

Documentación de código con Javadoc

Actividad:

Documentando código fuente con Javadoc

Los tres archivos siguientes con código fuente en Java forman el proyecto de una aplicación de consola. La aplicación se llama ProgramaComposicionCono, y crea cinco objetos cono e imprime por pantalla sus atributos. La clase Cono tiene varios atributos, y uno de ellos es un objeto de la clase Circulo (la base del cono).

En esta actividad se debe:

- 1) Crear el proyecto y compilarlo.

```
1 public class ProgramaComposicionCono {
2
3     private static void imprimir (Object datos) {
4         System.out.println(datos);
5     }
6
7     public static void main (String[] args) {
8         java.util.Random aleatorio = new java.util.Random();
9         for (int i=1; i<=5; i++) {
10             int radio = aleatorio.nextInt( bound: 9)+1;
11             Cono cono = new Cono( cX: 0, cY: 0, radio, i, color: "Azul");
12             cono.imprimir();
13         }
14     }
15 }
```

```
1 public class Cono {
2
3     private Circulo base;
4     private float altura;
5     private String color;
6     public Cono (float cX, float cY, float r, float h, String color) { // constructor
7         this.base = new Circulo(cX, cY, r);
8         this.altura = h;
9         this.color = color;
10    }
11    public Circulo getBase() {
12        return base;
13    }
14    public void setBase(Circulo base) {
15        this.base = base;
16    }
17    public float getAltura() {
18        return altura;
19    }
20    public void setAltura(float altura) {
21        this.altura = altura;
22    }
23    public String getColor() {
24        return color;
25    }
26    public void setColor(String color) {
27        this.color = color;
28    }
29    public void imprimir () {
30        base.imprimir();
31        System.out.println(" h=" + altura + " c=" + color);
32    }
33
34 }
```

```

1 public class Circulo {
2
3     private float x, y;
4     private float radio;
5     public Circulo (float cX, float cY, float r) { // constructor
6         this.x = cX;
7         this.y = cY;
8         this.radio = r;
9     }
10    public float getX() { return x; }
11    public void setX(float x) { this.x = x; }
12    public float getY() { return y; }
13    public void setY(float y) { this.y = y; }
14    public float getRadio() { return radio; }
15    public void setRadio(float radio) { this.radio = radio; }
16    public float area() { return (float)Math.PI*radio*radio; }
17    public void imprimir () {
18        System.out.print(" x=" + x);
19        System.out.print(" y=" + y);
20        System.out.print(" r=" + radio);
21        System.out.print(" A=" + area());
22    }
23 }

```

2) Generar las plantillas de comentarios Javadoc para todas las clases y métodos del programa.

3) Completar los comentarios generados por el IDE.

```

1 /**
2  * Clase que representa el programa principal. Se trata de una aplicación de consola
3  * y crea cinco objetos cono e imprime por pantalla sus atributos.
4  * @Author Desconocido
5  * @Version Desconocida
6  */
7 public class ProgramaComposicionCono {
8
9     /**
10     * <h1>Método estatico que imprime datos por consola</h1>
11     * @param datos datos a imprimir por consola
12     */
13     private static void imprimir (Object datos) {
14         System.out.println(datos);
15     }
16
17     /**
18     * <h1>Punto de entrada/arranque de la aplicación</h1>
19     * @param args
20     */
21     public static void main (String[] args) {
22         java.util.Random aleatorio = new java.util.Random();
23         for (int i=1; i<=5; i++) {
24             int radio = aleatorio.nextInt( bound: 9)+1;
25             Cono cono = new Cono( cX: 0, cY: 0, radio, i, color: "Azul");
26             cono.imprimir();
27         }
28     }
29 }
30 }

```

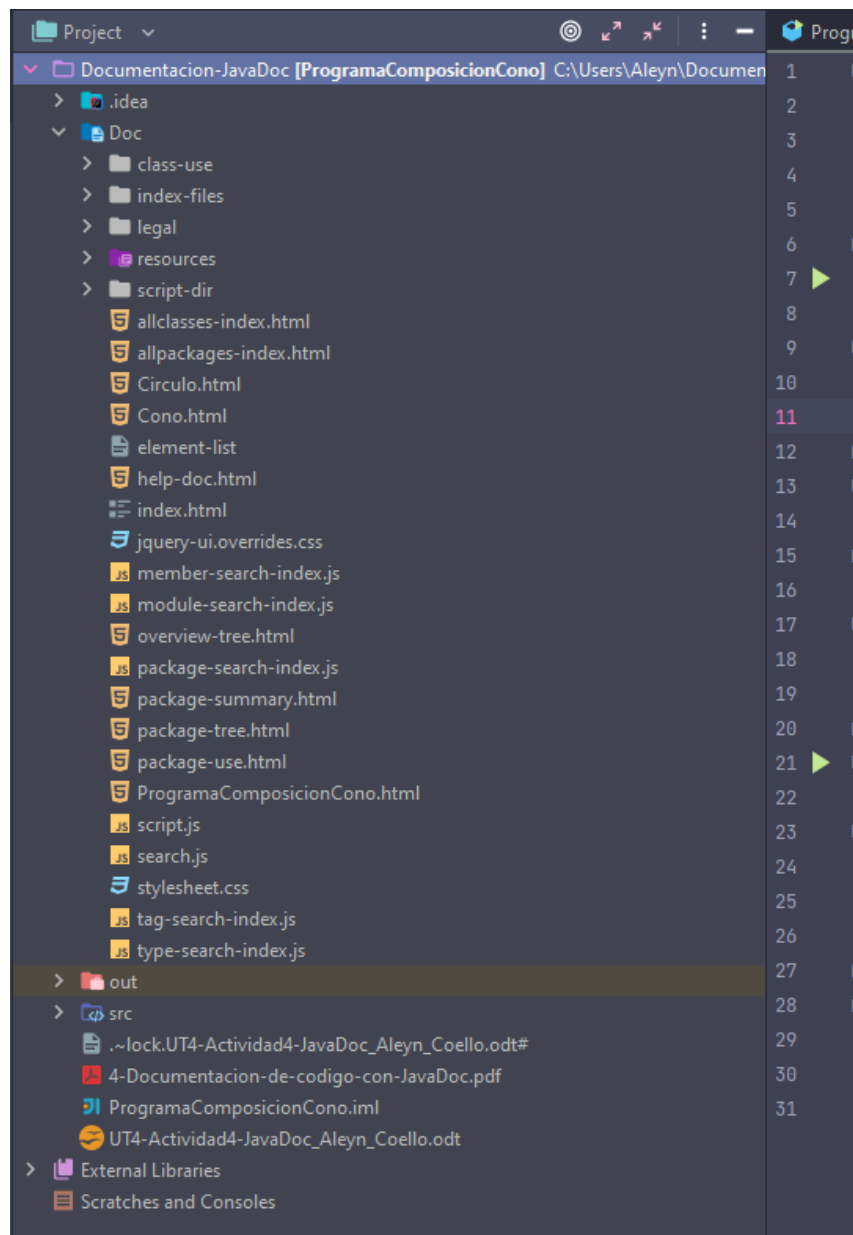
```
1  /**
2   * <h1>Clase que representa un cono</h1>
3   * @Author Desconocido
4   * @Version Desconocida
5   */
6   public class Cono {
7
8       private Circulo base;
9       private float altura;
10      private String color;
11
12      /**
13       * <h1>Constructor de la clase Cono</h1>
14       * @param cX cX
15       * @param cY cY
16       * @param r Radio del circulo
17       * @param h Altura del cono
18       * @param color Color
19       */
20      public Cono (float cX, float cY, float r, float h, String color) {
21          this.base = new Circulo(cX, cY, r);
22          this.altura = h;
23          this.color = color;
24      }
25
26      /**
27       * <h3>Método para obtener la base del circulo</h3>
28       * @return la base del circulo
29       */
30      public Circulo getBase() {
31          return base;
32      }
33
34      /**
35       * <h3>Método para establecer la base del circulo</h3>
36       * @param base base del circulo
37       */
38      public void setBase(Circulo base) {
39          this.base = base;
40      }
41  }
```

```
42  /**
43   * <h3>Método para obtener la altura del cono</h3>
44   * @return altura del cono
45   */
46   public float getAltura() {
47       return altura;
48   }
49
50  /**
51   * <h3>Método para establecer la altura del cono</h3>
52   * @param altura altura del cono
53   */
54   public void setAltura(float altura) {
55       this.altura = altura;
56   }
57
58  /**
59   * <h3>Método para obtener el color del cono</h3>
60   * @return color del cono
61   */
62   public String getColor() {
63       return color;
64   }
65
66  /**
67   * <h3>Método para establecer el color del cono</h3>
68   * @param color color del cono
69   */
70   public void setColor(String color) {
71       this.color = color;
72   }
73
74  /**
75   * <h3>Método para imprimir la base, altura y color</h3>
76   */
77   public void imprimir () {
78       base.imprimir();
79       System.out.println(" h=" + altura + " c=" + color);
80   }
81
82 }
```

```
1  /**
2   * <h1>Clase que representa un círculo</h1>
3   * @Author Desconocido
4   * @Version Desconocida
5   */
6   public class Circulo {
7
8       private float x, y;
9       private float radio;
10
11      /**
12       * <h1>Constructor de la clase Circulo</h1>
13       * @param cX cX
14       * @param cY cY
15       * @param r radio del círculo
16       */
17      public Circulo (float cX, float cY, float r) { // constructor
18          this.x = cX;
19          this.y = cY;
20          this.radio = r;
21      }
22
23      /**
24       * <h3>Método para obtener el valor de x</h3>
25       * @return valor de x
26       */
27      public float getX() { return x; }
28
29      /**
30       * <h3>Método para establecer el valor de x</h3>
31       * @param x x
32       */
33      public void setX(float x) { this.x = x; }
34
35      /**
36       * <h3>Método para obtener el valor de y</h3>
37       * @return valor de y
38       */
39      public float getY() { return y; }
```

```
39     public float getY() { return y; }
40
41     /**
42      * <h3>Método para establecer el valor de y</h3>
43      * @param y y
44      */
45     public void setY(float y) { this.y = y; }
46
47     /**
48      * <h3>Método para obtener el valor del radio del círculo</h3>
49      * @return valor del radio
50      */
51     public float getRadio() { return radio; }
52
53     /**
54      * <h3>Método para establecer el valor del radio del círculo</h3>
55      * @param radio radio del círculo
56      */
57     public void setRadio(float radio) { this.radio = radio; }
58
59     /**
60      * <h3>Método para hallar el área del círculo</h3>
61      * @return valor del área del círculo
62      */
63     public float area() { return (float)Math.PI*radio*radio; }
64
65     /**
66      * <h3>Método para imprimir por consola los valores de x, y, radio y área</h3>
67      */
68     public void imprimir () {
69         System.out.print(" x=" + x);
70         System.out.print(" y=" + y);
71         System.out.print(" r=" + radio);
72         System.out.print(" A=" + area());
73     }
74 }
```

- 4) Generar los archivos HTML de documentación y visualizarlos.



PACKAGE

CLASS

USE

TREE

INDEX

HELP

PACKAGE: DESCRIPTION | RELATED PACKAGES | CLASSES AND INTERFACES

SEARCH:

Unnamed Package

Classes

Class	Description
Circulo	Clase que representa un cÁrculo
Cono	Clase que representa un cono
ProgramaComposicionCono	Clase que representa el programa principal.

PACKAGE

CLASS

USE

TREE

INDEX

HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD

DETAIL: FIELD | CONSTR | METHOD

SEARCH:

Class Circulo

java.lang.Object[Ⓜ]
Circulo

public class Circulo
extends Object[Ⓜ]

Clase que representa un cÁrculo

Field Summary

Fields

Modifier and Type	Field	Description
private float	radio	
private float	x	
private float	y	

Constructor Summary

Constructors

Constructor	Description
Circulo(float cX, float cY, float r)	Constructor de la clase CÁrculo

Method Summary