

JAVA Programación Orientada a Objetos

Introducción

Documentar tu código es una práctica fundamental para mantenerlo legible y comprensible, tanto para ti como para otros desarrolladores que puedan trabajar en el proyecto en el futuro. En Java, la documentación se realiza principalmente mediante comentarios dentro del código fuente. Aquí encontrarás algunos consejos básicos para documentar tus proyectos en Java de manera efectiva.

1. Comentarios Descriptivos

Utiliza comentarios descriptivos para explicar el propósito de las clases, métodos y variables. Esto ayuda a comprender rápidamente qué hace cada parte del código.

```
// Este método calcula el área de un triángulo
public double calcularAreaTriangulo(double base, double altura) {
    return (base * altura) / 2;
}
```

2. Javadoc

Javadoc es una herramienta que genera documentación automáticamente a partir de comentarios en el código fuente. Utiliza etiquetas específicas para documentar clases, métodos y variables.

Funciona analizando los comentarios especialmente formateados en el código y generando archivos HTML que contienen la documentación correspondiente.

Aquí hay una explicación básica de cómo funciona Javadoc:

- **Comentarios Especiales:** Javadoc interpreta los comentarios que comienzan con `/**` y terminan con `*/`. Estos comentarios pueden incluir etiquetas especiales que Javadoc utiliza para generar la documentación.

```

/**
 * Clase que representa un Triángulo.
 */
public class Triangulo {
    /**
     * Base del triángulo.
     */
    private double base;
    /**
     * Altura del triángulo.
     */
    private double altura;
    /**
     * Constructor de la clase Triángulo.
     *
     * @param base    La base del triángulo.
     * @param altura  La altura del triángulo.
     */
    public Triangulo(double base, double altura) {
        this.base = base;
        this.altura = altura;
    }
    /**
     * Calcula el área del triángulo.
     *
     * @return El área del triángulo.
     */
    public double calcularArea() {
        return (base * altura) / 2;
    }
}

```

- **Etiquetas Javadoc:** Las etiquetas Javadoc comienzan con el símbolo @ seguido de un nombre. Estas etiquetas se utilizan para proporcionar información adicional sobre los elementos del código, como métodos, clases o variables. Algunas etiquetas comunes incluyen:
 - **@param:** Descripción de un parámetro del método.
 - **@return:** Descripción del valor de retorno del método.
 - **@throws:** Descripción de las excepciones que pueden ser lanzadas por el método.

- **@see**: Referencia a otros elementos relacionados.
- **@deprecated**: Indica que un elemento está obsoleto y debe ser evitado.
- **Generación de Documentación**: Una vez que se han agregado los comentarios Javadoc al código, la herramienta Javadoc se ejecuta en los archivos fuente. Esta herramienta analiza los comentarios y genera archivos HTML que contienen la documentación correspondiente.
- **Visualización de la Documentación**: La documentación generada por Javadoc puede ser visualizada en un navegador web. Los archivos HTML generados contienen información detallada sobre las clases, métodos, variables y otros elementos del código, junto con las descripciones y etiquetas Javadoc asociadas.

En resumen, Javadoc es una herramienta poderosa que facilita la generación de documentación clara y detallada para proyectos en Java, lo que ayuda a mejorar la comprensión y mantenibilidad del código. Mediante comentarios adecuadamente formateados y etiquetas Javadoc, los desarrolladores pueden crear documentación coherente y fácilmente accesible para sus proyectos.

3. Nombres Explicativos

Utiliza nombres de variables, métodos y clases que sean descriptivos. Esto facilita la comprensión del código sin necesidad de comentarios adicionales.

```
public class Calculadora {  
    public double sumar(double numero1, double numero2) {  
        return numero1 + numero2;  
    }  
  
    public double restar(double numero1, double numero2) {  
        return numero1 - numero2;  
    }  
}
```

4. Actualización de Documentación

Mantén la documentación actualizada a medida que el código cambia. Si realizas cambios significativos en el código, asegúrate de actualizar también los comentarios correspondientes.

Conclusión

Documentar tu código en Java es una práctica esencial para mejorar su mantenibilidad y comprensión. Utiliza comentarios descriptivos, Javadoc y nombres explicativos para hacer que tu código sea más fácil de entender para ti y para otros desarrolladores.

Recuerda que una buena documentación no solo ayuda en el presente, sino que también facilita el mantenimiento y la colaboración en el futuro.

💡 Siempre es recomendable utilizar la **documentación oficial** para obtener más detalles sobre el material proporcionado, su uso y su implementación. Puedes acceder a la información sobre javadoc desde [aquí](#).