Inheritance: Hierarchies

CPSC 219: Introduction to Computer Science for Multidisciplinary Studies II
Fall 2023

Jonathan Hudson, Ph.D.
Instructor
Department of Computer Science
University of Calgary

Thursday, 20 October 2023

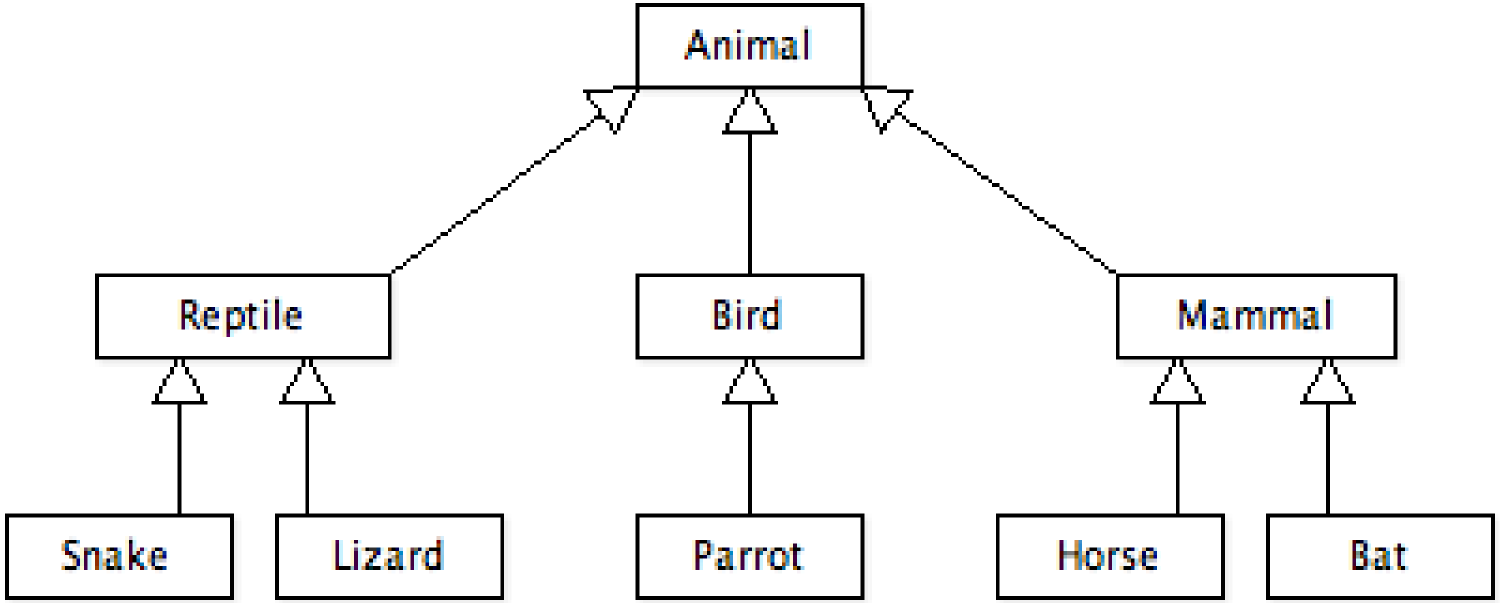
UNIVERSITY OF CALGARY

Outline

- Creating Subclasses
- Overriding Methods
- Class Hierarchies
- Designing for Inheritance

Class Hierarchies

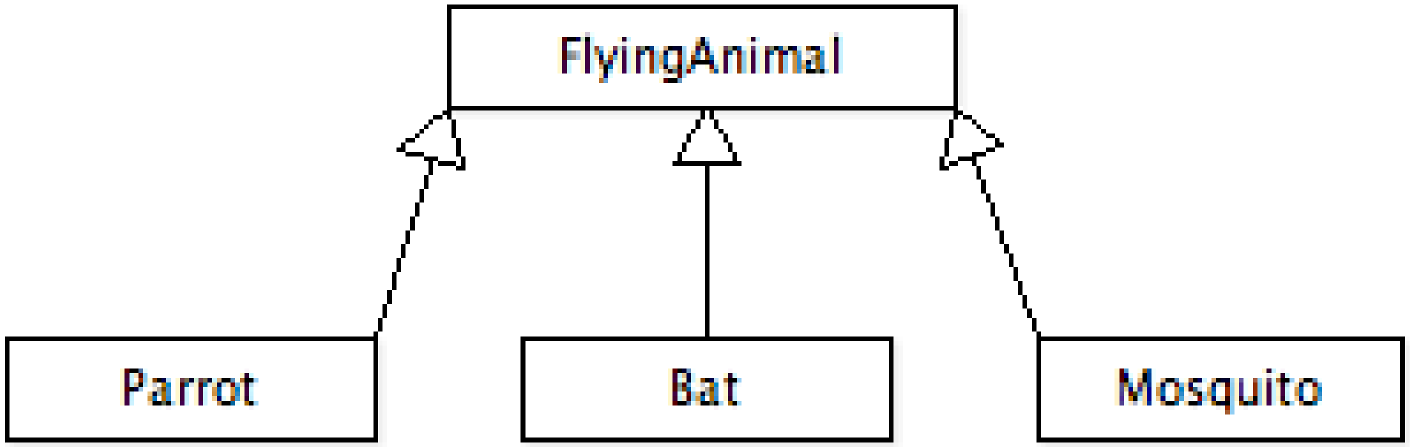
- A child class of one parent can be the parent of another child, forming a *class hierarchy*



Class Hierarchies

- Common features should be put as high in the hierarchy as is reasonable
- An inherited member is passed continually down the line
- i.e., a child class inherits from all its ancestor classes

An Alternate Class Hierarchy



- There is no single class hierarchy that is appropriate for all situations

The Object Class

- All classes are derived from the `Object` class (defined in `java.lang`)
  - the `Object` class is the ultimate ancestor of all classes
- recall `Object` has methods `toString()` and `equals()`
  - normally should override these when defining classes

Abstract Classes

- An abstract class is a placeholder in a class hierarchy that represents a generic concept
- An abstract class cannot be instantiated
- We use the modifier abstract on the class header to declare a class as abstract:

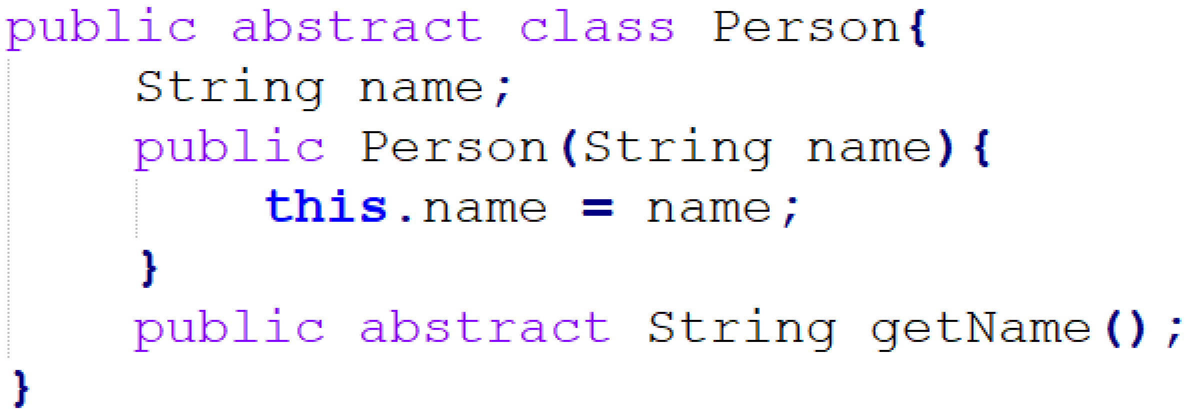
abstract

Prevent someone from instantiating the class itself

No Person exists without Staff, Student, Faculty role

We can make abstract methods as well and child class must create that method

If child class doesn’t implement abstract method it must be abstract as well



```
public abstract class Person{
    String name;
    public Person(String name){
        this.name = name;
    }
    public abstract String getName();
}
```

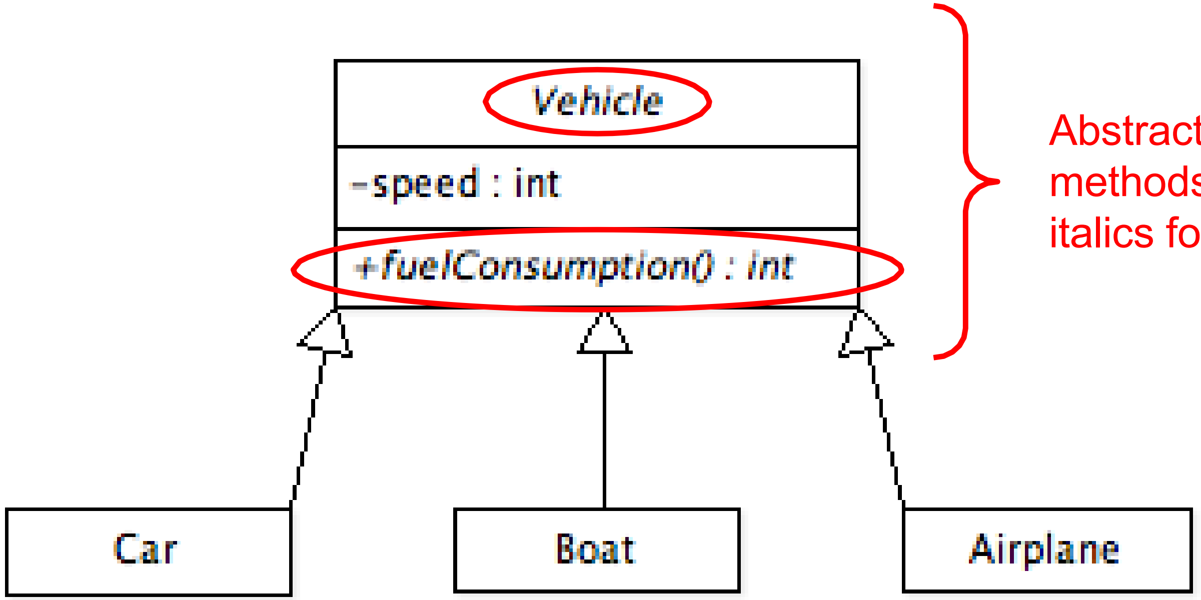
abstract

- An abstract class often contains abstract methods with no definitions (like an interface)
- Unlike an interface, the `abstract` modifier must be applied to each abstract method
- Also, an abstract class typically contains non-abstract methods with full definitions
- A class declared as abstract does not have to contain abstract methods – simply declaring it as abstract makes it so

abstract

- The child of an abstract class must override the abstract methods of the parent, or it too will be considered abstract
- An abstract method cannot be defined as `final` or `static`
- The use of abstract classes is an important element of software design – it allows us to establish common elements in a hierarchy that are too generic to instantiate

An Abstract Class in UML



Abstract classes and methods are shown in italics font.

- Car/Boat/Airplane MUST create a fuelConsumption() method for code to compile
- new Vehicle() cannot be used for code to compile

Onward to … Designing

Jonathan Hudson
jwhudson@ucalgary.ca
https://pages.cpsc.ucalgary.ca/~jwhudson/

UNIVERSITY OF CALGARY