



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Programação Orientada a Objetos**

**AP2 1º semestre de 2019.**

**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 o programa principal abaixo que manipula dicionários:

```
public class AP2_2019_1_Q1 {  
    public static void main(String[] args) {  
        Tradutor dic = new Tradutor();  
        dic.adiciona(new Entrada("carro", "car", "pt", "en"));  
        dic.adiciona(new Entrada("table", "mesa", "en", "pt"));  
        dic.adiciona(new Entrada("voiture", "carro", "fr", "pt"));  
        dic.adiciona(new Entrada("dog", "cachorro", "en", "pt"));  
        dic.adiciona(new Entrada("pao", "pane", "pt", "it"));  
        System.out.println(dic.geraEntradas("pt", "en"));  
    }  
}
```

- a) (1.0) Defina a classe Entrada, cujo construtor recebe como parâmetro a palavra no idioma de origem, a palavra no idioma traduzido, o idioma de origem e o idioma de tradução. Por exemplo, “table” e “mesa” como palavras e “en” e “pt” como respectivos idiomas.
- b) (3.0) Defina a classe Tradutor, a qual possuirá os métodos adiciona() e geraEntradas(). Na implementação do método geraEntradas() são fornecidos os idiomas de origem e destino. Este método deve retornar todas as entradas existentes no tradutor para os idiomas passados. De forma a produzir um resultado completo, entradas inseridas de forma invertida, destino e origem, também devem ser listadas. Por exemplo, para os idiomas “pt” e “en” (última linha da main()), a saída segue abaixo:

```
pt -> en
carro -> car
mesa -> table
cachorro -> dog
```

## RESPOSTA:

```
class Entrada {
    private String fonte;
    private String traducao;
    private String idioma_fonte;
    private String idioma_traducao;

    public Entrada(String fonte, String traducao, String idioma_fonte, String idioma_traducao) {
        this.fonte = fonte;
        this.traducao = traducao;
        this.idioma_fonte = idioma_fonte;
        this.idioma_traducao = idioma_traducao;
    }

    public String getFonte () {
        return this.fonte;
    }

    public String getTraducao () {
        return this.traducao;
    }

    public String getIdioma_fonte() {
        return idioma_fonte;
    }

    public String getIdioma_traducao() {
        return idioma_traducao;
    }
}

class Tradutor {
    private Entrada entradas[];
    private int tam;
    private static int MAX = 1000;

    public Tradutor() {
        entradas = new Entrada [MAX];
        tam = 0;
    }

    public void adiciona (Entrada e) {
        if (tam < MAX-1) {
            entradas[tam] = e;
            tam++;
        }
    }

    public String geraEntradas(String origem, String destino) {
        String saida = origem + " -> " + destino + "\n";
        for (int i=0; i<tam; i++)
            if ((entradas[i].getIdioma_fonte() == origem) && (entradas[i].getIdioma_traducao() ==
destino))
                saida += entradas[i].getFonte() + " -> " + entradas[i].getTraducao() + "\n";
            else
                if ((entradas[i].getIdioma_fonte() == destino) &&
(entradas[i].getIdioma_traducao() == origem))
                    saida += entradas[i].getTraducao() + " -> " + entradas[i].getFonte()
+ "\n";
        return saida;
    }
}
```

### Questão 2) (3.0 pontos)

Considere um programa em Java que ajude pessoas a estudar como calcular a área de figuras geométricas fornecidas num arquivo de entrada. O programa deve ser capaz de calcular a área de diferentes figuras, armazenadas em uma única coleção de figuras, ordenando essa coleção em ordem decrescente de área. Cada figura pode pertencer a um dos três tipos: triângulos, retângulos, quadrados. Um triângulo tem como atributos a sua

base b e a sua altura h. Um retângulo é descrito por dois lados a e b e um quadrado por seu único tamanho de lado l. Utilizando polimorfismo e os demais conceitos de orientação a objetos, implemente um programa em Java que calcule as áreas de uma coleção de figuras e ordene essas áreas em ordem crescente. Por exemplo, se o arquivo de entrada for:

```
R 1 2
Q 2
T 3 3
```

Seu programa deve responder:

```
2.0   R 1 2
4.0   Q 2
4.5   T 3 3
```

**LEMBRE-SE: SEU PROGRAMA SÓ PODE LER O ARQUIVO DE ENTRADA UMA ÚNICA VEZ, E ELE DEVE EXECUTAR COM QUAISQUER DADOS INFORMADOS COMO PARÂMETROS DE ENTRADA. SE O SEU PROGRAMA RESOLVER SOMENTE O EXEMPLO SUPRACITADO, SUA QUESTÃO SERÁ TOTALMENTE DESCONTADA.**

**RESPOSTA:**

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

```
class No {
    Figura fig;
    No prox;

    public No(Figura fig) {
        this.fig = fig;
        this.prox = null;
    }

    public String toString(){ return fig.toString(); }
}
```

```
class lista{
    No prim;

    lista(){ prim = null; }

    void insere(Figura fig){
        No novo = new No(fig);
        novo.prox = prim;
        prim = novo;
    }

    public String toString(){
        String resp = "";
        No p = prim;
        while (p != null){
```

```

        resp += p.toString();
        p = p.prox;
    }
    return resp;
}
}

```

```

abstract class Figura {
    abstract float calculaArea();
}

```

```

class Quadrado extends Figura {
    private int l;

    public Quadrado(int l){ this.l = l; }

    public String toString() {
        return calculaArea() + "\tQ " + l + "\n";
    }

    float calculaArea(){ return l*l; }
}

```

```

class Retangulo extends Figura {
    private int a, b;

    public Retangulo(int a, int b) { this.a = a; this.b = b; }

    public String toString() {
        return calculaArea() + "\tR " + a + " " + b + "\n";
    }

    float calculaArea() { return a*b; }
}

```

```

class Triangulo extends Figura {
    private int b, h;

    public Triangulo(int b, int h){ this.b = b; this.h = h; }

    public String toString() {
        return calculaArea() + "\tT " + b + " " + h + "\n";
    }

    float calculaArea() { return (b*h)/2.0f; }
}

```

```

public class Q2_AP2_POO_2019_1{
    public static void main(String[] args) throws IOException{

```

```

BufferedReader in;
in = new BufferedReader(new FileReader(args[0]));
try {
    lista l = new lista();
    String s = in.readLine();
    while(s != null){
        String vs[] = s.split(" ");
        Figura f;
        if(vs[0].equals("Q"))
            f = new Quadrado(Integer.parseInt(vs[1]));
        else if(vs[0].equals("R"))
            f = new Retangulo(Integer.parseInt(vs[1]),
Integer.parseInt(vs[2]));
        else
            f = new Triangulo(Integer.parseInt(vs[1]),
Integer.parseInt(vs[2]));
        l.insere(f);
        s = in.readLine();
    }
    in.close();
    Ordena(l);
    System.out.print(l);
}
catch (Exception e){
    System.out.println("Excecao\n");
}
}

static void Ordena(lista l){
    for(No p = l.prim; p != null; p = p.prox){
        No menor = p;
        for(No q = p.prox; q != null; q = q.prox)
            if(menor.fig.calculaArea() > q.fig.calculaArea()) menor = q;
        if(menor != p){
            Figura temp = menor.fig;
            menor.fig = p.fig;
            p.fig = temp;
        }
    }
}
}
}

```

### Questão 3) (3.0 pontos)

Em um curso de graduação, as disciplinas são oferecidas em turmas semestrais com até 50 alunos inscritos. Ao final de cada semestre, os professores lançam no sistema as notas dos alunos. Um aluno pode solicitar, a qualquer momento, a visualização do seu histórico na tela do computador, que é composto pelo nome do aluno, uma listagem de todas as disciplinas cursadas até então juntamente com as notas obtidas e o CR acumulado (média das notas) do aluno. Crie as classes em Java necessárias para implementar essa funcionalidade de visualização de histórico usando adequadamente o paradigma orientado a objetos, implementando as seguintes classes:

```

class Aluno{ ... }

class Inscricao{ ... }

class Turma{ ... }

class Disciplina{ ... }

```

**CONTUDO, VOCÊ NÃO DEVE IMPLEMENTAR A CLASSE DE TESTE, QUE INCLUI O MÉTODO MAIN.**

**RESPOSTA:**

```

public class Aluno {
    private String nome;
    private Inscricao[] inscricoes;

    public void visualizaHistorico() {
        System.out.println("*** História Escolar ***");
        System.out.println("Aluno: " + nome);
        System.out.println("Disciplinas e notas:");
        float cr = 0;
        for (int i = 0; i < inscricoes.length; i++) {
            String disciplina = inscricoes[i].getNomeDisciplina();
            float nota = inscricoes[i].getNota();
            System.out.println(disciplina + " --> " + nota);
            cr += nota;
        }
        cr /= inscricoes.length;
        System.out.println("CR --> " + cr);
    }
}

```

---

```

class Inscricao {
    private Turma turma;
    private float nota;

    public String getNomeDisciplina() {
        return turma.getNomeDisciplina();
    }

    public float getNota() {
        return nota;
    }
}

```

---

```

class Turma {
    private Disciplina disciplina;

    public String getNomeDisciplina() {
        return disciplina.getNome();
    }
}

```

---

```

class Disciplina {
    private String nome;

```

```
public String getNome() {  
    return nome;  
}  
}
```