

Práctica 05

Introducción a Ciencias de la Computación

Objetos: Programación de Métodos

Profesor: Mat. López Mendoza Salvador
Ayudante teoría: M. en C. Manuel Alcántara Juárez
Ayudante Laboratorio: Miranda Sánchez Luis Eduardo

15 de Septiembre de 2016

1. Marco Teórico

Aunque no es parte de la sintaxis de un método, es buena practica de programación incluir un comentario antes del encabezado. En este comentario se especifica la tarea que se realiza, cuáles son los perímetros que utiliza y cuál es el valor que devuelve, si es que lo hace. Para la documentación generada con **javadoc** es recomendable que en cada método se especifique como parte de los comentarios cada parámetro con **@param** seguido del nombre del mismo, y para el valor de retorno es recomendable usar **@return** seguido del tipo de valor que devuelve.

Métodos Modificadores

Los métodos modificadores tienen el propósito de modificar el valor de los atributos privados de los objetos, el estado del objeto antes de llamar a un método modificador debe ser diferente una vez que se ejecuta el método.

Los métodos modificadores en su forma mas simple, reciben como parámetro el nuevo valor para el atributo correspondiente, por lo tanto el tipo de parámetro debe coincidir con el tipo del atributo. Estos métodos no devuelven calor alguno, su objetivo es mantener la integridad de los objetos de la clase al ser la forma única en que un objeto cliente puede modificar el valor de los atributos de cualquier otro objeto. El nombre de estos métodos suele empezar con la palabra en ingles **set**, seguida por el nombre del atributo iniciado con letra mayúscula.

Métodos de Acceso

Los métodos de acceso o inspectores son el medio de conocer el valor de los atributos privados de los objetos. No reciben parámetros, pues su objetivo es obtener el valor de un atributo. Generalmente su nombre empieza con la palabra en ingles **get** seguido del nombre del atributo.

Si se especifica que un método va a devolver un valor, se debe usar en el cuerpo del mismo la instrucción **return**, pues con ella se devuelve el valor y se termina la ejecución del método. La instrucción **return** va seguida de una expresión cuyo valor es el que devolverá el método que la contiene.

Métodos Calculadores

Estos métodos se emplean para implementar cualquier comportamiento deseado de los objetos. Pueden recibir y/o devolver valores y trabajan con el estado del objeto para calcular un valor.

Documentación con Javadoc

Documentar un proyecto es algo fundamental de cara a su futuro mantenimiento. Cuando programamos una clase, debemos generar documentación lo suficientemente detallada sobre ella como para que otros programadores sean capaces de usarla sólo con su interfaz. No debe existir necesidad de leer o estudiar su implementación, lo mismo que nosotros para usar una clase del API Java no leemos ni estudiamos su código fuente.

Javadoc es una utilidad de Oracle para la generación de documentación de APIs en formato HTML a partir de código fuente Java. Javadoc es el estándar para documentar clases de Java.

Esto se debe incluir al documentar una clase.

- a) Nombre de la clase, descripción general, número de versión, nombre de autores.
- b) Documentación de cada constructor o método (especialmente los públicos) incluyendo: nombre del constructor o método, tipo de retorno, nombres y tipos de parámetros si los hay, descripción general, descripción de parámetros (si los hay), descripción del valor que devuelve.

Para que javadoc sea capaz de generar documentación automáticamente han de seguirse estas reglas:

- a) La documentación para javadoc debe de incluirse entre símbolos de comentario que deben de empezar con una barra y doble asterisco, y terminar con un asterisco y barra simple.
- b) La ubicación le define a javadoc qué representa el comentario: si está incluido justo antes de la declaración de clase se considerará un comentario de clase, y si está incluido justo antes de la signature de un constructor o método se considerará un comentario de ese constructor o método.
- c) Para alimentar javadoc se usan ciertas palabras reservadas (tags) precedidas por el carácter "@", dentro de los símbolos de comentario javadoc. Si no existe al menos una línea que comience con @ no se reconocerá el comentario para la documentación de la clase.

TAG	DESCRIPCIÓN	COMPRENDE
@author	Nombre del desarrollador	Nombre autor o autores
@deprecated	Indica que el método o clase es obsoleto (propio de versiones anteriores) y que no se recomienda su uso.	Descripción
@param	Definición de un parámetro de un método, es requerido para todos los parámetros del método.	Nombre de parámetro y descripción
@return	Informa de lo que devuelve el método, no se aplica en constructores o métodos "void".	Descripción del valor de retorno
@see	Asocia con otro método o clase.	Referencia cruzada
@version	Versión del método o clase.	Versión

Ejercicio

Programar lo necesario dentro de la clase Alumno para que se cumpla lo siguiente:

- Debe tener los siguientes atributos:
 - nombre
 - apellido materno
 - apellido paterno
 - edad
 - fecha de nacimiento
 - nombre de la carrera que cursa
 - sexo
 - telefono
 - correo electrónico y
 - número de cuenta
- La clase debe tener un constructor por omisión. Este constructor por omisión deberá iniciar los atributos con **tus** datos.
- La clase debe tener un constructor para crear un nuevo Alumno.
- Deberá tener **métodos de acceso y métodos modificadores**.
- Deberá respetar el **encapsulamiento**.
- Deberá tener los comentarios necesarios para poder crear la documentación de las clases involucradas.
- El método **toString** debe mostrar el estado del objeto.
- Un método que indique si el alumno creado estudia la carrera de Ciencias de la Computación o no.
- Un método que devuelva el CURP del alumno. ***
- Se deben hacer las pruebas necesarias dentro de la clase MainAlumno.

Ejercicio Extra (1 Punto)

Programar un método que calcule la edad del Alumno tomando la fecha de nacimiento como parámetro.

***Basarse en el programa que calcula el RFC y adaptarlo a la clase Alumno para crear un CURP.

Entrega

La practica es individual.

Si se va a entregar la práctica 5 y mi número de cuenta es 309199814 entonces el nombre del archivo sería [Practica 5]_309199814.zip

Subir la carpeta al sitio como comprimido (zip, rar, tar, tar.gz).