

1. Estándar de codificación

Nombres de Classes, Interfaces y Types

- PrimerLetraMayúsculaConPalabrasInternasPrimerLetraEnMayúscula
- Los nombres de las interfaces deben ser adjetivos
- Los nombres de las clases deben iniciar con l mayúscula
- Se usan sustantivos en singular para los nombres de las clases
- Los nombres de excepciones personalizadas deben terminar con el postfijo “Exception”
- Namespaces or Packages deben seguir el mismo nombre que los folders que contienen el código fuente y en minúscula

Methods

- MinúsculaConPalabrasInternasPrimerLetraEnMayúscula
- Para un método que es un procedimiento utilice un verbo seguido de un objeto, por ejemplo saveCustomer, moverFicha, insertarNodo
- Para el nombre de una función use una descripción del valor de retorno de la función, por ejemplo nextCustomerId, getName, retornarListaAutos
- Utilice una propiedad en lugar de métodos cuando lo que se va a retornar es un dato de la clase. En el caso de Java use los métodos llamados “setters” y “getters”
- Los parámetros se escriben en minúsculaConPalabrasInternasEnMayúscula iniciando con la letra “p”
- Un método no debe tener más de 6 parámetros ni ser mayor a 100 líneas de código

Variables

- Variables de instancia se inician con underscore y luego todas sus palabras con la primer letra de cada palabra en mayúscula, _ValorInmediatoAlterado
- No utilice el término “temp” o “index” para nombres de variables, tampoco los nombres de índices i,j,k ni cualquier uso de nombres de una letra o silábicos, en su lugar utilice nombres largos bien descriptivos
- No utilice nombres ambiguos en variables booleanas, ejemplo: done, status
- No utilice nombres negativos en las variables como NotFound, NoListado
- Nunca declare variables de instancia públicas, todas deben ser protected o private
- Variables locales se escriben con la primer letra en minúscula

Constants

- MAYUSCULAS_SEPARADAS_POR_UNDERSCORE
- Siempre utilice constantes en lugar de literales hardcoded
- Hay ciertos casos en los que no es necesario declarar una constante para ciertas literales hardcoded, tales se muestran en la siguiente tabla

Nombre	Descripción	Ejemplo
Cero	Use el valor cero cuando inicializa índices, variables o requiere comparar el largo o cantidad de elementos de una colección	<pre>_Variable = 0;</pre> <pre>for (int IndexXYZ=0;</pre> <pre>AnyKindCollection.Length > 0</pre> <pre>AnyKindCollection.Length < 0</pre> <pre>AnyKindCollection.Length == 0</pre> <pre>AnyKindCollection.Length >= 0</pre> <pre>AnyKindCollection.Length <= 0</pre> <p>Lo anterior aplica para arrays, arraylist, collections, hashtable, datarows, etc cuando usan las propiedades o métodos Length, Size, Count</p>
Length-1 Size-1	Algunos Collections requieren algún ajuste de tamaño para hacer comparaciones o para ajustarse algún algoritmo	<pre>for(int indexXYZ=0; indexXYZ<array.Length-1...</pre>
Inicio de índice 1,2,3	Cuando un ciclo inicia en los valores 1, 2, 3	<pre>for (int indexXYZ=1; indexXYZ<array.length</pre>
Evaluación par - impar	Cuando se requiere saber si un numero es par o impar	<pre>Var%2 == 0 Even</pre> <pre>Var%2 != 0 Odd</pre>

Properties

- MayusculaConLaPrimerLetraDePalabrasInternasEnMayuscula
- Los properties son la única forma de acceder a las variables de instancia tanto para las clases como para la clase misma que la contiene
- Si el lenguaje no soporta properties utilice los métodos de Set y Get que sean necesarios

Comentarios

- Los comentarios deben ser hechos de forma que los generadores y procesadores de documentación los entiendan y sepan procesarlos debidamente

Distribución del Código

El código fuente debe tener el siguiente orden en su estructura:

- Encabezado
- Declaración de clase debidamente comentada
- Constantes: de la visibilidad menos restrictiva a la más restrictiva. Se recomienda crear clases o interfaces únicamente para almacenar constantes.
- Atributos: de la visibilidad menos restrictiva a la más restrictiva
- Propiedades: de la visibilidad menos restrictiva a la más restrictiva
- Constructor (es)
- Destructor en caso de que exista
- Métodos: de la visibilidad menos restrictiva a la más restrictiva