**Estándar Codificación Java**

|  |  |
| --- | --- |
| **Propósito** | Guía para la codificación de programas en Java |
| **Encabezado del programa** | Todas las clases e interfaces deben contener dos encabezados uno antes de la declaración del paquete y de los imports, y otro antes de la declaración de la clase. |
| **Formato del encabezado** | /\*\*  \* <nombre\_clase>.java  \* <nombre\_programa>  \* Copyright (c) <entidad\_empresa\_dueña>  \*/  package ...  import ...  /\*\*  \* <nombre\_clase>  \*  \* **@**author <nombre\_autor> <mailto:<correo> />  \* **@**version<versión>  \* **@**since <fecha de la creación>  \*/ |
| **Indentación** | Se debe indentar cada nivel. Cada indentación debe ser de 4 espacios.  Cada línea no debe tener más de 80 caracteres.  Cuando una declaración supere los 80 caracteres, se debe dividir de acuerdo a estos principios:   * Separar después de una coma. * Separar antes de un operador. |
| **Ejemplo Indentación** | String nombreCompletoPersona = nombre1 + nombre2  + apellido1 + apellido2; |
| **Comentarios** | Hay dos tipos de comentarios: los de implementación y los de documentación.  Los comentarios de implementación son:   * Multilinea: Utilizan los delimitadores /\* \*/ * Una sola línea: Utilizan el delimitador //   Los comentarios de documentación utilizan los delimitadores /\*\* \*/ y también etiquetas como @return, @param, @see, @author, etc. |
| **Ejemplo de comentarios** | Comentarios de implementación:  int numeroLineas; //Contiene el valor total de las líneas de la clase  Comentario de documentación:  /\*\*  \* Método que obtiene los datos del archivo.  \* **@param** ruta ubicación del archivo que contiene los datos  \*/ |
| **Nombres** | Se debe tener en cuenta:   * Paquetes: Se define en minúsculas y debe corresponder a un dominio de alto nivel y al departamento o dependencia de la organización. * Clases e Interfaces: Deben ser sustantivos. Se debe utilizar el estilo de UpperCamelCase[[1]](#footnote-1) para su nombrado. * Métodos: Deben ser verbos y deben utilizar el estilo de LowerCamelCase.[[2]](#footnote-2) * Variables: Deben ser descriptivos y se debe utilizar el estilo LowerCamelCase. * Constantes: Deben ir totalmente en mayúsculas y si son varias palabras deben ir separadas por guion bajo (“\_”) |
| **Ejemplo de nombres** | * Paquetes: co.edu.uniandes.ecos * Clases: Profesor * Método: obtenerNombreProfesor * Variable: nombreProfesor * Constante: SALARIO\_PROFESOR |
| **Lineas y espacios en blanco** | Se debe utilizar líneas en blanco para separar declaraciones de métodos. Entre variables o contantes y métodos, pero no se debe usar para separar declaraciones de variables y constantes.  Los espacios en blanco se deben utilizar para separar un paréntesis de una palabra reservada. |
| **Ejemplo de líneas y espacios en blanco** | Linea:  public double sumarNumeros(int num1, int num2) {  ...  }  public double restarNumeros(int num1, int num2){  ...  }  Espacios  **while** (d++ = s++) {  n++;  } |

1. (Wikipedia, 2017) [↑](#footnote-ref-1)
2. (Wikipedia, 2017) [↑](#footnote-ref-2)