

Objetos, Clases y Programas

Félix Oscar Fernández Peña

Objeto





Clase

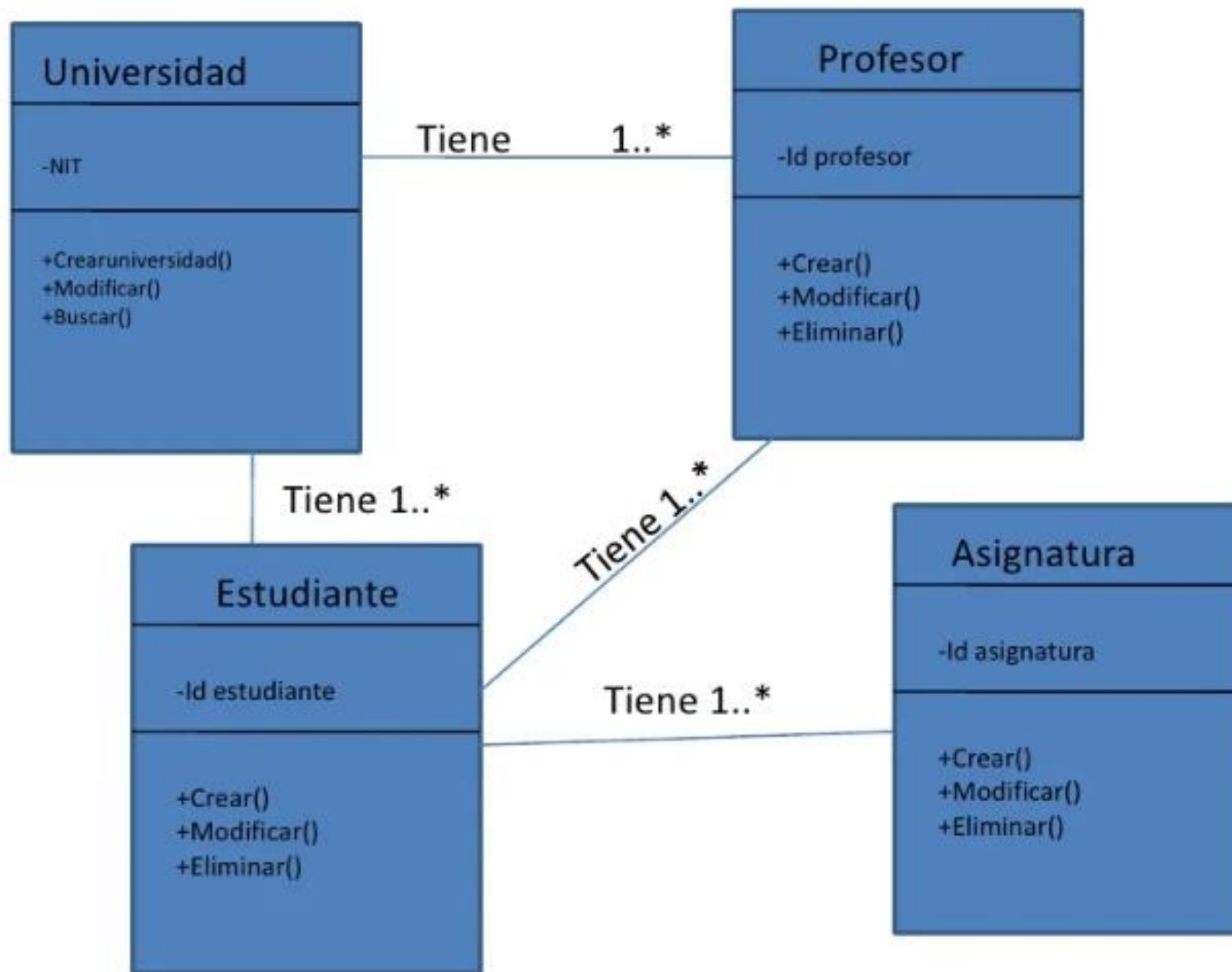


Programa


```
<li><a href="/home-events/">Home Events</a></li>
<li><a href="/multi-col-menu.html">Multiple Column Menu on Larger View</a></li>
<li><a href="/has-children">Has Children</a></li>
<li><a href="/tall-button-header/">Tall Button Header</a></li>
<li><a href="/image-logs/">Image Logs</a></li>
<li><a href="/tall-logs/">Tall Logs</a></li>
</li>
<li><a href="/carousel/">Carousel</a></li>
<li><a href="/variable-image-slider/">Variable Image Slider</a></li>
<li><a href="/testimonial-slider/">Testimonial Slider</a></li>
<li><a href="/featured-work-slider/">Featured Work Slider</a></li>
<li><a href="/equal-column-slider/">Equal Column Slider</a></li>
<li><a href="/video-slider/">Video Slider</a></li>
<li><a href="/mini-bootstrap-carousel/">Mini Bootstrap Carousel</a></li>
```

Clase Tipo de Dato





Clase Programa

Makes it class method so that it can be called using class name without creating an object of the class.

Name of the method which is called by JVM.

public

static

void

main(String[] args)

To call by JVM from anywhere

main method does not return value to JVM.

The main() method accepts one argument of type String array.

Clase de Utilidades

```
1 public class JavaMathExample1{
2     public static void main(String[] args){
3         double x = 28;
4         double y = 4;
5         System.out.println("Maximum number of x and y is: " +Math.max(x, y))
6         System.out.println("Square root of y is: " + Math.sqrt(y));
7         System.out.println("Power of x and y is: " + Math.pow(x, y));
8         System.out.println("Logarithm of x is: " + Math.log(x));
9         System.out.println("Logarithm of y is: " + Math.log(y));
10        System.out.println("log10 of x is: " + Math.log10(x));
11        System.out.println("log10 of y is: " + Math.log10(y));
12        System.out.println("log1p of x is: " +Math.log1p(x));
13        System.out.println("exp of a is: " +Math.exp(x));
14        System.out.println("expm1 of a is: " +Math.expm1(x));
15    }
16 }
```

Definición de una Clase


```
[modificadores] class NombreDeLaClase [extends OtraClase] {
```

```
    [[modificadores] tipo nomVar1;  
    [modificadores] tipo nomVar2;  
    .      ...      ...      ...  
    [modificadores] tipo nomVarN; ]
```

Atributos

```
    [[modificadores] tipo nomMetodo1 ([listaParams]) { cuerpo }  
    [modificadores] tipo nomMetodo2 ([listaParams]) { cuerpo }  
    ...      ...      ...  
    [modificadores] tipo nomMetodoM ([listaParams]) { cuerpo } ]
```

```
}
```

Métodos

Creación y Uso de un Objeto

```
Pizarra miPizarra = new Pizarra("ESPACIO DIBUJO", 300, 300);  
Circulo c1 = new Circulo(50, "amarillo", 100, 100);  
Rectangulo r1 = new Rectangulo(30, 30, "azul", 125, 125);  
Rectangulo r2 = new Rectangulo(100, 10, "rojo", 50, 155);
```

Creación

```
miPizarra.add(c1);  
miPizarra.add(r1);  
miPizarra.add(r2);
```

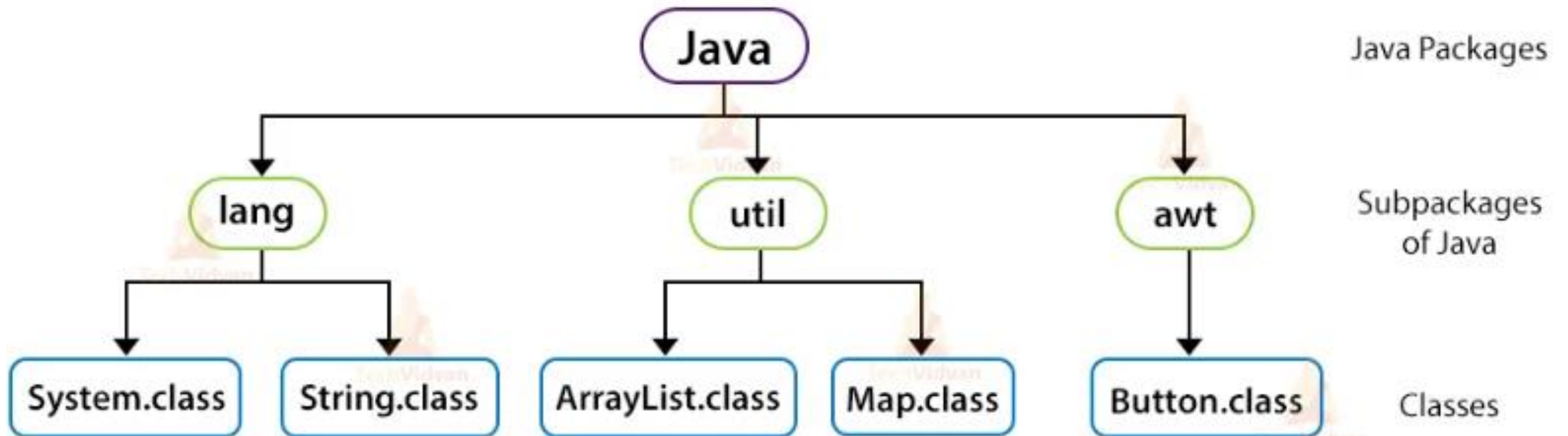
Uso

```
// Crear un Circulo estándar c1 e incrementar su radio un 30%
Circulo c1 = new Circulo();
c1.crece();
// Crear un Circulo c2 con el radio, color y centro actuales de c1
double radioC2 = c1.getRadio();
String colorC2 = c1.getColor();
int centroXC2 = c1.getCentroX(), centroYC2 = c1.getCentroY();
Circulo c2 = new Circulo(radioC2, colorC2, centroXC2, centroYC2);
// Crear una Pizarra estándar pEstandar y dibujar c2 en ella
Pizarra pEstandar = new Pizarra();
pEstandar.add(c2);
```

Reutilización y
ocultación de la info

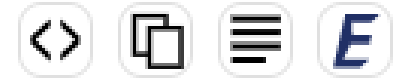


Built-in Packages in Java



Definición

```
1. package com.techvidvan.packagedemo; //package
2. class Example
3. {
4.     public static void main(String args[])
5.     {
6.         System.out.println("Welcome to Techvidvan's Java Tutorial");
7.     }
8. }
```



```
1. import packageName.*;
```

Uso

```
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JFrame;
import javax.swing.JPanel;
```

```
/**
```

```
 * Clase Pizarra: define una pizarra de determinados tamaño y título,
 * en la que se pueden dibujar objetos de tipo Circulo y Rectangulo.
```

```
 * @author Libro IIP-PRG
```

```
 * @version 2016
```

```
 */
```

```
public class Pizarra extends JFrame { ... }
```

```
public Pizarra(String t, int ancho, int alto) {
    super(t);
    setSize(ancho, alto);
    setContentPane(initPanel());
    setVisible(true);
}
```

Reutilización de Código

```
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JFrame;
import javax.swing.JPanel;
/**
 * Clase Pizarra: define una pizarra de determinados tamaño y título,
 * en la que se pueden dibujar objetos de tipo Circulo y Rectangulo.
 * @author Libro IIP-PRG
 * @version 2016
 */
public class Pizarra extends JFrame { ... }
```

Herencia

```
public Pizarra(String t, int ancho, int alto) {
    super(t);
    setSize(ancho, alto);
    setContentPane(initPanel());
    setVisible(true);
}
```

Object

`java.lang.Object`

|

+--`java.awt.Component`

|

+--`java.awt.Container`

|

+--`java.awt.Window`

|

+--`java.awt.Frame`

|

+--`javax.swing.JFrame`

Ejercicios Propuestos

1. Dada la clase `Punto` que se muestra en la figura 2.7 se pide identificar sus elementos y en concreto:
 - a)* Indicar sus atributos, de qué tipo son cada uno y cuál es su nivel de visibilidad.
 - b)* Escribir el perfil de los métodos constructores. ¿En qué se diferencian del resto de métodos? ¿Para qué se utilizan?
 - c)* Identificar los métodos modificadores.
 - d)* Identificar los métodos consultores.


```

/**
 * Clase Punto: define puntos en un espacio bidimensional entero
 * con la funcionalidad que se indica a continuacion.
 * @author Libro IIP-PRG
 * @version 2016
 */
public class Punto {
    private int x; // abscisa del punto
    private int y; // ordenada del punto

    /** Crea un Punto (abs, ord). */
    public Punto(int abs, int ord) { x = abs; y = ord; }

    /** Crea un punto (0, 0). */
    public Punto() { this(0, 0); }

    /** Devuelve la abscisa del Punto. */
    public int getX() { return x; }

    /** Devuelve la ordenada del Punto. */
    public int getY() { return y; }

    /** Actualiza la abscisa del Punto a nuevaX. */
    public void setX(int nuevaX) { x = nuevaX; }

    /** Actualiza la ordenada del Punto a nuevaY. */
    public void setY(int nuevaY) { y = nuevaY; }

    /** Actualiza las coordenadas del Punto a nuevaX y nuevaY. */
    public void asignar(int nuevaX, int nuevaY) { x = nuevaX; y = nuevaY; }

    /** Devuelve la distancia del Punto al origen. */
    public double distOrigen() { return Math.sqrt(x * x + y * y); }

    /** Devuelve un String con las coordenadas del punto en el formato
     * matemático habitual: (abscisa, ordenada). */
    public String toString() { return "(" + x + ", " + y + ")"; }
}

```

2. Dada la clase `Punto` del ejercicio anterior se pide escribir las instrucciones Java para:
- a)* Declarar y crear un objeto de tipo `Punto` cuyo nombre sea `p1`.
 - b)* Mostrar por pantalla la distancia al origen de dicho punto.
 - c)* Escribir la clase `PruebaPunto` en cuyo `main` se deben incluir las instrucciones anteriores. Compilar y ejecutar el programa.

3. ¿Qué error tiene el siguiente programa?

```
public class PruebaCirculo {  
    public static void main(String[] args) {  
        Circulo c = new Circulo(2.5, "rojo", 1, 1);  
        System.out.println("El radio del circulo es: " + c.radio);  
    }  
}
```

```

public class Circulo {
    private double radio;
    private String color;
    private int centroX, centroY;

    /** Crea un Circulo de radio r, color c y centro en (px, py). */
    public Circulo(double r, String c, int px, int py) {
        radio = r; color = c; centroX = px; centroY = py;
    }
    /** Crea un Circulo de radio 50, negro y centro en (100, 100). */
    public Circulo() {
        radio = 50; color = "negro"; centroX = 100; centroY = 100;
    }

    /** Devuelve el radio del Circulo. */
    public double getRadio() { return radio; }
    /** Devuelve el color del Circulo. */
    public String getColor() { return color; }
    /** Devuelve la abscisa del centro del Circulo. */
    public int getCentroX() { return centroX; }
    /** Devuelve la ordenada del centro del Circulo. */
    public int getCentroY() { return centroY; }
}

```

```

    /** Actualiza el radio del Circulo a nuevoRadio. */
    public void setRadio(double nuevoRadio) { radio = nuevoRadio; }
    /** Actualiza el color del Circulo a nuevoColor. */
    public void setColor(String nuevoColor) { color = nuevoColor; }
    /** Actualiza el centro del Circulo a (nuevoCentroX, nuevoCentroY). */
    public void setCentro(int nuevoCentroX, int nuevoCentroY) {
        centroX = nuevoCentroX; centroY = nuevoCentroY;
    }
    /** Desplaza el centro del Circulo a la derecha 10 unidades. */
    public void aLaDerecha() { centroX = centroX + 10; }
    /** Incrementa un 30% el radio del Circulo. */
    public void crece() { radio = radio * 1.3; }
    /** Decrementa un 30% el radio del Circulo. */
    public void decrece() { radio = radio / 1.3; }

    /** Devuelve el área del Circulo. */
    public double area() { return 3.14 * radio * radio; }
    /** Devuelve el perímetro del Circulo. */
    public double perimetro() { return 2 * 3.14 * radio; }

    /** Devuelve un String con las componentes del Circulo en el
     * siguiente formato: Circulo de radio r, color c y centro (x, y). */
    public String toString() {
        return "Circulo de radio " + radio + ", color " + color
            + " y centro (" + centroX + ", " + centroY + ")";
    }
}

```

}

4. Se pide completar el código de la clase Cuadrado (figura 2.8) para que tenga una funcionalidad similar a la clase Circulo.

```
public class Cuadrado {
    private ... lado;
    private ... color;
    private ... centroX;
    private ... centroY;

    /** Crea un Cuadrado de lado 50, negro y centro en (100, 100). */
    public Cuadrado() { ... }
    /** Crea un Cuadrado de lado l, color c y centro en (px, py). */
    public Cuadrado(double l, String c, int px, int py) {
        ...
    }

    /** Devuelve el lado del Cuadrado. */
    public double getLado() { ... }
    /** Devuelve el color del Cuadrado. */
    public String getColor() { ... }
    /** Devuelve el centro del Cuadrado. */
    public int getCentroX() { ... }
    /** Devuelve el centro del Cuadrado. */
    public int getCentroY() { ... }

    /** Actualiza el lado del Cuadrado a nuevoLado. */
    public void setLado(double nuevoLado) { ... }
    /** Actualiza el color del Cuadrado a nuevoColor. */
    public void setColor(String nuevoColor) { ... }
    /** Actualiza el centro del Cuadrado. */
    public void setCentro(int px, int py) { ... }

    /** Desplaza el centro del Cuadrado a la derecha 10 unidades. */
    public void aLaDerecha() { ... }
    /** Incrementa un 30% el lado del Cuadrado. */
    public void crece() { ... }
    /** Decrementa un 30% el lado del Cuadrado. */
    public void decrece() { ... }

    /** Devuelve el área del Cuadrado. */
    public double area() { ... }
    /** Devuelve el perímetro del Cuadrado. */
    public double perimetro() { ... }
    /** Devuelve un String con las componentes del Cuadrado en el
     * formato: Cuadrado de lado l, color c y centro (x, y). */
    public String toString() { ... }
}
```

5. Modificar el programa PrimerPrograma (figura 2.3) para usar Cuadrado como tipo de r1 en lugar de Rectangulo.

```
public class PrimerPrograma {
    public static void main(String[] args) {
        // Crear la Pizarra miPizarra, de título "ESPACIO DIBUJO" y
        // tamaño 300 x 300 píxeles
        Pizarra miPizarra = new Pizarra("ESPACIO DIBUJO", 300, 300);
        // Crear un Circulo c1 de radio 50, amarillo y con centro
        // en (100, 100)
        Circulo c1 = new Circulo(50, "amarillo", 100, 100);
        // Crear un Rectangulo r1 30 x 30, azul y con centro
        // en (125, 125)
        Rectangulo r1 = new Rectangulo(30, 30, "azul", 125, 125);
        // Crear un Rectangulo r2 100 x 10, rojo y con centro
        // en (50, 155)
        Rectangulo r2 = new Rectangulo(100, 10, "rojo", 50, 155);

        // Añadir c1 a miPizarra, dibujándolo
        miPizarra.add(c1);
        // Añadir r1 a miPizarra, dibujándolo
        miPizarra.add(r1);
        // Añadir r2 a miPizarra, dibujándolo
        miPizarra.add(r2);
    }
}
```

6. Modificar la clase Circulo para sustituir los dos atributos centroX y centroY por un único atributo centro de tipo Punto.

```
public class Circulo {
    private double radio;
    private String color;
    private int centroX, centroY;

    /** Crea un Circulo de radio r, color c y centro en (px, py). */
    public Circulo(double r, String c, int px, int py) {
        radio = r; color = c; centroX = px; centroY = py;
    }
    /** Crea un Circulo de radio 50, negro y centro en (100, 100). */
    public Circulo() {
        radio = 50; color = "negro"; centroX = 100; centroY = 100;
    }

    /** Devuelve el radio del Circulo. */
    public double getRadio() { return radio; }
    /** Devuelve el color del Circulo. */
    public String getColor() { return color; }
    /** Devuelve la abscisa del centro del Circulo. */
    public int getCentroX() { return centroX; }
    /** Devuelve la ordenada del centro del Circulo. */
    public int getCentroY() { return centroY; }
```

```
    /** Actualiza el radio del Circulo a nuevoRadio. */
    public void setRadio(double nuevoRadio) { radio = nuevoRadio; }
    /** Actualiza el color del Circulo a nuevoColor. */
    public void setColor(String nuevoColor) { color = nuevoColor; }
    /** Actualiza el centro del Circulo a (nuevoCentroX, nuevoCentroY). */
    public void setCentro(int nuevoCentroX, int nuevoCentroY) {
        centroX = nuevoCentroX; centroY = nuevoCentroY;
    }
    /** Desplaza el centro del Circulo a la derecha 10 unidades. */
    public void aLaDerecha() { centroX = centroX + 10; }
    /** Incrementa un 30% el radio del Circulo. */
    public void crece() { radio = radio * 1.3; }
    /** Decrementa un 30% el radio del Circulo. */
    public void decrece() { radio = radio / 1.3; }

    /** Devuelve el área del Circulo. */
    public double area() { return 3.14 * radio * radio; }
    /** Devuelve el perímetro del Circulo. */
    public double perimetro() { return 2 * 3.14 * radio; }

    /** Devuelve un String con las componentes del Circulo en el
     * siguiente formato: Circulo de radio r, color c y centro (x, y). */
    public String toString() {
        return "Circulo de radio " + radio + ", color " + color
            + " y centro (" + centroX + ", " + centroY + ")";
    }
}
```

}

Bibliografía

- Prieto, N. et al. (2016). *Empezar a programar utilizando Java*. Universitat Politècnica de Valencia. 3ª edición. <https://0110o2ybj-y-https-elibro-net.uta.metaproxy.org/es/ereader/uta/57434/>