



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação III

AP2 1º semestre de 2015.

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 uma lista duplamente encadeada de valores inteiros. A interface das classes que implementam essa lista é dada por:

```
public class Lista {  
    private No primeiro = null;  
    private No ultimo = null;  
    public void insereOrdenado(int valor) { ... }  
    public boolean estaVazia() { ... }  
    public void retiraPrimOcorrencia(int valor) { ... }  
}  
  
class No {  
    int valor;  
    No anterior, proximo;  
    No(int valor) { ... }  
}
```

Nesta questão, você deverá substituir “{ ... }” pela implementação correta dos respectivos métodos. O método `insereOrdenado` recebe um valor e insere um novo nó na posição que garanta ordenação crescente dos elementos da lista. O método `estaVazia` retorna se existem valores armazenados na lista. Finalmente, o método

retiraPrimOcorrencia remove o primeiro nó que armazene o valor passado como parâmetro de entrada. Você deve implementar, também, o construtor da classe No. **Para a implementação dos métodos da classe Lista, é permitido que a lista seja percorrida no máximo uma vez.**

RESPOSTA:

```
class No{  
    int valor;  
    No anterior, proximo;  
  
    No(int valor){  
        this.valor = valor;  
        anterior = proximo = null;  
    }  
  
    public String toString(){ return valor + " "; } //TESTE  
}
```

```
class Lista {  
    private No primeiro = null;  
    private No ultimo = null;  
  
    public void insereOrdenado(int valor){  
        No p = primeiro;  
        No novo = new No(valor);  
        if(p == null){  
            primeiro = ultimo = novo;  
            return;  
        }  
        while((p != null) && (p.valor <= valor)) p = p.proximo;  
        if(p == null){  
            ultimo.proximo = novo;  
            novo.anterior = ultimo;  
            ultimo = novo;  
            return;  
        }  
        if(p.anterior == null){  
            novo.proximo = primeiro;  
            primeiro.anterior = novo;  
            primeiro = novo;  
        }  
    }
```

```

        return;
    }
    novo.proximo = p;
    novo.anterior = p.anterior;
    p.anterior.proximo = novo;
    p.anterior = novo;
}

```

```

public boolean estaVazia(){ return primeiro == null; }

```

```

No busca(int info){
    No p = primeiro;
    while((p != null) && (p.valor != info)) p = p.proximo;
    return p;
}

```

```

public void retiraPrimOcorrencia(int valor){
    No p = busca(valor);
    if(p == null) return;
    if(p == primeiro) primeiro = primeiro.proximo;
    else p.anterior.proximo = p.proximo;
    if(p.proximo != null) p.proximo.anterior = p.anterior;
    else ultimo = p.anterior;
}

```

```

public String toString(){ //TESTE
    String resp = "";
    No p = primeiro;
    while(p != null){
        resp += p.toString();
        p = p.proximo;
    }
    return resp + "\n";
}
}

```

```

public class Q1_AP2_2015_1 { //TESTE
    public static void main(String[] args){
        Lista l = new Lista();
        l.insereOrdenado(5);
    }
}

```

```

l.insereOrdenado(1);
l.insereOrdenado(3);
l.insereOrdenado(7);
l.insereOrdenado(1);
System.out.print(l);
l.retiraPrimOcorrencia(1);
System.out.print(l);
l.retiraPrimOcorrencia(7);
System.out.print(l);
l.retiraPrimOcorrencia(3);
System.out.print(l);
}
}

```

Questão 2) (2.0 pontos)

Escreva um programa que, no método **main**, receba do usuário um arquivo que possui uma sequência de números inteiros maiores ou iguais a zero (um em cada linha), invoque uma rotina que processe o número da vez e, por fim, exiba o resultado retornado pela rotina de processamento. O final do arquivo é marcado pela palavra **“Fim”**.

Ao receber uma informação do arquivo, o programa deve garantir que o texto seja igual a **“Fim”** ou que o texto possa ser convertido para um valor inteiro maior ou igual à zero. Caso contrário, o programa deverá alertar ao usuário que o valor informado é inválido, lançando uma exceção (classe criada por você).

O processamento de um número válido consiste em convertê-lo da base decimal para a base binária. A conversão deve ser implementada por uma função que recebe um único argumento (um número inteiro) e retorne uma instância de String com o resultado da conversão. **Não é permitido o uso de rotinas nativas da Java API para a conversão de base.**

RESPOSTA:

```

import java.io.*;
public class Q2_AP2_2015_1 {
    public static void main(String[] args) throws Exception{
        FileReader in = new FileReader(args[0]);
        BufferedReader br = new BufferedReader(in);
        int num;
        String name = "";
        do {
            try {
                name = br.readLine();
                if (!name.equals("Fim")) {
                    num = Integer.parseInt(name);

```

```

        name = convertBin(num);
        System.out.println(num + " = " + name);
    }
} catch (NumberFormatException e) {
    System.out.println("ERRO");
} catch (IOException e) {
}
} while (!(name.equals("Fim")));
}

public static String convertBin(int n) {
    int k = 0;
    String c = "";
    k = n % 2;
    if (n/2 > 0) {
        c = convertBin(n/2);
    }
    return c+k;
}
}

```

Questão 3) (4.0 pontos)

Considere o programa abaixo:

```

interface Figura3D {
    public double getVolume();
    public String toString();
    public boolean equals(Object obj);
}

class Esfera implements Figura3D {
    private double raio;

    public Esfera(double r) {
        raio = r;
    }

    public double getRadius() {
        return raio;
    }

    public double getVolume() {
        return 4.0 * Math.PI * Math.pow(raio, 3) / 3.0;
    }

    public boolean equals(Object obj) {
        if (obj instanceof Esfera) {
            Esfera other = (Esfera) obj;
            return raio == other.raio;
        }
        else
    }
}

```

```

        return false;
    }
}

public class Principal {
    public static final int MAX = 3;
    public static void main(String[] args) {
        Figura3D[] shapes = new Figura3D[MAX];
        shapes[0] = new Esfera(20);
        shapes[1] = new Cubo(10);
        shapes[2] = new Cilindro(10, 20);

        for (int i = 0; i < shapes.length; i++) {
            switch(i) {
                case 0:
                    System.out.print("\nEsta é uma esfera. Seu volume é: " +
shapes[0].getVolume());
                    break;
                case 1:
                    System.out.print("\nEste é um cubo. Seu volume é: " +
shapes[1].getVolume());
                    break;
                case 2:
                    System.out.print("\nEste é um cilindro. Seu volume é: " +
shapes[2].getVolume());
                    break;
            }
        }
    }
}

```

A interface `Figura3D` modela objetos em 3 dimensões. Logo abaixo temos a classe `Esfera`, a qual é um exemplo de figura 3D. Mais abaixo temos uma classe (`Principal`) que utiliza a classe e interface definidas.

- Implemente a classe `Cubo` utilizada na classe `Principal`, lembrando que um cubo é uma caixa com lados iguais. O argumento passado na construção do objeto `Cubo` é o tamanho do lado do cubo.
- Ainda na classe `Principal`, temos a impressão do tipo de cada objeto e o seu volume. Esta impressão utiliza o conceito de polimorfismo (sim ou não)? Se não, o que é necessário fazer para que utilize o conceito. Altere o programa de forma que este uso aconteça.

RESPOSTA:

```

class Esfera implements Figura3D {
    private double raio;

    ...

    public String toString() {
        return "Esta é uma esfera. Seu volume é: " + this.getVolume();
    }
}

```

```

class Cubo implements Figura3D {
    private double lado;

    public Cubo(int l) {
        this.lado = l;
    }

    public double getArea() {
        return 6 * lado * lado;
    }

    public double getVolume() {
        return lado * lado * lado;
    }

    public String toString() {
        return "Esta é um cubo. Seu volume é: " + this.getVolume();
    }

    public boolean equals(Object obj) {
        if (obj instanceof Cubo) {
            Cubo other = (Cubo) obj;
            return this.lado == other.lado;
        }
        else
            return false;
    }
}

public class AP2_2015_1_Q3 {
    public static final int MAX = 3;
    public static void main(String[] args) {

        ...

        // No lugar do for antigo
        for (Figura3D f : shapes) {
            System.out.println(f);
        }
    }
}

```