

PAC 3. UF4.

*Utilización avanzada de
clases en el diseño de
aplicaciones.*

La clínica veterinaria Vetllerna nos ha pedido realizar una aplicación de escritorio con Java para la gestión de sus clientes. Durante todo el módulo, vamos a realizar poco a poco, el desarrollo del proyecto.

Ahora vamos a diferenciar los tipos de mascota, por lo que vamos a crear tres nuevas subclases de la creada en la PAC anterior.

- Gato

- Atributos: color
 - Constructor vacío
 - Constructor con parámetros
 - Métodos get y set
 - Método toString:
- Ejemplo: Luna es un gato de color blanco.

- Perro

- Atributos: raza
 - Constructor vacío
 - Constructor con parámetros
 - Métodos get y set o Método toString:
- Ejemplo: Luna es un perro de raza labrador.

- Roedor

- Atributos: tipo (Puede ser un String)
 - Constructor vacío
 - Constructor con parámetros
 - Métodos get y set
 - Método toString:
- Ejemplo: Luna es un conejo.

Continuando el proyecto anterior en el Paquete Vetllerna, vamos a desarrollar tres clases nuevas que serán las subclases de la clase que ya tenemos definida como Mascota.

Si recordamos, el Paquete Vetllerna se componía de dos clases, Cliente y Mascota, además de un VetllernaMain creado para demostrar la funcionalidad del proyecto realizado.

Cada una de estas subclases heredarán los métodos y atributos de la clase Mascota y a su vez tendrán atributos y métodos propios de cada una.

A continuación se muestra una captura de pantalla de la clase Mascota resumida en eclipse, es decir, con sus métodos plegados a modo de resumen:

```
1 package Vetllerna; //paquete
2
3 public class Mascota { //clase
4
5     private String nombre, genero; //atributos o propiedades de la clase
6     private int codigo;
7
8     public Mascota() {} //construcción vacía sin ningunos parametros
9
10    public Mascota(String nombre, String genero, int codigo) // constructor con parametros de tipo string e int
11
12    public String getNombre() // metodo get []
13
14    public String getGenero()
15
16    public int getCodigo()
17
18    public void setNombre(String nombre) // metodo set[]
19
20    public void setGenero(String genero)
21
22    public void setCodigo(int codigo)
23
24    public String toString() // metodo toString []
25 }
26
```

En las siguientes imágenes podremos ver las tres subclases nuevas creadas para este proyecto:

Subclase Gato:

```
package VetIlerna;

public class Gato extends Mascota { //con extends hacemos herencia de la clase Mascota a Gato

    private String color; //atributo propio de la clase Gato

    public Gato() {}

    public Gato(String color) //constructor con parametro de tipo String
    {
        super(); //llamar al constructor de la clase Mascota
        this.color=color;
    }

    public String getColor() //metodo get color
    {
        return color;
    }

    public void setColor(String color) //metodo set color
    {
        this.color=color;
    }

    public String toString() //metodo toString
    {
        return " es un gato de color " + color;
    }

}
```

Como podemos ver en esta primera subclase, y posteriormente en las siguientes, aparecen nuevos conceptos utilizados como que lo que hace es herencia de todo aquello que posee la clase Mascota, en la nueva subclase Gato.

Después creamos el atributo de la subclase, recordando que la nueva subclase además tiene los atributos de la clase Mascota.

Creamos un constructor vacío y luego otro con parámetros, donde utilizaremos la palabra super() para llamar al constructor de la clase Mascota.

El resto de métodos se realizan igual que anteriormente. Y como se demuestra en las siguientes imágenes las subclases Perro y Roedor se realizan de la misma manera, teniendo en cuenta sus atributos propios.

```
package VetIlerna;

public class Perro extends Mascota { //con extends hacemos herencia de la clase Mascota a Perro

    private String raza; //atributo propio de la clase Perro

    public Perro() {}

    public Perro(String raza) //constructor con parametro de tipo String
    {
        super(); //llamar al constructor de la clase Mascota
        this.raza=raza;
    }

    public String getRaza() // metodo get raza
    {
        return raza;
    }

    public void setRaza(String raza) //metodo set raza
    {
        this.raza=raza;
    }

    public String toString() //metodo toString
    {
        return " es un perro de raza " + raza;
    }

}
```

```
package VetIlerna;

public class Roedor extends Mascota { //con extends hacemos herencia de la clase Mascota a Roedor

    private String tipo; //atributo propio de la clase Roedor

    public Roedor() {}

    public Roedor(String tipo) //constructor con parametro de tipo String
    {
        super(); //llamar al constructor de la clase Mascota
        this.tipo=tipo;
    }

    public String getTipo() // metodo get tipo
    {
        return tipo;
    }

    public void setTipo(String tipo) //metodo set tipo
    {
        this.tipo=tipo;
    }

    public String toString() //metodo toString
    {
        return " es un " + tipo;
    }

}
```

Por último haremos mención al main realizado que no es más que una continuación del proyecto, donde podremos ver como funcionan todas las clases y sus métodos perfectamente.

En este caso mostraremos sólo como se ha llevado a cabo la creación de los objetos, sus clases y métodos.

Para los tres ejemplos hemos utilizado estos parámetros:

```

Caso gato=new Gato(); // definimos el estado nuevo

gato.setNombre("Luna"); //parametro de la clase Mascota
gato.setColor("blanco"); //parametro de la subclase Gato

System.out.println(gato.getNombre() + gato.toString()); //imprimen de subclase con métodos de Mascota y su subclase

Perro perro1=new Perro(); //Resetamos el proceso con los siguientes ejemplos viendo las diferencias en los atributos de cada subclase
|
perro1.setNombre("Luna");
perro1.setRaza("podenco");

System.out.println(perro1.getNombre() + perro1.toString());

Ratón raton1=new Ratón();

raton1.setNombre("Luna");
raton1.setTipo("ratón");

System.out.println(raton1.getNombre() + raton1.toString());

}
}

```

El resultado por pantalla es el siguiente:

```

Luna es un gato de color blanco
Luna es un perro de raza podenco
Luna es un ratón

```

En el archivo que se adjunta, además del presente documento explicando resumidamente el proceso y documentación del proyecto, encontraremos la integridad del proyecto comentado.