

Práctica Diagrama de Clases

FECHA LÍMITE DE ENTREGA: lunes 23 de marzo





ENUNCIADO: Se necesita realizar una aplicación para el control de una clínica veterinaria. El sistema gestiona la siguiente información:

- Los datos sobre los dueños de las mascotas y sobre los médicos. De ellos se sabe su nombre, su número de teléfono y su dirección. Además, de los dueños se sabe su fecha de ingreso y de los médicos su especialidad.
- De los animales se sabe su nombre, edad y dueño (que es único). También se guarda datos sobre su historial de visitas. De cada visita se guarda la fecha y el diagnóstico (normalmente una cadena de texto indicando que le ocurría al animal).
- Por último, se guarda información de las citas. Cada cita se da entre una mascota y un médico, y de ella se guarda la fecha de la cita y la hora.
- El programa debe permitir: crear una cita para una mascota, cambiar la fecha y hora de una cita existente y, por último, dado un día recuperar el diagnóstico de ese día para una mascota.

SE PIDE: Crear el diagrama de clases que gestione el sistema anterior descrito utilizando StarUML.

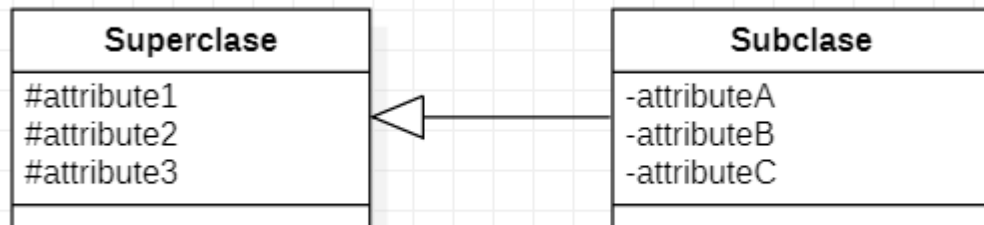
- Se debe crear las clases con los atributos y métodos de cada una indicando su visibilidad. Crear las relaciones entre clases, indicando la cardinalidad de cada una y marcando los atributos de relación.
- Utilizar la herramienta de StarUML para generar las clases de Java. Después, utilizando NetBeans crear *getters* y *setters* para todos los atributos de todas las clases. Generar los constructores y el método *toString()* para los atributos que no son de relación para todas las clases.
- Subir el diagrama y el proyecto de NetBeans a vuestra cuenta de GitHub. El nombre del repositorio debe ser "DAW1-Sauces". La estructura de directorios del repositorio es la siguiente.

 README.md	Archivo MD	1 KB
 PracticaFinal.mdj	StarUML Model	34 KB
 PracticaFinalNB	Carpeta de archivos	
 .git	Carpeta de archivos	

 Pablo and Pablo Cambio segundo rama pruebas		Latest commit 8e23af2 1 minute ago
 PracticaFinalNB	Cambio segundo rama pruebas	1 minute ago
 PracticaFinal.mdj	Cambio segundo rama pruebas	1 minute ago
 README.md	Initial commit	16 minutes ago

NOTAS: Tened en cuenta las siguientes pautas a la hora de realizar la práctica.

- La visibilidad de todos los métodos debe ser `public` y la de los atributos `private` con una excepción. Los atributos de las superclases en una herencia deben ser `protected` (se indica mediante un “#”, esto se me pasó en los ejemplos de clase). Mira que los atributos de una subclase siguen siendo privados.



- A la hora de crear el proyecto de NB debéis crear un proyecto nuevo de NB (solo con el Main) y luego copiar los archivos de las clases generadas con StarUML dentro de la carpeta “sources” del proyecto de NB.
- A la hora de subir los ficheros a GitHub, debéis crear un repositorio desde la página e importarlo en NB. Una vez importando simplemente copiar el archivo del diagrama y el proyecto de NB dentro de la carpeta siguiendo el árbol de directorios que se indicó arriba. Después ya se puede subir el repositorio a GitHub como hemos visto en clase.

Si tenéis dudas mi correo es pablo.rodper.6@educa.jcyl.es