

```
// ***** FIGURA *****
public abstract class Figura {
    protected double x, y;
    private String color;

    public Figura(){
        this.x = 0;
        this.y=0;
    }
    public Figura(double x, double y){
        this.x = x;
        this.y = y;
    }

    public abstract double area(); // Método abstracto
}

// ***** CIRCULO hereda de Figura *****
public class Circulo extends Figura{
    private double radio;

    Circulo(double radio) {
        this.radio = radio;
    }
    Circulo(double x, double y, double radio) {
        super(x,y); // Llamada al constructor de Figura
        this.radio = radio;
    }
    public double getRadio(){
        return this.radio;
    }
    public void setRadio(double radio){
        this.radio = radio;
    }

    @Override
    public double area(){
        return Math.PI*radio*radio;
    }
}

//***** RECTANGULO hereda de Figura *****
public class Rectangulo extends Figura{
    private double base, altura;

    Rectangulo(double base, double altura) {
        this.base = base;
        this.altura= altura;
    }

    public double getBase(){
        return base;
    }
    public double getAltura(){
        return altura;
    }
}
```

```
@Override
public double area(){
    return base * altura;
}

// ***** INICIO DEL PROGRAMA. Metodo main *****
public class Principal {

    public static void main(String[] args) {

        // Lista de objetos de tipo Figura, permite almacenar juntos objetos de tipo Circulo y Rectangulo
        List<Figura> lista_figuras = new ArrayList<>();

        lista_figuras.add( new Circulo(2,3,5) );
        lista_figuras.add( new Rectangulo(3,6) );
        lista_figuras.add( new Rectangulo(5,8) );
        lista_figuras.add( new Circulo(7,7,5) );

        Circulo c;
        Rectangulo r;

        // Recorremos la lista y mostramos los atributos específicos y las áreas
        for(Figura f: lista_figuras){

            // getClass() devuelve la clase del objeto f
            String nombreClase = f.getClass().getSimpleName();

            //metodo abstract area(), permite que todos los elementos lo tengan
            System.out.print("Area " + nombreClase + " " + f.area() + "--> ");

            if(f instanceof Circulo){ //true si f es de la clase Circulo
                //conversion explícita a Circulo para poder acceder a las propiedades específicas
                c = (Circulo) f;
                System.out.println("radio: " + c.getRadio());
            }

            if (f instanceof Rectangulo) { // true si f es de la clase Rectangulo
                //conversion explícita a Rectangulo para poder acceder a las propiedades específicas
                r = (Rectangulo) f;
                System.out.println("base, altura: " + r.getBase() + "," + r.getAltura());
            }
        } // end for
    } // end main()
} // end class
```