

Tarefa 1. Identificación de elementos da programación orientada a obxectos (POO)

No seguinte exemplo de código identifica todos os conceptos e termos que podas relacionados coa programación orientada a obxectos: clase, obxecto, construtor, atributos, método, encapsulado de campos, abstracción, encapsulamento, herdanza, polimorfismo, sobrecarga, envío de mensaxes, agregación/composición, xeneralización/especialización, asociación.

```
//creacion de una clase abstracta
abstract class FiguraGeometrica {
    public abstract double area();
}

//creacion de una clase hija que hereda de la clase FiguraGeometrica
class Circulo extends FiguraGeometrica {
    //atributo privado de la clase
    private int radio;

    //setter & getter
    public int getRadio() {
        return this.radio;
    }

    public void setRadio(int radio) {
        this.radio = radio;
    }

    //Constructor
    public Circulo(int radio) {
        this.radio = radio;
    }

    //Override del metodo abstrato
    @Override
    public double area() {
        return (double) (radio * radio * 3.14);
    }
}
```

```
//Clase que hereda de figuraGeometrica
class Triangulo extends FiguraGeometrica {
    // atributos privados de la clase
    private int ladoA;
    private int ladoB;
    private int ladoC;

    //setters & getters
    public int getLadoA() {
        return this.ladoA;
    }
    public void setLadoA(int ladoA) {
        this.ladoA = ladoA;
    }
    public int getLadoB() {
        return this.ladoB;
    }
    public void setLadoB(int ladoB) {
        this.ladoB = ladoB;
    }
    public int getLadoC() {
        return this.ladoC;
    }
    public void setLadoC(int ladoC) {
        this.ladoC = ladoC;
    }

    //constructor
    public Triangulo(int a, int b, int c) {
        ladoA = a;
        ladoB = b;
        ladoC = c;
    }

    //sobrescritura del metodo abstracto
    @Override
    public double area() {
        double num, mPer;
        mPer = (double) (ladoA + ladoB + ladoC) / 2;
        num = mPer * (mPer - ladoA) * (mPer - ladoB) * (mPer - ladoC);
        if (num <= 0) {
            System.out.println("Posiblemente o triángulo non exista, volve tentalo de novo");
            return 0;
        } else {
            return Math.pow(num, 0.5);
        }
    }
}

//creacion de una clase hija que hereda de FiguraGeometrica
class Cadrado extends FiguraGeometrica {
    //declaracion del atributo privado de la clase
    private int lado;
    //setter getter
    public int getLado() {
        return this.lado;
    }
    public void setLado(int lado) {
        this.lado = lado;
    }
    //constructor
    public Cadrado(int lado) {
        this.lado = lado;
    }
    @Override //sobrescritura del metodo
    public double area() {
        return lado * lado;
    }
}
```