



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação III

AP1 2º semestre de 2009.

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) (2.5 pontos)

Implemente um programa que receba, como parâmetro de entrada, um número **n** e retorne, num vetor, os **n** primeiros números primos existentes depois de **n**. Por exemplo, se **n = 2**, o vetor será composto pelos dois números primos **{3, 5}**. É necessário salientar que **n** não precisa ser primo.

RESPOSTA:

```
public class calcPrimo{
    public static void main(String[] args){
        int n = Integer.parseInt(args[0]);
        int vet[] = new int[n];

        int cont = 0;
        int j = n + 1;
        while(cont < n){
            while(!ePrimo(j)) j++;
            vet[cont++] = j++;
        }

        for(int i = 0; i < n; i++)
            System.out.println(vet[i]);
    }
}
```

```

static boolean ePrimo(int n){
    for(int i = 2; i <= Math.sqrt(n); i++)
        if((n % i) == 0) return false;

    return true;
}
}

```

Questão 2) (2.5 pontos)

Implemente um programa que derive polinômios. Cada polinômio é definido por um vetor que contém seus coeficientes. Por exemplo, o polinômio de grau dois $3x^2 + 2x + 12$ terá um vetor de coeficientes $v = \{12, 2, 3\}$. Sua derivada será $\{2, 6\}$, equivalente ao polinômio $6x + 2$. O programa deve receber o valor do maior grau g do polinômio, seguido de g coeficientes. O programa calcula qual é o polinômio derivado e o imprime na tela.

RESPOSTA:

```

class derivaPol{
    public static void main (String[] args){
        int g = Integer.parseInt(args[0]);
        float vet[] = new float[g + 1];
        int i;

        for (i = 1; i <= (g + 1); i++)
            vet[i - 1] = Float.parseFloat(args[i]);

        float der[] = new float[g];

        for(i = 1; i < vet.length; i++)
            der[i - 1] = vet[i] * i;

        for(i = 0; i < der.length; i++)
            System.out.println(der[i]);
    }
}

```

Questão 3) (2.5 pontos)

Considere as classes abaixo, as quais modelam figuras geométricas.

```

class Quadrilatero {
    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 exhibe() {
        System.out.println("Retangulo com lados " + lado1 + " e " +

```

```

lado2);
    }
}

class Circulo {
    double raio;
    public Circulo(double r) {
        raio = r;
    }
    public void exhibe() {
        System.out.println("Circulo com raio " + raio);
    }
}

```

- Crie uma classe para modelar um quadrado
- 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)
- 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.

Obs: Sempre que possível, utilize os conceitos de orientação a objetos vistos.

RESPOSTA: O código apresentado contém comentários para cada item pedido na questão

```

// Interface criada como resposta ao item c). Uma outra possibilidade seria através da criação de uma classe genérica
interface ObjetoManipulavel {
    double perimetro();
}

// As classes de objetos precisam implementar a interface definida
class Quadrilatero implements ObjetoManipulavel {
    double lado1, lado2, lado3, lado4;
    public Quadrilatero(double l1, double l2, double l3, double l4) {
        lado1 = l1; lado2 = l2; lado3 = l3; lado4 = l4;
    }

    // Método implementado em resposta ao item b). Observe que, como Retângulo e Quadrado possuem a mesma forma de cálculo do perímetro, estas não deveriam redefinir este método
    public double perimetro () {
        return lado1 + lado2 + lado3 + lado4;
    }
}

class Retangulo extends Quadrilatero {
    public Retangulo(double b, double h) {
        super(b, h, b, h);
    }
}

```

```

    }
    public void exhibe() {
        System.out.println("Retangulo com lados " + lado1 + " e " +
lado2);
    }
}

// Classe solicitada no item a) desta questão
class Circulo implements ObjetoManipulavel {
    double raio;
    public Circulo(double r) {
        raio = r;
    }
    public void exhibe() {
        System.out.println("Circulo com raio " + raio);
    }

    // Método implementado em resposta ao item b).
    public double perimetro () {
        return 2 * 3.14 * raio;
    }
}

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

```

Questão 4) (2.5 pontos)

Construa uma classe Ponto3D capaz de representar um ponto qualquer em um espaço tridimensional XYZ. Considere que os valores de x, y e z são reais. Além disso, implemente:

- Dado um outro ponto como argumento (um outro objeto Ponto3D ou as coordenadas x, y e z deste outro ponto), retorne o objeto Ponto3D referente à diferença entre as coordenadas.
- Supondo que as coordenadas têm uma imprecisão (erro) de 3 unidades (para mais e para menos), forneça uma operação que indica se este ponto se intercepta a outro ponto dado.

RESPOSTA:

```

class Ponto3D {
    // Estes campos são declarados como privados de forma a encapsular
    recursos da classe. Ou seja, apenas a classe Ponto3D conhece detalhes
    de seu funcionamento interno. Classes externas obtém os valores de um
    objeto através de métodos get/set (isso não era solicitado na questão)
    private double x, y, z;
    public Ponto3D(double x, double y, double z) {
        this.x = x;
        this.y = y;
        this.z = z;
    }
}

```

```

    }
    public double getX() {
        return x;
    }
    public void setX(double x) {
        this.x = x;
    }
    public double getY() {
        return y;
    }
    public void setY(double y) {
        this.y = y;
    }
    public double getZ() {
        return z;
    }
    public void setZ(double z) {
        this.z = z;
    }

    // Métodos referentes ao item a) da questão
    public Ponto3D diferenca (Ponto3D ponto) {
        return new Ponto3D(this.getX() - ponto.getX(),
                           this.getY() - ponto.getY(),
                           this.getZ() - ponto.getZ());
    }
    public Ponto3D diferenca (double x, double y, double z) {
        return this.diferenca(new Ponto3D(x, y, z));
    }

    // Métodos referentes ao item b) da questão
    public boolean intercepta (Ponto3D ponto) {
        if ((this.getX() >= ponto.getX()-3 && this.getX() <=
ponto.getX()+3) &&
            (this.getY() >= ponto.getY()-3 && this.getY() <=
ponto.getY()+3) &&
            (this.getZ() >= ponto.getZ()-3 && this.getZ() <=
ponto.getZ()+3))
            return true;
        return false;
    }
    public boolean intercepta (double x, double y, double z) {
        return this.intercepta(new Ponto3D(x, y, z));
    }
}

```