



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Programação III**

**AP1 2º semestre de 2011.**

**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)**

Escreva um programa que receba, como parâmetro de entrada, uma string e que imprima, na tela, TODAS as N palavras geradas pelos deslocamentos a esquerda dos caracteres da string. Sabe-se que N é o tamanho da string.

Por exemplo, se o usuário digitar `java Q1 AMOR`, seu programa deve imprimir na tela:

`AMOR`  
`MORA`  
`ORAM`  
`RAMO`

Se o usuário digitar `java Q1 A`, seu programa deve imprimir na tela:

`A`

**RESPOSTA:**

```
public class Q1{
    public static void main(String[] args){
        String alphabet = args[0];
        int N = alphabet.length();
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                char c = alphabet.charAt((i + j) % N);
                System.out.print(c);
            }
            System.out.println();
        }
    }
}
```

```

    }
}

```

## Questão 2) (3.0 pontos)

Suponha a classe Livro definida abaixo, a qual será utilizada num sistema que manipula informações de publicações:

```

public class Livro {
    int isbn;
    String titulo;
    String autor;
    String editora;
    java.util.GregorianCalendar datapublicacao;
    float preco_compra;
}

```

(Obs.: A classe `java.util.GregorianCalendar` é utilizada para a manipulação de datas em Java)

Além de livros, o sistema deve armazenar também informações de revistas (título, isbn, mês, ano e preço de compra), gibis (isbn, título, edição e preço de compra) e cds e dvds (issn, título, tamanho e preço de compra).

- Crie novas classes de forma que estas informações possam ser manipuladas.
- Implemente construtores que inicializem todos os atributos das classes.
- Crie um objeto do tipo revista no método main e imprima na tela o preço deste objeto revista criado.
- Imagine que, futuramente, pretendemos fazer um levantamento do valor total do acervo. Faça com que as classes criadas sejam obrigadas a definir um método para a obtenção do preço de um item (por exemplo, *float retornaPreco()*).

Reutilize construções, utilizando os mecanismos de O.O. em Java, sempre que possível. A classe fornecida também pode ser modificada.

## RESPOSTA:

```

// Classe abstrata que modela a estrutura básica de uma publicação
class Publicacao {
    int cod;
    String titulo;
    float preco_compra;

    public Publicacao (int c, String t, float p) {
        cod = c; titulo = t; preco_compra = p;
    }

    public float retornaPreco() {
        return preco_compra;
    }
}

// Os campos isbn (cod), titulo e preço de compra passam a ser
//herdados da classe 'Publicacao'

```

```

class Livro extends Publicacao {
    String autor;
    String editora;
    GregorianCalendar datapublicacao;

    public Livro (int i, String t, String a, String e, GregorianCalendar d,
float p) {
        super(i, t, p);
        autor = a; editora = e; datapublicacao = d;
    }
}

class Revista extends Publicacao {
    int mes, ano;

    public Revista (int i, String t, float p, int m, int a) {
        super(i, t, p);
        mes = m;
        ano = a;
    }
}

class Gibi extends Publicacao {
    String edicao;

    public Gibi (int i, String t, float p, String e) {
        super(i, t, p);
        edicao = e;
    }
}

class CdDvd extends Publicacao {
    int tamanho;

    public CdDvd (int i, String t, float p, int tam) {
        super(i, t, p);
        tamanho = tam;
    }
}

// Classe principal que exemplifica o uso das classes declaradas
// Não é necessária na solução da questão !!
public class AP1_2011_2_Q3 {
    public static void main(String[] args) {
        Publicacao revista = new Revista(1, "Auto Esporte", 10.0f, 9,
2010);
        System.out.println("Preço da revista: " +
revista.retornaPreco());
    }
}

```

### Questão 3) (3.0 pontos)

Defina classes e/ou interfaces que modelem objetos geométricos. Os objetos geométricos são: retângulo, caixa, círculo e cilindro. Crie classes para estes objetos de forma a

aproveitar ao máximo suas características comuns. Nestas classes, crie construtores parametrizados, os quais devem ser utilizados para inicialização dos atributos dos objetos. Para cada objeto, crie métodos para obter seu perímetro (figuras 2D), sua área (externa, no caso das 3D) e volume (figuras 3D).

Retângulo: Área = base\*altura, Perímetro = 2\*base + 2\*altura

Caixa: Volume = base1\*base2\*altura, Área = 2\*(base1\*base2 + base1\*altura + base2\*altura)

Círculo: Área =  $3.14*(raio)^2$ , Perímetro =  $2*3.14*raio$

Cilindro: Volume =  $3.14*(raio)^2*altura$ , Área =  $2*3.14*(raio)^2 + 2*3.14*raio*altura$

Imaginando que objetos geométricos serão fabricados, e que o material utilizado deve ser economizado, defina um método que compare um objeto com outro por sua área ocupada e outro método que compare suas capacidades volumétricas.

### RESPOSTA:

```
class Retangulo {
    double lado1, lado2;
    public Retangulo(double l1, double l2) {
        lado1 = l1; lado2 = l2;
    }
    public double perimetro () {
        return 2*lado1 + 2*lado2;
    }
    public double area() {
        return lado1 * lado2;
    }
}

class Circulo {
    double raio;
    public Circulo(double r) {
        raio = r;
    }
    public double perimetro () {
        return 2 * 3.14 * raio;
    }
    public double area() {
        return 3.14 * raio * raio;
    }
}

class Cilindro extends Circulo {
    private double altura;
    public Cilindro(double a, double r) {
        super(r);
        altura = a;
    }
    public double volume() {
        return this.area() * altura;
    }
}
```

```
    }  
}  
  
class Caixa extends Retangulo {  
    private double altura;  
    public Caixa(double l1, double l2, double a) {  
        super(l1, l2);  
        altura = a;  
    }  
    public double volume() {  
        return this.area() * altura;  
    }  
    public double area() {  
        return 2*(lado1*lado2 + altura*lado1 + altura*lado2);  
    }  
}
```