Estandar para la codificación JAVA

Juan Pablo Ospino Solano

versión 1.0

Proposito	Suministrar la estructura semántica como deben estar implementados los programanas para el lenguaje Java.
Encabezado	Ubicado en la parte superior del archivo. Contienen la siguiente información: 1. Nombre de la clase 2. Descripción general de la implementación. 2. Nombre del autor. 3. Versión Deben estar en la parte superior del archivo .java
Formato encabezado	/** * App * vista de usuario y contenedora del método main * de la solución. * * @author Juan Pablo Ospino Solano * @version 1.0 * @since 2017-02-20 */
Pakages e Imports	Ubicados debajo del ecabezado separados por una rengón. Deben estar continuos en rengones diferentes. Primero se debe ubicar la definición del packages, seguidamente separados por un renglón la declaración de los imports.
Ejemplo Pakages e Imports	package universidad.andes.ecos.tarea2; import java.IO; import java.string.*;
Declaración clases	Ubicados debajo de la declaración de *packge e imports* separados por una rengón. En la parte superior de la declaración de la clase debe estar documentado: descripción de la clase, fecha de creación y autor. La declaración de las clases siempre debe tener el nivel de ocultamiento (public, private). Su nombre debe empezar con letras mayúsculas; únicamente pueden ser utilizadas las letras del abecedario excepto vocales con tilde y la letra Ñ. El

	simbolo “{“ ubicarse en el siguiente renglón del texto. Las líneas de código y comentarios que conforman la clase debe ir con sangria de cuatro (4) espacios a la derecha. Las sentencia herencia se deben realizar en la línea siguiente y tabuladas.
Ejemplo declaración de clases	

```
/**
* <h1>Empleado</h1>
* Clase empleados
*
* @author Pepito perez
* @version 1.0
* @since   2018-11-20
*/
public class Empleado
    implements IEmpleado
{
    public int idEmpleado;
    ...
    /** code here **/
}

/**
* <h1>DocumentosCartera</h1>
* Clase que representan los registros
* correspondiente a la cartera de la compañía
*
* @author Martín Martínez
* @version 1.0
* @since   2005-10-07
*/
private class DocumentosCartera
{
    /**
    * Constructor que toma el identificador de la cartera
    * para hacer instancia de la misma
    * @param  idCartera número que identifica la cartera
```

	*/ public DocumentosCartera(int idCartera) { /** code here **/ } }
Atributos de la clase	Los atributos de la clase siempre deben contener el tipo de ocultamiento (public, private). La primera vocal del nombre debe estar en minusculas. Si el nombre es compuesto de varias palabras no deben estar separadas por simbolos; se debe utilizar la letra mayúcula al iniciar la siguiente palabra. Las propiedades deben estar en la parte superior del codigo, deben estar continuar una debajo de otras y separadas de los constructores, métodos o delegados por un renglón. Luego de las propiedades deben declararse los constructores y por último los métodos.
	/** * Estudiante * Clase que representan el estudiante del salón * * @author Armando Puertas * @version 9.2 * @since 2013-08-11 */ public class Estudiante { private string nombreEstudiante; public static DateTime fechaHoraActual; /** * Constructor que toma el identificador del estudiante * para hacer instancia de la clase * @param nombreEstudiante nombre estudiante a crear */ public Estudiante(String nombreEstudiante) { this. NombreEstudiante = nombreEstudiante; }

	`` /** * errola el esudiante con la huella * @param estudianteEnrrolar estudiante a ingresar al * sistema * @return indica si el estudiante está enrrolado */ public bool enrrolarEstudiante(Estudiante estudianteEnrrolar) { /*code here*/ return true; } } ```
Métodos y constructores	Los metodos de las clases siempre deben contener el nivel de ocultamiento. La primera letra del nombre siempre debe ser minúscula y Si el nombre es compuesto de varias palabras no deben estar separadas por simbolos; se debe utilizar la letra mayúcula al iniciar la siguiente palabra. El simbolo “{“ ubicarse en el siguiente renglón del texto. En la parte superior del metodo debe estar documentado: descripción del método, fecha de creación, autor,descripción de las entradas y descripción de el resultado. Los metodos debe ir separados verticalmente por un renglon de cualquier otro método, delegado o propiedad de la clase. Las variables de entradas no pueden tener el mismo nombre de una propiedad de la clases; es necesario utilizar una pablabra adicional o letra que pueda hacer distinción. Las sentencia de retorno de excepciones se deben realizar en la línea siguiente y tabuladas.
Ejemplo de implementación de los métodos y constructores	``` /** * Calcula la media de dos número enteros * * @param valor1 número uno a calcular * @param valor2 número dos a calcular * @return media de los dos números */ public decimal calcularMedia(int valor1, int valor2) throws IOException { ```

	return (valor1 + valor2) / 2; }
Variables	Las variables se crean en renglones independientes y pueden contener un valor por defecto. Si el nombre es compuesto de varias palabras no deben estar separadas por simbolos; se debe utilizar la letra mayúcula al iniciar la siguiente palabra. Los nombres de las variables únicamente pueden ser utilizadas las letras del abecedario excepto vocales con tilde y la letra Ñ. Nunca el nombre de una variable puede coincidir con el de una propiedad de la misma clase. Se pueden implementar comentarios en las variables del código, pero es necesario que dichos comentarios se hagan utilizando la notación “//”
Ejemplo Variables	// valor resultado de la operación Int resultadoMedia; Empleado empleadoEmpresa = new Empleado(1, “Chespirito”); DateTime anhioVentaFactura = null;