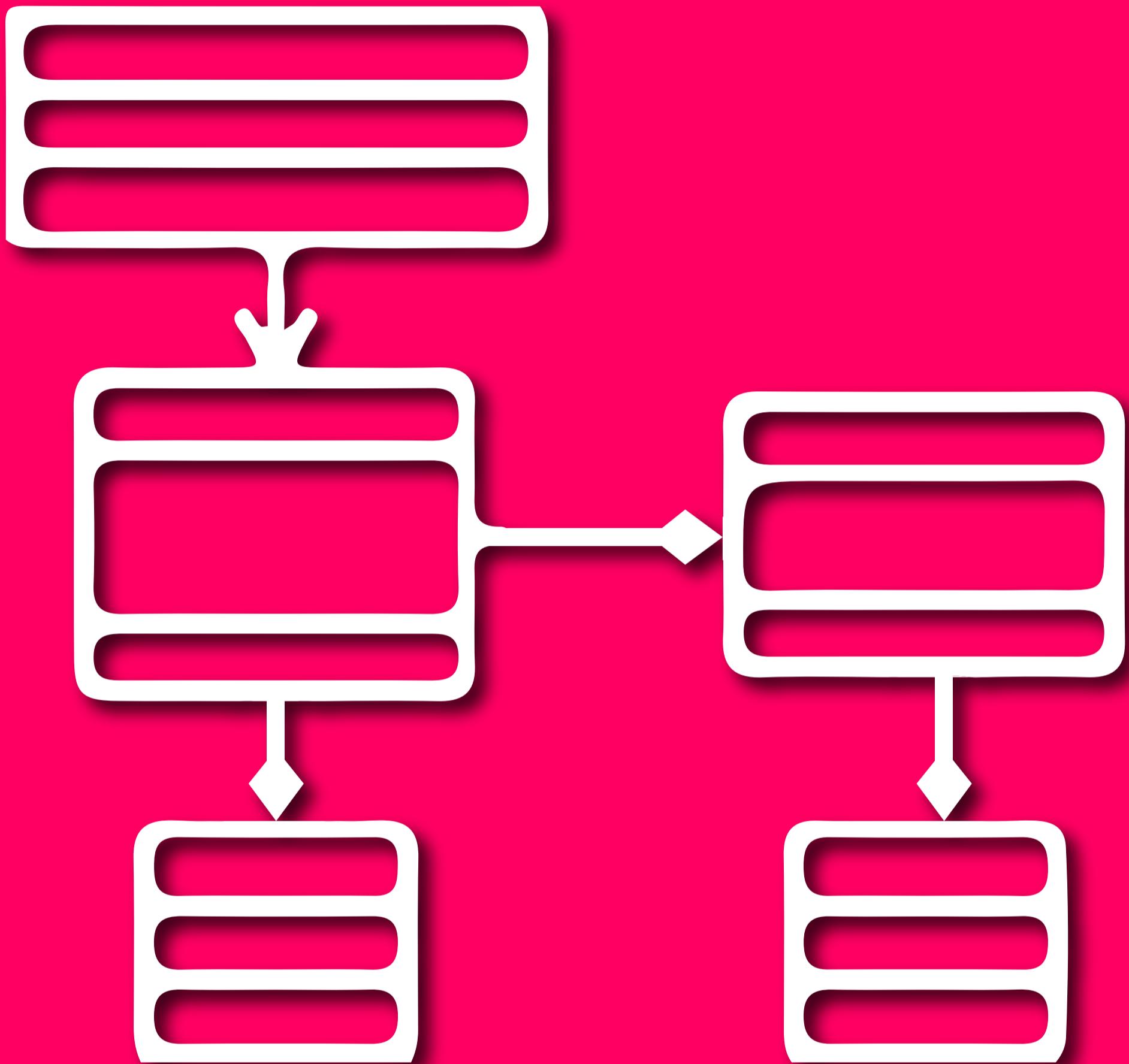




El futuro digital
es de todos

MinTIC



Relación entre clases

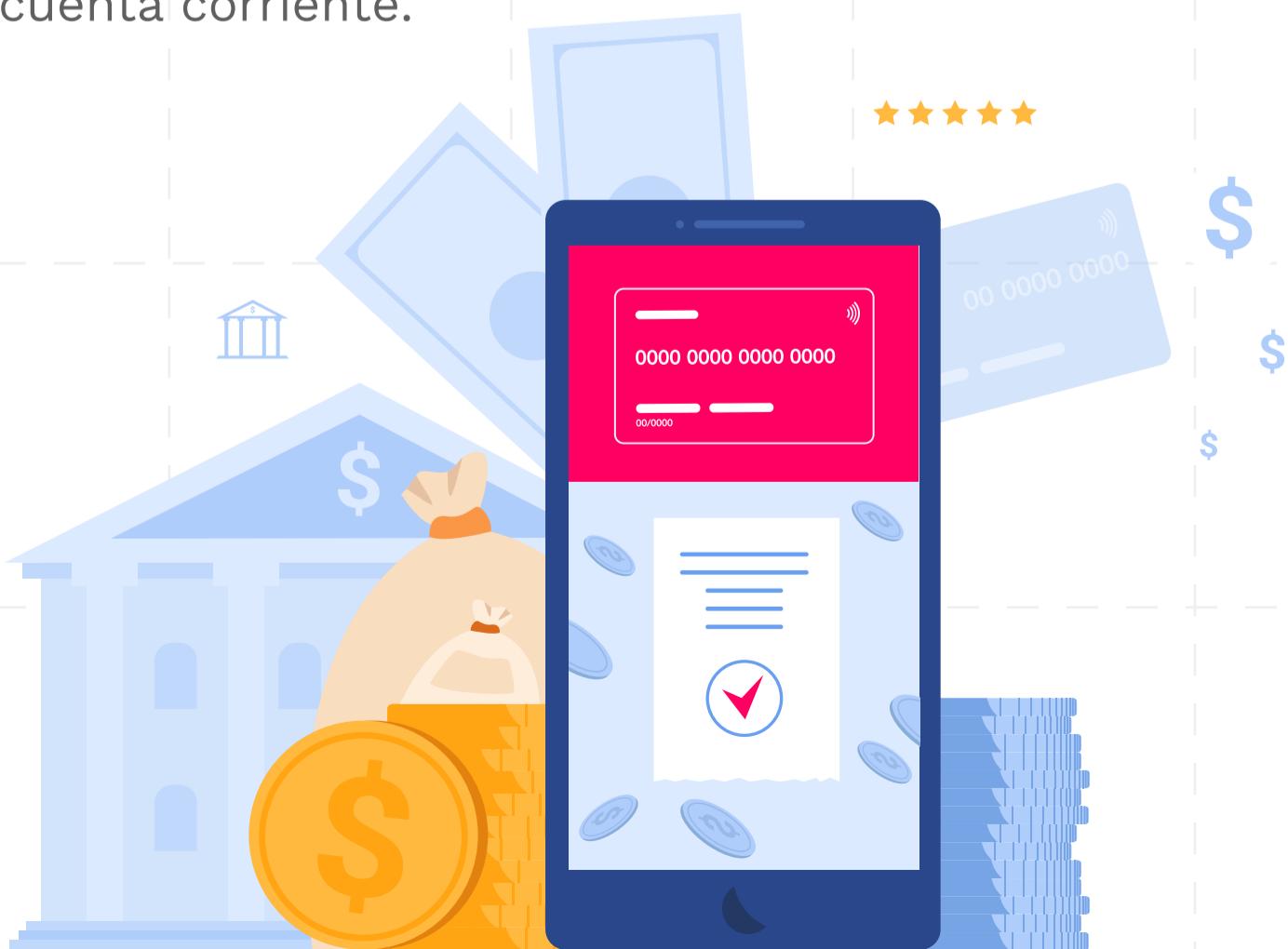
Parte 1

Hola:

Hasta el momento ya conocemos los conceptos básicos que debemos emplear para avanzar hacia los temas de esta semana. Los temas principales que veremos serán las relaciones entre clases, UML avanzado y estructura de métodos compartidos entre clases. Esto nos ayudará a personalizar mucho más el *software* que estemos desarrollando. Además, estas temáticas permitirán que se realice un desarrollo de *software* ordenado y de buena calidad.

Inicialmente les explicaremos todo lo que tiene que ver con relaciones entre clases, es decir, no solo vamos a tener una clase en nuestro sistema como lo hicimos la semana pasada. En esta ocasión veremos cómo se define un conjunto de clases, a las cuales les definiremos unas relaciones que poseen en la vida real.

Las relaciones que existen entre las diferentes clases indican cómo estas se comunican entre sí. Les mostraremos las principales relaciones que existen. La primera es la **asociación**, que es una relación estructurada entre clases, donde se asocian dos (2) o más clases de manera unidireccional o bidireccional (flecha o línea continua respectivamente), un ejemplo de la asociación de clases se puede ver cuando un cliente de un banco tiene asociada una cuenta. El cliente puede tener asociados varios productos del banco y entre ellos una cuenta de ahorros o una cuenta corriente.



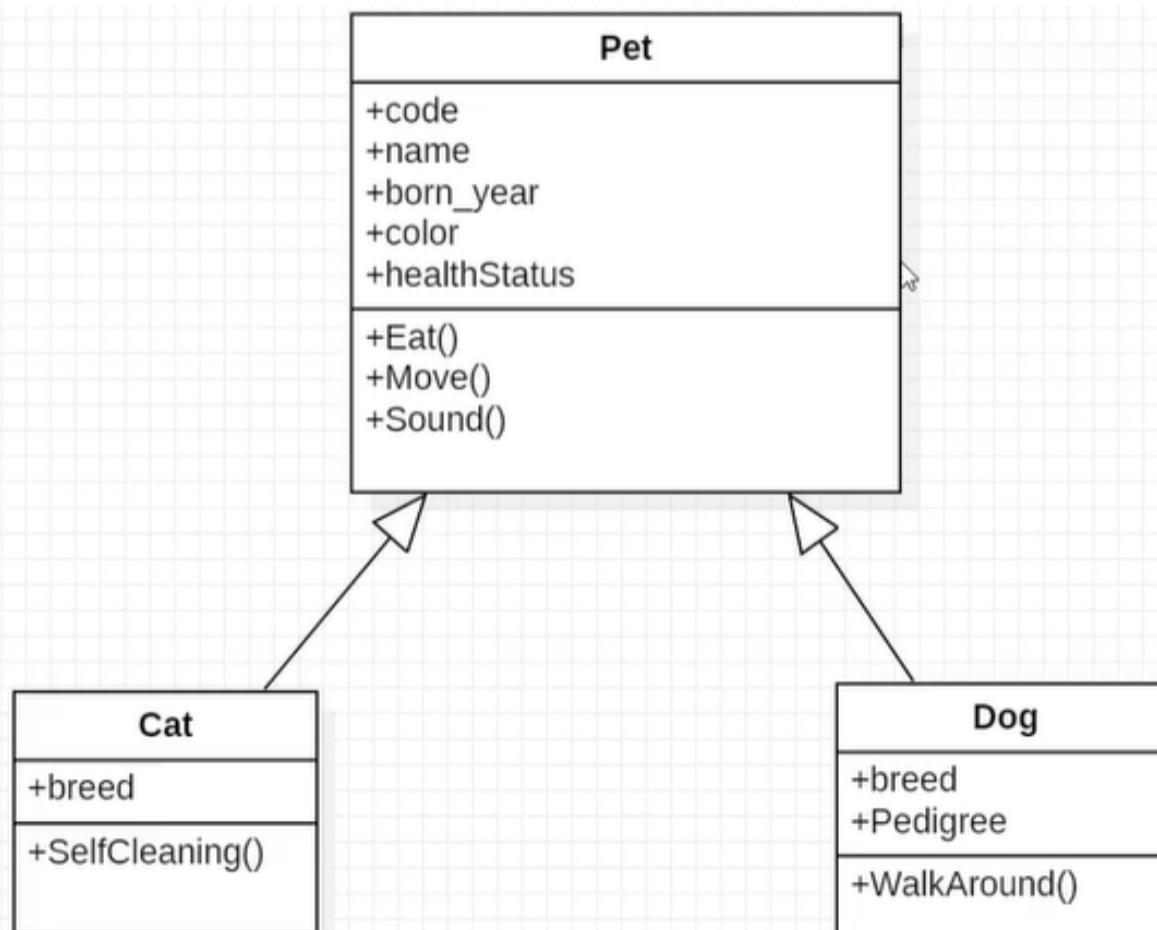
También existen unas relaciones que pertenecen a la asociación, pero son casos muy específicos, estas son las relaciones de **agregación** y de **composición** (rombo vacío y relleno respectivamente). La relación de agregación consiste en una **clase** que puede ampliarse con otra, sin embargo, la segunda no es fundamental para que la primera exista. Por ejemplo, una empresa puede agregar clientes, sin embargo, si la empresa no tiene clientes aún así sigue siendo una empresa que a futuro puede agregar clientes. Por su parte la relación de composición si es una relación muy estrecha entre las clases y como su nombre lo dice una **clase** se compone de otra u otras clases. Por ejemplo, la clase avión se puede componer de la clase "Ala", la clase "Motor" y la clase "Cola". Haciendo un análisis un avión sin alas deja de ser un avión, un avión sin motor no podrá volar, por lo cual tampoco será un avión. Esa es la gran diferencia entre estas clases, la agregación no es obligatoria, pero la asociación si.

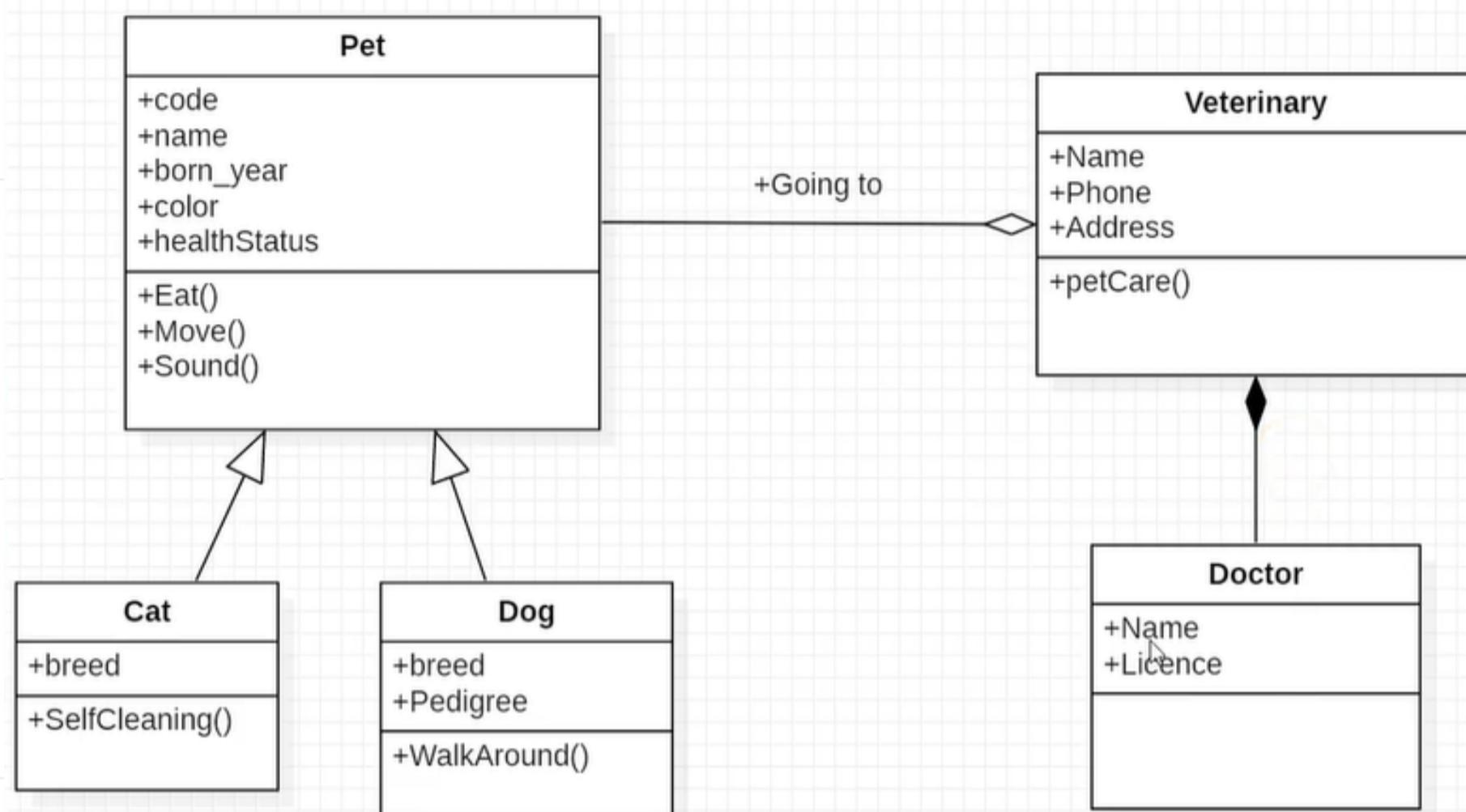


Existe una relación llamada **dependencia**, la cual es más débil que la **asociación** y muestra una interacción entre dos clases, donde la primera hace uso de una funcionalidad expuesta por la segunda para ejecutar una acción propia. Un ejemplo de este tipo de relación es el cálculo de una ecuación de segundo grado, la cual necesita internamente el cálculo de la raíz cuadrada, operación que se encuentra en una **clase** de Operaciones Matemáticas Básicas.

Finalmente, la relación que es más utilizada se llama **herencia**, como su nombre lo dice, consiste en la herencia que se da desde una clase padre hacia sus hijas, donde se pueden compartir atributos y métodos dependiendo de su nivel de acceso. Recordemos que existen tres (3) niveles de acceso: el público, que es para que sea accedido desde cualquier parte, el privado, que es solo para que sea accedido desde la misma clase y el protegido, para que sea accedido solo desde las clases que heredan de la clase que lo posee, es decir, estos elementos protegidos son los que serán heredados específicamente.

Video de diseño de diagrama de clases en UML (Ejemplo de clase gato y perro que heredan de mascota, y mascota se relaciona con veterinaria con agregación, y veterinaria se relaciona con veterinario mediante la composición)





Ya tenemos diseñado el diagrama de clases, el cual además de presentar los atributos y métodos de cada **clase**, define las relaciones que existen entre ellas. Los invito a practicar con este nuevo conocimiento dando solución al reto de la semana.



Universidad de Caldas