**Estándar de codificación en Java**

|  |  |
| --- | --- |
| **Propósito** | Guiar el desarrollo de la codificación en el lenguaje de programación Java |
| **Cabecera del programa** | Comienza todos los programas con un encabezado donde se describan datos generales. |
| **Formato de la cabecera** | /\*\*  \* Nombre: Nombre del programa a desarrollar  \* Versión: Versión actual del programa  \* Autor: Nombre del autor(es) del programa  \* Fecha: Fecha en que se inició el desarrollo  \* Descripción: Una corta descripción de lo que hace el programa  \*/ |
| **Lista de contenidos** | Proporciona un resumen de la lista de contenidos |
| **Ejemplo de contenidos** | /\*\*  \* Lista de contenidos:  \* Instrucciones de reutilización  \* Instrucciones de modificación  \* Importaciones de paquetes  \* Declaraciones de clases:  \*  \* ClaseA  \*  \* ClaseB  \*  \*  \* Código fuente en: C:/User/Documents/NetbeansProjects/…:  \* Clase A  \* Código fuente en: C:/User/Documents/NetbeansProjects/…:  \* Clase B  \*/ |
| **Instrucciones de reutilización** | - Describe cómo se utiliza el programa: formato de declaración, valores de parámetros, tipos, formatos, propósito y limitaciones del método.  - Proporciona advertencias sobre valores no soportados, excepciones u otras condiciones que podrían resultar en una operación errónea. |
| **Ejemplo de Instrucciones de reutilización** | /\*\*  \* Instrucciones de reutilización:  \* int printLine(String linea)  \* Propósito: Imprimir la línea de texto ‘linea”, en una sola línea \* Limitaciones: El tamaño de la línea no deberá exceder el tamaño del \* renglón  \* Regresa 0 si no se está listo para imprimir, si no, regresa 1  \*/ |
| **Identificadores** | Usa nombres descriptivos para las variables, nombres de funciones, constantes, y otros identificadores. Evita las abreviaciones o letras independientes. |
| **Ejemplo de identificadores** | int numeroDeAnimales; */\*Esto está bien\*/*  float a, b */\*Esto está mal\*/* |
| **Comentarios** | -Documenta el código para que el lector pueda entender cada operación que realizas.  -Los comentarios deben explicar el propósito y el comportamiento del código.  -Comenta la declaración de las variables para indicar su propósito en el programa. |
| **Buenos comentarios** | If(flag == true) */\*Comprueba si la variable flag es verdadera, si lo es, realiza lo que está dentro de la condición. Flag proviene de la comprobación de archivos\*/* |
| **Malos comentarios** | If(flag == true) */\* Condicional if \*/* |
| **Bloques grandes** | Precede los bloques grandes de código con un comentario donde describas lo que realizas en esa sección. |
| **Ejemplo** | /\*\*  \* La siguiente sección del programa contiene la impresión en pantalla de \* las facturas de ventas realizadas hasta el momento  \*  \*/ |
| **Espacios en blanco** | -Escribe programas con suficiente espacio entre líneas (donde haga falta) para que no se vea amontonado.  -Separa cada constructor del programa con al menos un espacio en blanco. |
| **Indentación** | -Indenta las líneas del código que lo requieran y hagan falta. (Por ejemplo, en ciclos)  -Abre y cierra las llaves asegurando que queden alineadas verticalmente.  -Se le asignarán cuatro espacios a la indentación. |
| **Ejemplo de indentación** | If(flag == true) {  suma += 8;  if(suma == 10) {  return 10;  }  } |
| **Uso de mayúsculas** | -Las clases empiezan siempre con letra mayúscula  -Las variables y funciones empiezan con minúscula, si se llegarán a unir palabras, a partir de la segunda palabra comenzarían con mayúscula.  -Las constantes van con letras mayúsculas |
| **Ejemplo de uso de mayúsculas** | int variableUno;  final double PI = 3.1415  public void realizarSumaVariables(); |