

Directrices de codificación para Java

Este documento es un resumen de directrices para crear código Java elegante.

Estas directrices no forman parte de la especificación Java: un pedazo de código Java dado puede violar todas estas directrices y aún así ser un programa Java completamente correcto.

No obstante, la experiencia nos enseña que seguir estas directrices lleva a largo plazo a mejores programas. El código correctamente formateado incrementa drásticamente la legibilidad, haciendo que encontrar los problemas sea mucho más sencillo.

Por ejemplo, estos dos pedazos de código son exactamente el mismo código... ¿cuál crees que es más fácil de leer y entender?

```
public class Fofito {
public static

void main(String[] args) {
    String mensaje1="Fofito es feo"
    ;
    String mensaje2
    = "Tu si que eres feo";
    boolean interruptor = true;

    int acumulador;
    if (
    interruptor
    ==
    true
    )
    {

        System.out.println(mensaje1);
    }
}
}
```

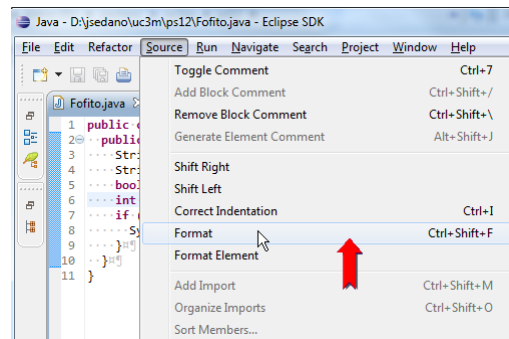
```
public class Fofito {
    public static void main(String[] args) {
        String mensaje1 = "Fofito es feo";
        String mensaje2 = "Tu si que eres feo";
        boolean interruptor = true;

        int acumulador;
        if (interruptor == true) {
            System.out.println(mensaje1);
        }
    }
}
```

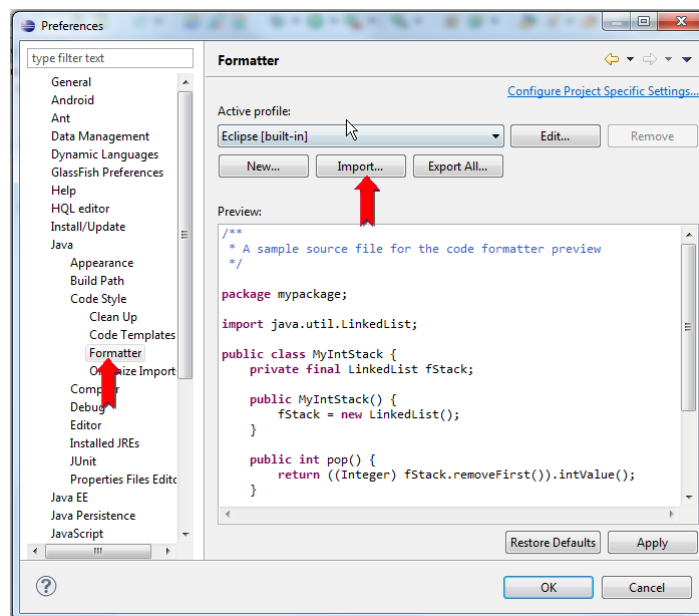
Oracle publica una guía llamada *Code Conventions for the Java programming Language* (<http://www.oracle.com/technetwork/java/codeconv-138413.html>) con muchas convenciones que están ampliamente extendidas entre la comunidad de programadores Java. Dicha guía te proporcionará consejo sobre el nombrado de las clases, métodos y propiedades, cómo indentar código, dónde añadir los retornos de carro, cómo escribir las llaves,...

No sigas leyendo hasta que hayas leído la guía de Oracle. Es un poco larga pero merecerá el tiempo que inviertas en ella.

Eclipse puede ayudarte a formatear correctamente el código. Con un fichero .java abierto, usa *Source* → *Format* para formatear el código con las reglas de formateo incorporadas.



Las reglas de formateo incorporadas por defecto cumplen en su mayoría las reglas definidas por Oracle. No obstante, los profesores de este curso utilizan un conjunto ligeramente distinto de reglas (en realidad la única diferencia es que las líneas se indentan con 2 espacios en lugar de con un tabulador). Si quieres usar esas reglas, puedes descargarlas de este link (http://www.it.uc3m.es/java/2012-13/resources/j_java.xml) e importarlas en Eclipse a través de sus preferencias.



La experiencia nos enseña que se deberían seguir también algunas otras reglas:

- Intenta evitar los caracteres que no sean US-ASCII. US-ASCII contiene las letras normales (a, b, c,...), los números (1, 2, 3,...) y los signos ortográficos estándar (?, :, !, \$, %,...) pero no letras tildadas (á, é, ñ, í, ô, ù,...) ni los signos ortográficos específicos del Español (¿ y ¡).
- Una excepción a esta regla podrían ser los literales que se muestren al usuario.
- Usa nombres significativos para las clases, métodos, propiedades y variables. Puede parecerse divertido usar nombres graciosos para las variables, o llamar ejercicio1() a los

métodos... pero te complicará la vida en el futuro. Y también le complicará la vida al profesor encargado de evaluar tu código, por cierto.

- Usa los comentarios para explicar qué hace el código. Léelo de nuevo: QUÉ hace el código, no CÓMO lo hace. Una excepción podría ser un algoritmo especialmente complicado. Usa nombres significativos para las clases, métodos, propie... un momento, ¿esto no lo hemos dicho ya? Sí. Muchas veces, si usas nombres significativos, no necesitarás añadir comentarios. Por ejemplo, si un método se llama `ajustarAltura()` es obvio que ajusta la altura; en cambio, si lo llamarás `ajustarA()`, tendrás que añadir un comentario para explicarlo.
- Escribe métodos cortos (no los nombres de los métodos, sino los propios métodos). Si el método cabe en una pantalla, serás capaz de verlo, leerlo y entenderlo en un vistazo. En cambio, si no cabe, estar continuamente moviendo el scroll arriba y abajo te confundirá.