CIFP César Manrique.

Entornos de desarrollo 1º de Desarrollo de Aplicaciones Web

Profesora: Sofía del Carmen Hernández González

Refactorización y Documentación.

Juan Carlos Francisco Mesa

Índice

Ejercicio

Enunciado

Tarea UT4 Optimización y documentación

Crea un archivo en algún procesador de textos, para que puedas **añadir, paso a paso, las capturas, comentarios y respuestas** a las dos actividades propuestas a continuación:

Actividad 1: Revisa el código del archivo **Dog.java** y encuentra todas las malas prácticas de programación que están presentes. Deberás realizar las correcciones, ya sea de forma manual o utilizando las herramientas de refactorización propias de tu IDE. También puedes utilizar un analizador de código si lo crees conveniente.

Debes añadir al informe, el código totalmente corregido, la explicación de los cambios que has realizado y en base a qué lo has hecho. Recuerda **personalizar el entorno** y mostrar claramente **todo lo solicitado.**

Actividad 2: Añadirás al código ya corregido **etiquetas Javadocs** para que puedes **generar la documentación** correspondiente. Recuerda usar los comentarios. Deberás **añadir al informe el código fuente con las etiquetas y comentarios** añadidos. **Adjunta** también, en una carpeta comprimida, **toda la documentación Javadocs** generada.

Volver al índice

Documento a revisar

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dog;

/**
 *
 * @author Alejandro
 */
public class Dog
{
   // Instance Variables
   String name, breed, color;
   int age;

   // Constructor Declaration of Class
   public Dog(String name,String breed,int
         age, String color)
   {
      this.name = name;
      this.breed = breed;
      this.age = age;
      this.color = color;
   }

   // method 1
   public String GetName()
   {
      return name;
   }

   // method 2
   public String GetBreed()
   {
      return breed;
   }
```

```
// method 3
    public int GetAge()
    {
        return age;
    }

    // method 4
    public String GetColor()
    {
        return color;
    }

    @Override
    public String toString()
    {
        return("Hi my name is "+this.GetName()+
                ".\nMy breed,age and color are "+
                this.GetBreed()+","+this.GetAge()+
                ","+ this.GetColor());
    }

    public static void main(String[] args)
    {
        Dog tuffy = new Dog("tuffy","papillon", 5, "white");
        System.out.println(tuffy.toString());
    }
}
```

Volver al índice

Actividad 1

Descripción del trabajo realizado

Voy a proceder a hacer un análisis pormenorizado del código usando la guía que nos fuera suministrada en la plataforma.

Nos encontramos en primer lugar que antes de la clase lo único que vemos es un comentario acerca del autor, lo cual no aporta mayor cosa acerca del uso de la clase, por lo que se procede a corregir; Nos encontramos con la sentencia package y la ausencia de la sentencia import ya que no hay librerías adicionales que importar, se observa también la definición de una única clase, y su nombre es igual al nombre del fichero.
Dentro de la definición de la clase encontramos que no hay variables de clase, y las variables de instancia se estaban declarando agrupadas por lo que procedemos a declarar cada una en una línea diferente, hay un único constructor y hay override del método toString() que lo pasaremos al final.

Con respecto a los nombres de los identificadores, el de Package y el de clase están usando correctamente la convención UpperCamelCase, pero los métodos no están haciendo uso del lowerCamelCase, por lo que se procede a corregir.

No se aprecian condicionales ni bucles en el archivo para análisis.

En referencia a la visibilidad de los atributos podemos ver que los mismos no tienen modificadores de alcance por lo que procederemos a agregar el modificador private ya que para obtener sus valores podemos acceder a través de los getters.

Volver al índice

Resultados

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dog;

/**
 * Clase que permite la instanciación de objetos dog y la utilización de sus
 * métodos para mostrarlos por pantalla
 *
 * @author Alejandro
 */
public class Dog
{
   // Instance Variables
   private String name;
   private String breed;
   private String color;
   private int age;

   // Constructor Declaration of Class
   public Dog(String name,String breed,int
         age, String color)
   {
      this.name = name;
      this.breed = breed;
      this.age = age;
      this.color = color;
   }

   // method 1
   public String getName()
   {
      return name;
   }

   // method 2
   public String getBreed()
   {
      return breed;
   }
```

```
    // method 3
    public int getAge()
    {
        return age;
    }

    // method 4
    public String getColor()
    {
        return color;
    }

    public static void main(String[] args)
    {
        Dog tuffy = new Dog("tuffy","papillon", 5, "white");
        System.out.println(tuffy.toString());
    }

    @Override
    public String toString()
    {
        return("Hi my name is "+this.getName()+
                ".\nMy breed,age and color are "+
                this.getBreed()+","+this.getAge()+
                ","+ this.getColor());
    }
}
```

Volver al índice

Actividad 2

Descripción del trabajo realizado

Se procede a crear todas las etiquetas útiles para la generación de la documentación a través de la herramienta Javadocs.

Volver al índice

Resultados

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dog;

/**
 * Clase que permite la instanciación de objetos dog y la utilización de sus
 * métodos para mostrarlos por pantalla
 *
 * @author Alejandro
 */
public class Dog {

    /**
     * El nombre del perro.
     */
    private String name;

    /**
     * La raza del perro.
     */
    private String breed;

    /**
     * El color del pelaje del perro.
     */
    private String color;
```

```java
    /**
     * La edad del perro en años.
     */
    private int age;

    /**
     * Crea un nuevo objeto Dog con los datos especificados.
     *
     * @param name el nombre del perro.
     * @param breed la raza del perro.
     * @param age la edad del perro en años.
     * @param color el color del pelaje del perro.
     */
    public Dog(String name, String breed, int age, String color) {
        this.name = name;
        this.breed = breed;
        this.age = age;
        this.color = color;
    }

    /**
     * Devuelve el nombre del perro.
     *
     * @return el nombre del perro.
     */
    public String getName() {
        return name;
    }

    /**
     * Devuelve la raza del perro.
     *
     * @return la raza del perro.
     */
    public String getBreed() {
        return breed;
    }
    /**
     * Devuelve la edad del perro en años.
     *
     * @return la edad del perro en años.
     */
    public int getAge() {
        return age;
    }
```

```java
    /**
     * Devuelve el color del pelaje del perro.
     *
     * @return el color del pelaje del perro.
     */
    public String getColor() {
        return color;
    }

    /**
     * El método main crea una instancia de Dog y llama al método toString().
     *
     * @param args los argumentos de la línea de comandos (no se utilizan).
     */
    public static void main(String[] args) {
        Dog tuffy = new Dog("tuffy", "papillon", 5, "white");
        System.out.println(tuffy.toString());
    }

    /**
     * Devuelve una representación en cadena del objeto Dog.
     *
     * @return una cadena que describe el perro, incluyendo su nombre, raza,
     * edad y color.
     */
    @Override
    public String toString() {
        return ("Hi my name is " + this.getName()
                + ".\nMy breed,age and color are "
                + this.getBreed() + "," + this.getAge()
                + "," + this.getColor());
    }
}
```

Volver al índice

Notas

Se adjunta documentación generada con javadocs.

Volver al índice