

A dark blue vertical bar is positioned on the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the date '3-9-2018'. In the bottom-left corner, there are several thin, curved, light gray lines that sweep upwards and to the right.

3-9-2018

Google Java Style Guide

Resumen

GOMEZ SAN GABRIEL JETHRAN ENRIQUE
UNIVERSIDAD VERACRUZANA

Contenido

1. Introducción	3
1.1 Notas de terminología.....	3
1.2 Notas de guía.....	3
2. Conceptos básicos sobre archivos de origen	3
2.1 Nombre del archivo.....	3
2.2 UTF – 8.....	3
2.3 Caracteres especiales	3
2.3.1 Personajes de espacios blancos	3
2.3.2 Secuencia de escape especiales	4
2.3.3 Caracteres no ASCII	4
3. Estructuras del archivo fuente	4
3.1 Licencia o copyright.....	4
3.2 Declaración del paquete	4
3.3 Declaraciones de importación.....	5
3.3.1 Sin importación de comodines.....	5
3.3.2 Sin ajuste de línea	5
3.3.3 Pedido y espaciado.....	5
3.4 Declaración de clases	5
3.4.1 Clase de primer nivel.....	5
3.4.2 Ordenar los contenidos de la clase	5
4. Formateo	5
4.1 Refuerzos.....	5
4.1.1 Las llaves se usan cuando es opcional.....	5
4.1.2 Bloques no vacíos: estilo K & R	5
4.1.3 Bloques vacíos	6
4.2 Indentación de bloques.....	6
4.3 Una declaración por línea	6
4.4 Limite de columna: 100	6
4.5 Embalaje de línea	6
4.5.1 Donde romper	6
4.5.2 Sancionar las líneas de continuación al menos 4 espacios	7
4.6 Espacio en blanco	7

4.6.1 Espacio en blanco vertical	7
4.6.2 Espacios en blanco horizontales	7
4.6.3 Alineación horizontal: nunca se requiere	7
4.7 Agrupar Paréntesis	7
4.8 Construcciones específicas.....	7
4.8.1 Declaración de variables	7
4.8.3 Matrices.....	7
4.8.4 Cambiar declaraciones	8
4.8.5 Anotaciones.....	8
4.8.6 Comentarios	8
5. Nombrado	8
5.1 Reglas comunes.....	8
5.2 Reglas por tipo.....	8
5.2.1 Nombres de paquete.....	8
5.2.2 Nombre de clase.....	8
5.2.3 Nombres de métodos.....	8
5.2.4 Nombre constantes.....	9
5.2.5 Nombres de campos no constantes.....	9
5.2.6 Nombres de parámetros	9
5.2.7 Nombres de variables locales.....	9
6. Prácticas de programación.....	9
7. Formateo	9

1. Introducción

1.1 Notas de terminología

El nombre clases sobre una clase ordinaria, interfaz o tipo de anotación.

El termino miembro es para una clase anidada, campo, método o constructor.

El termino comentario solo son los de javadoc.

1.2 Notas de guía

Las elecciones opcionales de formato hechas en ejemplos no se deben aplicar como reglas.

2. Conceptos básicos sobre archivos de origen

2.1 Nombre del archivo

El nombre del archivo origen es de mayúsculas y minúsculas de la clase nivel superior más la extensión .java

2.2 UTF – 8

Todos los archivos están codificados en UTF – 8

2.3 Caracteres especiales

2.3.1 Personajes de espacios blancos

Carácter de espacio horizontal ASCII (0x20) es el único carácter de espacio en blanco que aparece en cualquier parte de un archivo origen. Los caracteres de tabulación no se usan para sanfría-

2.3.2 Secuencia de escape especiales

Se encuentran una secuencia de escape especial (`\b`, `\t`, `\n`, `\f`), la secuencia se utiliza en lugar de la octal correspondiente o Unicode de escape.

2.3.3 Caracteres no ASCII

Se debe de utilizar los caracteres Unicode real. La elección depende de que el código sea más fácil de leer y comprender. Utilizar un comentario explicativo puede ser muy útil.

Nunca hacer que el código sea menos legible por temor que otros programas no manejen caracteres que no son ASCII.

3. Estructuras del archivo fuente

Orden del archivo fuente:

1. Información de licencia o copyright.
2. Extracto del paquete
3. Importar declaraciones
4. Clase de nivel superior

Una línea en blanco para separar cada sección.

3.1 Licencia o copyright

Las licencias deben de estar en cada uno de los archivos, en todo el proyecto.

3.2 Declaración del paquete

El límite de columna no se aplica en los paquetes.

3.3 Declaraciones de importación

3.3.1 Sin importación de comodines

No se utilizan.

3.3.2 Sin ajuste de línea

El límite de columna no se aplica a las declaraciones de importación

3.3.3 Pedido y espaciado

1. Las importaciones estáticas en un solo bloque

2. Las importaciones no estáticas en un solo bloque

3.4 Declaración de clases

3.4.1 Clase de primer nivel

Cada clase de nivel superior reside en un archivo de origen propio.

3.4.2 Ordenar los contenidos de la clase

Las clases deben tener un orden lógico, que su mantenedor podría explicar si se le pregunta. Los nuevos métodos no se agregan al final de la clase.

3.4.2.1 Sobrecargas: nunca dividir

Cuando se tienen múltiples constructores o múltiples métodos con el mismo nombre.

4. Formateo

4.1 Refuerzos

4.1.1 Las llaves se usan cuando es opcional

Para las sentencias if, else, for, do y while incluso cuando el cuerpo está vacío.

4.1.2 Bloques no vacíos: estilo K & R

- Sin saltos de línea antes de la llave de apertura
- Salto de línea después de la llave de apertura
- Salto de línea antes del corsé de cierre

4.1.3 Bloques vacíos

Puede estar con el estilo K & R, puede cerrarse inmediatamente después de abrirse, sin caracteres ni salto de línea, al menos que cuente con una instrucción

4.2 Indentación de bloques.

Se aumenta dos espacios cada vez que se abre un nuevo bloque o construcción similar. Cuando se termina vuelve al sangrado anterior. También se aplica en los comentarios.

4.3 Una declaración por línea

Cada declaración es seguida por salto de línea

4.4 Limite de columna: 100

Un carácter significa cualquier punto de código Unicode.

Excepciones:

1. Un URL en javadoc o referencia.
2. Package y import
3. Líneas de comando en un comentario

4.5 Embalaje de línea

Cuando el código que de otro modo podría ocupar legalmente una sola línea se divide en varias líneas, llamada ajuste de línea.

4.5.1 Donde romper

Deben romperse a un nivel sintáctico más alto

1. La ruptura viene antes del símbolo cuando el operador no es de asignación
2. Cuando se rompe una línea en un operador de asignación, la ruptura normalmente viene después del símbolo.
3. Un método o nombre de constructor permanece adjunto al paréntesis abierto
4. Por una coma (,)

4.5.2 Sancionar las líneas de continuación al menos 4 espacios

Cada línea después de la primera se sangra al menos 4 espacios desde la línea original.

4.6 Espacio en blanco

4.6.1 Espacio en blanco vertical.

Entre miembros consecutivos o inicializadores de una clase: campos, constructores, métodos, clases anidadas, inicializadores estáticos.

4.6.2 Espacios en blanco horizontales

1. Separar cualquier palabra reservada, como if, for o catch desde un paréntesis abierto.
2. Separar cualquier palabra reservada, else o catch de una llave de cierre.
3. Antes de cualquier llave abierta
4. En cualquier lado de operadores binario o temario.
5. El paréntesis de cierre
6. Después de una matriz (opcional)

4.6.3 Alineación horizontal: nunca se requiere

Ayuda a legibilidad, pero crea problemas para el mantenimiento futuro.

4.7 Agrupar Paréntesis

Se pueden omitir cuando el autor y el revisor están de acuerdo en que no hay posibilidades razonables de que el código.

4.8 Construcciones específicas

4.8.1 Declaración de variables

Se declara solo una variable, no se valen las declaraciones múltiples. Solo se pueden en los ciclos for. Se deben declarar cuando sea necesario, aunque es recomendable que sea al inicio.

4.8.3 Matrices

Cualquier inicializador de matriz puede formatearse opcionalmente como si fuera una "construcción similar a un bloque".

4.8.4 Cambiar declaraciones

Sangría: el contenido de un bloque de conmutadores tiene sangría +2. Después de un salto de línea el nivel de la línea aumenta +2.

Fall through: quiere decir que después de un break, continue, return o excepción, podría continuar.

Default: en caso de que no se cumpla alguna otra sentencia.

4.8.5 Anotaciones

Se aplican a una clase, método o constructor después del bloque de documentación y cada anotación se enumera en una línea propia.

4.8.6 Comentarios

Estilos tipo de javadoc, cualquier salto de línea deberá tener un asterisco.

5. Nombrado

5.1 Reglas comunes

Los identificadores usan solo letras y dígitos y en un pequeño número de casos indicados a continuación.

5.2 Reglas por tipo

5.2.1 Nombres de paquete

Los nombres de los paquetes son minúsculos, con palabras consecutivas simplemente concatenados juntos.

5.2.2 Nombre de clase

Los nombres de las clases están escritos en UpperCamelCase.

Son nombres sustantivos o frases nominales. Los nombres de interfaz también pueden ser sustantivos o sintagmas nominales.

5.2.3 Nombres de métodos

Los nombres de los métodos están escritos en LowerCamelCase

Los nombres de los métodos son verbos o frases verbales.

5.2.4 Nombre constantes

Los nombres constantes deben ir todas las letras en mayúsculas con cada palabra separada por un guion bajo.

5.2.5 Nombres de campos no constantes

Se escriben en lowerCamelCase, estos son sustantivos.

5.2.6 Nombres de parámetros

Se escriben en lowerCamelCase, se deben de evitar nombres de carácter en los métodos públicos

5.2.7 Nombres de variables locales

Se escriben en lowerCamelCase, no se consideran constante

6. Prácticas de programación

Los métodos con override debe estar marcado siempre y cuando sea legal

Las excepciones atrapadas no deben ser ignoradas

7. Formateo

Forma general debe ser:

```
/**
```

```
*
```

```
*/ o
```

```
/** .... */ (Siempre y cuando sea un fragmento)
```

Las etiquetas del bloque aparecen en orden y nunca deben tener una descripción vacía.

El javadoc se utiliza para cada clase publica y public o protected miembro de una clase.