

CLASE RECTÁNGULO PASO A PASO

EJERCICIO 1 - Creación de una Clase y atributos sencillos

Imagina que quieres escribir una clase que represente un rectángulo en el plano. Para ello has pensado en los siguientes atributos:

- Atributos x1, y1, que representan la coordenadas del vértice inferior izquierdo del rectángulo. Ambos de tipo double (números reales).
- Atributos x2, y2, que representan las coordenadas del vértice superior derecho del rectángulo. También de tipo double (números reales).

Con estos dos puntos (x1, y1) y (x2, y2) se puede definir perfectamente la ubicación de un rectángulo en el plano. Escribe una clase que contenga todos esos atributos teniendo en cuenta que queremos que sea una clase visible desde cualquier parte del programa y que sus atributos sean también accesibles desde cualquier parte del código.

SOLUCIÓN

```
public class Rectangulo {  
    public double x1, y1; // Vértice inferior izquierdo  
    public double x2, y2; // Vértice superior derecho  
}
```

EJERCICIO 2 - Añadiendo atributos estáticos

Ampliar el ejercicio anterior incluyendo los siguientes atributos:

- Atributo numRectangulos, que almacena el número de objetos de tipo rectángulo creados hasta el momento.
- Atributo nombre, que almacena el nombre que se le quiera dar a cada rectángulo.
- Atributo nombreFigura, que almacena el nombre de la clase, es decir, "Rectángulo".
- Atributo PI, que contiene el nombre de la constante PI con una precisión de cuatro cifras decimales.

No se desea que los atributos nombre y numRectangulos puedan ser visibles desde fuera de la clase.

SOLUCIÓN

```
class Rectangulo {  
    // Atributos de clase  
    private static int numRectangulos;  
    public static final String nombreFigura= "Rectángulo";  
    public static final double PI= 3.1416; // Constante PI  
    // Atributos de objeto  
    private String nombre;  
    public double x1, y1;  
    public double x2, y2;  
}
```

EJERCICIO 3 - Métodos

Vamos a seguir ampliando la clase mediante los siguientes métodos públicos:

- Métodos obtenerNombre y establecerNombre, que permiten el acceso y modificación del atributo nombre del rectángulo.
- Método calcularSuperficie, que calcula el área encerrada por el rectángulo.
- Método calcularPerímetro, que calcula la longitud del perímetro del rectángulo.
- Método desplazar, que mueve la ubicación del rectángulo en el plano en una cantidad X (para el ejeX) y otra cantidad Y (para el eje Y). Se trata simplemente de sumar el desplazamiento X a las coordenadas x1 y x2, y el desplazamiento Y a las coordenadas y1 e y2. Los parámetros de entrada de este método serán por tanto X e Y, de tipo double.
- Método obtenerNumRectangulos, que devuelve el número de rectángulos creados hasta el momento.

Incluye la implementación de cada uno de esos métodos en la clase Rectangulo.

SOLUCIÓN

```
public String obtenerNombre() {
    return nombre;
}

public void establecerNombre(String nom) {
    nombre = nom;
}

public double calcularSuperficie() {
    double area, base, altura; // Variables locales
    // Cálculo de la base
    base = x2 - x1;
    // Cálculo de la altura
    altura = y2 - y1;
    // Cálculo del área
    area = base * altura;
    // Devolución del valor de retorno
    return area;
}

public double calcularPerimetro() {
    double perimetro, base, altura; // Variables locales
    // Cálculo de la base
    base = x2 - x1;
    // Cálculo de la altura
    altura = y2 - y1;
    // Cálculo del perímetro
    perimetro = 2 * base + 2 * altura;
    // Devolución del valor de retorno
```

```
        return perimetro;
    }

    public void desplazar(double X, double Y) {
        // Desplazamiento en el eje X
        x1 = x1 + X;
        x2 = x2 + X;
        // Desplazamiento en el eje Y
        y1 = y1 + Y;
        y2 = y2 + Y;
    }

    public static int obtenerNumRectangulos() {
        return numRectangulos;
    }
}
```

EJERCICIO 4 - Utilización de Métodos

Utilizando la clase Rectangulo indica como declararías tres objetos de esa clase llamados r1, r2, r3.

SOLUCIÓN

```
Rectangulo r1;
Rectangulo r2;
Rectangulo r3;
```

EJERCICIO 5 - Instanciación de Objetos

Ampliar el ejercicio anterior instanciando los objetos r1, r2, r3 mediante el constructor por defecto.

SOLUCIÓN

```
Rectangulo r1, r2, r3;
r1 = new Rectangulo();
r2 = new Rectangulo();
r3 = new Rectangulo();
```

EJERCICIO 6 - Manipulación de objetos

Crea un rectángulo r1, asignarle los valores x1=0, y1=0, x2=10, y2=10, calcula su área y su perímetro y mostrarlos por pantalla.

SOLUCIÓN

```
public class EjemploRectangulos {
    public static void main(String[] args) {
        Rectangulo r1, r2;
        r1 = new Rectangulo ();
        r2 = new Rectangulo ();
        r1.x1 = 0;
```

```
        r1.y1 = 0;
        r1.x2 = 10;
        r1.y2 = 10;
        r1.establecerNombre ("rectangulo1");
        System.out.printf ("PRUEBA DE USO DE LA CLASE RECTÁNGULO\n");
        System.out.printf ("-----\n\n");
        System.out.printf ("r1.x1: %4.2f\nr1.y1: %4.2f\n", r1.x1, r1.y1);
        System.out.printf ("r1.x2: %4.2f\nr1.y2: %4.2f\n", r1.x2, r1.y2);
        System.out.printf ("Perimetro: %4.2f\nSuperficie:%4.2f\n",
            r1.CalcularPerimetro(), r1.CalcularSuperficie());
        System.out.printf ("Desplazamos X=3, Y=3\n");
        r1.desplazar (3,3);
        System.out.printf ("r1.x1: %4.2f\nr1.y1: %4.2f\n", r1.x1, r1.y1);
        System.out.printf ("r1.x2: %4.2f\nr1.y2: %4.2f\n", r1.x2, r1.y2);
    }
}
```

EJERCICIO 7 - Constructores

Añade a Rectangulo tres constructores:

1. Un constructor sin parámetros (para sustituir al constructor por defecto) que haga que los valores iniciales de las esquinas del rectángulo sean (0,0) y (1,1);
2. Un constructor con cuatro parámetros, x1, y1, x2, y2, que rellene los valores iniciales de los atributos del rectángulo con los valores proporcionados a través de los parámetros.
3. Un constructor con dos parámetros, base y altura, que cree un rectángulo donde el vértice inferior derecho esté ubicado en la posición (0,0) y que tenga una base y una altura tal y como indican los dos parámetros proporcionados.

SOLUCIÓN

```
// Constructor 1
public Rectangulo () {
    x1 = 0.0;
    y1 = 0.0;
    x2 = 1.0;
    y2 = 1.0;
}

// Constructor 2
public Rectangulo (double x1, double y1, double x2, double y2) {
    this.x1 = x1;
    this.y1 = y1;
    this.x2 = x2;
    this.y2 = y2;
}
```

```
// Constructor 3
public Rectangulo (double base, double altura) {
    this.x1= 0.0;
    this.y1= 0.0;
    this.x2= base;
    this.y2= altura;
}
```

EJERCICIO 8 - Constructor de copia

Añade a Rectangulo un constructor copia.

SOLUCIÓN

```
// Constructor copia
public Rectangulo (Rectangulo r) {
    this.x1 = r.x1;
    this.y1 = r.y1;
    this.x2 = r.x2;
    this.y2 = r.y2;
}
```