

Java & OOP

9. Inheritance

Part 1. Basics

Inheritance

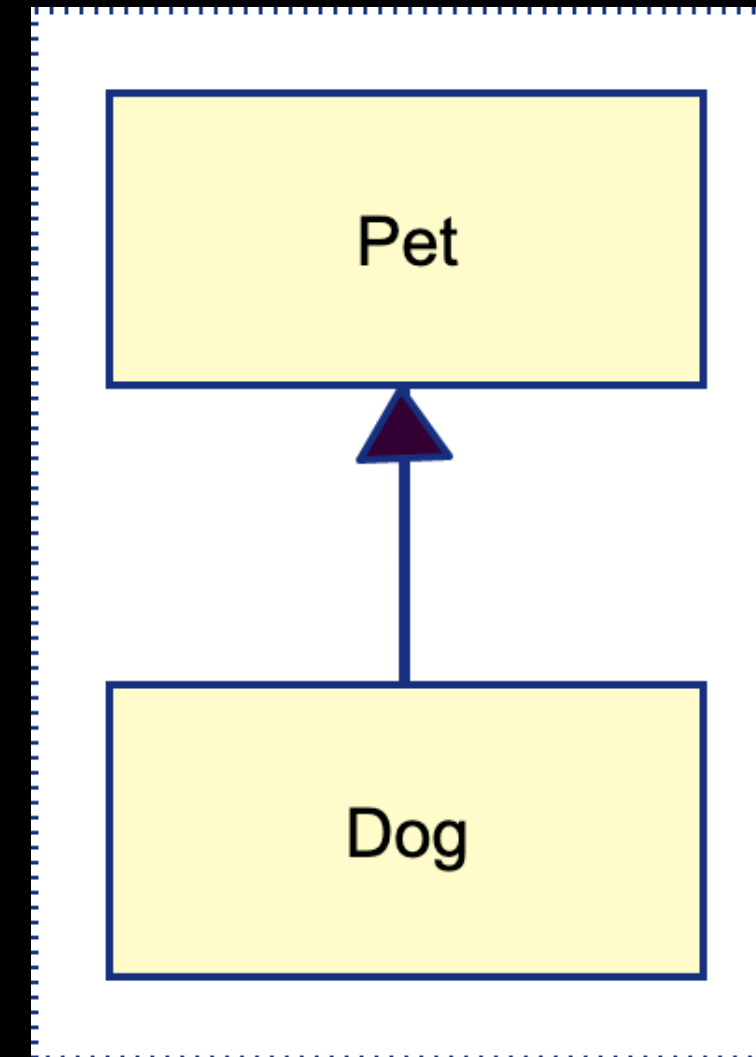
- Inherit wealth (property) and/or character (Behaviour) - real world
- Higher level of abstraction than classification
- Is-A relationship
 - Cat ***is a*** Pet
 - Dog ***is a*** Pet

Inheritance in Java

- A class inherits another class by **extending** behaviour
- Pet is the base class and Dog is the derived class

```
class Dog extends Pet {  
  
}
```

- A class can inherit from or extend **only one** class

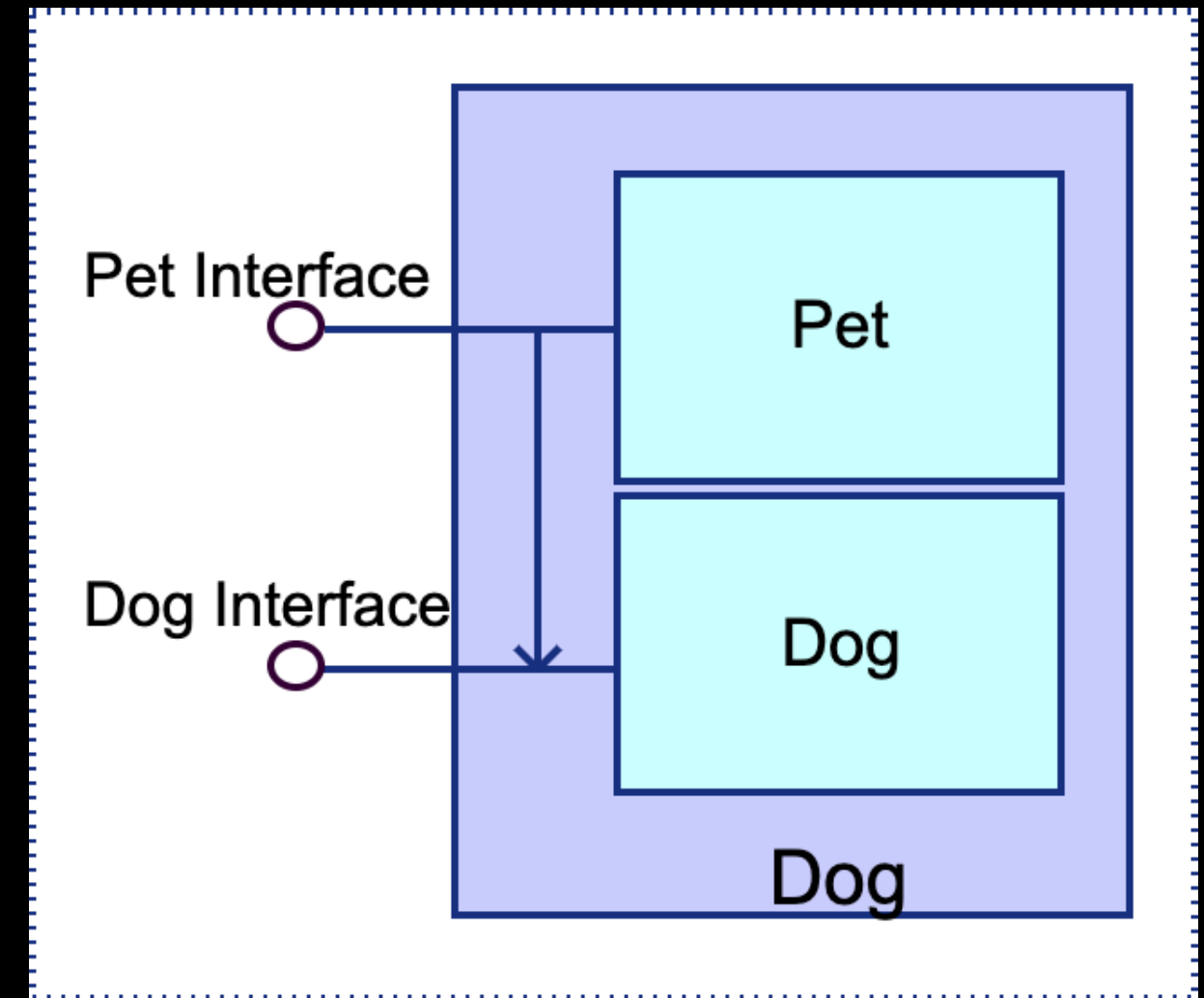


Inheritance in Java

- is-a relationship

```
Dog dog = new Dog();
```

```
Pet pet = new Dog();
```



Members in inheritance

- Only Object members will take part
- static members are never inherited - they are part of classes
- Access specifier is the deciding authority
- Derived class can hide the base class variable - not done usually

Members in inheritance

- Member functions are `virtual` in Java
- For function it is overriding - Base class interface exposes the overridden version
- Derived class constructor must access the base class constructor
- `super` is used to access the base class object from derived class

Points to remember

- `final` functions cannot be overridden
- abstract functions dont have body, must be overridden by derived class
- A class with atleast one abstract function is an abstract class
- An abstract class need not have any abstract methods
- abstract classes cannot be instantiated by others but derived classes

Some more points

- A `final` class cannot be inherited
- A class with inaccessible constructor is `final` - Singleton
- A `final` class cannot have an abstract method
- An abstract class can have `final` method(s)

Next

Using Inheritance

Code reuse myth buster