



Buenas prácticas

▼ Convenciones de nomenclatura

Esto incluye la creación de una serie de normas sobre cómo nombrar clases e interfaces, métodos, variables, constantes, etc.

Un identificador en Java debe tener las siguientes características:

- Se explica por sí mismo
- Distinciones significativas
- Pronunciables

▼ Reglas de nomenclatura en Java

Estas son algunas de las reglas de nomenclatura que se suelen incluir dentro de las buenas prácticas de programación.

- Los nombres de clase e interfaz deben ser sustantivos, comenzando con una letra mayúscula. → NombrePaquete
- Los nombres de las variables deben ser sustantivos, comenzando con una letra minúscula. → contador
- Los nombres de los métodos deben ser verbos, comenzando con una letra minúscula. → genPalabra
- Los nombres constantes deben tener todas las letras mayúsculas. → PI
- Las palabras compuestas deben seguir la regla CamelCase:
 - NombreClase.
 - nombreVariable o nombreMetodo.

▼ Ordenar miembros de cada clase por ámbitos

Una de las buenas prácticas de programación con Java es la de organizar cada una de las variables de una determinada clase en función de sus ámbitos. Esto se hace desde el más restrictivo al menos restrictivo.

Pues que debemos ordenar las variables en función de la visibilidad de los modificadores de acceso: privado, predeterminado (paquete), protegido y público. Y cada grupo separado por una línea en blanco.

▼ Los campos deben ser privados

Una buena práctica en Java es minimizar la accesibilidad de los campos (de cada clase) de la forma más inaccesible posible.

Esto significa que el programador Java debe usar el modificador de acceso más bajo posible para proteger los campos.

Esta práctica está especialmente recomendada para hacer cumplir la ocultación o encapsulación de información en el diseño de software.

▼ Evitar los bloques de captura vacíos

Uno de los malos hábitos de la programación con Java es dejar bloques de captura vacíos. Esta mala práctica puede hacer que el programa de fallos silenciosos, lo que hace que la depuración del código sea mucho más complicada.

Por lo tanto, es una de las buenas prácticas de programación con Java evitar los bloques de captura vacíos. ¿Cómo se puede hacer? Pues detectado y corrigiendo las excepciones que se generan cuando se crean bloques de captura vacíos.

▼ Don't Repeat Yourself (DRY)

Principio de “No te repitas”, también conocido como “Una vez y sólo una”. Según este principio, toda pieza de información (código fuente, comentarios, documentación, etc ...) nunca debería ser duplicada debido a que la duplicación incrementa la dificultad en los cambios y evolución posterior, puede perjudicar la claridad y crear un espacio para posibles inconsistencias.

```

NombreClase.java x
1 package nombrePaquete;
2
3 import java.util.Scanner;
4
5 public class NombreClase {
6
7     public static void main(String[] args) {
8
9         //Declaración de variables
10        Scanner ent = new Scanner(System.in);
11        String palabraAdivinar = "";
12        int resultado = 0, intentos = 0, aciertos=0, contador=0;
13
14        System.out.println("¿Desea introducir una palabra?");
15
16        //Condición
17        if(resultado == 1) {
18            System.out.println("Introduce la palabra a adivinar");
19            palabraAdivinar = ent.next();
20        }else {
21            palabraAdivinar = genPalabra();
22        }
23
24        //Bucle
25        do {
26            contador = 0;
27            System.out.println("Introduzca una letra "
28                + " -> intentos disponibles " + (10 - intentos));
29        }while(intentos < 10 );
30
31    } // fin main
32
33    public static String genPalabra() {
34
35        String lista[] = new String[] {"teclado", "ordenador", "presidente",
36
37        return lista[(int) (Math.random() * lista.length)];
38    }
39
40    public static void toString(char array[]) {
41        System.out.print("(");
42
43        for(int i=0 ; i< array.length ; i++) {
44            System.out.print(array[i] + " ");
45        }
46
47        System.out.print(")");
48    }
49 } //Fin clase
50
51

```

IDENTIFICADORES CON NOMBRE CLAROS Y SIN CARACTERES EXTRAÑOS

LOS NOMBRES DE LAS CLASES EMPIEZAN EN MAYUSCULAS. NOTACIÓN CAMEL CASE SI SON DOS PALABRAS

PUNTO Y COMA OBLIGATORIO AL FINAL DE CADA LÍNEA DE INSTRUCCIÓN

ESPACIO EN BLANCO ENTRE BLOQUES LÓGICOS

CADA BLOQUE LÓGICO SE HA DE AGRUPAR EN BLOQUES LIMITADOS CON LLAVES

LÍNEAS NO MUY LARGAS. SI NO ES POSIBLE, SALTO DE LÍNEA

CÓDIGO COMENTADO CON CRITERIO