MT SOFTECH

SOLUCIONES EN INFORMÁTICA

Buenas prácticas de escritura en java

Autor: Diego Hernández

Introducción

La escritura de código es una tarea que como desarrolladores realizaremos a diario y aunque podamos generar500 líneas de código por día, nos pasaremos la mayor parte del tiempo leyendo código e identificando tareas, acciones o errores de lógica dentro de él. Es por eso que debemos cuidar la estructura, limpieza y flujo del proyecto.

Lo básico

Para empezar es bueno definir algún estándar para la definición de todo lo que vayamos a utilizar, para java suele usarse el CamelCase, consiste en separar las palabras dentro de nuestras variables, clases, métodos, etc.colocando la inicial de cada palabra con una mayúscula, por ejemplo:

```
int miPrimerNumeroEntero = 0;
```

En este ejemplo definimos un número entero y separamos cada una de las palabras involucradas por una mayúscula, pero si notamos, la primer letra del identificador de la variable comenzó por una minúscula, esto noslleva al siguiente tema.

Uso del camel case según su contexto

Si bien seguimos la misma línea al usar el CamelCase tenemos una pequeña variación según el sitio donde lousemos, la variación es básicamente iniciar con mayúscula o no.

- Iniciar con mayúscula.
- Nombres de clases.
- Nombres de interfacesIniciar con minúsculas.
- Nombres/identificadores de variables.

Nombres de instancias de clases (objetos).

Nombres de métodos o funciones.

Aunque sea algo simple, nos permitirá diferenciar más fácil si se trata de un elemento u otro.

Caracterización de elementos en mi código

Cuando vamos a escribir o nombrar una clase, no utilizamos las mismas reglas a comparación de variables, constantes o interfaces, debemos mantener un margen entre cada uno de estos elementos empecemos por...

¿Cómo escribir nombres de variables?

- Priorizar nombres cortos y auto explicativos.
- Auto explicativo refiere a nombres que con solo leerlos, nos da a entender en que contexto esutilizado.
- La inicial de cada una debe ser minúscula.
- Su nombre debe definir una propiedad (dependiendo el contexto en el que se utilice).

```
int posicionEnX = 100;
```

¿Cómo escribir nombres de constantes?

Todas el nombre debe estar escrito en mayúsculas, para este no utilizamos Camel Case y separamos cadapalabra por un guion bajo.
Su nombre debe ser muy claro y de usos principalmente de referencia.

```
final int POSICION_INICIAL_X = 100;
```

¿Cómo escribir nombres de clases?

- Utilizamos Camel Case e iniciamos su nombre con una mayúscula.
- Su nombre debe ser un sujeto/sustantivo, recordemos que una clase es un molde que representarádiferentes tipos de entidades.

```
public class automovil {

}
```

¿Cómo escribir nombres de interfaces?

Utilizamos Camel Case e iniciamos su nombre con una mayúscula.
Su nombre debe ser un adjetivó, recordemos que podemos implementar más de una interfaz por clase yque además indica a un tipo de comportamiento abstracto que se debe definir en cada clase implementada.

```
public interface contable{

}
```

Características en común

Todos estos elementos deben compartir una definición o nombre breve y que al leerlo ya nos permita identificar cuál es su labor

Con excepción de las constantes, todas utilizan Camel Case

Una práctica adicional es agregar comentarios sobre lo que ocurre con ciertos fragmentos de código, pero no nos referimos a cosas como "definimos la variable entera" sino más bien al objetivo de ese fragmento, con el ejemplo anterior es mejor decir "definimos la variable entera para almacenar el total delcosto de productos".

Pero ¿Por qué tomarlas?

El mero hecho de tomarlas ya hará que el código sea mucho más fácil y cómodo de leer a que si no lo hiciéramos, pero debemos pensar que en un futuro es posible que alguien más tenga que leer el código (si trabajamos en equipos) o si es personal, es seguro que después de mucho tiempo no recordemos lo que hacíael código y tendríamos que invertir más tiempo en recordar todas sus funcionalidades.