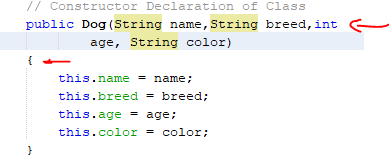
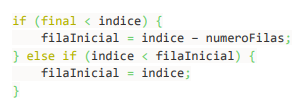
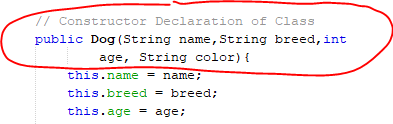
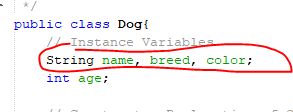
El primer error que podemos observar es el incorrecto uso de las llaves.



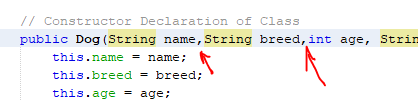
Por consenso, la llave debe abrirse al inicio de la línea, y debe cerrarse en la línea debajo de la última expresión, tal que:

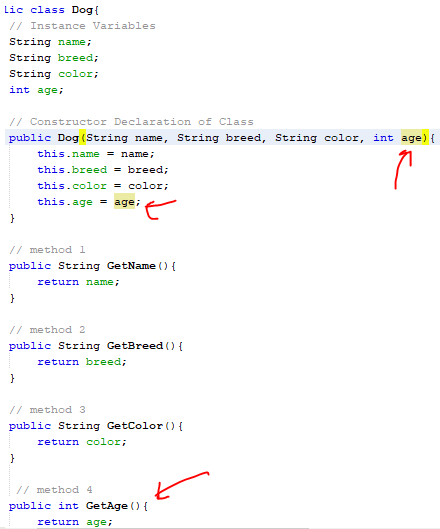


No hay motivo para realizar un salto de línea en medio de la definición de parámetros, puede resultar confuso al lector.

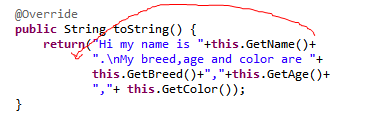
  
  
  
  
  
  
  
  


Solo una declaración de variables por línea, podemos observar 3 en este caso.

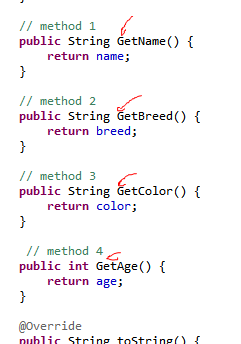
Los espacios entre operadores, signos, comas… Mejoran la legibilidad del programa.

Como detalle, podemos mantener el orden de las variables a la hora de definirlas, de pasarlas como parámetros y de programar los getters y setters de la variable. Tal que:  
  


Es más correcto empezar las líneas con el carácter de concatenación. Y dejar espacios entre los diferentes elementos:



Los métodos normalmente empiezan en minúsculas.



El método toString() y el main se encuentran al final del código, si el código tuviera 500 líneas tendríamos que bajar abajo del todo para encontrar el main y toString, por lo que no es óptimo.  
Por lo tanto, colocaremos el main en primer lugar, después los métodos y por último los getters y setters.

**package** refactorizacion;

/\*\*

\*Esta clase contiene los atributos de un perro

\*Y un método para devolver todos los atributos

\* **@author** Daniel Castro

\* **@version** 1.0

\*/

**public** **class** Dog {

// Instance Variables

String name;

String breed;

String color;

**int** age;

/\*\*

\*

\* **@param** name Nombre del perro

\* **@param** breed Raza del perro

\* **@param** color Color del perro

\* **@param** age Edad del perro

\*/

// Constructor Declaration of Class

**public** Dog(String name, String breed, String color, **int** age) {

**this**.name = name;

**this**.breed = breed;

**this**.color = color;

**this**.age = age;

}

/\*\*

\* Método main de la clase

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

Dog tuffy = **new** Dog("tuffy","papillon","white",5);

System.***out***.println(tuffy.toString());

}

/\*\*

\* Método para devolver un String con la información del perro

\*/

@Override

**public** String toString() {

**return**("Hi my name is " + **this**.getName()

+ ".\nMy breed,age and color are "

+ **this**.getBreed() + "," + **this**.getAge()

+ "," + **this**.getColor());

}

/\*\*

\* Método para devolver el nombre del perro

\* **@return** Devuelve el nombre del perro

\*/

// method 1

**public** String getName() {

**return** name;

}

/\*\*

\* Método para devolver la raza del perro

\* **@return** Devuelve la raza del perro

\*/

// method 2

**public** String getBreed() {

**return** breed;

}

/\*\*

\* Método para devolver el color del perro

\* **@return** Devuelve el color del perro

\*/

// method 3

**public** String getColor() {

**return** color;

}

/\*\*

\* Método para devolver la edad del perro

\* **@return** Devuelve la edad del perro

\*/

// method 4

**public** **int** getAge() {

**return** age;

}

}