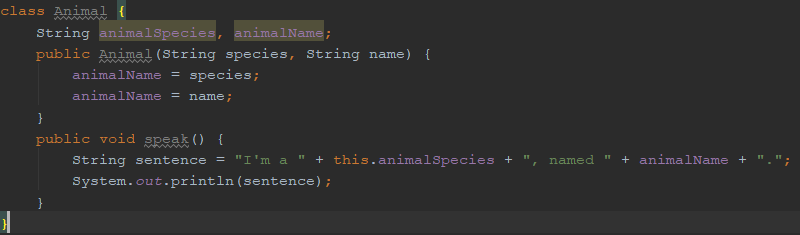
1. Object

An object is an instance of a class. Every object has its own data and has methods defined by its class.



1. Class

A class is like a blueprint to create objects. A class has constructors, fields and methods. Every object of that class will have the same methods, constructors and fields, but contain different data. A class can inherit properties of another class, or can be used as a parent to create another subclass.



1. Instantiation of object (creating an object)

Using the “new” keyword, a new instance of a class can be created with desired parameters.

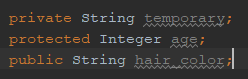


1. Visibility (public / private / protected)

Public: property/method can be accessed from anywhere

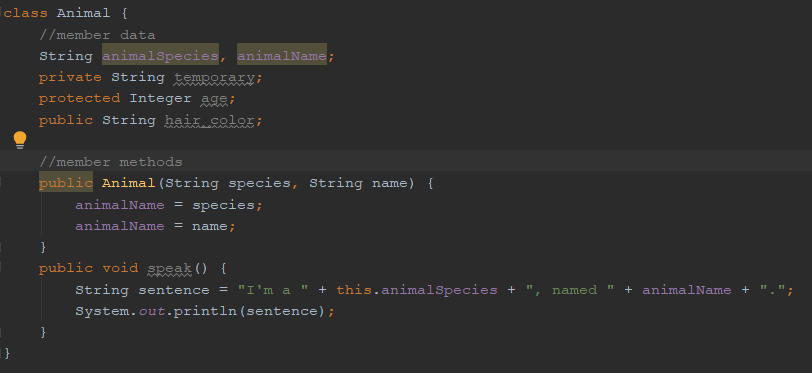
Private: property/method can only be accessed within the class it was declared

Protected: property/method can be accessed from subclasses



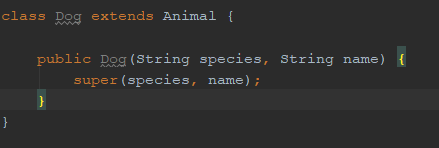
1. Member data/ methods

Member data and methods are variables and methods declared inside a class.



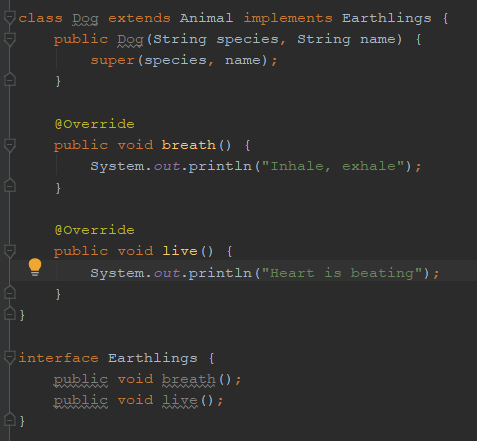
1. Inheritance

When a subclass is created from a parent class, it automatically has all the member data and methods that the parent class have.



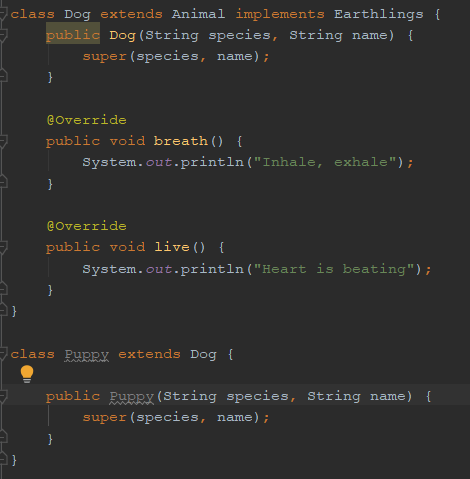
1. Interface

An interface is similar to an abstract class, it cannot be used to create objects. Interfaces are used to group related methods with empty bodies.



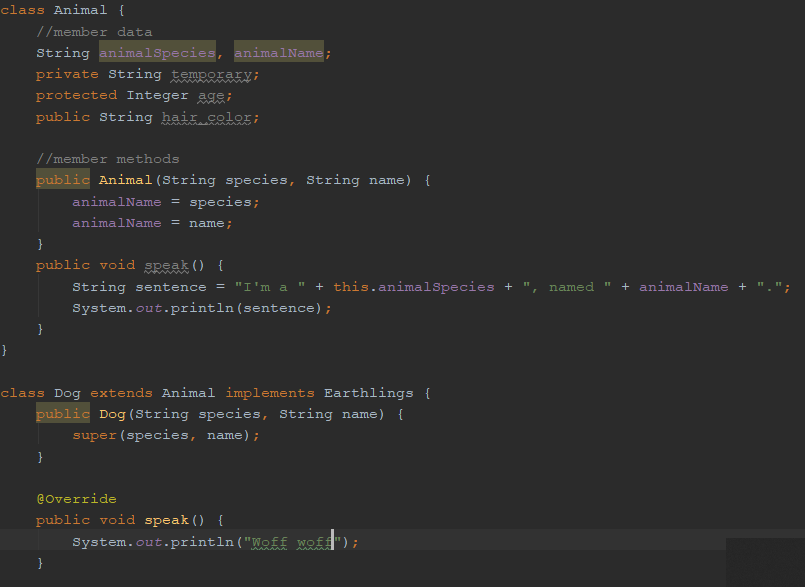
1. Polymorphism

Polymorphism means a class can be a parent class while being a subclass of another class.



1. Overriding

Overring is when a subclass changes a method inherited from its parent class so that method will work differently just for itself.



1. Abstract classes

Abstract classes are classed used solely as parent class and cannot be used to create objects.

