out.println(q);

Conj uniao = p.uniao(q);
System.out.println(uniao);

Conj inter = p.intersecao(q);
System.out.println(inter);

Conj dif1 = p.dif(q); //dif1 = p - q
System.out.println(dif1);

Conj dif2 = q.dif(p); //dif2 = q - p
System.out.println(dif2);

Conj difsim = dif1.uniao(dif2); //difsim = (p - q) uniao (q - p)
System.out.println(difsim);
}
}

```

RESPOSTA:

```

class Conj{
    int tam;
    float elem[];

    Conj(int t){ tam = t; elem = new float[tam]; }

    void modElem(int g, float valor){ elem[g] = valor; }

    boolean pertence(float x){
        for(int i = 0; i < tam; i++)
            if(elem[i] == x) return true;
        return false;
    }

    public String toString(){
        String resp = "{";
        for(int i = 0; i < tam - 1; i++) resp = resp + elem[i] + ", ";
        resp = resp + elem[tam - 1] + "}\n";
        return resp;
    }

    Conj uniao(Conj c){
        Conj resp = new Conj (tam + c.tam);
    }
}

```

```

    int i;

    for(i = 0; i < tam; i++) resp.modElem(i, elem[i]);
    resp.tam = tam;
    for(i = 0; i < c.tam; i++)
        if(resp.pertence(c.elem[i]) == false)
            resp.elem[resp.tam++] = c.elem[i];
    return resp;
}

Conj intersecao(Conj c){
    Conj resp;
    if(c.tam < tam) resp = new Conj (c.tam);
    else resp = new Conj (tam);
    resp.tam = 0;
    for(int i = 0; i < c.tam; i++)
        if(pertence(c.elem[i]))
            resp.elem[resp.tam++] = c.elem[i];
    return resp;
}

Conj dif(Conj c){
    Conj resp = new Conj (tam);
    resp.tam = 0;
    for(int i = 0; i < c.tam; i++)
        if(pertence(c.elem[i]) == false)
            resp.elem[resp.tam++] = c.elem[i];
    return resp;
}
}

```

Questão 3) (4 pontos)

Suponha um sistema de controle de pedidos num restaurante. Neste sistema, um cliente pode ser identificado pelo número de sua mesa ou pelo número do seu telefone, no caso do pedido ser entregue em domicílio. Um pedido contém o código do prato, o preço deste e o cliente correspondente. Os pedidos a serem entregues em domicílio tem acréscimo de 10% (Para acrescentar um valor de x%, basta multiplicar este valor por $(1 + x/100)$). Baseado no código do método main(), informado abaixo, apresente uma implementação para as classes utilizadas neste código.

```

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

public class AP2_2011_2_Q3 {
    public static void main(String[] args) {
        List<Pedido> pedidos = new ArrayList<Pedido>();
        pedidos.add(new PedidoLocal(1, 50.0, 1)); // Código do
prato, valor e código do cliente
        pedidos.add(new PedidoLocal(2, 20.0, 1));
        pedidos.add(new PedidoLocal(3, 100.0, 1));
        pedidos.add(new PedidoViagem(1, 50.0, "9999-9999")); //
Código do prato, valor e telefone do cliente
    }
}

```

```

        pedidos.add(new PedidoViagem(2, 20.0, "8888-8888"));
        double soma = 0;
        for (Pedido p : pedidos) {
            soma = soma + p.getPrecoPrato();
        }
        System.out.println("A soma é: " + soma);
    }
}

```

RESPOSTA:

```

class Pedido {
    int codPrato;
    double precoPrato;
    public Pedido(int codPrato, double precoPrato) {
        this.codPrato = codPrato;
        this.precoPrato = precoPrato;
    }

    public double getPrecoPrato() {
        return precoPrato;
    }
}

class PedidoViagem extends Pedido {
    String telefone;

    public PedidoViagem(int codPrato, double precoPrato, String tel) {
        super(codPrato, precoPrato);
        this.telefone = tel;
    }

    public double getPrecoPrato() {
        return precoPrato*1.1;
    }
}

class PedidoLocal extends Pedido {
    int mesa;

    public PedidoLocal(int codPrato, double precoPrato, int mesa) {
        super(codPrato, precoPrato);
        this.mesa = mesa;
    }
}

```