



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação Orientada a Objetos

AP2 2º semestre de 2018.

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) (4.0 pontos)

Considere a interface *Colecao* definida a seguir e seu uso no método *main()*.

```
interface Colecao {
    void adiciona (Object e);
    void remove (Object e);
    void lista();
}

public class AP2_2018_2_Q1 {
    public static void main(String[] args) {
        Colecao colecao;
        colecao = new Vetor();
        String a = "A";
        String b = "B";
        String c = "C";
        colecao.adiciona(a);
        colecao.adiciona(b);
        colecao.adiciona(c);
        colecao.remove(b);
        colecao.lista();
    }
}
```

Defina *Vetor* (2ª linha da *main*) usando um **vetor** como sua representação interna para armazenar valores. O método *lista()* imprime os valores na tela. Os métodos *adiciona()* e *remove()* funcionam como os seus nomes sugerem. Obs.: (1) Crie um vetor que permita alocação de até 1000 objetos; (2) Não permita o desperdício de espaço no vetor.

RESPOSTA:

```
class Vetor implements Colecao {
    Object objetos[];
    int contador;

    public Vetor() {
        objetos = new Object[1000];
        contador = 0;
    }

    public void adiciona(Object e) {
        objetos[contador] = e;
        contador++;
    }

    public void remove(Object e) {
        boolean achou = false;
        for (int i=0; i<contador; i++)
            if (!achou && objetos[i] == e) {
                achou = true;
                objetos[i] = objetos[i+1];
            }
            else
                // Move objetos para eliminar buraco
                if (achou)
                    objetos[i] = objetos[i+1];

        if (achou)
            contador--;
    }

    public void lista() {
        for (int i=0; i<contador; i++)
            System.out.println(objetos[i]);
    }
}
```

Questão 2) (3.0 pontos)

Escreva um programa em Java que receba um número n ($n > 0$), n arquivos de inteiros ordenados em ordem crescente, um número em cada linha, e o nome do arquivo de saída, e retorne, no arquivo de saída todos os números dos demais arquivos ordenados de maneira crescente, um número por linha.

LEMBRE-SE: SEU PROGRAMA SÓ PODE LER O ARQUIVO DE ENTRADA UMA ÚNICA VEZ.

RESPOSTA:

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

public class Q2_AP2_POO_2018_2{
    public static void main(String[] args) throws IOException{
        int n = Integer.parseInt(args[0]), i;
        if(n <= 0) return;
        String narq[] = new String[n];
```

```

        for(i = 1; i <= n; i++) narq[i - 1] = args[i];
        Merge(n, narq, args[n + 1]);
    }

    static void Merge(int n, String[] narq, String saida) throws
IOException{
        BufferedReader in[] = new BufferedReader[n];
        for(int i = 0; i < n; i++)
            in[i] = new BufferedReader(new FileReader(narq[i]));
        BufferedWriter out;
        out = new BufferedWriter(new FileWriter(saida));
        String resp = "", s;
        int vet[] = new int[n];
        try {
            for(int i = 0; i < n; i++){
                s = in[i].readLine();
                if(s != null) vet[i] = Integer.parseInt(s);
                else vet[i] = Integer.MAX_VALUE;
            }
            int ind = menorInd(vet);
            while(ind != Integer.MAX_VALUE){
                resp = resp + vet[ind] + "\n";
                s = in[ind].readLine();
                if(s != null) vet[ind] = Integer.parseInt(s);
                else vet[ind] = Integer.MAX_VALUE;
                ind = menorInd(vet);
            }
            for(int i = 0; i < n; i++) in[i].close();
            out.write(resp);
            out.close();
        }
        catch (Exception e){
            System.out.println("Excecao\n");
        }
    }

    static int menorInd(int[] vet){
        int resp = Integer.MAX_VALUE, menor = Integer.MAX_VALUE;
        for(int i = 0; i < vet.length; i++)
            if(menor > vet[i]){
                menor = vet[i];
                resp = i;
            }
        return resp;
    }
}

```

Questão 3) (3.0 pontos)

Nesta questão sua tarefa será ler vários números de um arquivo de entrada, um número em cada linha, cujo o nome é passado como parâmetro de entrada, e em seguida informar quantas vezes cada número aparece nesse arquivo de entrada, em ordem decrescente de frequência. Se dois ou mais números tiverem o mesmo valor de frequência, a ordenação será em ordem crescente do número lido.

LEMBRE-SE: SEU PROGRAMA SÓ PODE LER O ARQUIVO DE ENTRADA UMA ÚNICA VEZ.

RESPOSTA:

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

class No {
    int valor, ocorr;
    No prox;

    public No(int valor) {
        this.valor = valor;
        ocorr = 1;
        this.prox = null;
    }

    public String toString(){ return valor + " (" + ocorr + ")"; }
}

class lista{
    No prim;

    lista(){ prim = null; }

    void insere(int valor){
        No novo = busca(valor);
        if(novo == null){
            novo = new No(valor);
            novo.prox = prim;
            prim = novo;
        }
        else novo.ocorr++;
    }

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

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

public class Q3_AP2_POO_2018_2{
```

```

public static void main(String[] args) throws IOException{
    BufferedReader in;
    in = new BufferedReader(new FileReader(args[0]));
    BufferedWriter out;
    out = new BufferedWriter(new FileWriter("saida-" + args[0]));
    try {
        lista l = new lista();
        String s = in.readLine();
        while(s != null){
            l.insere(Integer.parseInt(s));
            s = in.readLine();
        }
        in.close();
        Ordena(l);
        out.write(l.toString());
        out.close();
    }
    catch (Exception e){
        System.out.println("Excecao\n");
    }
}

static void Ordena(lista l){
    for(No p = l.prim; p != null; p = p.prox){
        No maior = p;
        for(No q = p.prox; q != null; q = q.prox)
            if((maior.ocorr < q.ocorr) || ((maior.ocorr == q.ocorr) &&
(maior.valor > q.valor))) maior = q;
        if(maior != p){
            int temp = maior.valor;
            maior.valor = p.valor;
            p.valor = temp;
            temp = maior.ocorr;
            maior.ocorr = p.ocorr;
            p.ocorr = temp;
        }
    }
}
}

```