

# Abstract Classes and Interfaces

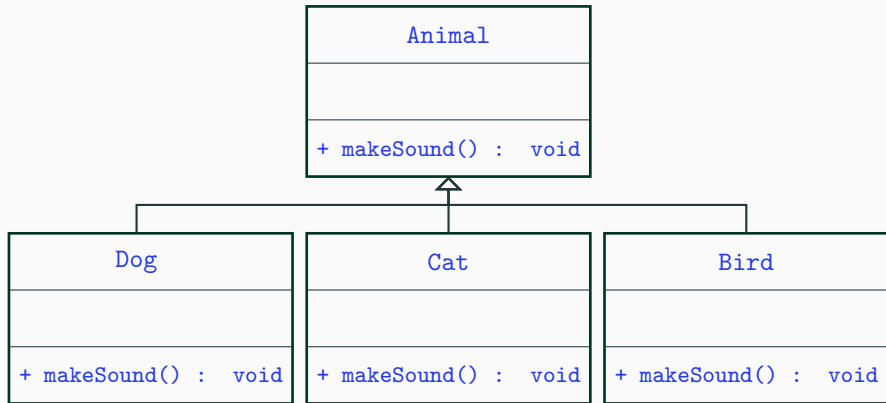
---

Shirley B. Chu

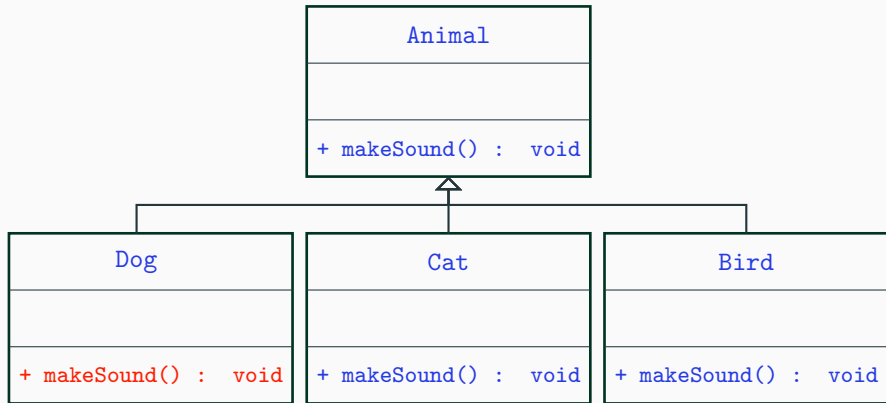
June 22, 2020

De La Salle University  
College of Computer Studies

# Animals

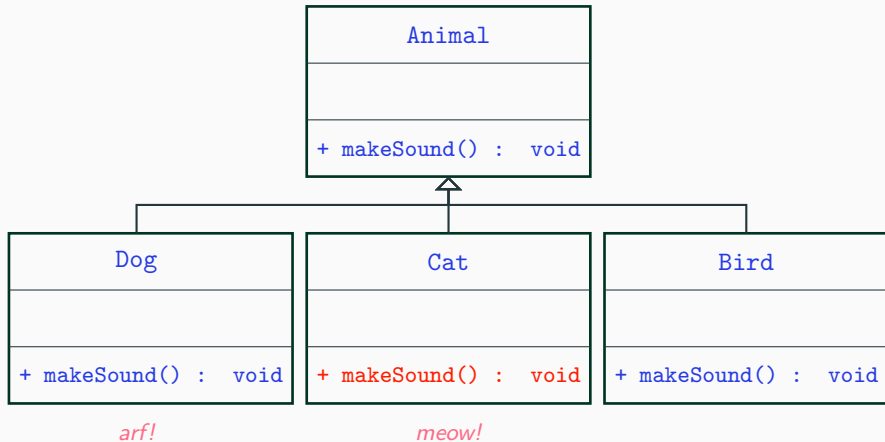


# Animals

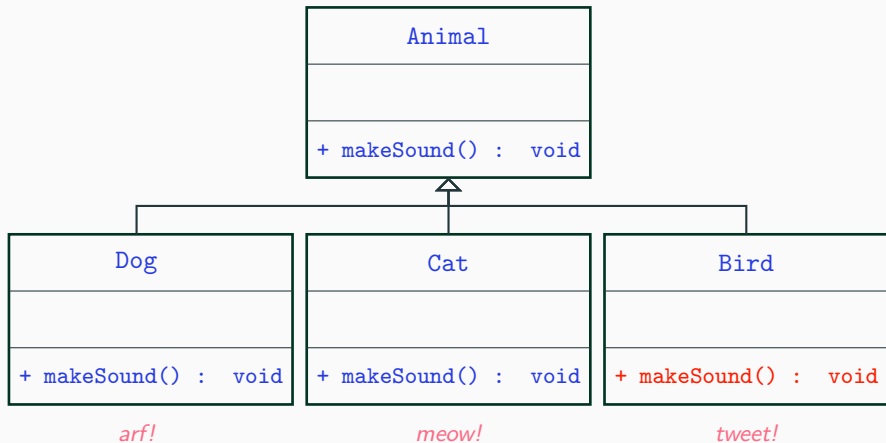


*arf!*

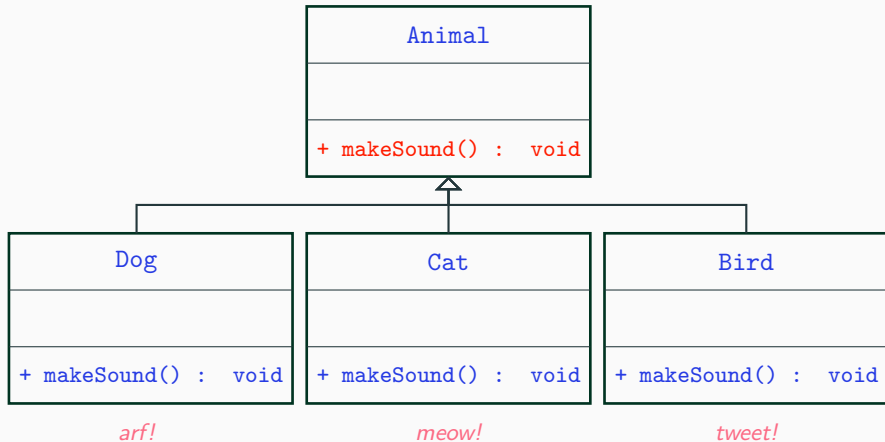
# Animals



# Animals



# Animals



## Abstract method

a method that does not have an implementation.

## Abstract method

a method that does not have an implementation.

Its implementation depends on the subclasses that **must** override it.



## Abstract method

a method that does not have an implementation.

Its implementation depends on the subclasses that **must** override it.

Once a class contains an abstract method, it is automatically an abstract class.

## **Abstract class**

a class that cannot be instantiated.

## **Abstract class**

a class that cannot be instantiated.

## **Abstract class**

a class that cannot be instantiated.

Its only purpose is for it to be a parent of a more specific class.

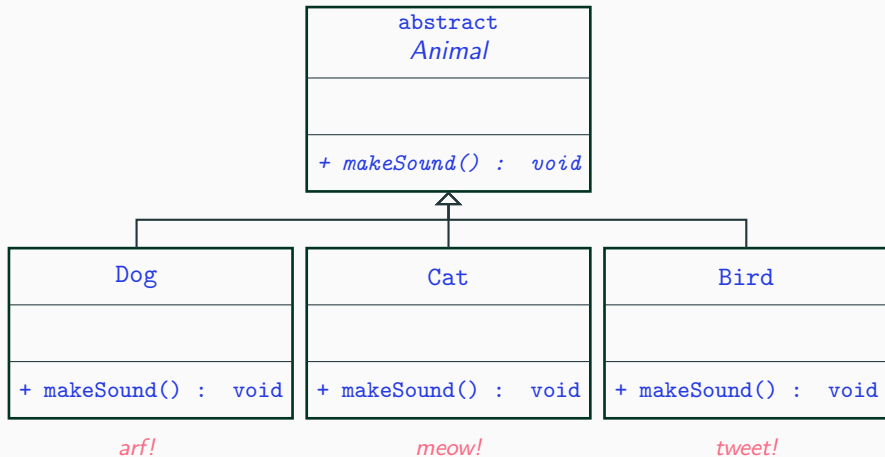
## **Abstract class**

a class that cannot be instantiated.

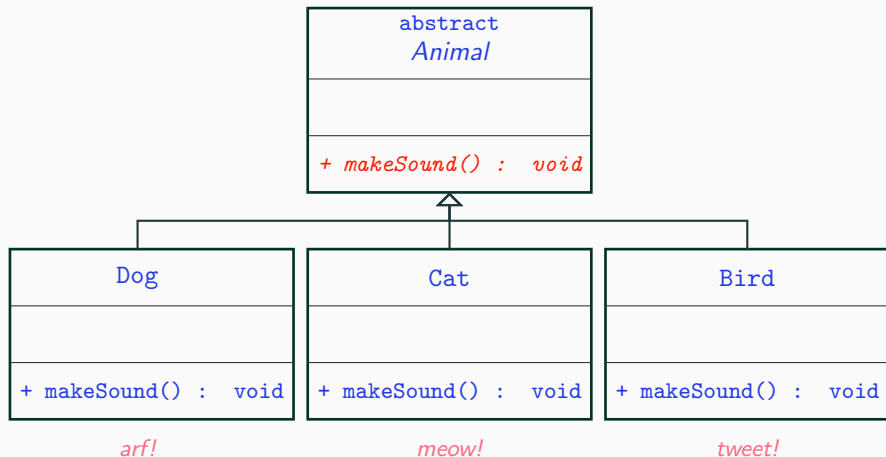
Its only purpose is for it to be a parent of a more specific class.

It can have zero or more abstract methods in it.

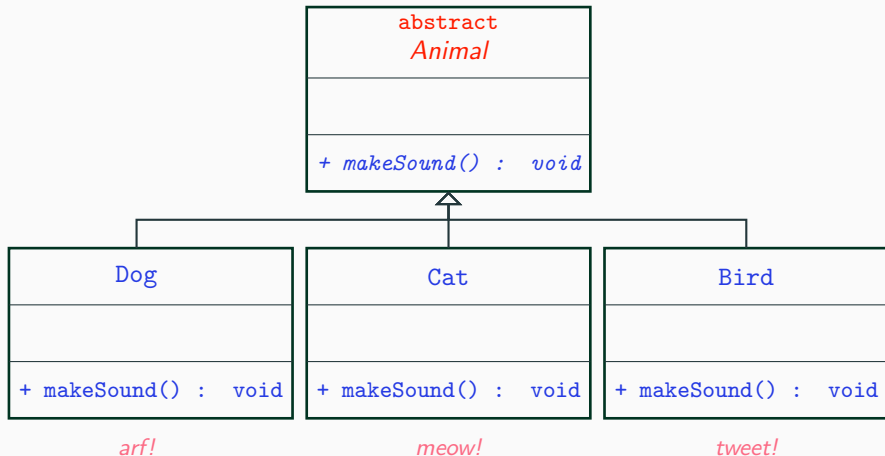
# Animals



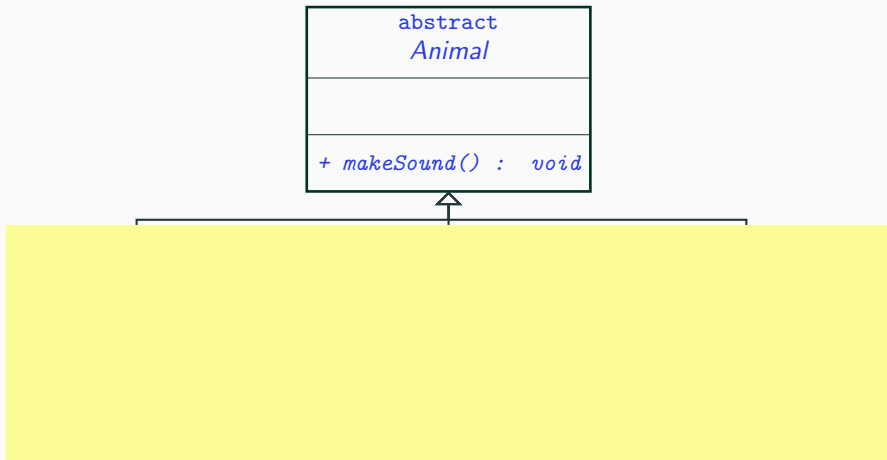
# Animals

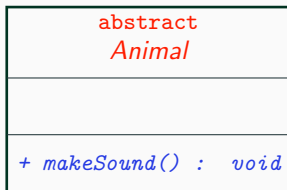


# Animals

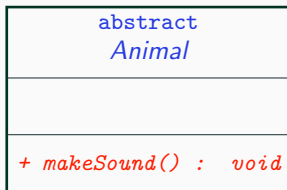






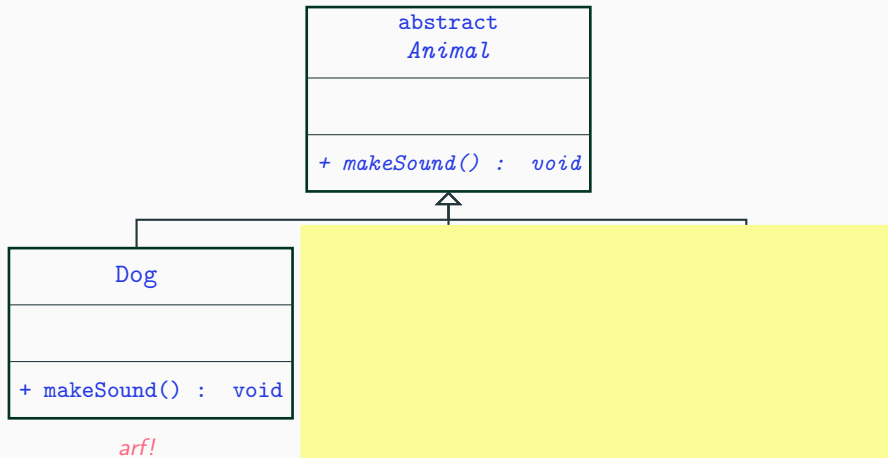


```
public abstract class Animal
{
}
}
```

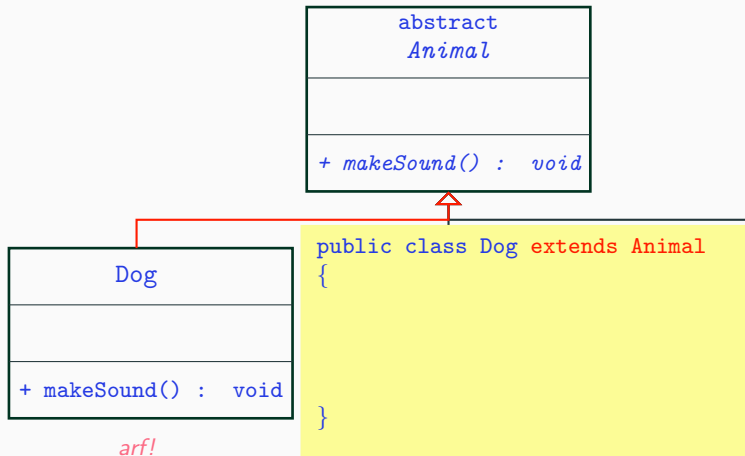


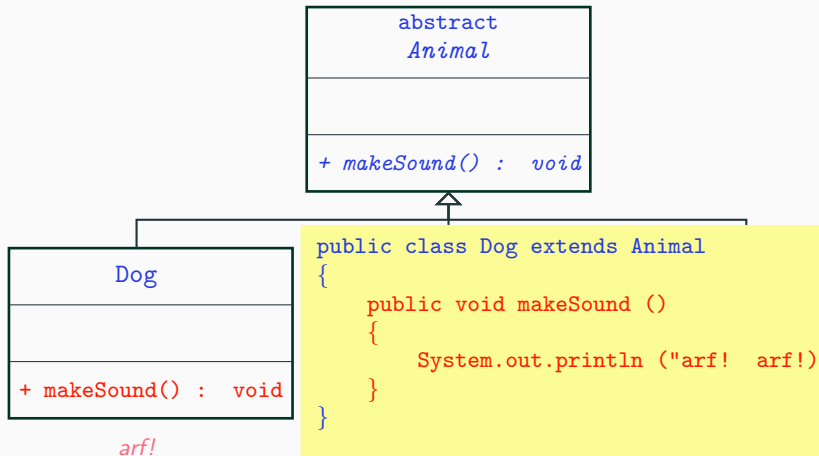
```
public abstract class Animal
{
    public abstract void makeSound ();
}
```

# Animals

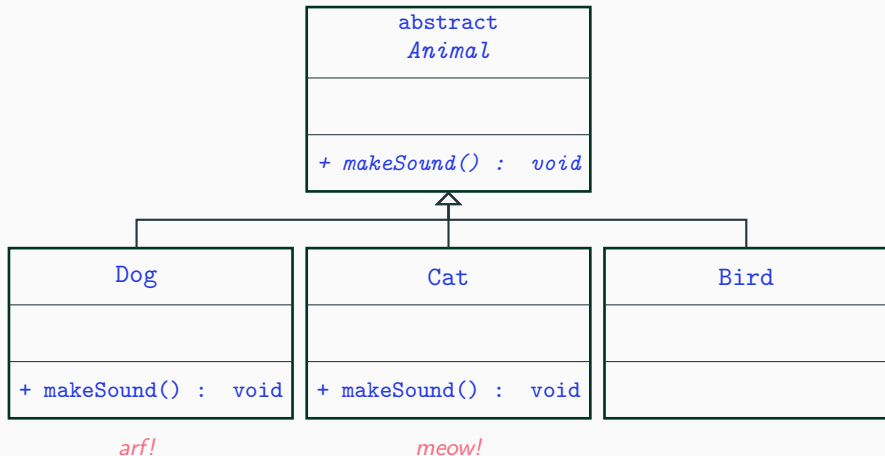


# Animals

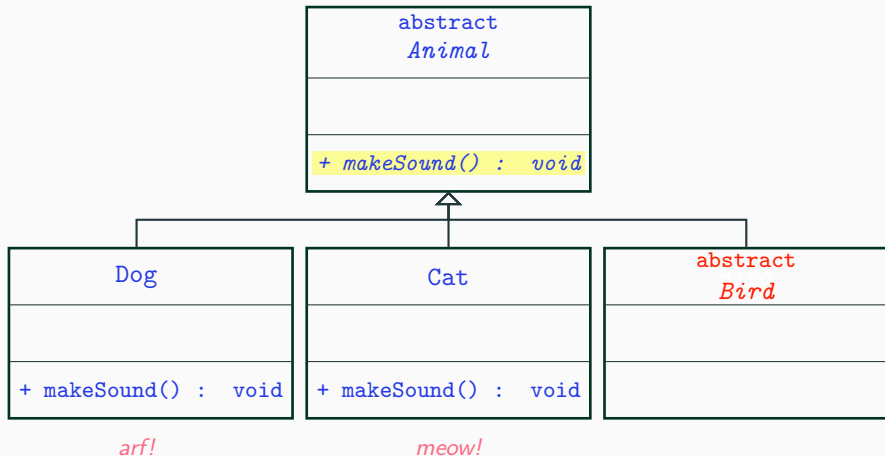




# What if...



# What if...





An **interface**

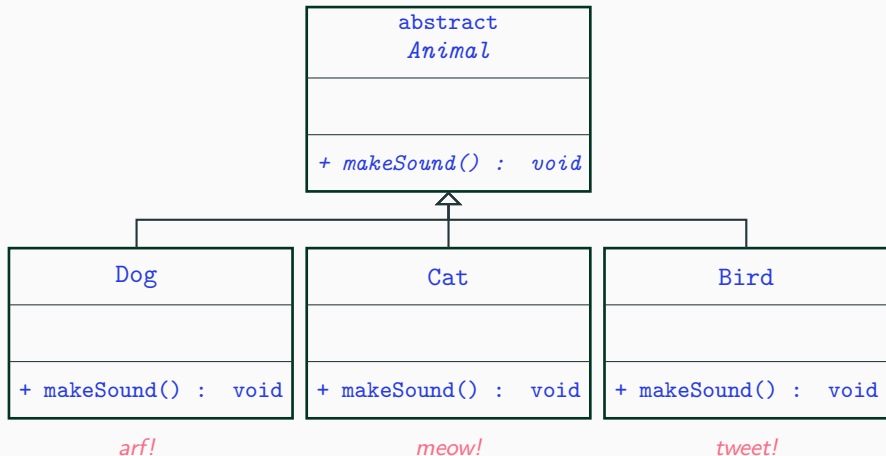
a collection of abstract methods.

An **interface**

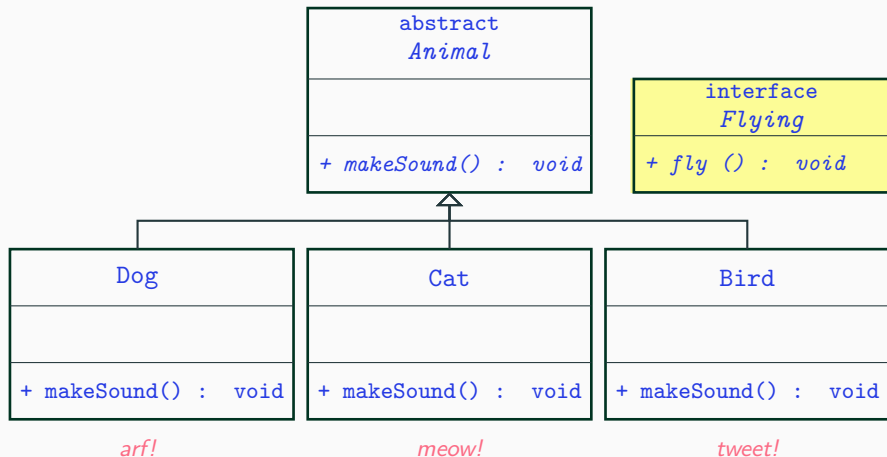
a collection of abstract methods.

The purpose is to ensure that the classes that will implement it must have the listed abilities.

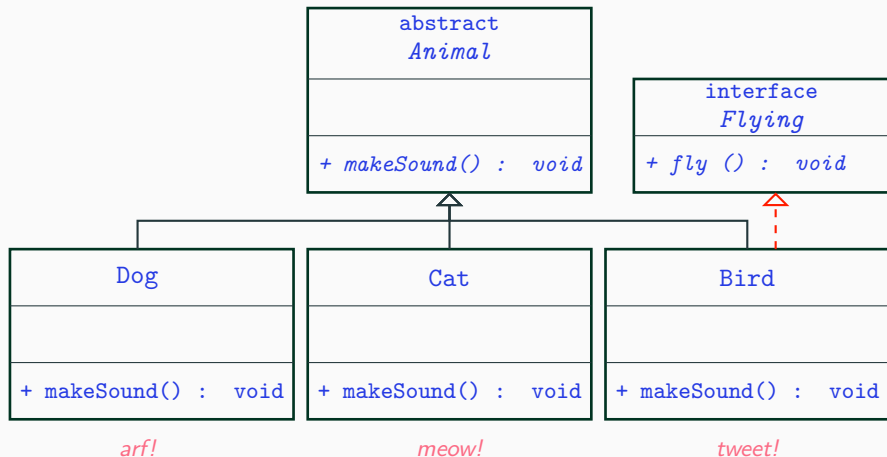
## Example



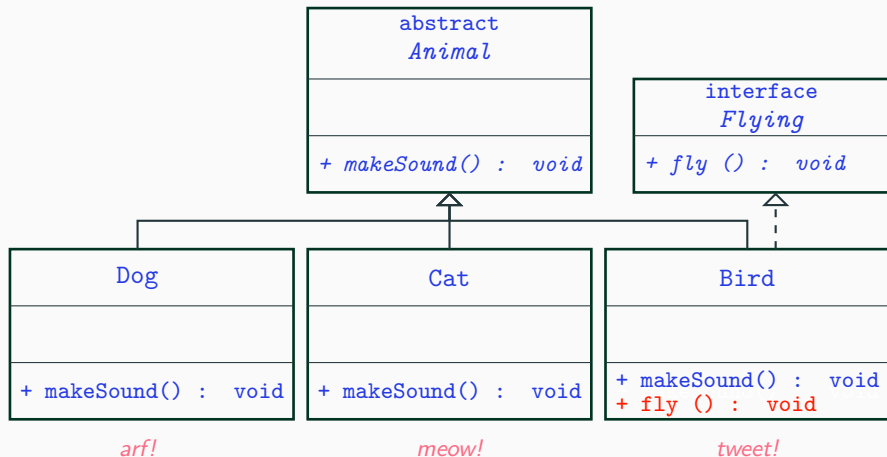
# Example

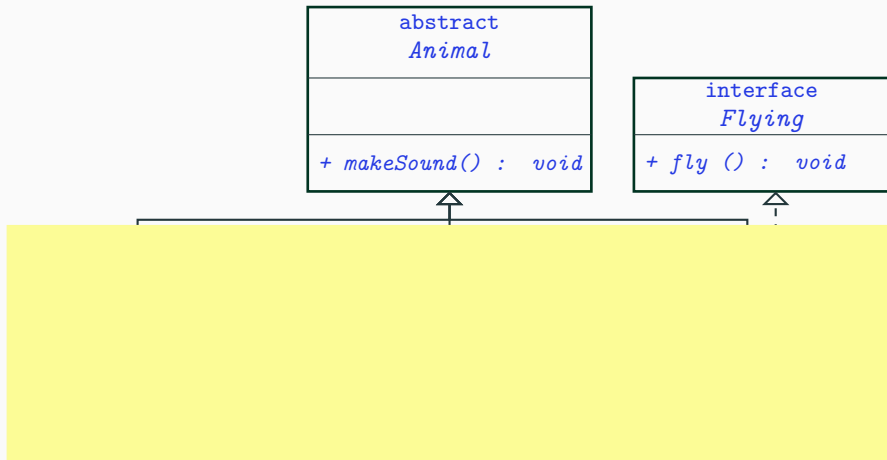


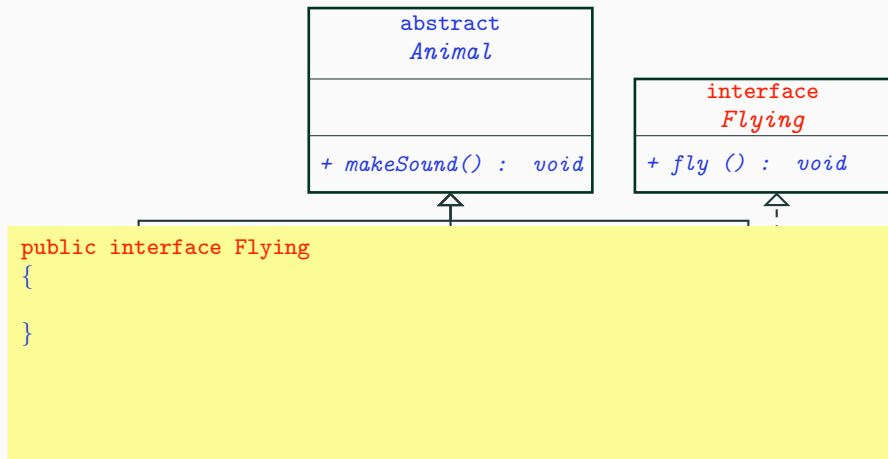
# Example



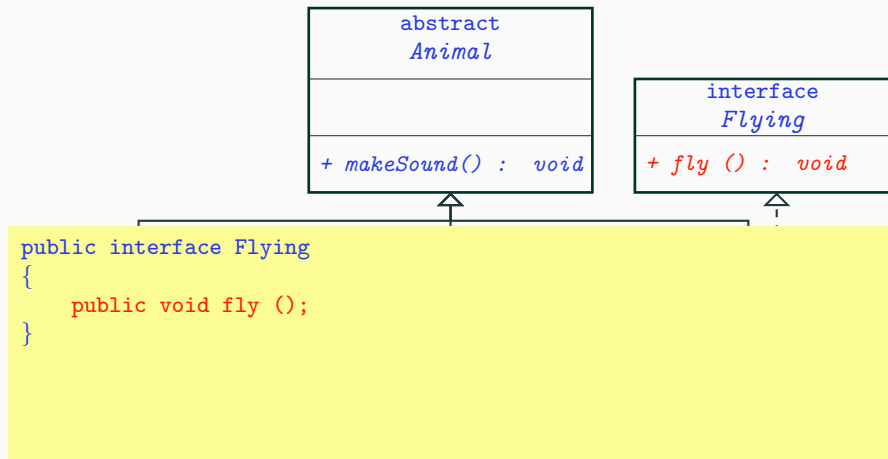
# Example



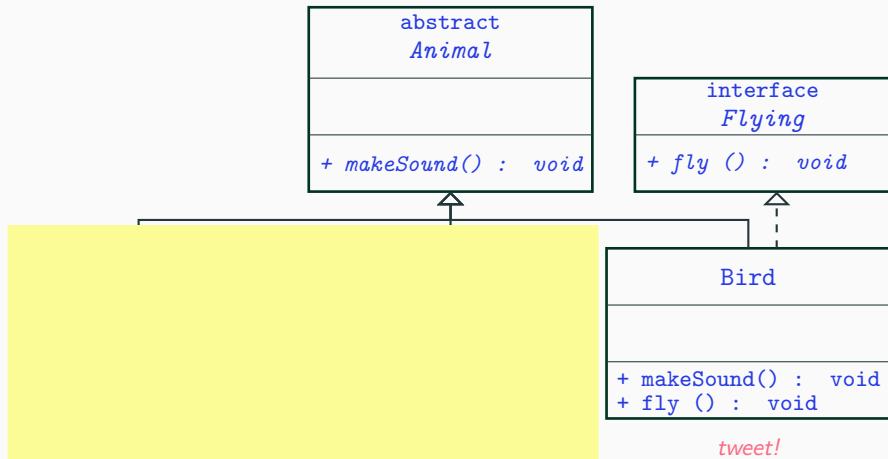




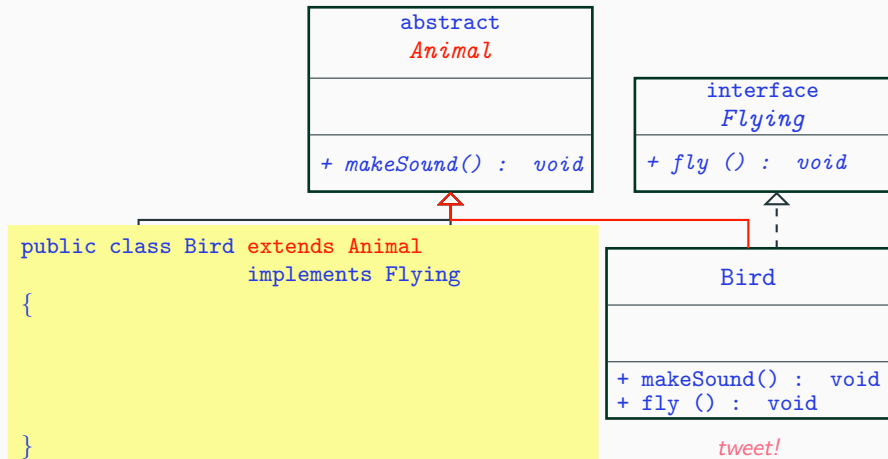


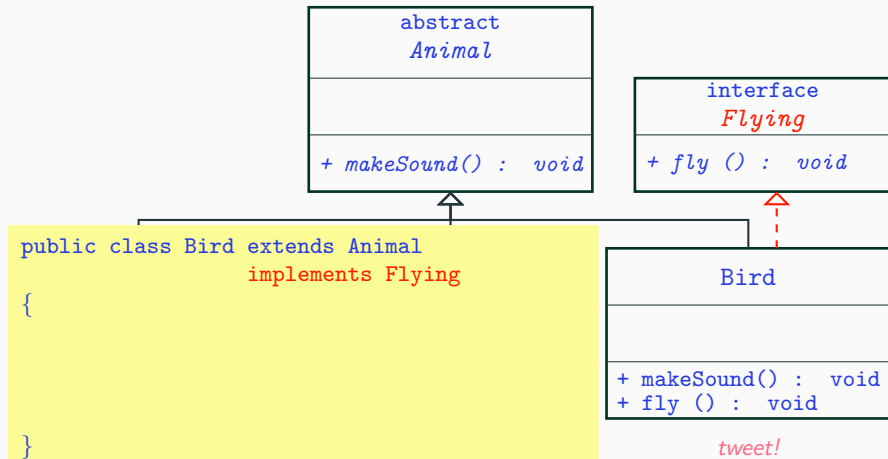


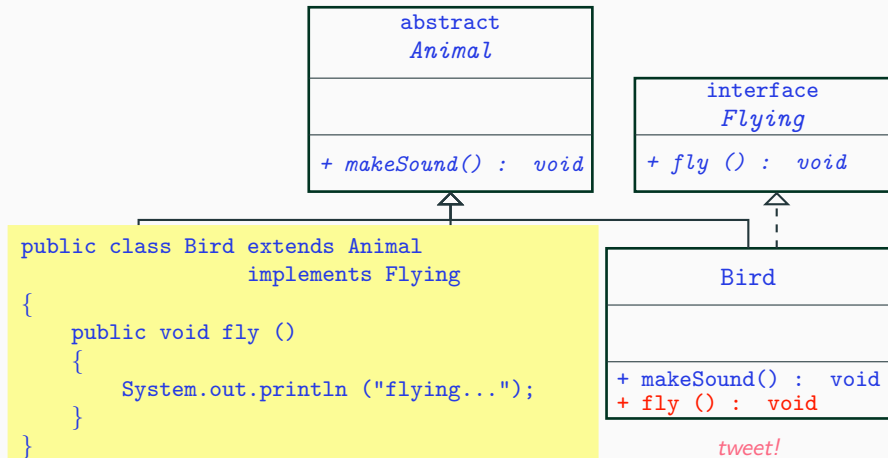
# Animals



# Animals







# Abstract class vs. Interface

Abstract Class	Interface
----------------	-----------

# Abstract class vs. Interface

Abstract Class	Interface
contains have 0 or more abstract methods, and 0 or more concrete methods	contains abstract methods only

# Abstract class vs. Interface

Abstract Class	Interface
contains have 0 or more abstract methods, and 0 or more concrete methods	contains abstract methods only
can have <code>static</code> , <code>final</code> , or <code>static final</code> attributes with any access specifier	can only have <code>public static final</code> attributes



# Abstract class vs. Interface

Abstract Class	Interface
contains have 0 or more abstract methods, and 0 or more concrete methods	contains abstract methods only
can have <code>static</code> , <code>final</code> , or <code>static final</code> attributes with any access specifier	can only have <code>public static final</code> attributes
the keyword <code>abstract</code> is mandatory to declare a method abstract	the keyword <code>abstract</code> is optional when declaring a method

# Abstract class vs. Interface

Abstract Class	Interface
contains have 0 or more abstract methods, and 0 or more concrete methods	contains abstract methods only
can have <code>static</code> , <code>final</code> , or <code>static final</code> attributes with any access specifier	can only have <code>public static final</code> attributes
the keyword <code>abstract</code> is mandatory to declare a method abstract	the keyword <code>abstract</code> is optional when declaring a method
can have <code>public</code> or <code>protected</code> abstract methods	the keyword <code>abstract</code> can only have <code>public</code> methods

# Abstract class vs. Interface

Abstract Class	Interface
can extend one (abstract or concrete) class at a time	can extend 1 or more interfaces

# Abstract class vs. Interface

Abstract Class	Interface
can extend one (abstract or concrete) class at a time	can extend 1 or more interfaces
can implement 0 or more interfaces	cannot implement

😊 Thank you! 😊