

| Nombre y apellidos | Nota |
|--------------------|------|
| | |

18/04/2024

RECUPERACIÓN 2ª EVALUACIÓN PROGRAMACIÓN

1. Se desea realizar una aplicación Java para la gestión de una clínica veterinaria, almacenando las mascotas, los veterinarios y las especialidades.

De las **especialidades** hay que almacenar su identificador y su nombre.

Sobre los **veterinarios** se deberá almacenar los datos propios de **persona** (el DNI, el nombre, la edad y la dirección), su sueldo mensual, la especialidad a la que se dedica y la fecha de contratación. Además, se deberá llevar un registro de las mascotas a las que ha ido atendiendo cada uno a lo largo del tiempo, en una estructura de clave (identificador de tipo numérico), valor.

En cuanto a las **mascotas**, se almacenará su nombre, especie (perro, gato, ave, exótico), raza, fecha de nacimiento y fecha de registro en la plataforma.

Ambas entidades, veterinarios y mascotas, deberán implementar obligatoriamente un método `calcularCoste`. En el caso de los veterinarios, el resultado de dicho método será el cálculo de su sueldo mensual, multiplicado por 14 pagas, más un plus de un 10% si su edad es superior a 40 años. En el caso de las mascotas, el resultado se obtendrá multiplicando los años desde la fecha de registro de la mascota en la plataforma por una media de 150€ al año. *Por ejemplo, si el Loro Rubí se ha registrado en la plataforma el 16 de julio de 2021, el resultado de `calcularCoste` será $3 \text{ años} * 150€ = 450€$.*

Puntuación jerarquía de clases: 2.5 puntos.

Por último, desarrolla una clase ejecutable que realice lo siguiente:

- Crear una lista de veterinarios. **(0.25 pts)**
- Deberá existir un menú con las opciones a continuación.
- Dar de alta veterinarios. Los datos se recogerán por teclado (únicamente los personales). **(2.25 puntos)**

Se deberá comprobar la validez de los datos introducidos y lanzar una excepción informando de lo sucedido en los siguientes casos:

- En caso de que la edad introducida sea un número negativo.
- En caso de que el DNI no posea una longitud de 9 caracteres.
- Para ello, se creará una clase para cada excepción y se lanzará la que corresponda en cada caso.
- Elimina de la lista aquellos veterinarios cuya especialidad sea dermatología, ya que no se atiende en la clínica. **(1.5 punto)**
- Calcula el salario medio anual (sueldo mensual * 14 pagas) de los veterinarios especialistas en traumatología. **(1.5 puntos)**
- Muestra por pantalla cuántas mascotas de las atendidas en la clínica tienen más de 6 años. **(2 puntos)**

NOTA 1: Se proporciona un fichero ejecutable (*VeterinarioPrincipal.java*) que se deberá completar donde corresponda, para llevar a cabo las funcionalidades solicitadas.

NOTA 2: Se proporcionan también las documentaciones de algunas clases Java necesarias para el desarrollo del programa.

NOTA 3: Estas son algunas instrucciones útiles para el manejo de fechas en alguno de los apartados.

```
// CONVERTIR STRING A LOCALDATE

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd/MM/yyyy");

String date = "16/08/2022";

LocalDate localDate = LocalDate.parse(date, formatter);

// COMPROBAR QUE UNA FECHA ESTÁ ANTES O DESPUÉS QUE OTRA

LocalDate d1 = LocalDate.of(2022, Month.AUGUST, 13);

LocalDate d2 = LocalDate.of(2022, Month.APRIL, 13);

System.out.println("d1.isAfter(d2 ) : " + d1.isAfter(d2));

System.out.println("d1.isBefore(d2 ) : " + d1.isBefore(d2));

// OBTENER NÚMERO DE DÍAS ENTRE DOS FECHAS

LocalDate dt = LocalDate.parse("2018-11-21");

LocalDate dt1 = LocalDate.parse("2018-11-14");

System.out.println(dt1.compareTo(dt));

// COMPARAR 2 FECHAS LOCALDATE Y OBTENER AÑOS

int anos = Period.between(dt, dt1).getYears();
```