



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**AD2 de Programação III**

**2º semestre de 2009.**

**Nome:**

**Matrícula:**

**Pólo:**

**Observações:**

1) *ADs enviadas pelo correio, devem ser postadas cinco dias antes da data final de entrega estabelecida no calendário de entrega de ADs.*

2) *A solução para o exercício proposto deve ser entregue por escrito e em formato digital.*

**Questão Única)**

Refaça a questão 3) da AP1 deste semestre, a qual foi estendida para esta AD. Para as classes abaixo:

```
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);
    }
}
class Circulo {
    double raio;
    public Circulo(double r) {
        raio = r;
    }
}
```

- a) 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, 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)
- b) Adicione um método para o cálculo de áreas (recorde que a área do retângulo é a multiplicação do seu lado menor pelo maior, a de um círculo é dada por PI multiplicado pelo raio ao quadrado, enquanto que a área de um quadrilátero qualquer é dada, aproximadamente, por  $(lado1 + lado2)/2 * (lado3 + lado4)/2$  ).

- c) Crie, no método `main()`, os seguintes objetos: um retângulo com lados 2 e 3, outro retângulo com lados 7 e 15, um círculo com raio 5 e um quadrilátero com lados 2, 3, 8 e 10. Adicione estes objetos de classes diferentes a uma lista única. Utilize, por exemplo, a classe `java.util.ArrayList`. Percorra esta lista e imprima o perímetro e a área de cada objeto encontrado.
- d) Adicione uma classe para modelar um quadrado. No método `main()`, crie um quadrado com lado 3 e outro com lado 8 e os adicione à lista.
- e) 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 e área de seus objetos. Modifique a classe quadrado de forma adequada.

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

Resposta:

Do programa abaixo, observem que definimos uma interface `ObjetoManipulavel`, a qual define o comportamento padrão de um objeto. Neste caso, o comportamento é o cálculo de área e perímetro dos objetos. Observem no método `main()` (final do programa) a lista de objetos criada é do tipo `ObjetoManipulavel`, o que obriga que só adicionemos objetos deste tipo, ou de suas subclasses, na lista.

Na definição das classes, observem se só declaramos o necessário. Por exemplo, a classe `Quadrado` só define o construtor, informando que os lados são iguais, pois os métodos de perímetro e área são herdados da classe `Quadrilátero` e `Retângulo`, respectivamente (observem que quadriláteros são quaisquer polígonos de 4 lados).

Para encerrar, ainda na `main()`, recuperamos o tipo da figura geométrica através do método `figura.getClass().getSimpleName()`, o qual não foi pedido na AD. Esta chamada exemplifica uma característica interessante da linguagem Java, também existente em algumas linguagens de programação, chamada reflexão. Esta característica nos permite saber informações sobre o próprio programa durante a execução deste. Por exemplo, da mesma maneira que descobrimos o nome da classe do objeto `figura`, também poderíamos descobrir que métodos a classe possui, listá-los, executá-los, etc.

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

interface ObjetoManipulavel {
    double perimetro();
    double area(); // AD2
}

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;
    }
    public double perimetro () {
        return lado1 + lado2 + lado3 + lado4;
    }
    public double area() {
```

```

        return ((lado1 + lado2) / 2) * ((lado3 + lado4) / 2); //
Cálculo aproximado do quadrilátero
    }
}

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);
    }
    public double area() {
        return lado1 * lado2;
    }
}

class Circulo implements ObjetoManipulavel {
    double raio;
    public Circulo(double r) {
        raio = r;
    }
    public void exhibe() {
        System.out.println("Circulo com raio " + raio);
    }
    public double perimetro () {
        return 2 * 3.14 * raio;
    }
    public double area() {
        return 3.14 * raio * raio;
    }
}

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

public class AD2_2009_2 {
    public static void main(String[] args) {
        List <ObjetoManipulavel> objetos = new
ArrayList<ObjetoManipulavel> ();
        objetos.add(new Retangulo(2, 3));
        objetos.add(new Retangulo(7, 15));
        objetos.add(new Circulo(5));
        objetos.add(new Quadrilatero(2, 3, 8, 10));
        objetos.add(new Quadrado(3));
        objetos.add(new Quadrado(8));

        Iterator <ObjetoManipulavel> it = objetos.iterator();
        while (it.hasNext()) {
            ObjetoManipulavel figura = (ObjetoManipulavel)
it.next();
            System.out.println("Tipo: " +
figura.getClass().getSimpleName());

```

```
                System.out.println("Área figura: " + figura.area());  
                System.out.println("Perímetro: " +  
figura.perimetro());  
            }  
        }  
    }
```