

Ejercicio Propuesto

Bloque 2

Objetivo

- El objetivo de este ejercicio es aprender a identificar clases y aplicar los conceptos de:
 - Herencia. Identificando aquella clase que depende jerárquicamente de otra, extendiéndola y dotándola de nuevos atributos y métodos.
 - Polimorfismo. Sobrescribiendo desde la clase hija los métodos de la clase padre.

Enunciado del ejercicio

- Crear las clases necesarias para clasificar una serie de animales de compañía. Una vez implementadas, las clases podrán ser importadas en aplicaciones veterinarias.
- Es necesario modelizar las siguientes entidades con la información que aparece a continuación.
 - Mascota, nombre.
 - Perro, se deberá almacenar la raza.
 - Pájaro, que deberá almacenar la especie.
- Las clases que se implementen deberán disponer de los métodos apropiados para consultar los datos que almacenan, y un método que permita obtener la cadena de texto que represente al objeto.

Paso 1

- Crear la clase Mascota con:
 - el atributo nombre (de tipo String)
 - un constructor con parámetros y otro sin parámetros.
 - los siguientes métodos:
 - `public String toString():` Este método devuelve una cadena de texto que representa al objeto. Este método, se hereda por defecto de la clase `Object`. Por lo tanto es necesario sobrescribirlo, y que devuelva el nombre de clase seguido del nombre de la mascota, el cual se almacena en el atributo `nombre`.

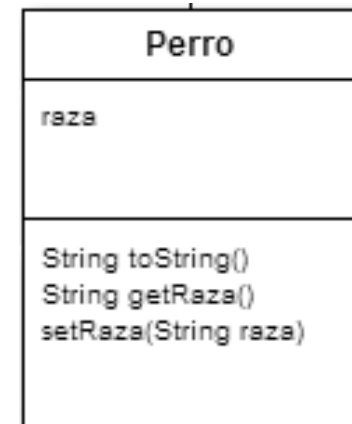
Por ejemplo: “Mascota: Toby”

- `public String getNombre():` Devolverá una cadena de texto que contenga el nombre de la mascota.
- `public void setNombre(String nombre):` Permite asignarle un nombre a la mascota.

Mascota
nombre
<code>String toString()</code> <code>String getNombre()</code> <code>void setNombre(int nombre)</code>

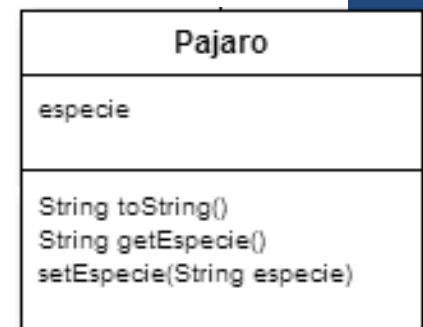
Paso 2

- Crear la clase Perro que herede de la clase Mascota. Añadirle el atributo raza, los constructores y los siguientes métodos:
 - `public String toString()` → Será necesario sobrescribir el método para que diga también la raza del Perro, concatenando el nombre de la clase, el nombre del animal (atributo heredado), y el texto raza seguido del valor del atributo raza. Por ejemplo:
“Perro: Toby Raza: caniche”
 - `public String getRaza()` → Devuelve un String con la raza del perro.
 - `public void setRaza(String raza)` → Le asigna al atributo de la clase el valor del parámetro recibido por el método.



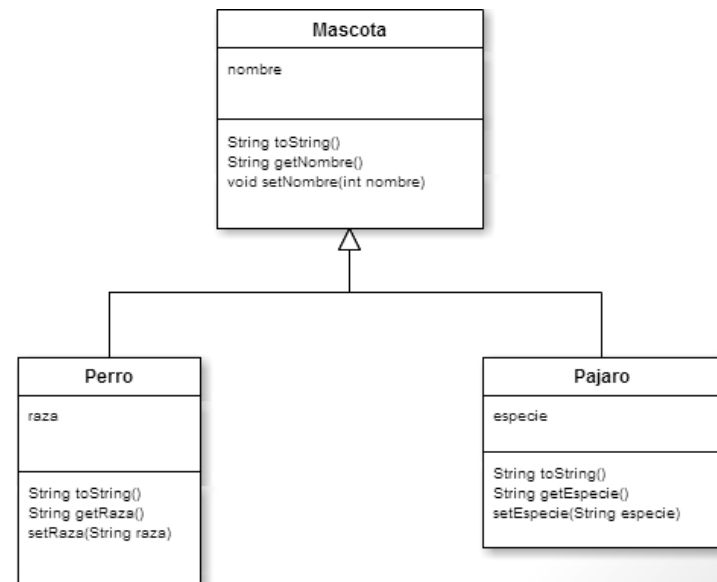
Paso 3

- Crear la clase Pajaro que extienda de la clase Mascota, Añadir a esta clase el atributo especie, los constructores y los siguientes métodos:
 - `public String toString()` → Será necesario sobrescribir el método para que diga también la especie del pájaro, de forma semejante al de la clase Perro. En este caso deberá devolver un texto semejante al siguiente:
“Pajaro: Piolin Especie: canario”
 - `public String getEspecie()` → Devuelve un String con la especie del pájaro.
 - `public void setEspecie(String especie)` → Le asigna al atributo de la clase el valor del parámetro recibido por el método.



Resumen de situación del ejercicio

- Llegados a este punto del ejercicio, el alumno habrá modelado tres tipos de objetos, que nos permiten clasificar: Mascotas (desde un punto de vista genérico), Perros y Pájaros.
- A continuación se creará una clase que reciba un array de Mascotas, y muestre por pantalla los textos que representan los objetos. Lógicamente hará uso del método toString que se ha implementado en las clases



Paso 4

- Crear una clase IdentificadorMascotas, con un método estático denominado identificarMascotas que recibe por parámetro un array de Mascotas y muestra por pantalla la cadena de texto que representa cada una de las mascotas del array.

```
public static void identificarMascotas(Mascota[] animales)
```

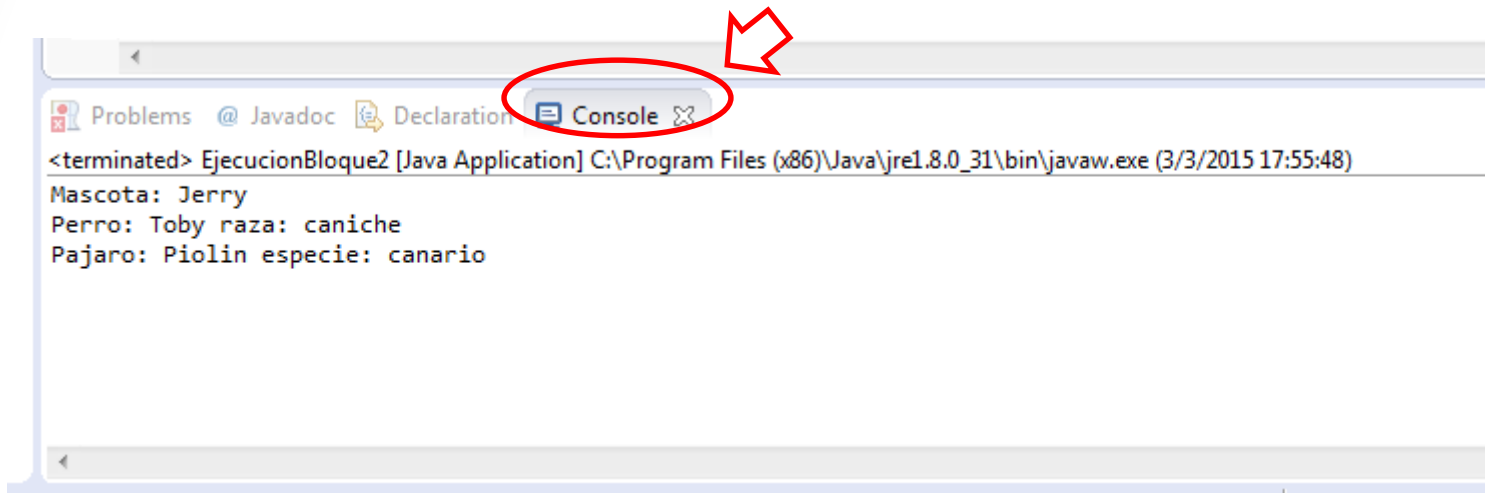
- Este método se recorrerá el array de Mascotas de forma semejante a como se realizaba en el ejercicio del Bloque 1
- Dado que todas las clases implementadas sobrescriben el método toString para obtener la cadena de texto que representa dichas clases. Se deberá hacer uso de dicho método para mostrar por pantalla el texto que lo representa.

Ejecución del ejercicio

- Para ejecutar las clases implementadas, se utilizará el siguiente fragmento de código:

```
public class EjecucionBloque2 {  
  
    public static void main(String[] args) {  
  
        Mascota unaMascota = new Mascota("Jerry");  
        Perro unPerro = new Perro("Toby", "caniche");  
        Pajaro unPajaro = new Pajaro();  
        unPajaro.setNombre("Piolin");  
        unPajaro.setEspecie("canario");  
  
        Mascota[] animales = new Mascota[3];  
        animales[0]=unaMascota;  
        animales[1]=unPerro;  
        animales[2]=unPajaro;  
  
        IdentificadorMascotas.identificarMascotas(animales);  
  
    }  
  
}
```

Resultado de la ejecución en Eclipse



```
<terminated> EjecucionBloque2 [Java Application] C:\Program Files (x86)\Java\jre1.8.0_31\bin\javaw.exe (3/3/2015 17:55:48)
Mascota: Jerry
Perro: Toby raza: caniche
Pajaro: Piolin especie: canario
```