Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação III
AP2 1° semestre de 2016.

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 pontos)
Considere as classes abaixo, as quais modelam figuras geométricas.

```
class Quadrilatero {
	private double lado1, lado2, lado3, lado4;
	public Quadrilatero(double l1, double l2, double l3, double l4) {
		lado1 = l1; lado2 = l2; lado3 = l3; lado4 = l4;
	}
}

class Retangulo extends Quadrilatero {
	public Retangulo(double b, double h) {
		super(b, h, b, h);
	}
	public void exibe() {
		System.out.println("Retangulo com lados " + lado1 + " e " +
lado2);
	}
}
```

a) Crie classes para modelar quadrados e círculos.
b) Modifique as classes acima de forma que possamos, para cada objeto de uma classe criado, calcular seu perímetro (recorde que o perímetro de um quadrilátero, quadrado, retângulo, etc., é dado pela soma de seus lados,

enquanto que o perímetro de um círculo é o dobro do seu raio multiplicado por PI – aproximadamente 3,14). Dica: Utilize a constante java.lang.Math.PI

c) Forneça uma maneira de obrigarmos que novas classes a serem inseridas na hierarquia (novos objetos geométricos) sejam obrigadas a definir uma forma de calcular o perímetro de seus objetos.

d) Declare uma classe que contenha um método main() (sintaxe aproximada, apenas), a qual declare e instancie uma coleção de objetos dentro deste método. Insira nesta coleção 1 instância exemplo, com qualquer valor para os seus atributos, **para cada classe criada,** percorra esta coleção somando os perímetros de todos os objetos desta coleção e exiba esta soma.

Obs: 1) Sempre que possível, utilize os conceitos de orientação a objetos vistos; 2) A solução da questão pode ser apresentada como um programa completo

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

// item c)
interface FiguraGeom {
	double perimetro();
	double area();
}

abstract class Quadrilatero implements FiguraGeom {
	double lado1, lado2, lado3, lado4;
	public Quadrilatero(double l1, double l2, double l3, double l4) {
		lado1 = l1; lado2 = l2; lado3 = l3; lado4 = l4;
	}
	// item b)
	public double perimetro() {
		return lado1 + lado2 + lado3 + lado4;
	}
}

class Retangulo extends Quadrilatero {
	public Retangulo(double b, double h) {
		super(b, h, b, h);
	}
	// item b)
	public double area() {
		return lado1 * lado2;
	}
}

// item a)
class Quadrado extends Retangulo {
	public Quadrado(double l) {
		super(l, l);
	}
}

// item a)
class Circulo implements FiguraGeom {
	double raio;
	public Circulo(double r) {
		raio = r;
```

```
        }
        // item b)
        public double area() {
                return Math.PI * Math.pow(raio, 2);
        }
        // item b)
        public double perimetro() {
                return 2 * Math.PI * raio;
        }
}

// item d)
public class AP3_2013_1_Q2 {
        public static void main(String[] args) {
                double soma = 0;
                List<FiguraGeom> objetos = new ArrayList();
                objetos.add(new Quadrado(5));
                objetos.add(new Retangulo(2,3));
                objetos.add(new Circulo(7));
                for (FiguraGeom fig : objetos) {
                        soma = soma + fig.perimetro();
                }
                System.out.println("A soma dos perimetros e: " + soma);
        }
}
```

Questão 2) (3 pontos)

Considerando as seguintes declarações de nó e de uma lista encadeada (estes métodos já estão prontos. Isto é, você não precisa escrevê-los):

```
class no{
      int info;
      no prox;
      no(int info){ this.info = info; }
      public String toString(){ return "info: " + info; }
}
class lista{
      no prim;
      lista(){ ... };
      void insere_inicio(int info){ ... }
      void insere_fim(int info){ ... }
      void imprime(){ ... }
      no busca(int info){ ... }
      void retira(int info){ ... }
}
```

Escreva um método que, dada uma lista, remova todos os elementos repetidos desta lista. O protótipo deste método é `void remove_todos_repetidos(lista l)`.

RESPOSTA:

```
class No{
  int info;
  No prox;
  No (int info){ this.info = info; }
  public String toString(){ return "info: " + info; }
}
```

```
class Lista{
  No prim;

  Lista(){ prim = null; }

  void insere_inicio (int info){
    No novo = new No(info);
    if(prim == null) prim = novo;
    else{
      novo.prox = prim;
      prim = novo;
    }
  }

  void insere_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;
  }

  void retira(int info){
    No p = prim, ant = null;
    while((p != null) && (p.info != info)){
      ant = p; p = p.prox;
    }
    if(p == null) return;
    if(ant == null) prim = p.prox;
    else ant.prox = p.prox;
  }

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

  void imprime(){ System.out.println (this); }
}
```

```
public class AP2_Q2_2016_1{
  public static void main(String[] args) {
    Lista l = new Lista();
    l.insere_fim(1);
    l.insere_fim(2);
    l.insere_fim(1);
    l.insere_fim(1);
    l.insere_fim(2);
    System.out.println("Antes de remove_todos_repetidos...");
    l.imprime();
    remove_todos_repetidos(l);
    System.out.println("Depois de remove_todos_repetidos...");
    l.imprime();
  }

  //ÚNICA OPERAÇÃO QUE DEVERIA SER FEITA NA Q2 DA AP2 DE 2016/1
  static void remove_todos_repetidos(Lista l){
    No p = l.prim;
    while(p != null){
      No q = p.prox, ant = p;
      while((q != null) && (q.info != p.info)){
        ant = q;
        q = q.prox;
      }
      if(q != null) ant.prox = q.prox;
      else p = p.prox;
    }
  }
}
```

Questão 3) (3 pontos)

Supondo que a memória secundária do seu computador possui um arquivo que indica os segmentos de memória que compõem os arquivos já armazenados. Cada linha deste arquivo é composta pelo nome do arquivo, a quantidade de bytes armazenadas naquele setor e um índice para o próximo segmento do arquivo. Se a informação utilizada como próximo segmento for igual a -1, o arquivo chegou ao seu fim. Um exemplo deste arquivo seria:

```
arq1.txt 500 3
arq2.dat 100 4
arq1.txt 100 -1
arq2.dat 200 5
arq2.txt 300 -1
xxx.odt 600 -1
```

Escreva um programa que receba, como parâmetro de entrada, um arquivo como descrito anteriormente e que insira, num arquivo de resposta cujo o nome é `saida-` acrescido do nome do arquivo de entrada,o nome do arquivo, a quantidade total de bytes e o índice de início de cada arquivo existente na memória ao fim dos dados copiados do arquivo original. O resultado da aplicação no arquivo de exemplo seria, **APÓS SEU PROGRAMA LER O ARQUIVO DE ENTRADA UMA ÚNICA VEZ**:

```
arq1.txt 500 3
```

```
arq2.dat 100 4
arq1.txt 100 -1
arq2.dat 200 5
arq2.dat 300 -1
xxx.odt 400 -1
arq1.txt 600 1
arq2.txt 600 2
xxx.odt 400 6
```

SE SEU PROGRAMA LER MAIS DE UMA VEZ O ARQUIVO DE ENTRADA OU NÃO RESPONDER CORRETAMENTE PARA QUALQUER ARQUIVO QUE SIGA O FORMATO ANTERIORMENTE CITADO, SUA RESPOSTA SERÁ SEVERAMENTE DESCONTADA.

RESPOSTA:

```
import java.io.*;

class No{
  String nome;
  int ini, tam;
  No prox;
  No (String n, int bytes, int inicio){
    nome = n;
    tam = bytes;
    ini = inicio;
  }
  public String toString(){ return nome + " " + tam + " " + ini +
"\n"; }
}


class Lista{
  No prim;

  Lista(){ prim = null; }

  No busca(String nome){
    No p = prim;
    while((p != null) && (!p.nome.equals(nome))) p = p.prox;
    return p;
  }

  void insere_fim (String nome, int tam, int ini){
    No novo = new No(nome, tam, ini);
    if(prim == null) prim = novo;
    else{
      No p = prim;
      while(p.prox != null) p = p.prox;
      p.prox = novo;
    }
  }
```

```
  void insere_total (String nome, int tam, int ini){
    No p = busca(nome);
    if(p == null) insere_fim(nome, tam, ini);
    else p.tam += tam;
  }

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


public class AP2_Q3_2016_1{
  public static void main(String[] args) throws IOException{
            BufferedReader   in      =   new   BufferedReader(new
FileReader(args[0]));
             BufferedWriter   out   =   new   BufferedWriter(new
FileWriter("saida-" + args[0]));
    Lista lista_arq = new Lista(), lista_total = new Lista();

    try {
      int n = 1;
      String s, vs[];
      while((s = in.readLine()) != null){
        vs = s.split(" ");
             lista_arq.insere_fim(vs[0],  Integer.parseInt(vs[1]),
Integer.parseInt(vs[2]));
          lista_total.insere_total(vs[0],  Integer.parseInt(vs[1]),
n++);
      }
      out.write(lista_arq.toString());
      out.write(lista_total.toString());
    } catch (Exception e){
      System.out.println("Excecao\n");
    } finally {
      in.close();
      out.close();
    }
  }
}
```