Notes - Inheritance/Polymorphism

polymorphism means many forms it is where you can make an attribute change depending on the inputs.

Inheritance allows you to share attributes/methods that are common between two classes

The relationship between the Super Class and sub class is an “is a”. For example a car “ is a” vehicle. or a eagle “is a” bird.

Open close principle - Abstraction

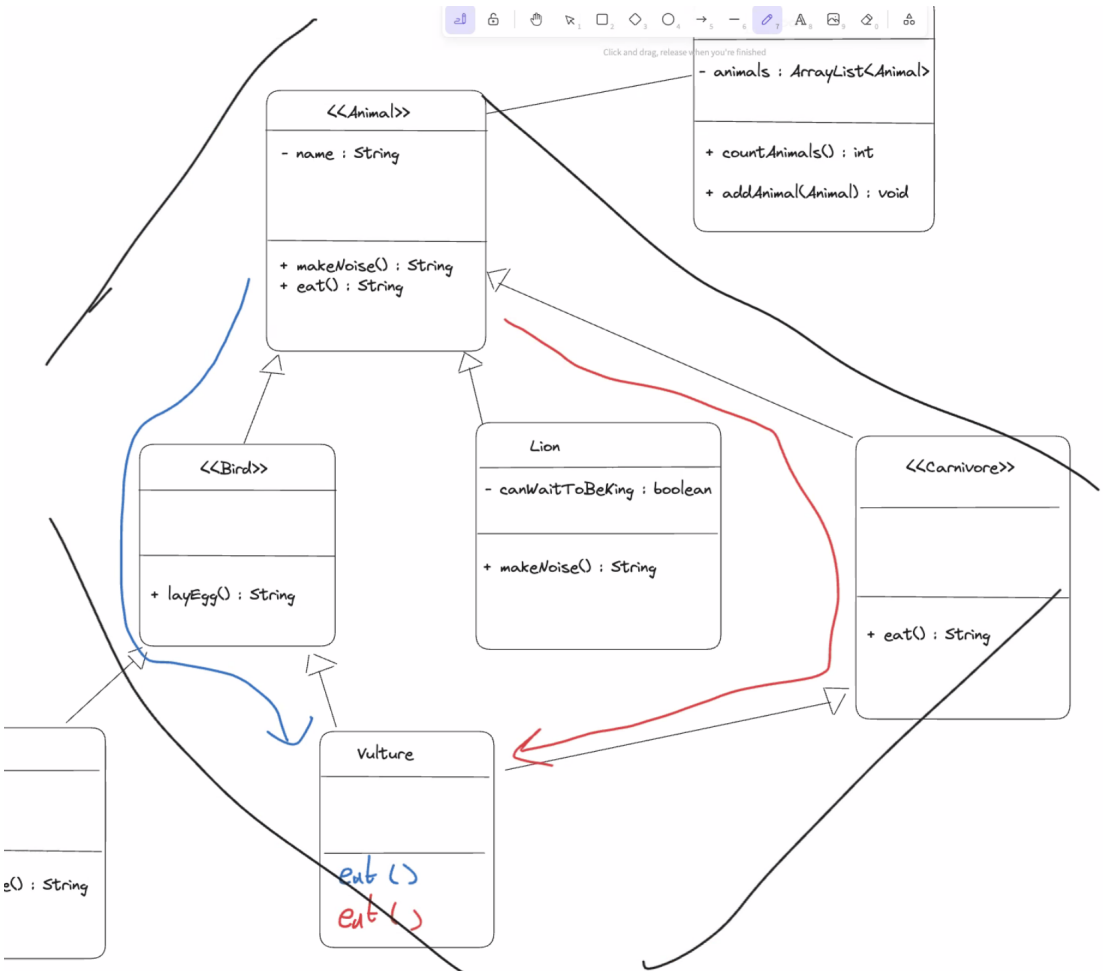
Abstract classes cannot be instantiated, you can still use it as a “datatype” but cannot use new constructor. you would have to assign the parent class datatype with the child class object. This is to stop users instantiating an object that doesn't have completed functionality, its used mainly to give all the child classes common attributes.

Cant technically test abstract classes because they cant be instantiated so should test child classes.

On class diagram you can do **<<ClassName>>** to indicate it is abstract. or put abstract before.

Inheritance chain A class ← B class ←C class. Be careful in doing long chains it makes it difficult to check variables and indicates architecture should be changed.

The diamond problem - **eat()** gets overridden in carnivore so vulture gets 2 different ones which one does it use.



You can only inherit one class to avoid this. C,Python lets you do it but java doesn't. Interfaces (Abstraction) fixes the issue.

Protected variables allows us to access the variables of the super class in the sub classes. When you want to access methods of the super class in the child class you can do super.method()

if you have a method in the parent class and its overriden in the child classes. you still cannot delete the method from the parent class as when you call the object parent class it wont recognise that the child classes have the method.