

# Java Doc y E/S gráfica



# JavaDoc en NetBeans

---

## □ NetBeans lo genera

- Run > Generate Javadoc (*nombre\_paquete*)

## □ Se indica con palabras clave

/ \*\*

\* Descripción programa

\* @author nombre

\* @param explicación parámetros

\* @return explicación valor de salida

\* @exception explicación excepción

\* @version x.x

\* /

# Código con comentarios javadoc

---

```
/**
 * Implementacion de diversos constructores para un circulo
 * @autor rene martinez
 * @version 1.0, 1/marzo/17
 */
public class CirculoDoc {
    private final double PI = 3.141592653589793238;
    private double x,y,r ;
    /**
     * Crea un circulo a partir de su centro y radio
     * @param x coordenada x del centro
     * @param y coordenada y del centro
     * @param r radio
     */
    public CirculoDoc (double x, double y, double r) {
        this.x=x; this.y = y; this.r=r;
    }
    /**
     * Crea un circulo en el origen
     * @param r radio
     */
    public CirculoDoc (double r) {this(0.0,0.0,r);}
}
```

# Código con comentarios javadoc (cont)

```
/**
 * Crea un circulo copia de otro
 * @param c circulo del que se van a tomar centro y radio
 */
public CirculoDoc (CirculoDoc c) {this(c.x,c.y,c.r);}
/**
 * Crea un circulo en el origen de radio 1
 */
public CirculoDoc () {this(0.0,0.0,1.0);}
/**
 * Obtiene el area de un circulo
 * @return el valor del area
 */
public double area() { return PI * r * r;}
/**
 * Proporciona las coordenadas del centro de un circulo
 */
public void imprimirCentro() {
    System.out.println("(" + x + ", " + y + ")");
}
}
```

# Código con comentarios javadoc (cont)

The screenshot shows a web browser window titled "CirculoDoc" displaying the Javadoc documentation for the `CirculoDoc` class. The browser's address bar shows the file path: `file:///localhost/Users/mina/NetBeansProjects/EstructuraDatos/dist/javadoc/index.html`. The left sidebar contains a list of "All Classes" and "Packages". The main content area shows the "Class" tab selected, displaying the class hierarchy, a description, and summaries for constructors and methods.

**Overview Package Class Use Tree Deprecated Index Help**

[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#)  
[SUMMARY: NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

Practicas  
**Class CirculoDoc**

`java.lang.Object`  
└─ `Practicas.CirculoDoc`

`public class CirculoDoc`  
`extends java.lang.Object`

Este programa implementa diversos constructores para un circulo

**Constructor Summary**

<a href="#">CirculoDoc()</a>	Crea un circulo en el origen de radio 1
<a href="#">CirculoDoc(CirculoDoc c)</a>	Crea un circulo copia de otro
<a href="#">CirculoDoc(double r)</a>	Crea un circulo en el origen
<a href="#">CirculoDoc(double x, double y, double r)</a>	Crea un circulo a partir de su centro y radio

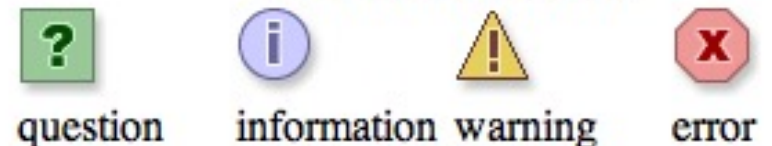
**Method Summary**

double	<a href="#">area()</a>	Obtiene el area de un circulo
void	<a href="#">imprimirCentro()</a>	Proporciona las coordenadas del centro de un circulo
static void	<a href="#">main(java.lang.String[] args)</a>	

# Java Foundation Classes (JFC)

- Bibliotecas Java para desarrollo de GUIs en múltiples plataformas
  - Swing
  - Java 2D
    - Gráficas 2D
  - Pluggable Look-and-Feel Support
    - Vista y sentido del programa
  - Accesability
    - Dispositivos
  - Internationalization
    - Idiomas

Icons used by JOptionPane  
(Java look and feel)



(Windows look and feel)



# Programar con GUI

---

- GUI (Graphical User Interface)
  - Interfaz Gráfica de Usuario
    - Usa ventanas para E/S o comunicación con el usuario
  - Se crean a partir de componentes de la GUI
    - Controles o widgets (accesorios de ventana)
- Java nace con biblioteca de componentes
  - awt (Abstract Window Toolkit)
    - Paquete: java.awt
    - Componentes pesados: Enlazados con las herramientas de interfaz gráfica de la plataforma (windows, unix, MacOS)
  - swing
    - Paquete: javax.swing
    - Componentes ligeros: Mas independientes del sistema operativo

# Entrada/Salida de datos

---

## □ JOptionPane

```
java.lang.Object
├── java.awt.Component
│   ├── java.awt.Container
│       ├── javax.swing.JComponent
│           └── javax.swing.JOptionPane
```

## □ Salida

- JOptionPane.showMessageDialog  
(Componente\_padre, mensaje, título, tipo mensaje, ícono);

## □ Entrada

- JOptionPane.showInputDialog (mensaje);



# Mensaje default

---

```
import javax.swing.JOptionPane;

public class HolaPanel {

    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null,
                                     "Hola \n Java con GUI \n");
        System.exit(0);
    }
}
```



# Tipos mensaje

- ❑ JOptionPane.INFORMATION\_MESSAGE
- ❑ JOptionPane.WARNING\_MESSAGE
- ❑ JOptionPane.QUESTION\_MESSAGE
- ❑ JOptionPane.ERROR\_MESSAGE
- ❑ JOptionPane.PLAIN\_MESSAGE



# Java Doc y E/S gráfica



# Usar una variable para el mensaje

```
import javax.swing.JOptionPane;
public class VariableString {
    public static void main(String[] args) {
        String salida="";
        for (int i=0; i<10; i++)
            salida += i+" \n";
        JOptionPane.showMessageDialog(null,salida);
    }
}
```



# Formato a los datos

□ En paquete java.text

□ Clase: DecimalFormat

```
DecimalFormat id = new DecimalFormat( patron );
```

□ Método

■ `format(tipo dato):String`

□ Patrón

■ 0 un dígito

■ # un digito, si el valor es de 0 no se muestra

■ . separador decimales

■ , separador miles

123456.789

###,###.###

123,456.789

123456.789

###.##

123456.79

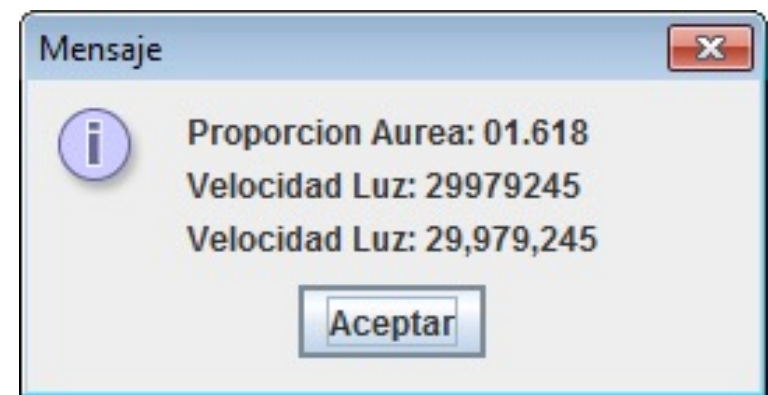
123.78

000000.000

000123.780

# Ejemplo DecimalFormat

```
import java.text.*;
import javax.swing.JOptionPane;
public class FormatoDF {
    public static final float NUM_AUREO = 1.618033f;
    public static final double VEL_LUZ = 29979245;
    public static void main(String[] args) {
        DecimalFormat dfAureo = new DecimalFormat("00.###");
        DecimalFormat dfLuz = new DecimalFormat("###,###");
        String salida="Proporcion Aurea: "+dfAureo.format(NUM_AUREO);
        salida += "\nVelocidad Luz: "+dfAureo.format(VEL_LUZ);
        salida += "\nVelocidad Luz: "+dfLuz.format(VEL_LUZ);
        JOptionPane.showMessageDialog(null,salida);
    }
}
```



# Entrada de datos

---

```
import javax.swing.JOptionPane;
public class EntradaDatos {
    public static void main(String[] args) {
        String linea;
        int entero;
        double flotante;
        linea= JOptionPane.showInputDialog
            ("Introduce un número entero ");
        entero=Integer.parseInt(linea);
        linea= JOptionPane.showInputDialog
            ("Introduce un número real");
        flotante=Double.parseDouble(linea);
        JOptionPane.showMessageDialog(null,
            "El número entero es: "+entero
            +"\nEl numero flotante es:"+flotante);
    }
}
```

# Ventanas EntradaDatos

