

Curso de Tecnologia em Sistemas de Computação
AD1 de Programação Orientada a Objetos
1º semestre de 2017

Nome:

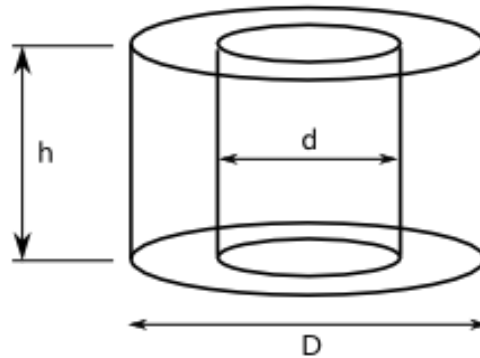
Matrícula:

Pólo:

Considere o programa abaixo que manipula figuras geométricas 3D:

```
1 public class Esfera {
2     private double raio;
3     public Esfera(double r) {
4         raio = r;
5     }
6     public double getRaio() {
7         return raio;
8     }
9     public double getVolume() {
10         return 4.0 * Math.PI * Math.pow(raio, 3) / 3.0;
11     }
12     public String toString() {
13         return "Esta é uma esfera. Seu volume é: " + this.getVolume();
14     }
15 }
16 public class Principal {
17     public static final int MAX = 3;
18     public static void main(String[] args) {
19         Figura3D[] shapes = new Figura3D[MAX];
20         shapes[0] = new Esfera(20);
21         shapes[1] = new Cubo(10);
22         shapes[2] = new Cilindro(10, 20);
23         for (int i = 0; i < shapes.length; i++) {
24             switch(i) {
25                 case 0:
26                     System.out.print("\nVolume da esfera: " + shapes[0].getVolume());
27                     break;
28                 case 1:
29                     shapes[1].imprimeVolume();
30                     break;
31                 case 2:
32                     System.out.print("\nVol. do cil.: " + shapes[2].obtemVolume());
33                     break;
34             }
35         }
36     }
37 }
```

- Implemente a classe Cubo utilizada na classe Principal (linha 21), 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.
- Implemente a classe Cilindro utilizada na classe Principal (linha 22), lembrando que um cilindro é um círculo com uma altura h . Os argumentos passados na construção do objeto Cilindro são o raio r da base do cilindro e a altura h deste.
- Ainda na classe Principal, temos a impressão do tipo de cada objeto e o seu volume (linhas 23 à 35). A obtenção do volume dos objetos e a impressão utilizam o conceito polimorfismo? Se não, o que é necessário fazer para que utilize o conceito. Altere o programa de forma que este uso aconteça.
- Crie uma nova classe, chamada CilindroOco, a qual modela objetos como a figura abaixo. Observe que a classe serve para modelar tubos vazados, onde o volume interessado nesta questão é o de massa para constituir o cilindro oco ($D - d$) e não a sua capacidade interna (d). Instancie um objeto desta classe na classe Principal, insira-o no vetor e imprima o seu volume.



RESPOSTA:

```
interface Figura3D {
    public double getArea();
    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 getArea() {
        return 4 * Math.PI * Math.pow(raio, 2);
    }
}
```

```

    }

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

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

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

// Item a)
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 "Este 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;
    }
}

// Item b)
class Cilindro implements Figura3D {
    double raio, altura;
    public Cilindro(double r, double h) {
        this.raio = r; this.altura = h;
    }

    public double getArea() {
        return 2 * Math.PI * this.raio * this.altura;
    }
}

```

```

        public double getVolume() {
            return Math.PI * Math.pow(this.raio, 2) * this.altura;
        }

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

// Item d)
class CilindroOco extends Cilindro {
    double raioInterno;
    public CilindroOco(int r, int R, int h) {
        super(R, h);
        this.raioInterno = r;
    }

    public double getVolume() {
        Cilindro interno = new Cilindro (this.raioInterno, this.altura);
        Cilindro externo = new Cilindro (this.raio, this.altura);
        return (externo.getVolume() - interno.getVolume());
    }

    public String toString() {
        return "Este é um cilindro oco. Seu volume é: " + this.getVolume();
    }
}

public class AD1_2017_1 {
    public static final int MAX = 4;
    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;
            }
        }

        shapes[3] = new CilindroOco(5, 10, 20);

// Item c)
        for (Figura3D f : shapes) {
            System.out.println(f);
        }
    }
}

```

```
}      }
```