Web Application Development using Python

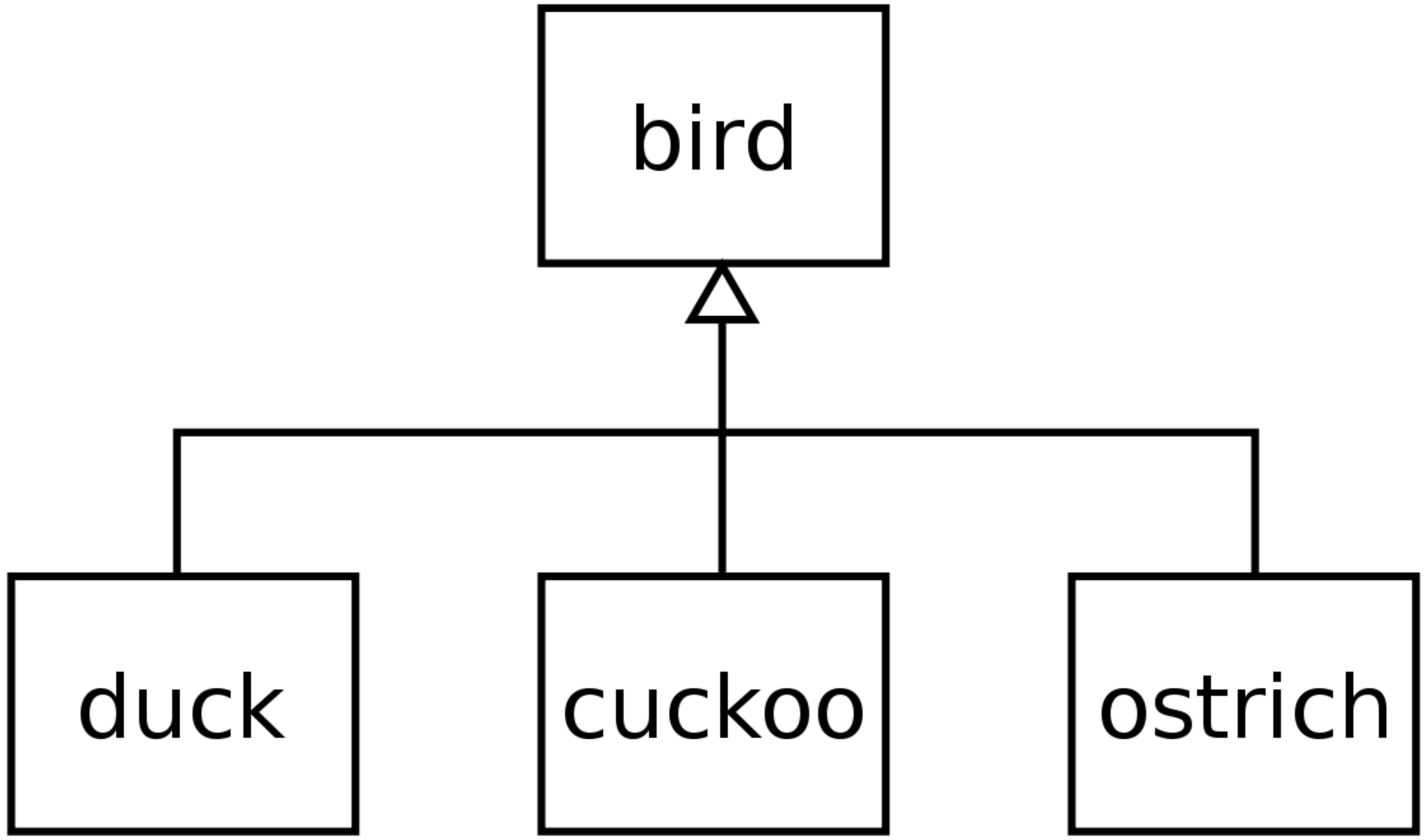
Introduction to Object Oriented Programming - Part 2

Prepared by George Khoury

Outline

- Inheritance
  - Inheriting from a class
  - Types of inheritance
- Polymorphism
- Method overriding
- Other important functions
  - `super()`
  - `instanceof()`
  - `issubclass()`

Inheritance



bird

duck

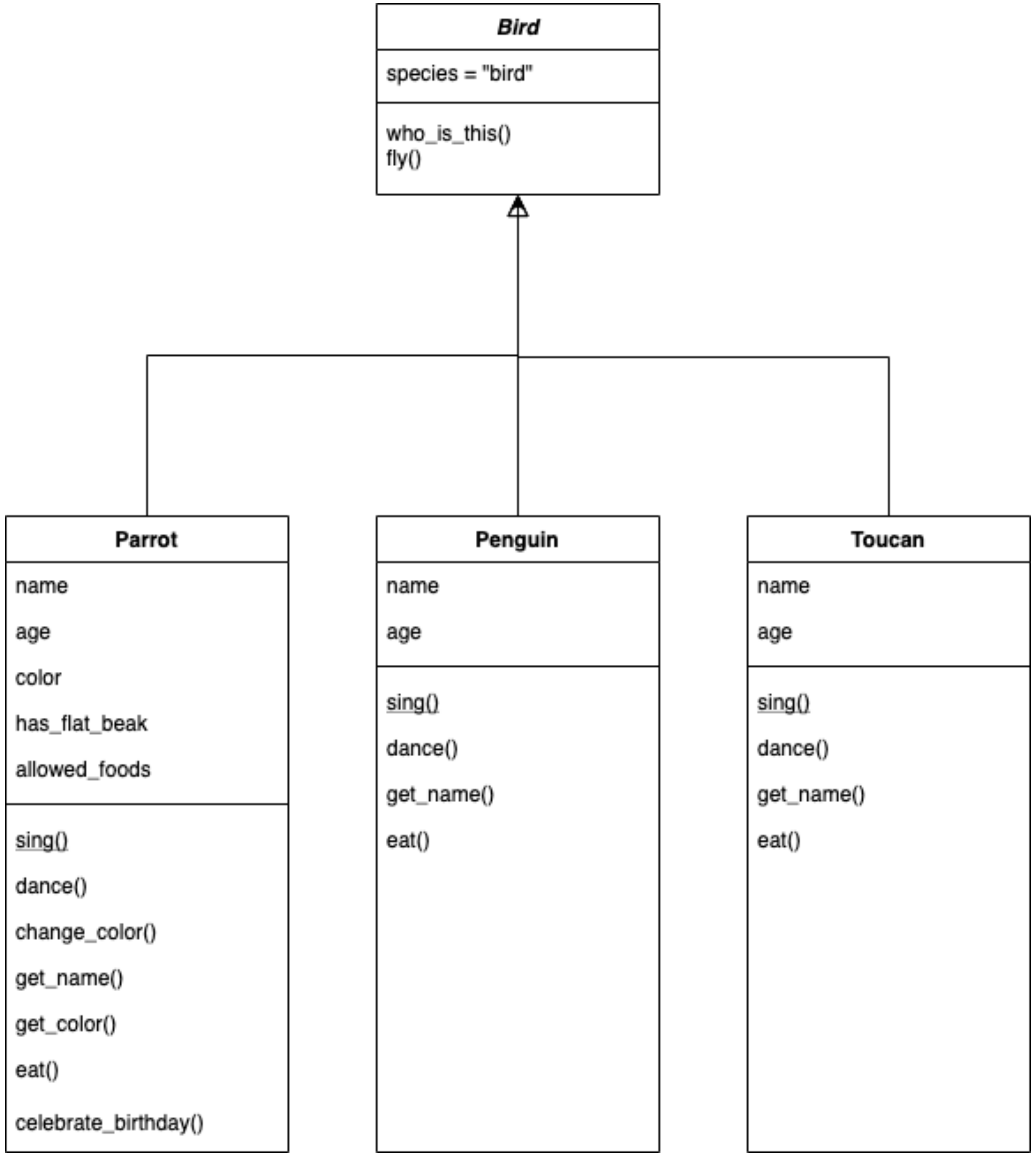
cuckoo

ostrich

What is inheritance?

W3/S3/Birds/

- One of the major advantages of Object Oriented Programming is re-useable code.
    - Inheritance is one of the mechanisms to achieve this.
- In inheritance, a class (usually called **superclass**) is inherited by another class (usually called **subclass**).
- The subclass adds some attributes to superclass.



Bird

species = "bird"

who_is_this()
fly()

Parrot

name
age
color
has_flat_beak
allowed_foods

sing()
dance()
change_color()
get_name()
get_color()
eat()
celebrate_birthday()

Penguin

name
age

sing()
dance()
get_name()
eat()

Toucan

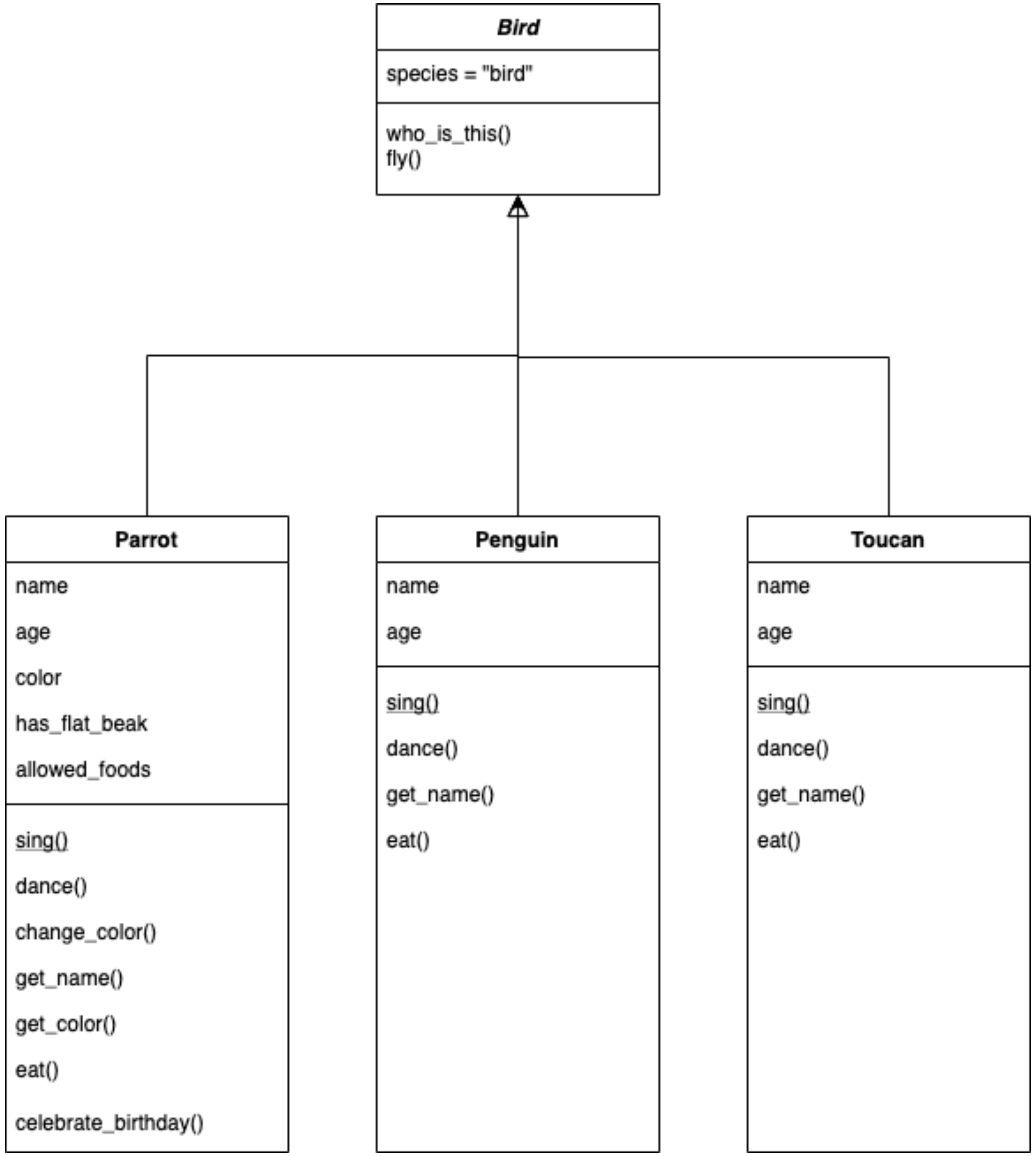
name
age

sing()
dance()
get_name()
eat()

Inheritance model

W3/S3/Birds/

- What is the base (or parent) class?
- Which attributes are inherited from the parent to each of the other child classes?
- Which methods are inherited from the parent to each of the other child classes?
- How can we use inheritance to enhance this class diagram?



Bird

species = "bird"

who_is_this()
fly()

Parrot

name

age

color

has_flat_beak

allowed_foods

sing()

dance()

change_color()

get_name()

get_color()

eat()

celebrate_birthday()

Penguin

name

age

sing()

dance()

get_name()

eat()

Toucan

name

age

sing()

dance()

get_name()

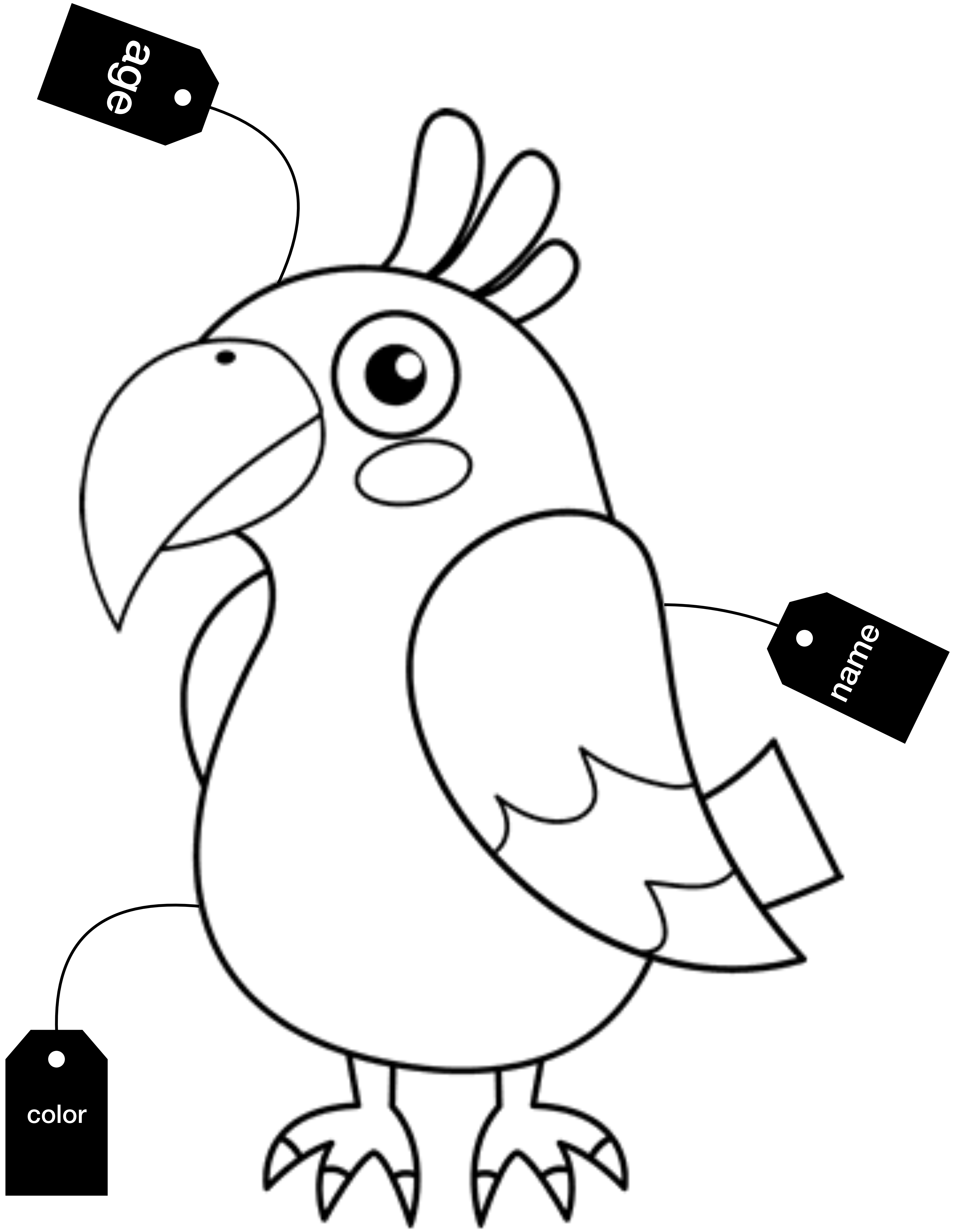
eat()

Inheriting from a class

Inheriting from a class

W3/S3/Birds/

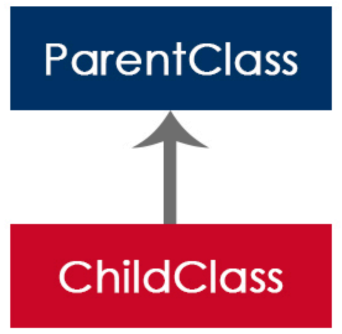
- A class can specify a parent class (or more) in parenthesis next to the class declaration.
- We must remember to initialise the parent class by calling the super init method.
  - We can use `super().__init__()`
  - We may need to pass parameters to the parent class.
- We may need to import our class from a different file.



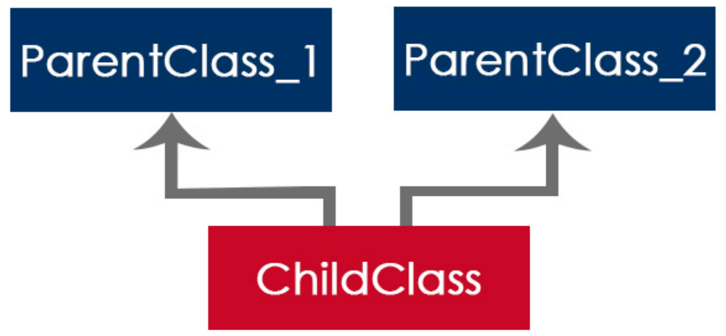
Types of inheritance

- Simple inheritance
- Multi-level inheritance
- Multiple inheritance
- Hierarchal inheritance

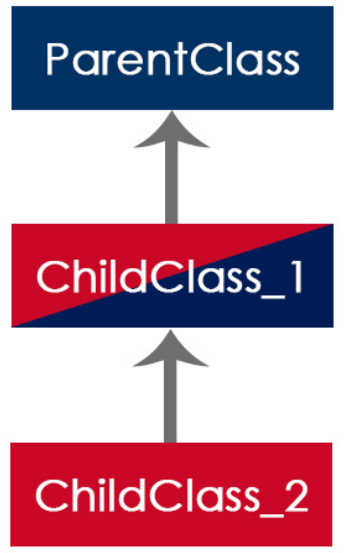
Simple Inheritance



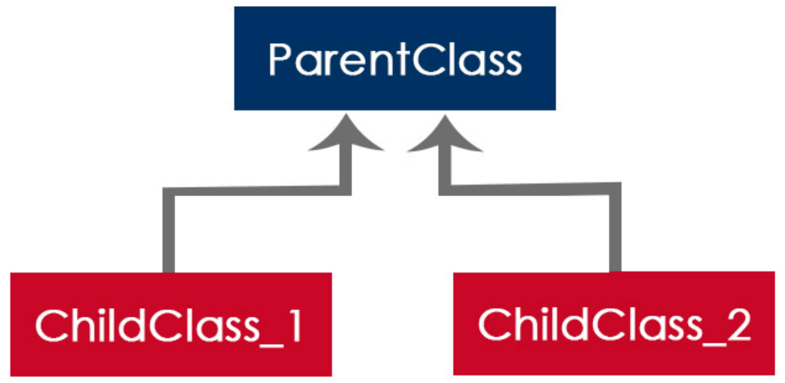
Multiple Inheritance



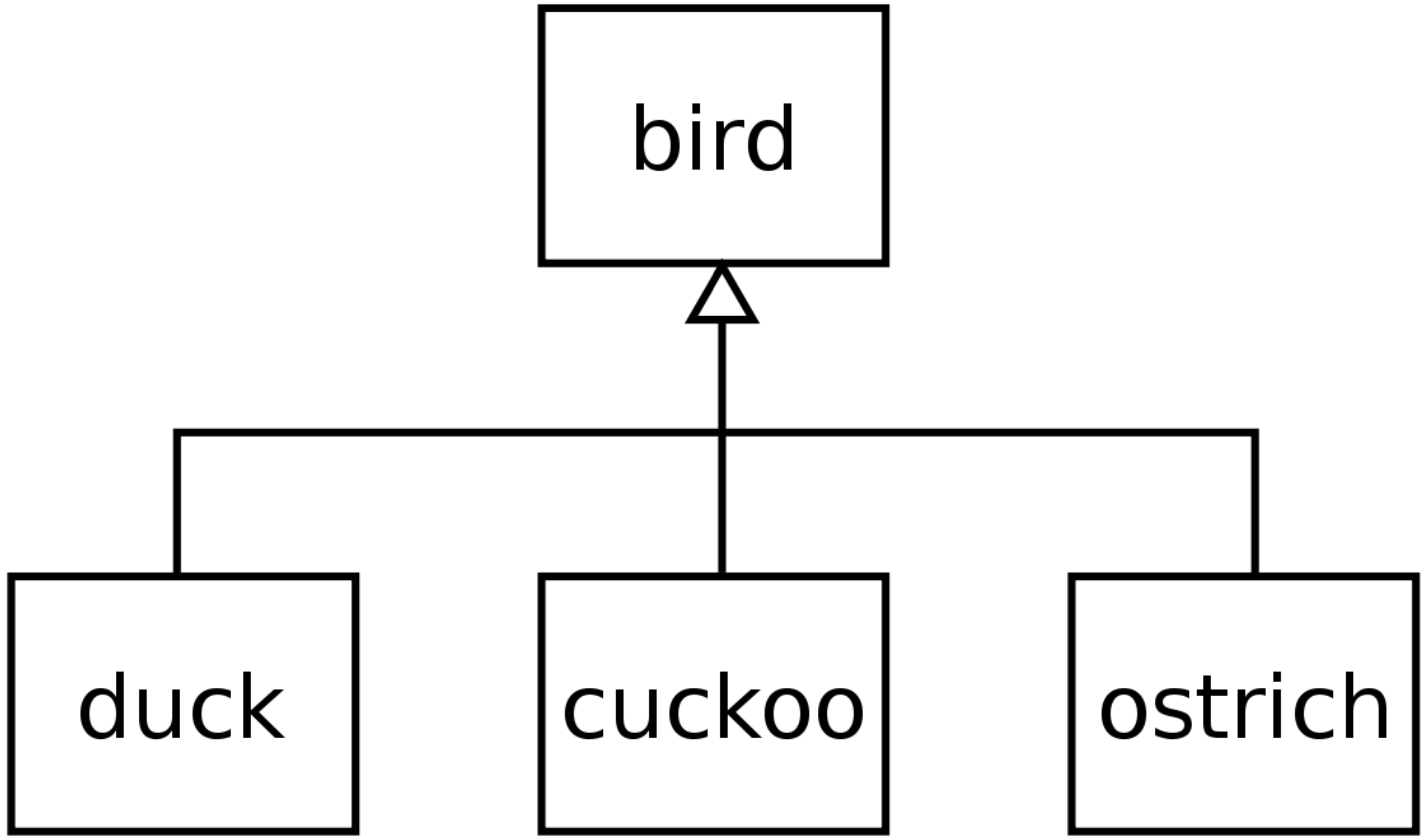
Multi Level Inheritance



Hierarchical Inheritance



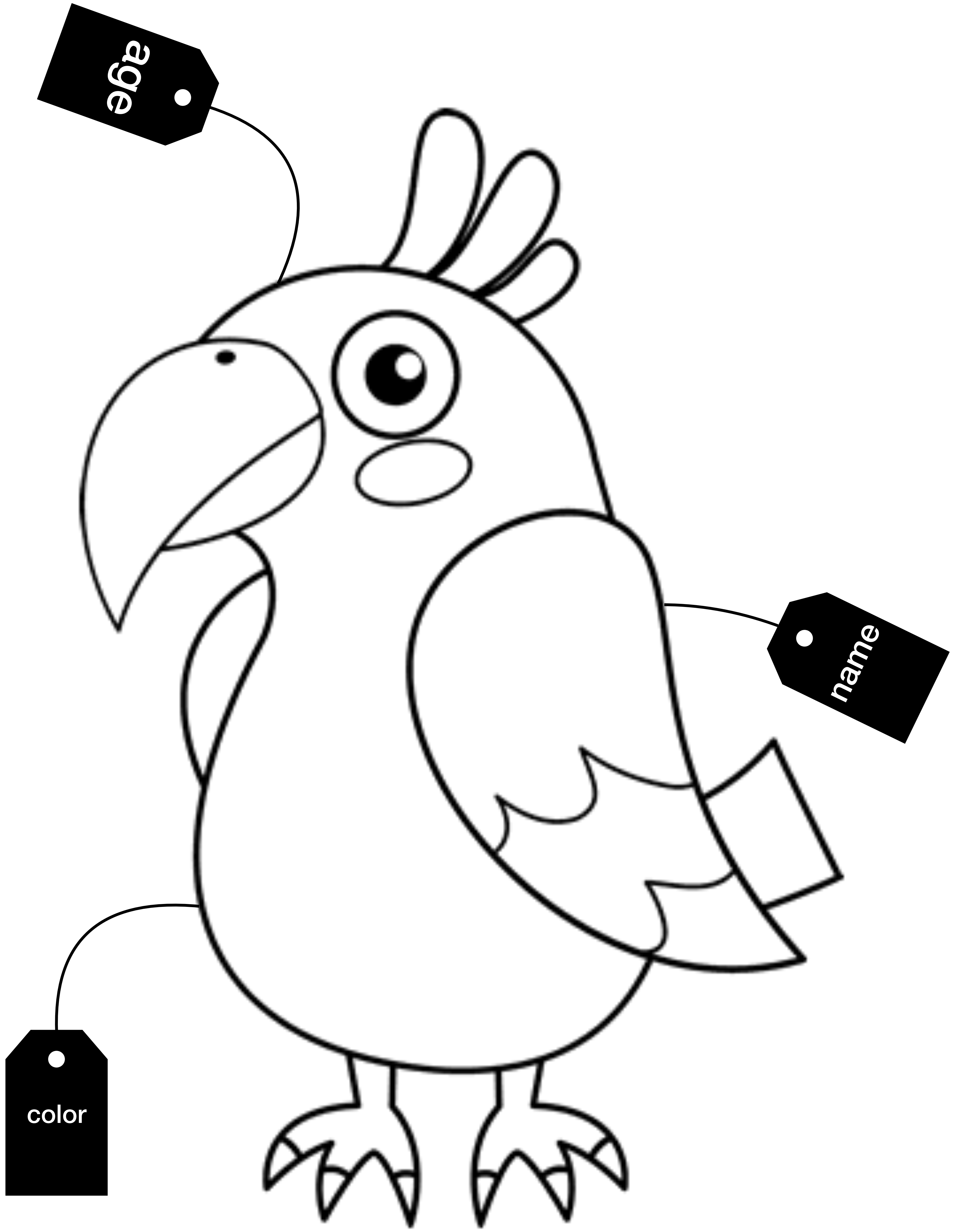
Polymorphism



Polymorphism

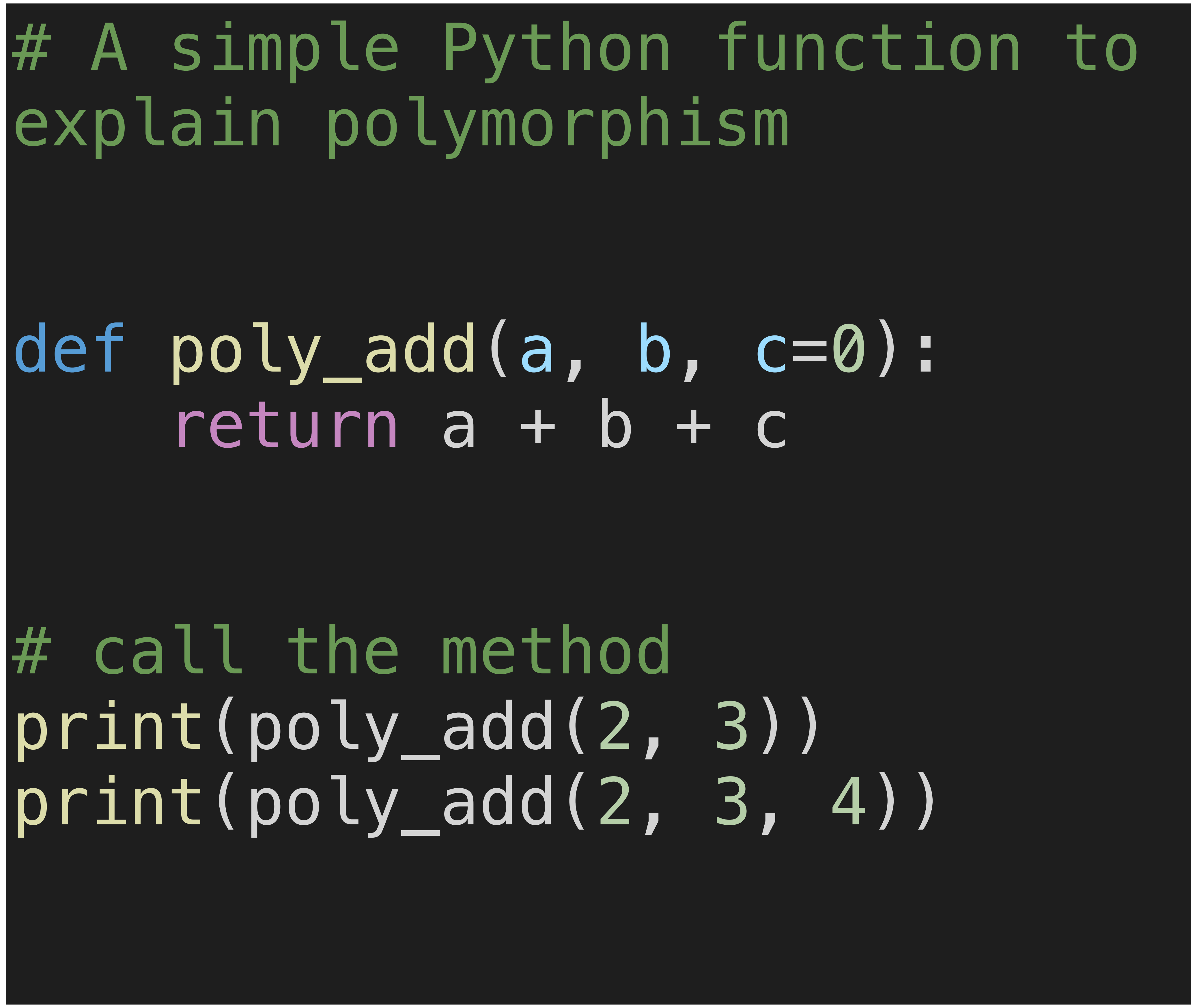
W3/S3/Birds/

- What is polymorphism?
    - The word polymorphism means having many forms.
- In programming, polymorphism means we can have the same function or method name but with different signatures.
- Built-in examples:
    - `len()`



Polymorphism

W3/S3/polymorphism.py



```
# A simple Python function to
explain polymorphism


def poly_add(a, b, c=0):
    return a + b + c


# call the method
print(poly_add(2, 3))
print(poly_add(2, 3, 4))
```

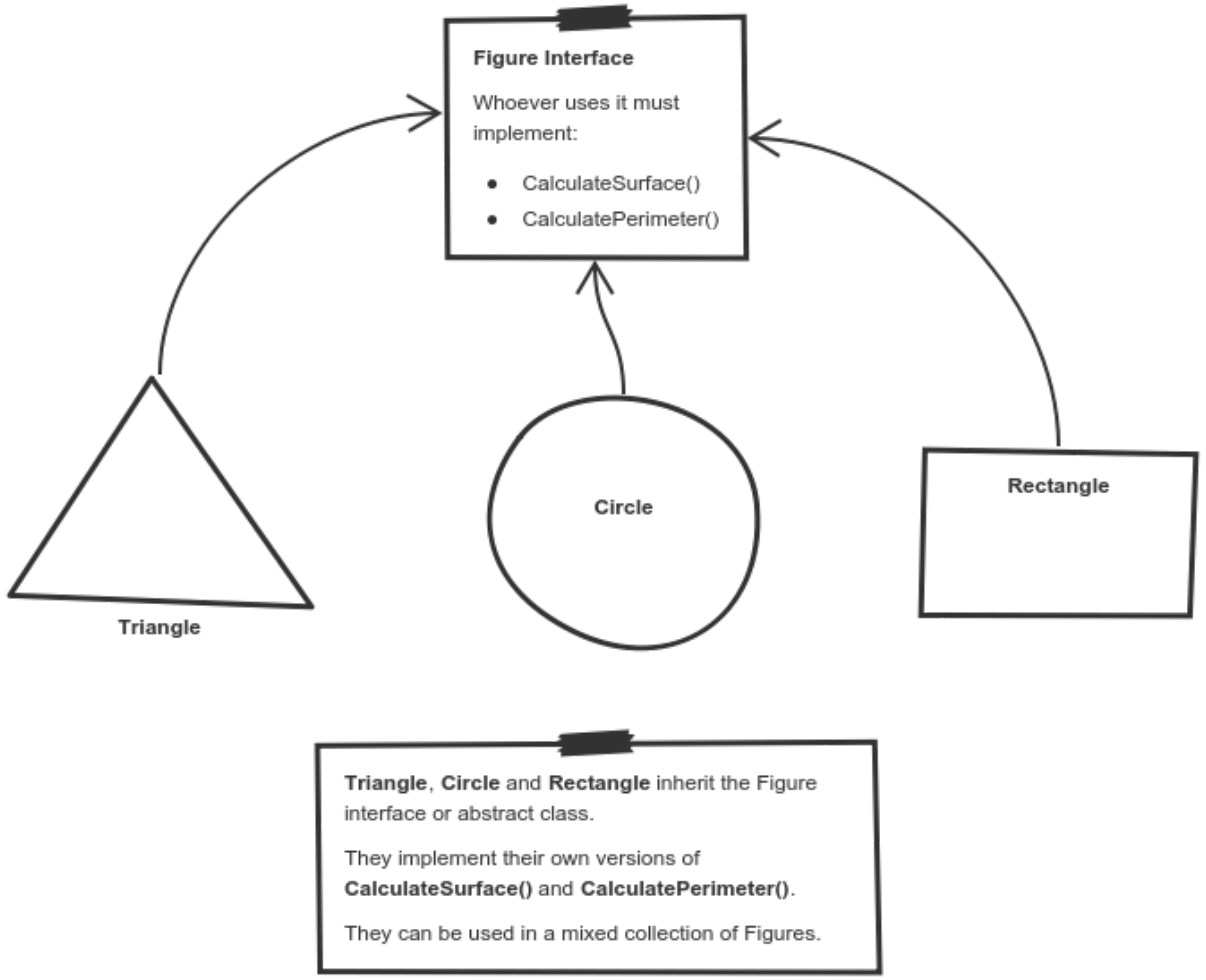


Figure Interface

Whoever uses it must implement:

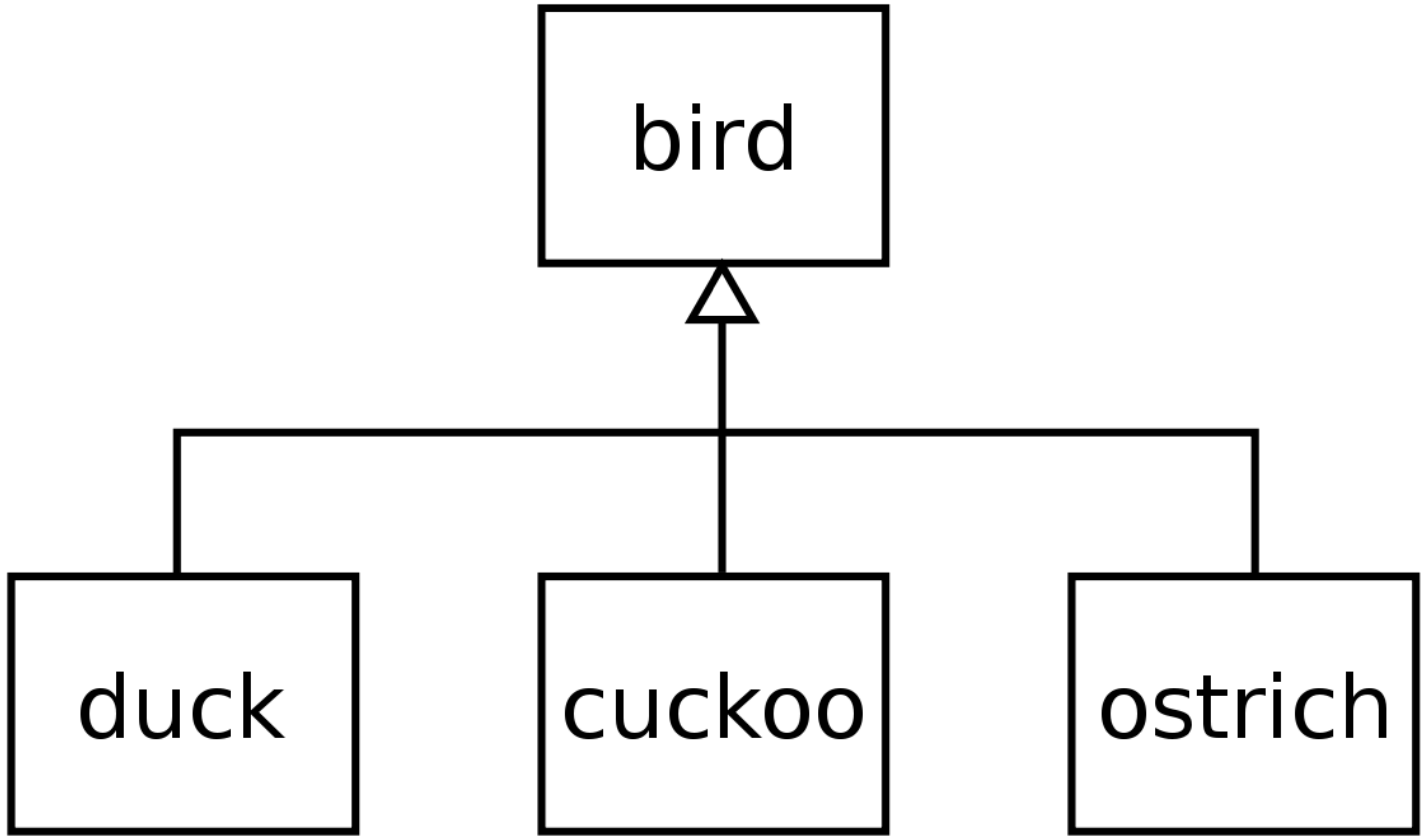
- CalculateSurface()
- CalculatePerimeter()

Triangle, **Circle** and **Rectangle** inherit the Figure interface or abstract class.

They implement their own versions of **CalculateSurface()** and **CalculatePerimeter()**.

They can be used in a mixed collection of Figures.

Method overriding



Method overriding

W3/S3/Birds/

- Method overriding is a concept of object oriented programming that allows us to change the implementation of a method in the **child class** that is defined in the **parent class**.
- To override a method we need,
  - Inheritance as we override the functionality of the parent class in the child class.
  - The overridden method to have the have the **same** method signature as in the parent class.
- Which methods can we override?
  - who_is_this()
  - fly()

Resources

- https://docs.python.org/3/library/abc.html
- https://www.python-course.eu/object_oriented_programming.php
- https://www.programiz.com/python-programming/multiple-inheritance
- https://www.w3schools.com/python/ref_func_isinstance.asp
- https://www.w3schools.com/python/ref_func_issubclass.asp