



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Programação III**

**AP3 1º semestre de 2014.**

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) (5.0 pontos)**

O ECAD (Escritório Central de Arrecadação e Distribuição) é um órgão público responsável por centralizar e distribuir os direitos autorais referentes à reprodução musical pública. Suponha que fomos contratados para elaborar um sistema de apoio a este controle. Considere a classe abstrata **Reprodutivel** dada abaixo:

```
abstract class Reprodutivel {  
    private int reproducoes;  
  
    public Reprodutivel() {  
        this.reproducoes = 0;  
    }  
  
    public void play() {  
        this.reproducoes++;  
    }  
  
    public float direitos () { return ((float)(this.reproducoes)) *  
this.ecad(); }  
  
    public abstract float ecad();  
}
```

Esta classe possui um atributo chamado **reproducoes**, o qual armazenará por quantas vezes uma dada música foi reproduzida. O método **play** simula uma reprodução, enquanto que o método **direitos** calcula o montante de direitos devidos por uma dada música. Para simplificar, a expressão para esse cálculo é a multiplicação do número de

reproduções pelo valor retornado pelo método **ecad**. Observe que este método é abstrato, assim como a classe.

Para esse sistema, devemos manipular 2 tipos de mídias: músicas isoladas e filmes, os quais normalmente contém uma trilha musical. Para cada destes tipos devemos guardar seus nomes e ano em que foram produzidos. Além desses, uma música deve conter a lista de compositores (strings apenas), enquanto que um filme deve conter o nome do seu diretor (string também) e uma lista de músicas que representa sua trilha musical. O valor do **ecad** será 0,01 (1 centavo) para a execução de uma música. Para o caso dos filmes, supondo que uma música num filme só é reproduzida 1 (uma) vez, o valor do **ecad** é a soma da quantidade de filmes multiplicado por 0,01.

Num método **main** crie 1 objeto como exemplo para cada classe concreta criada. Os dados desses objetos podem ser fictícios.

RESPOSTA:

```
abstract class Media extends Reprodutivel {
    String nome;
    int ano;

    public Media(String nome, int ano) {
        super();
        this.nome = nome;
        this.ano = ano;
    }
}

class Musica extends Media {
    List<String> compositores;
    static float vecad = 0.01f;

    public Musica(String nome, List<String> compositores, int ano) {
        super(nome, ano);
        this.nome = nome;
        this.compositores = compositores;
        this.ano = ano;
    }

    public float ecad() {
        return Musica.vecad;
    }
}

class Filme extends Media {
    String diretor;
    List<Musica> trilha;

    public Filme(String nome, int ano, String d, List<Musica> musicas) {
        super(nome, ano);
        diretor = d;
        trilha = musicas;
    }

    public float ecad() {
        return trilha.size() * Musica.vecad;
    }
}
```

```

    }
}

public class AP3_2014_1_Q2 {
    public static void main(String[] args) {
        List<String> comp1 = new ArrayList<String>();
        comp1.add("Pharrel Williams");
        Musica m1 = new Musica ("Happy", comp1, 2014);

        List<String> comp2 = new ArrayList<String>();
        comp2.add("Carlinhos Brown");
        Musica m2 = new Musica ("Dia de Festa, Salve a Floresta", comp2,
2011);

        List<Musica> trilha = new ArrayList<Musica>();
        trilha.add(m2);
        Filme f = new Filme ("Rio", 2011, "Carlos Saldanha", trilha);
    }
}

```

### Questão 2) (5.0 pontos)

Escreva uma interface **Numero** e um conjunto de classes que representem números racionais e complexos de acordo com as seguintes instruções:

(a) (2.5 pontos) Escreva a interface **Numero** que proponha a implementação dos métodos:

```

public Numero soma (Numero n);

public Numero subtracao (Numero n);

public Numero produto (Numero n);

public Numero divisao (Numero n);

```

A interface **Numero** deve ser implementada pela classe concreta **NumeroComplexo** e pela classe abstrata **NumeroRacional**.

(b) (2.5 pontos) A classe **NumeroRacional** deve conter o método **public abstract void imprimeNumero ()**, cujas implementações pertencem às subclasses **NumeroRacionalFracao** e **NumeroRacionalDecimal**. Na classe **NumeroRacionalFracao**, o método **imprimeNumero ()** deve imprimir na saída padrão o número racional usando a notação de frações, enquanto que na classe **NumeroRacionalDecimal** a impressão deve acontecer usando a notação decimal.

Observações:

- 1) Implemente apenas um construtor por classe, sendo que este recebe todos os parâmetros necessários para inicializar os atributos da classe respectiva.
- 2) Leve em conta os princípios do encapsulamento e da ocultação da informação.
- 3) Um número complexo é do tipo  $a+bi$  onde  $a, b \in \mathbb{R}$  e  $i = \sqrt{-1}$ .
- 4) Um número racional é do tipo  $\frac{c}{d}$  onde  $c, d \in \mathbb{Z}$  e  $d \neq 0$ .

**RESPOSTA:**

```

public interface Numero {
    public Numero soma(Numero n);
    public Numero subtracao(Numero n);
    public Numero produto(Numero n);
    public Numero divisao(Numero n);
}

public class NumeroComplexo implements Numero {
    private double real;
    private double imag;

    public NumeroComplexo(double real, double imag) {
        this.real = real; this.imag = imag;
    }

    public Numero soma(Numero n) {
        NumeroComplexo other = (NumeroComplexo) n;
        return new NumeroComplexo(this.real + other.real, this.imag +
other.imag);
    }

    public Numero subtracao(Numero n) {
        NumeroComplexo other = (NumeroComplexo) n;
        return new NumeroComplexo(this.real - other.real, this.imag -
other.imag);
    }

    public Numero produto(Numero n) {
        NumeroComplexo other = (NumeroComplexo) n;
        return new NumeroComplexo(
            this.real * other.real - this.imag * other.imag,
            this.real * other.imag + this.imag * other.real);
    }

    public Numero divisao(Numero n) {
        NumeroComplexo other = (NumeroComplexo) n;
        double aux = 1.0 / (other.real * other.real + other.imag *
other.imag);
        return new NumeroComplexo(
            (this.real * other.real + this.imag * other.imag) * aux,
            (this.imag * other.real - this.real * other.imag) * aux);
    }
}

```

```
}  
}
```

```
public abstract class NumeroRacional implements Numero {  
    protected int num;  
    protected int den;  
  
    public NumeroRacional(int num, int den) {  
        this.num = num;  
        this.den = den;  
    }  
    protected NumeroRacional somar(final NumeroRacional other) {  
        int m = mmc(this.den, other.den);  
        this.num = (this.num * (m / this.den)) + (other.num * (m / other.den));  
        this.den = m;  
        return this;  
    }  
  
    protected NumeroRacional subtrair(final NumeroRacional other){  
        int m = mmc(this.den, other.den);  
        this.num = (this.num * (m / this.den)) - (other.num * (m / other.den));  
        this.den = m;  
        return this;  
    }  
  
    protected NumeroRacional multiplicarPor(final NumeroRacional  
other) {  
        this.num *= other.num;  
        this.den *= other.den;  
        return this;  
    }  
  
    protected NumeroRacional dividirPor(final NumeroRacional other) {  
        this.num *= other.den;  
        this.den *= other.num;  
        return this;  
    }  
  
    private static int mmc(int x, int y) {
```

```

        int r = x * y;
        for (int i = 1; i <= y; i++) {
            if (((x * i) % x) == 0 && ((x * i) % y) == 0 && ((x * i) < r)) {
                r = x * i;
            }
        }
        return r;
    }

    public abstract void imprimeNumero();
}

public class NumeroRacionalDecimal extends NumeroRacional {
    public NumeroRacionalDecimal(int num, int den) {
        super(num, den);
    }

    public Numero soma(Numero n) {
        NumeroRacionalDecimal r = new NumeroRacionalDecimal(this.num,
this.den);
        return r.somar((NumeroRacional) n);
    }

    public Numero subtracao(Numero n) {
        NumeroRacionalDecimal r = new NumeroRacionalDecimal(this.num,
this.den);
        return r.subtrair((NumeroRacional) n);
    }

    public Numero produto(Numero n) {
        NumeroRacionalDecimal r = new NumeroRacionalDecimal(this.num,
this.den);
        return r.multiplicarPor((NumeroRacional) n);
    }

    public Numero divisao(Numero n) {
        NumeroRacionalDecimal r = new NumeroRacionalDecimal(this.num,
this.den);
        return r.dividirPor((NumeroRacional) n);
    }
}

```

```
public void imprimeNumero() {  
    System.out.print(((double) this.num) / this.den);  
}  
}
```

```
public class NumeroRacionalFracao extends NumeroRacional {  
    public NumeroRacionalFracao(int num, int den) {  
        super(num, den);  
    }  
  
    public Numero soma(Numero n) {  
        NumeroRacionalFracao r = new NumeroRacionalFracao(this.num,  
this.den);  
        return r.somar((NumeroRacional) n);  
    }  
  
    public Numero subtracao(Numero n) {  
        NumeroRacionalFracao r = new NumeroRacionalFracao(this.num,  
this.den);  
        return r.subtrair((NumeroRacional) n);  
    }  
  
    public Numero produto(Numero n) {  
        NumeroRacionalFracao r = new NumeroRacionalFracao(this.num,  
this.den);  
        return r.multiplicarPor((NumeroRacional) n);  
    }  
  
    public Numero divisao(Numero n) {  
        NumeroRacionalFracao r = new NumeroRacionalFracao(this.num,  
this.den);  
        return r.dividirPor((NumeroRacional) n);  
    }  
  
    public void imprimeNumero() {  
        System.out.print(this.num + "/" + this.den);  
    }  
}
```

