

Programación

UD 2: Introducción a JAVA

- Estilo de Codificación -

Estilo de Codificación

1.- Introducción

2.- Normas y estilo de codificación

2.1.- Nombres de ficheros

2.2.- Organización de ficheros

2.3.- Identación

2.4.- Comentarios

2.5.- Declaraciones

2.6.- Sentencias

2.7.- Separaciones

2.8.- Nombres

1.- Introducción

1.- Introducción

Una vez realizado el diseño se realiza el proceso de codificación.

En esta etapa, el programador recibe las especificaciones del diseño y las transforma en un conjunto de instrucciones escritas en un lenguaje de programación.

A este conjunto de instrucciones se le llama **código fuente**.

En cualquier proyecto en el que trabaja un grupo de personas debe haber unas **normas de codificación y estilo**, claras y homogéneas.

Estas normas facilitan las tareas de **corrección y mantenimiento** de los programas, sobre todo cuando se realizan por personas que no los han desarrollado.

1.- Introducción

```
import java.util.Scanner;

public class Suma {

    public static void main (String[] args){
        int suma = 0;
        int contador = 0;

        while (contador < 10) {
            contador++;
            suma = suma + contador;
        }
        System.out.println("Suma => " + suma);
    }
}
```

good code
is like a
good joke
- it needs no explanation

```
import java.util.Scanner;
public class
Suma {
    public static

    void main (String[] args){
        int suma = 0; int contador = 0;
        while (contador < 10)
        {    contador++;

            suma = suma + contador; }
        System.out.println("Suma => " + suma);
        }
}
```

2.- Normas y estilo de codificación

2.- Normas y estilo de codificación

2.1.- Nombres de ficheros:

La extensión para los ficheros de código fuente es *.java*

La extensión para los ficheros compilados es *.class*

2.- Normas y estilo de codificación

2.2.- Organización de ficheros:

Comentarios:

Todos los ficheros fuente deben comenzar con un comentario que muestre:

- El nombre de la clase
- Información de la versión
- La fecha de creación y/o modificación
- Aviso de derechos de autor

```
/**  
 * Nombre de la clase  
 *  
 * @version  
 *  
 * @since  
 *  
 * @author  
 *  
 */
```


2.- Normas y estilo de codificación

2.2.- Organización de ficheros:

Package e import:

Van después de los comentarios, la sentencia package va delante de import.

Ejemplo:

```
package paquete.ejemplo;  
  
import java.util.ArrayList;
```

2.- Normas y estilo de codificación

2.3.- Identación

Como norma general se usarán cuatro espacios.

La longitud de las líneas de código no debe superar 80 caracteres.

La longitud de las líneas de comentarios no debe superar 70 caracteres.

Cuando una expresión no cabe en una sola línea: romper después de una coma, romper antes de un operador, alinear la nueva línea al principio de la anterior.

IMPORTANTE:

ctrl + i (en Eclipse)

2.- Normas y estilo de codificación

2.4.- Comentarios

Los comentarios deben contener solo la información que es relevante para la lectura y la comprensión del programa.

Existen 2 tipos de comentarios:

De documentación: Están destinados a describir la especificación del código. Se utilizan para describir las clases Java, las interfaces, los constructores, los métodos, ...

De implementación: Son para comentar algo acerca de la aplicación particular, de qué está realizando el algoritmo, ...

2.- Normas y estilo de codificación

2.5.- Declaraciones

Se recomienda declarar una variable por línea.

Inicializar las variables locales donde están declaradas y colocarlas al comienzo del bloque.

En las clases e interfaces:

- No se ponen espacios en blanco entre el nombre del método y el paréntesis “(”.
- La llave de apertura “{” se coloca en la misma línea que el nombre del método o clase.
- La llave de cierre “}” aparece en una línea aparte.

2.- Normas y estilo de codificación

2.6.- Sentencias

Cada línea debe contener una sentencia.

Si hay un bloque de sentencias, este debe ser sangrado con respecto a la sentencia que lo genera y debe estar entre llaves aunque solo tenga una sentencia.

2.- Normas y estilo de codificación

2.8.- Nombres

Si las variables o métodos tienen varias palabras para hacerlas mas entendibles, se puede utilizar una de las siguientes 3 notaciones:

CamelCase

camelCase

snake_case

kebab-case: no se puede utilizar en Java