**Nomenclatura Java, XML, Android**



**Java Estandar Recomendado por Sun:**

**-Clases:**Para todo nombre de clase, la primera letra debe de ser Mayúscula,  
si son varias palabras se debe de intercalar entre mayúsculas  
y minúsculas, este mecanismo de nombre es llamado carnelCase  
(Por ejemplo: NombreDeClase, Animal, Factura).

**-Interfaces:**  
Aplica la misma definición de clases, pero se debe colocar la  
terminación “able” al final del nombre de la  
clases (Por ejemplo Runn**able**,Serializ**able**).

**-Metodos:**  
Para los metodos de clases, la primera letra debe ser minúscula,  
si son varias palabras se debe de intercalar entre minúsculas  
y mayúsculas, para el caso de los métodos de clases  
aplica el mecanismo del **camelCase**(Porejemplo getNombre, setEdad, buscarPorIdPersona).

**-Variables:**  
Para las variables, se aplica el caso de los métodos, donde la  
primera letra es minúscula, y las demás se deben de  
guiar por el mecanismo de **camelCase**,  
lo que es importante de destacar, es que los nombres de las  
variables, además de cumplir lo anterior, deben ser cortos y  
descriptivos en si mismo. (Por ejemplo: nombre, edad, idPersona).

**-Constantes:**

Para la constantes, su nombre debe ser escrito completamente en

Mayúsculas, y para la separación de palabras se debe

usar el underscore/guión bajo (\_). (Por ejemplo MAX\_SUMA,  
VALOR\_MULTIPLO).

**-Paquetes:**  
El nombre de los paquetes se escriben completamente en minúsculas  
y sin caracteres de separación. El prefijo del paquete siempre corresponderá a un nombre de dominio de primer nivel, tal como: es, eu, org, com, net, etc. El resto de componentes del paquete se nombrarán de acuerdo a las normas internas de organización de la empresa: departamento, proyecto, máquina, sección, organismo, área, etc. (Por ejemplo es.provincia.organismo3.formacion, java.util.ArrayList)

**Sintaxis Nombre Objetos**

|  |  |
| --- | --- |
| **Objetos, Controles y Tipos** | **Sintaxis Nombre Objetos** |
| Clases | class<nombre> |
| Módulos | mod<nombre> |
| Formularios | fmr<nombre> |
| Combobox | cb<nombre> |
| Datagrid | dg<nombre> |
| Listbox | lb<nombre> |
| Radiobutton | radb<nombre> |
| CheckBoxes | chkb<nombre> |
| TextBoxes | txt<nombre> |
| Button | btn<nombre> |
| Table | tab<nombre> |
| Integer | int<nombre> |
| Long | long<nombre> |
| Boolean | bool<nombre> |
| Object | obj<nombre> |
| String | str<nombre> |
| Double | doub<nombre> |
| Constantes | CONST\_<nombre> |
| Variable Globales | GLOBAL\_<nombre> |

**Base de Datos**

|  |  |
| --- | --- |
| Tabla | t<nombre> |
| Relaciones entre Tablas | FK<correlativo>\_t<id tabla origen> |
| Vista | vst<id vista> |
| Procedimiento Almacenado | SP\_<nombre> |
| Funciones | FUNC\_<nombre> |

**Comentarios:**

Comentarios de inicio:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Name : Nombre del Objeto  
Module : Nombre del Modulo del Objeto  
Process : Breve descripcion del Objeto  
Date : Fecha  
Type : Tipo de Objeto  
—————————————————————–  
Updated by : Autor de Actualizacion  
Date : Fecha de Ultima Actualizacion  
Ref. : Breve descripcion del Cambio  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Comentarios de bloque:  
Permiten la descripción de clases, bloques, estructuras de datos y algoritmos.  
/\*\*  
\* Esto es un comentario  
\* de bloque  
\*/

Comentarios de línea:

Son comentarios cortos localizados en una sola línea y tabulados al mismo nivel que el código que describen. Si ocupa más de una línea se utilizará un comentario de bloque. Deben estar precedidos por una línea en blanco.

/\* Esto es un comentario de línea \*/  
// Esto es otro comentario de línea

**Sangria:**

Como norma general se establecen 4 caracteres como unidad de sangría.

**Longitud de línea:**

La longitud de línea no debe superar los 80 caracteres por motivos de visualización e impresión.

**Estandar para XML:**

Todos los elementos XML deben tener una etiqueta de cierre.

<p>This is a paragraph.</p>

Las etiquetas XML son sensibles a mayusculas

<Message>This is incorrect</message>  
<message>This is correct</message>

Los elementos XML deben estar correctamente anidados.

<b><i>This text is bold and italic</b></i>

Los comentarios en XML

La sintaxis para escribir comentarios en XML es similar a la de HTML.

<- Esto es un comentario ->

**Normas de denominación:**

Los elementos XML deben seguir las siguientes reglas de nomenclatura:

- Los nombres pueden contener letras, números y otros caracteres  
- Los nombres no pueden comenzar con un número o un carácter puntuacion  
- Los nombres no pueden comenzar con las letras xml (o XML, o XML, etc)  
- Los nombres no pueden contener espacios

**Directrices de codigo para Android:**

Android sigue las convenciones del estándar de Java. Pero añaden unas cuantas reglas:

**Reglas de estilo:**

Todas las clases y métodos públicos*deben* contener un comentario con al menos una frase que describe lo que la clase o el método hace.

Ejemplo:  
/\*\* Returns the correctly rounded positive square root of a double value. \*/  
static double sqrt(double a) {…}

**Escribir métodos abreviados**

En la medida que es posible, los métodos deben ser pequeños. Sin embargo, los métodos largos son a veces adecuados. Si un método es superior a 40 líneas, se debería pensar en dividirlo sin dañar la estructura del programa.

**Orden de importación**

El orden de las declaraciones de importación es:  
- Android imports  
- Las importaciones procedentes de terceros ( com , junit , net , org )  
- java y javax

**Use espacios para el sangrado:**

Se usan cuatro espacios para los bloques. Nunca se usa la tabulacion. En caso de duda, ser coherente con el código de su alrededor.

Se usan 8 espacios para nuevas lineas, incluyendo llamadas a funciones y tareas.

Por ejemplo:

Instrument i =

someLongExpression(that, wouldNotFit, on, one, line)

**Convenciones de nomenclatura de campos**

- Los nombres no públicos, no estáticos deben comenzar con m.

- Nombres de campos estáticos deben comenzar con s.

- Los demas campos deben comenzar con una letra minúscula.

- Nombres públicos estáticos finales (constantes) son TODO\_MAYÚSCULA\_CON\_UNDERSCORES.

Por ejemplo:

public class MyClass {

public static final int SOME\_CONSTANT = 42;

public int publicField;

private static MyClass sSingleton;

int mPackagePrivate;

private int mPrivate;

protected int mProtected;

}

**Límite de longitud de línea**

Cada línea de texto en el código debe ser como máximo 100 caracteres.