



Buenas Prácticas JAVA

Convención de nombres <http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html>

Paquetes	Clases/Interfaces	Métodos	Atributos/Variables	Constantes
<ul style="list-style-type: none">• Cualquier carácter del código ASCII en minúsculas separando las palabras por un punto y sin espacios.• Empezando siempre com, edu, gov, mil, net, org o el código de dos letras definido por ISO-3166.• Si palabra reservada se añade _ al final com.int_.example.	<ul style="list-style-type: none">• Los nombres deben ser capitalizados.• Nombres simples y descriptivos.• EN caso de clases de colecciones nombrarlas en plural.• Encapsular el nombre nunca cosas como CarArray.• Las interfaces NO deben anteponer I, evitar por tanto ICar.	<ul style="list-style-type: none">• Verbos con la primera letra de cada palabra en mayúsculas menos la primera palabra.• Breve, conciso e intuitivo.• Un getter de boolean debe usar is o has en lugar de get evitando usar la palabra Not.	<ul style="list-style-type: none">• Primera letra de cada palabra en mayúsculas menos la primera palabra.• no se debe usar _ o \$ como primera letra.	<ul style="list-style-type: none">• todas las letras de la constante debe ser en mayúsculas, separando cada palabra con _ .

Sangría y espaciado

- Las líneas no mayores de 110 caracteres (aunque el estándar dice 80, las pantallas son más grandes)
- Respecto a los espacios entre líneas
 - Debe considerarse 1 línea en blanco
 - Entre métodos
 - Entre las variables locales y su primera declaración
 - Entre un bloque de comentarios o una línea simple de comentarios
 - Dos líneas
 - Entre secciones lógicas dentro de un método para mejorar la lectura
 - Entre secciones (entre la declaración de imports y la definición de clases, por ejemplo)
 - Entre definiciones de clases e interfaces

Ordenación

Aunque la ordenación de variables y métodos se basa en su visibilidad también se permite la agrupación por funcionalidades siempre respetando la precedencia por visibilidad.

1. Comentarios de clases e interfaces
2. Definición de la propia clase o interfaz
3. Variables static
 1. Public/Protected/Default/Private
4. Variables instanciables
 1. Public/Protected/Default/Private
5. Constructores
6. Métodos (Agrupados por funcionalidades, no especifica nada acerca del orden de static o no)
 1. Public/Protected/Default/Private



**Y recordad amiguitos!!!
No olvidéis mantener un
bajo acoplamiento y una
alta cohesión!!**



Podéis encontrar toda la información en
<https://quark.everis.com/confluence/pages/viewpage.action?pageId=811303050>