



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância  
**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina: EAD05032 - Programação Orientada a Objetos**  
**AP1 – 1º semestre de 2023.**

---

**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)**

Nesta questão, você deverá implementar classes Java que atendam ao cenário do problema descrito abaixo. Os quesitos avaliados serão a aplicação adequada de (i) abstração, (ii) encapsulamento, (iii) herança e (iv) polimorfismo. Indique na folha de resposta onde cada um desses princípios foi empregado na solução entregue. Erros de sintaxe serão descontados.

Descrição do Cenário:

Projete e implemente as classes Empregado, Gerente, Secretario e Técnico. Todo empregado tem um nome, um salário e o ano de contratação. Para um dado empregado, deve ser possível obter seu nome, obter seu salário, aumentar seu salário de acordo com um percentual e obter o ano de contratação. Um gerente, que também é um empregado, realiza todas as operações realizadas por um empregado, mas diferentemente de empregado, ele possui um secretário e comanda um grupo de técnicos. A partir de um gerente, deve ser possível aumentar o salário de todos os seus subordinados, que incluem os seus técnicos e secretário, de um percentual passado como parâmetro de entrada. Secretários e técnicos também são empregados, logo, também possuem todas as responsabilidades de empregado. Todo empregado deve ser capaz de realizar a ação de imprimir na saída padrão qual seu papel na empresa por meio de uma frase formatada. Gerentes devem imprimir “Eu sou gerente e tenho <nome-secretário> como secretário(a) e mais <quantidade> técnicos(as) sob meu comando.”, onde < nome-secretário > deve ser substituído pelo nome do secretário(a) do gerente e <quantidade> pela quantidade de técnicos. Secretário, por sua vez, deve imprimir “Eu sou secretário(a)” e técnico deve imprimir “Eu sou técnico(a)”. Observe que “empregado” é um conceito abstrato neste cenário, enquanto que “gerente”, “secretário” e “técnico” são conceitos concretos.

## Gabarito

```
import java.util.ArrayList;

public abstract class Empregado { // Abstração
    private String nome;        // Uso de encapsulamento ao declarar o atributos como privado e
    private double salario; // mantê-los apenas por métodos desta classe.
    private int anoDeContratacao;

    public Empregado(String nome, double salario, int anoDeContratacao) {
        this.nome = nome;
        this.salario = salario;
        this.anoDeContratacao = anoDeContratacao;
    }

    public String getNome() {
        return this.nome;
    }

    public double getSalario() {
        return this.salario;
    }

    public int getAnoDeContratacao() {
        return this.anoDeContratacao;
    }

    public void aumentarSalario(double percentual) {
        this.salario += this.salario * percentual / 100;
    }

    public abstract void imprimirPaper(); // Abstração
}
```

```
public class Gerente extends Empregado { // Uso de herança ao estender a classe "Empregado" e não
                                           // cometer o erro de redeclarar os atributos herdados.
    private Secretario secretario; // Uso de encapsulamento ao declarar o atributos como privado e
    mantê-los apenas por métodos desta classe.
    private ArrayList<Tecnico> tecnicos = new ArrayList<Tecnico>();

    public Gerente(String nome, double salario, int anoDeContratacao, Secretario secretario) {
        super(nome, salario, anoDeContratacao);
        this.secretario = secretario;
    }

    public void aumentarSalarioEmpregados(double percentual) {
        if (secretario != null) {
            secretario.aumentarSalario(percentual);
        }
        for (Tecnico tecnico: tecnicos) {
            tecnico.aumentarSalario(percentual)
        }

        public void addTecnico(Tecnico tecnico) {
            this.tecnicos.add(tecnico);
        }
    }

    @Override
    public void imprimirPaper() { // Polimorfismo
        System.out.println("Eu sou gerente e tenho %s como secretário(a) e mais %d técnicos(as) sob
        meu comando.", this.secretario.getNome(), this.tecnicos.size());
    }
}
```

```
public class Secretario extends Empregado { // Uso de herança ao estender a classe "Empregado" e não
                                           // cometer o erro de redeclarar os atributos herdados.
    public Secretario(String nome, double salario, int anoDeContratacao) {
        super(nome, salario, anoDeContratacao);
    }

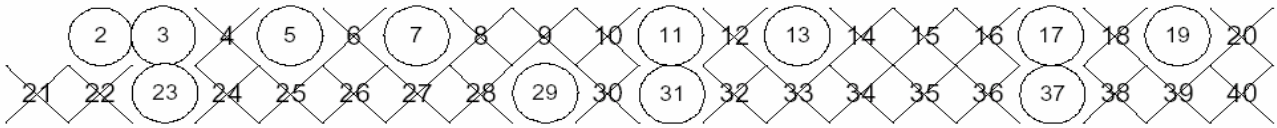
    @Override
    public void imprimirPaper() { // Polimorfismo
        System.out.println("Eu sou secretário(a).");
    }
}

public class Tecnico extends Empregado { // Uso de herança ao estender a classe "Empregado" e não
                                           // cometer o erro de redeclarar os atributos herdados.
    public Tecnico(String nome, double salario, int anoDeContratacao) {
        super(nome, salario, anoDeContratacao);
    }

    @Override
    public void imprimirPaper() { // Polimorfismo
        System.out.println("Eu sou técnico(a).");
    }
}
```

### Questão 2)

Uma forma simples e eficiente de calcular todos os números primos até um certo valor  $n$  é o método da Peneira de Eratosthenes. O processo é simples: escrevem-se todos os valores entre 2 e  $n$  (limite máximo). Em seguida, faz-se um círculo em volta do 2, marcando como primo e riscam-se todos os seus múltiplos. Continua-se a fazer círculos em volta do menor inteiro que se encontra, eliminando todos os seus múltiplos. Quando não restarem números sem terem círculos à volta ou traços por cima, os números com círculos à volta representam todos os primos até  $n$ . A figura seguinte apresenta o método para  $n = 40$ .



Escreva um programa em Java que implemente a Peneira de Eratosthenes. Você deve ler o valor  $n$  e mostrar todos os números primos encontrados.

Sua classe principal, que **NÃO PODE SER ALTERADA**, é a seguinte:

```
import java.util.Scanner;
```

```
public class Q2_AP1{
    public static void main (String args[]){
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        while(n > 1){
            PeneiraEratosthenes(n);
            n = sc.nextInt();
        }
        sc.close();
    }

    public static void PeneiraEratosthenes (int n){
        //IMPLEMENTE ESSE MÉTODO...
    }
}
```

## Gabarito

```
import java.util.Scanner;

public class Q2_AP1{
    public static void main (String args[]){
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        while(n > 1){
            PeneiraEratosthenes(n);
            n = sc.nextInt();
        }
        sc.close();
    }

    public static void PeneiraEratosthenes (int n){
        int i, j, div, tam = n - 1, vet[] = new int[tam];
        for(i = 0; i < tam; i++) vet[i] = i + 2;
        for(i = 0; i < tam; i++){
            if(vet[i] != 0){
                div = vet[i];
                for(j = i + 1; j < tam; j++)
                    if(vet[j] % div == 0) vet[j] = 0;
            }
        }
        for(i = 0; i < tam; i++)
            if(vet[i] != 0) System.out.print(vet[i] + " ");
        System.out.println();
    }
}
```