5.F. Constructores.

1. Constructores.

1.3. Utilización de constructores.

Una vez que dispongas de tus propios constructores personalizados, la forma de utilizarlos es igual que con el constructor por defecto (mediante la utilización de la palabra reservada `new`) pero teniendo en cuenta que si has declarado parámetros en tu método constructor, tendrás que llamar al constructor con algún valor para esos parámetros.

Un ejemplo de utilización del constructor que has creado para la clase Punto en el apartado anterior podría ser:

```
Punto p1;
```

```
p1= new Punto (10, 7);
```

En este caso no se estaría utilizando el constructor por defecto sino el constructor que acabas de implementar en el cual además de reservar memoria se asigna un valor a algunos de los atributos.

Para saber más

Puedes echar un vistazo al artículo sobre constructores de una clase Java en los manuales de Oracle (en inglés):

Providing Constructors for Your Classes.

Ejercicio resuelto

Ampliar el ejercicio de la clase **Rectangulo** añadiéndole 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

En el caso del primer constructor lo único que hay que hacer es "rellenar" los atributos x1, y1, x2, y2 con los valores 0, 0, 1, 1:

```
public Rectangulo ()
```

```
{
```

```
    x1= 0.0;
```

```
    y1= 0.0;
```

```
    x2= 1.0;
```

```
    y2= 1.0;
```

```
}
```

Para el segundo constructor es suficiente con asignar a los atributos x1, y1, x2, y2 los valores de los parámetros x1, y1, x2, y2. Tan solo hay que tener en cuenta que al tener los mismos nombres los parámetros del método que los atributos de la clase, estos últimos son ocultados por los primeros y para poder tener acceso a ellos tendrás que utilizar el operador de autorrerferencia `this`:

```
public Rectangulo (double x1, double y1, double x2, double y2)
```

```
{
```

```
    this.x1= x1;
```

```
    this.y1= y1;
    this.x2= x2;
    this.y2= y2;
}
```

En el caso del tercer constructor tendrás que inicializar el vértice (x1, y1) a (0,0) y el vértice (x2,y2) a (0 + base, 0 + altura), es decir a (base, altura):

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

El código completo lo compondrán los ficheros Rectangulo.java y EjemploRectangulos02.java que incorporaremos en un mismo proyecto:

Rectangulo.java

```
package ejemplorectangulos02;

/**----------------------------------------------------------------
 * Clase Rectangulo.
 * Incluye constructores.
 ----------------------------------------------------------------*/
public class Rectangulo {

   // Atributos de clase (estáticos)
   private static int numRectangulos;                    // Número total de rectángulos creados
   public static final String nombreFigura= "Rectángulo";     //  Nombre de la clase
   public static final double PI= 3.1416;                // Constante PI

   // Atributos de objeto
   private String nombre;       // Nombre del rectángulo
   public double x1, y1;        // Vértice inferior izquierdo
   public double x2, y2;        // Vértice superior derecho

//----------------
// Constructores
//----------------
```

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

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

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



//--------------------------------
// Métodos estáticos (de clase)
//--------------------------------

    // Métodos de estáticos públicos
    // ------------------------------

    // Método obtenerNumRectangulos
    public static int obtenerNumRectangulos () {
       return numRectangulos;
    }



//-------------------
```

```
// Métodos de objeto
//-------------------

   //Métodos públicos
   //-----------------

   // Método obtenerNombre
   public String obtenerNombre () {
      return nombre;
   }

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

   // Método CalcularSuperficie
   public double CalcularSuperficie () {
      double area, base, altura;

      // 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;
   }

   // Método CalcularPerimetro
   public double CalcularPerimetro () {
      double perimetro, base, altura;

      // 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;
    }

    // Método desplazar
    public void desplazar (double X, double Y) {

        // Desplazamiento en el eje X
        x1= x1 + X;
        x2= x2 + X;

        // Desplazamiento en el eje X
        y1= y1 + Y;
        y2= y2 + Y;
    }
}
```

EjemploRectangulos02.java

```
/*
 * Ejemplo de uso de la clase Rectangulo con constructores
 */
package ejemplorectangulos02;

/**
 *
 * Programa Principal (clase principal)
 */
public class EjemploRectangulos02 {


    public static void main(String[] args) {
```

```
        Rectangulo r1, r2, r3;

        System.out.printf ("PRUEBA DE USO DE LA CLASE RECTÁNGULO\n");
        System.out.printf ("-----------------------------------\n\n");
        System.out.printf ("Creando rectángulos...\n\n");
        r1= new Rectangulo ();
        r2= new Rectangulo (1,1, 3,3);
        r3= new Rectangulo (10, 5);

        System.out.printf ("Recángulo r1: \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 ("Recángulo r2: \n");
        System.out.printf ("r2.x1: %4.2f\nr2.y1: %4.2f\n", r2.x1, r2.y1);
        System.out.printf ("r2.x2: %4.2f\nr2.y2: %4.2f\n", r2.x2, r2.y2);
        System.out.printf ("Perimetro: %4.2f\nSuperficie: %4.2f\n", r2.CalcularPerimetro(), r2.CalcularSuperficie());
        System.out.printf ("Recángulo r3: \n");
        System.out.printf ("r3.x1: %4.2f\nr3.y1: %4.2f\n", r3.x1, r3.y1);
        System.out.printf ("r3.x2: %4.2f\nr3.y2: %4.2f\n", r3.x2, r3.y2);
        System.out.printf ("Perimetro: %4.2f\nSuperficie: %4.2f\n", r3.CalcularPerimetro(), r3.CalcularSuperficie());

    }
}
```