

Estándar de codificación

1 Nombre de variable

El tipo de variable a utilizar es el UpperCamelCase, consiste en que la primera letra de cada una de las palabras será en mayúscula.

Ejemplo:

VentanaNumero

2 Sangría

3 La sangría será una implementada por defecto de Apache NetBeans IDE.

4 Comentario

Comentarios de varias líneas

Para describir un método completo en un código o un fragmento, se utilizará el comentario de varias líneas.

Ejemplo

Sintaxis:

/*El comentario comienza

continúa

continúa

.

.

.

El comentario termina*/

Comentarios de comienzo

Todos los ficheros fuente deben comenzar con un comentario en el que se lista el nombre de la clase, información de la versión, fecha, y copyright:

/*

* Nombre de la clase

*

* Informacion de la version

*

* Fecha

*

* Copyright
*/

5 Indentación

Deben tomarse 4 espacios como unidad de indentación. La construcción exacta de la indentación (espacios o tabuladores) no es crítica. Los tabuladores deben estar fijados exactamente cada 8 espacios (no cada 4).

6 Funciones

7 Espacios

8 Encabezado

9 Número de caracteres por línea

Evitar las líneas de más de 80 caracteres, ya que no son manejadas bien por muchas terminales y herramientas.

Nota: Ejemplos para uso en la documentación deben tener una longitud inferior, generalmente no más de 70 caracteres.

10 Nombre de interfaces

Partes de la declaración de una clase o interface	Notas
---	-------

Comentario de implementación de la clase o interface si fuera necesario (*...*/)	Este comentario debe contener cualquier información aplicable a toda la clase o interface que no era apropiada para estar en los comentarios de documentación de la clase o interface.
Variables de clase (static)	Primero las variables de clase public, después las protected, después las de nivel de paquete (sin modificador de acceso) , y después las private.
Variables de instancia	Primero las public, después las protected, después las de nivel de paquete (sin modificador de acceso), y después las private.

Métodos	Estos métodos se deben agrupar por funcionalidad más que por visión o accesibilidad. Por ejemplo, un método de clase privado puede estar entre dos métodos públicos de instancia. El objetivo es hacer el código más legible y comprensible.
---------	---

11 Nombre de clases

Los nombres de las clases deben ser sustantivos, en mayúsculas y minúsculas, con **la primera letra de cada palabra interna en mayúscula**. El nombre de las interfaces también debe estar en mayúscula (la primera) al igual que los nombres de las clases. Use palabras completas y se debe evitar acrónimos y abreviaturas.

Por ejemplo:

```
Interface Bicycle
Class MountainBike implements Bicycle
```

```
Interface Sport
Class Football implements Sport
```

12 Nombre de Objetos

13 Constantes

Debería estar todo en mayúsculas con palabras separadas por guiones bajos (“_”).

- Hay varias constantes utilizadas en clases predefinidas como Float, Long, String, etc.

Por ejemplo:

```
static final int MIN_WIDTH = 4;
```

```
// Algunas variables constantes utilizadas en la clase Float predefinida
public static final float POSITIVE_INFINITY = 1.0f / 0.0f;
public static final float NEGATIVE_INFINITY = -1.0f / 0.0f;
public static final float NaN = 0.0f / 0.0f;
```

14 Listas

15 Objetos

16