Inheritance

(OOP Core Concept 2 - Inheritance)

Learning objectives

- Inheritance
- The extends keyword
- Overriding methods
- Inheritance and constructors
- Inheritance and encapsulation

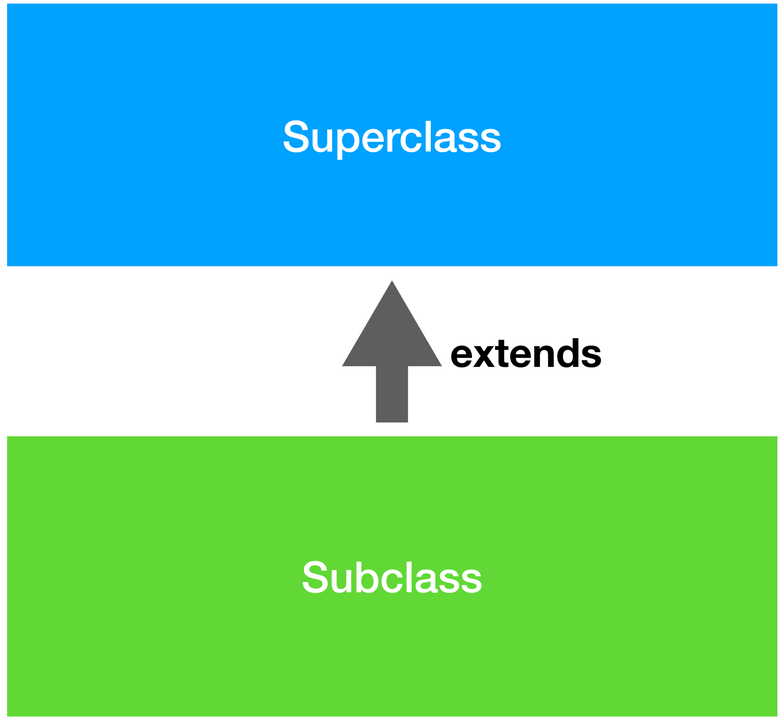
Inheritance

- Inheritance is an important part of OOP
- Inheritance supports the concept of reusability
- One class can inherit variables and methods from another class
- Inheritance builds on a is-a relationsship between subclass and superclass

Superclass and subclass

- A Superclass contains general functionality that is inherited by a subclass
- A Subclass extends the functionality of the Superclass
- Private variables and methods in the Superclass is not inherited by the Subclass
- The Subclass can override methods it inherit from the Superclass

Superclass and subclass



Code example

```
public class Animal {
   void eat() {
   }
}
```


```
public class Dog extends Animal {
   void wagTail() {
   }
}
```

Demo 1 - Inheritance

- Using the extends keyword
- Superclass and Subclass

Overriding methods

- Overriding methods means that the subclass has the same method as the superclass (same name and same input arguments)
- Then the method in the subclass replaces the method from the superclass that would otherwise have been inherited in the subclass
- The most specific method “wins”, and the further down the inheritance hierarchy the more specific it gets

Overriding methods

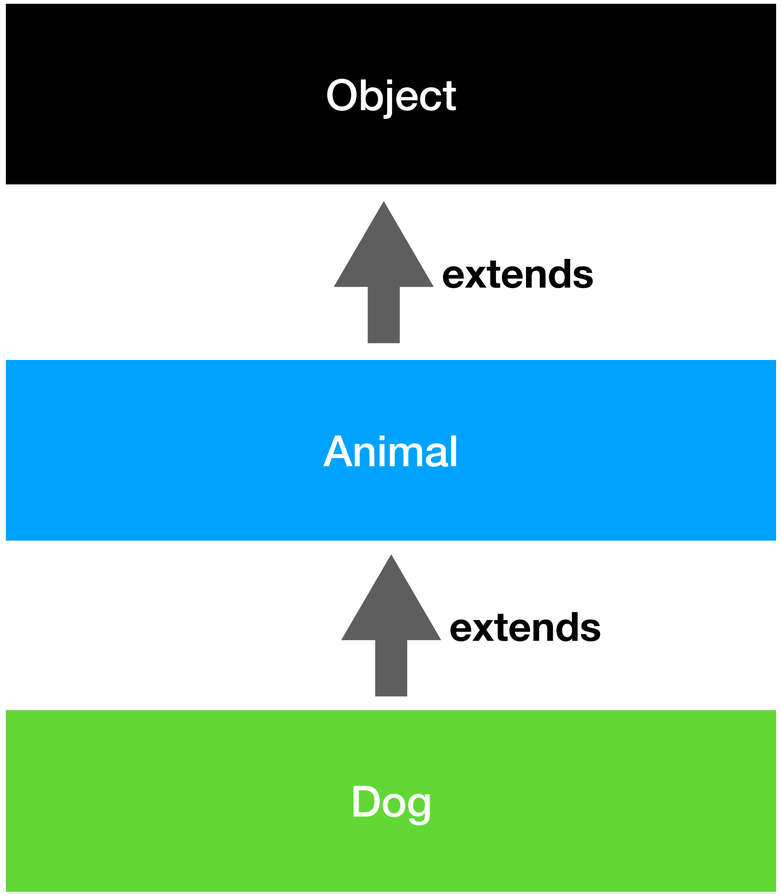
```
public class Animal {
   void eat() {
      System.out.println(”animal eats”);
   }
}
```


```
public class Dog extends Animal {
   @Override
   void eat() {
      System.out.println(”dog eats”);
   }
}
```

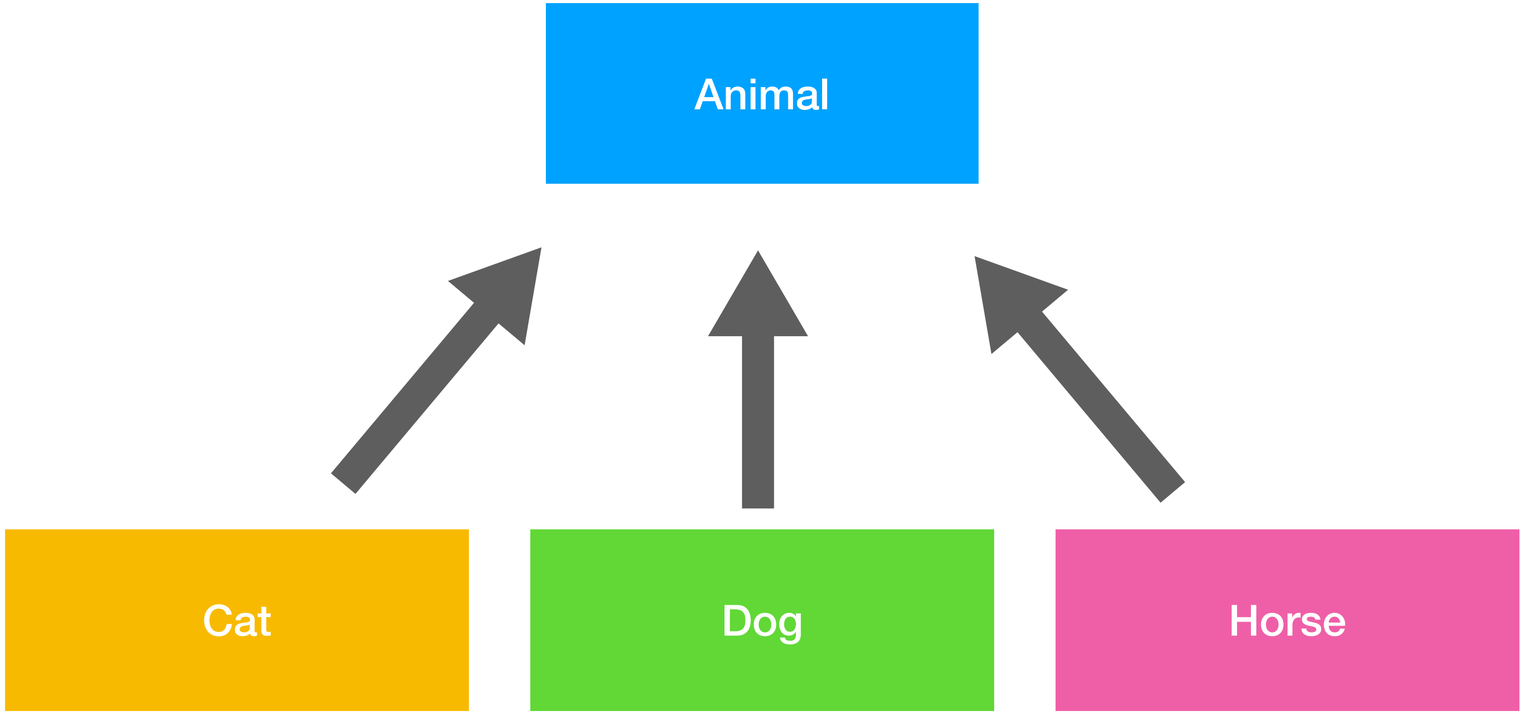
Demo 2 - Overriding methods

- Overriding methods

Object - Superclass of all classes



Object hierarchy



Demo 3 - Inheritance hierarchy

- Inheritance in many levels
- The Subclass points to its Superclass

Exercise 1 - Inheritance

- Reuse the solution from the previous exercise with the Book class
- You are selling Books, but also Movies and Video Games
- They are all Products and they all have a product id
- You can reuse the Book class, but also create classes for Movie and VideoGame
- Use inheritance and create a Product class with a productId, and let the Book, Movie and VideoGame inherit from Product

Inheritance and constructors

- Constructors in superclasses and subclasses must match in certain ways
- For example if the subclass has an empty constructor and the superclass has an empty constructor there is no problem
- But if the superclass does not have an empty constructor, then the subclass needs to call one of the constructors in the superclass from the empty constructor in the subclass

The super keyword

- The super keyword can be used to refer to the closest superclass
- It is similar to the this keyword, but always refer to the superclass of the object instead of the the object itself
- The super keyword can be used to call a constructor in the superclass from a constructor in the subclass
- If the super keyword is used in a constructor it must do so in the first line in the constructor

Demo 4 - Inheritance and constructors

- Constructors with different input arguments in superclass and subclass
- The super keyword

Inheritance and encapsulation

- Private variables and methods are not inherited from the superclass
- To provide an access modifier that means private for all external code except subclasses, use the protected access modifier

Access modifiers

Access modifier	Within Class	Within Package	Outside Package and in Subclass	Outside Package
Private	Y	N	N	N
Default	Y	Y	N	N
Protected	Y	Y	Y	N
Public	Y	Y	Y	Y

Demo 5 - Inheritance and encapsulation

- Private variables in the superclass is not inherited
- The protected access modifier

Exercise 2 - Inheritance

- You start selling ChildrensBooks, and they are exactly like normal Books but they also have a String variable called recommendedAgeInfo with information about the recommended age of readers

- Create a class for ChildrensBook and use inheritance to avoid repeating all the functionality in the ChildrensBook class

Learning objectives

- Inheritance
- The extends keyword
- Overriding methods
- Inheritance and constructors
- Inheritance and encapsulation