Estándar de codificación

|  |  |
| --- | --- |
| Propósito | Propuesta de síntesis del estándar de código de Java para su uso en proyectos de la materia |
| Encabezado | Debe contar con: 1.- Información de autor y el programa  2.- Declaración de clases |
| Ejemplo | /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\* @version Numero de Versión del Programa \*/  /\* @author Nombre del autor \*/  /\* @since Fecha de creación del programa \*/  /\* Nombre de la clase \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ |
| Orden de declaración de variables y métodos | Todas las variables de tipo public deben declararse primero, después los protected, después las de tipo private.  Los métodos deben agruparse de acuerdo a su funcionalidad sin importar su grado de alcance. |
| Nombres  (variables, identificadores, clases) | Paquetes: Todos en minúsculas y palabras juntas  Clases: Las iniciales de los nombres con mayúsculas. En caso de ser una unión de dos o mas palabras, se usará CamelCase.  Métodos: primera letra min, resto de palabras primera letra mayúscula.  Constantes: Todas mayúsculas, separar palabras con guion bajo  Variables: primera letra min, resto de palabras primera letra mayúscula  Tipo de variable genérica: Una letra mayúscula. |
| Ejemplo | Paquetes: com.example.deepspace  Clases: Persona Estudiante Auto  Métodos: sendMessage exit autoMover  Constantes: CONSTANT\_CASE  Variables : localLowerCase  Tipo de variable: T E L |
| Espacios en blanco horizontal | Una espacio horizontal en blanco aparece:  -Separando palabras reservadas de un paréntesis  -Separando palabras reservadas de llaves  -Antes de abrir una llave o corchete  -En ambos lados de un operador binario o ternario  -Después de:, : ; ( ) =  -En ambos lados de //  -Entre el tipo y la variable de una declaración  -Dentro de ambas llaves de una inicialización de un arreglo |
| Ejemplo | for ()  } else {  String x = {  (x | y)  X, b, c;  // Hola comentario  String x = { ‘e’, ‘b’ } |
| Espacios en blanco vertical | -Entre miembros o iniciadores de clase: campos, constructores, métodos, clases anidadas, iniciadores estáticos  -Entre enunciados, para organizar el código en subsecciones lógicas  -Entre bloques lógicos |
| Ejemplo | public class Auto {  } |
| Identación | -Aplicar una sangría cada que se cambia de nivel  -Las llaves que se abren deben estar al mismo nivel que su declaración |
| Ejemplo | for (int i=0;i<3;i++) {  aux[i];  } |
| Capitalización | -Con mayúsculas todas las definiciones  -Con minúsculas palabras reservadas e identificadores |
| Ejemplo | #define DEFAULT-NUMBER-OF-STUDENTS 15 |
| Condiciones if () | Se debe poner primero la variable y luego la constante. |
| Ejemplo | CORRECTO: if (var == 5) {  }    INCORRECTO: if (5 == var) {  } |
| Importaciones | Prohibido usar \*, se debe importar la clase que se necesita en especifico  Se separan importaciones, las estáticas van en un bloque, las no estáticas en otro bloque |
| Ejemplo | import util.java.\*  incorrecto  import util.java.List;  correcto |
| Llaves {} | Hay salto de línea en cuanto se cierra, excepto en else y en catch y finally.  En bloques vacíos, puede cerrarse inmediatamente después de abrirlo. |
| Ejemplo | if ( var == 5) {  return true;  }    if ( var == 5 ) { } |
| Columna | Límite de 100 columnas en Java, casos especiales: como en importación de paquetes y comentarios. |
| Variables | Cada declaración, declara solo una variable, en caso de no requerir un valor especifico inicial, se debe declarar como valor nulo. |
| Ejemplos | Incorrecto:  int a, b, c;  Correcto  int a = 0;  int cont = 0;  String aux = “”; |
| Arreglos | Los corchetes se escriben junto al tipo, no la variable, ya que forman parte del tipo |
| Ejemplo | String[] args  New int[] {  1, 2, 3, 4 } |
| Instancias | En caso de no requerirse un valor inicial se debe declarar como null. |
| Ejemplo | Estudiante estudiante = null; |