# /Users/jorgegarcia/Downloads/– Winslow Music Academy –.jpg

|  |  |
| --- | --- |
| **ESTÁNDAR GENERAL** | |
| Propósito. | Para guiar la implementación de programas en Java en la empresa “***El Solecito***”. |
| Encabezados de programa. | Comience todos los programas con un encabezado descriptivo, agregando el nombre de quién edito esa sección. |
| Formato del encabezado. | 1. **Asignación de programa:** Por lo regular deberá de llevar el nombre del programa o en su defecto el número de programa. 2. **Nombre:** Se agregará el nombre de quién creo cada sección del programa y de quién lo edita. 3. **Fecha:** Agregar siempre la fecha de creación y de edición. 4. **Descripción:** Una ligera descripción de lo que se realizó, o de lo que el programa pretende hacer. |
| Contenidos del listado. | - |
| Instrucciones de reutilización. | **DECLARACIÓN DE VARIABLES.** En java existen dos tipos de variables:   1. **Primitivas:** Pueden ser cualquiera de los 8 tipos:    1. char    2. boolean    3. byte    4. short    5. int    6. long    7. double    8. float 2. **De referencia:** Son usadas para referenciar (o acceder) a un objeto.  **PRIMITIVAS:** Las variables primitivas pueden ser declaradas como variables de clase (static), variables de instancia, parámetros de métodos, o variables locales:   1. byte b; 2. boolean myBooleanPrimitive; 3. int x,y,z;   Para los tipos enteros la secuencia desde más pequeños hasta más grandes es byte, short, int y long, y double es más grande que float. Ambos, enteros como tipos de puntos flotantes tienen signo, y esto afecta a sus rangos.    Para los boolean no hay rango; un booleano solo puede ser true o false.  El tipo char contiene un carácter singular de 16-bit Unicode. **DE REFERENCIA:** Las variables de referencia pueden ser declaradas como variables estáticas, variables de instancia, parámetros o métodos, o como variables locales. **DE INSTANCIA:** Las variables de referencia son definidas dentro de la clase, pero fuera de cualquier método, y son solo inicializadas cuando la clase es instanciada.   **VARIABLES LOCALES:** Las variables locales son declaradas dentro de un método. Esto significa que la variable no solo se inicializa dentro del método, sino que también se declara dentro del método. **DECLARACIÓN DE ARRAYS:** Los arrays son declarados poniendo el tipo de dato de los elementos que guardará (Objetos o primitivos), seguido de corchetes cerrados y seguido por el identificados. **DE CLASE:** Las variables de Clase o Miembro de dato estatico es una variable, al contrario que las variables de instancia, propias de la clase que la contiene y no de instancias de la misma. **DE ATRIBUTOS:** Las variables de Atributo se pueden definir como una característica que define un elemento. Los atributos se definen como una característica cualitativa. **DE MÉTODOS:** Las variables de Métodos pueden ser **Variables** dependientes e independientes. La clave para diseñar cualquier experimento es ver qué **variables** de investigación podrían afectar el resultado. Existen muchos tipos de **variables,** pero las más importantes para la gran mayoría de los **métodos** de investigación son las **variables** independientes y dependientes. |
| Ejemplos de instrucciones de reutilización. | **DE REFERENCIA:** Object o;  Dog myNewDogReferenceVariable;  String s1, s2, s3; **DE INSTANCIA:** Class Empoyee{  //Define los campos (variables de instancia)  private String name;  private String title;  } **VARIABLES LOCALES:** Class TestServer{  Public void logIn() {  int cont = 10;  }  } **ARRAYS:** Thread[] threads; //Recomendado.  Thread threads []; //Legal pero menos legible. **DE CLASE:** NombreClase.nombreVariable  Además es necesario que el metodo que la llame también sea estático(static). **DE ATRIBUTOS:** Por ejemplo:   * El nombre puede utilizer un byte como variable de instancia, porque el tamaño de un nombre no puede ser mayor que 255 caracteres, o se puede implementar un metodo que calculi el tamaño en un tiempo de ejecución. * La foto utilizara dos o cuatro bytes.  **DE MÉTODOS:** Static  This o super.  Package com.ecodeup.static; |
| Identificadores. | Use nombres descriptivos para todas las variables, nombres de funciones, constantes y otros identificadores. Evite abreviaturas o variables de una sola letra. |
| Ejemplos de identificadores. | int numero\_de\_estudiantes; //RECOMENDADO.  float: x4,j,ftave; //NO RECOMENDADO. |
| Comentarios. | Documente el código para que el lector pueda entender su funcionamiento.  - Los comentarios deben explicar la finalidad y el comportamiento del código.  – Comentarios declaraciones variables para indicar su propósito. |
| Buenos comentarios. | if (record\_count> limit) / \* ¿se han procesado todos los registros? |
| Malos comentarios. | If (record\_count> limit) / \* comprueba si el recuento de registros excede el límite |
| Secciones principales. | Preceda a las principales secciones del programa mediante un comentario en bloque que describa el procesamiento realizado en la siguiente sección. |
| Ejemplo. | La sección del programa examina los contenidos de la matriz 'calificaciones' y calcula la calificación promedio de la clase. |
| Espacios en blanco. | Escriba programas con suficiente espacio para que no aparezcan llenos.  - Separe cada construcción de programa con al menos un espacio. |
| Sangrado. | - Sangrar cada nivel de refuerzo desde el nivel anterior.  - Las llaves abiertas y cerradas deben estar en líneas por sí mismas y alineadas. |
| Ejemplo de sangrado. |  |
| Capitalización. | Minúsculas de todos los demás identificadores y palabras reservadas. Para que sean legibles, los mensajes del usuario pueden usar mayúsculas y minúsculas. |
| Ejemplo de capitalización. | #define DEFAULT-NUMBER-OF-STUDENTS 15  int class-size = DEFAULT-NUMBER-OF-STUDENTS; |