

Spring 2008 22C:021:A01/A03

Computer Science II : Data Structures

Instructor: Wenli He (wenli-he@uiowa.edu)

Supervisor: Jim Cremer (cremer@cs.uiowa.edu)

Department of Computer Science

University of Iowa

02/14/2008

Basic Object-Oriented Programming Concepts

- Interface
- Inheritance

Interface in the generic sense of the term

Real Life Example: TV



The buttons on the front of your television set, for example, are the interface between you and the electrical wiring on the other side of its plastic casing.

note: picture from www.umich.edu/~ktonez/pictures/Cartoon_TV.gif

Interface in Java Programming Language

- An abstract type
- Declared using **interface** keyword
- May only contain method signatures and constant declarations

(variable declarations which are declared to be both static and final)

- They cannot be directly instantiated

Object references in Java may be specified to be of an interface type

- either be null
- or be bound to an object which implements the interface

- The keyword **implements** is used to declare that a given class implements an interface.

A class which implements an interface must either implement all methods in the interface, or be an abstract class.

Interface example

```
interface Thing {
    public int age();
}

class Animal implements Thing {
    String name;
    int age;

    public Animal () {
        age = 0;
        System.err.println("An animal is born.");
    }

    public int age () {
        return age;
    }

    public void setName (String givenName) {
        name = givenName;
    }

    public String toString() {
        return "I'm an animal";
    }
}
```

Interface example (2)

Another class that implements **interface**:

```
class Publication implements Thing {
    String name;
    int age;
    String publisher;

    public Publication () {
        age = 0;
        System.err.println("I'm published.");
    }

    public int age () {
        return age;
    }

    public void setName (String givenName) {
        name = givenName;
    }

    public String toString() {
        return "I'm kind of publication.";
    }

    ....
}
```

Inheritance

- Different kinds of objects often have a certain amount in common with each other.

Dog, cat, rat, for example, all share the characteristics of animals.

- Java allows classes to inherit commonly used state and behavior from other classes.

In this example, Animal now becomes the superclass of Dog, and Cat.

- One-to-many relationship

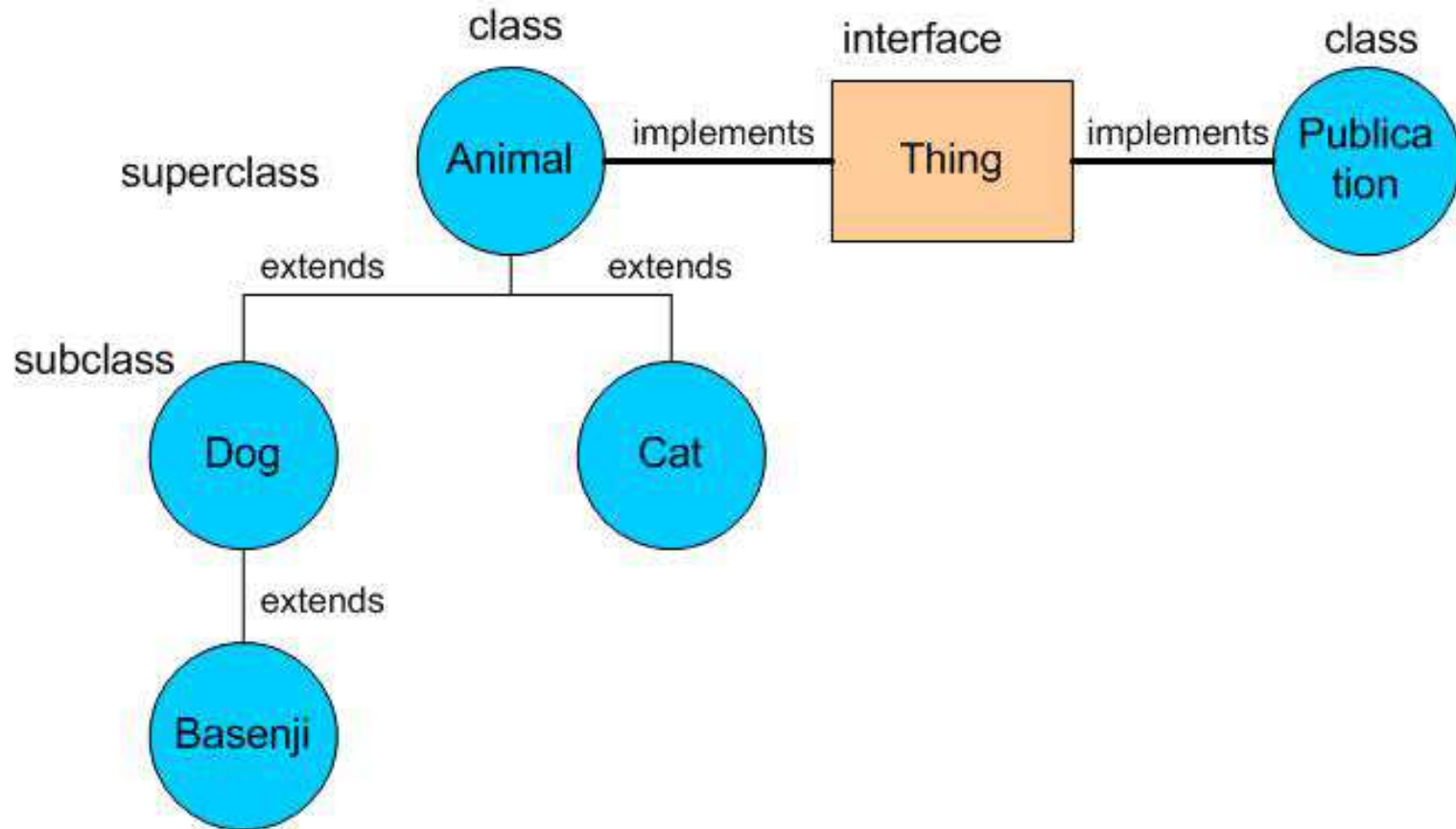
In the Java programming language, each class is allowed to have one direct superclass, and each superclass has the potential for an unlimited number of subclasses.

- Example Source code on course Website:

<http://www.cs.uiowa.edu/~cremer/courses/cs21-s08/Animals.java>

Inheritance and Interface Example

A hierarchy of animal classes:



Why do we need Interface?

- Interfaces are used to collect similarities which classes of various types share.
- simulate multiple inheritance

All classes in Java (other than `java.lang.Object`, the root class of the Java type system) must have exactly one base class; multiple inheritance of classes is not allowed. However, a Java class may implement any number of interfaces.

Demonstration of the Usage

- Instantiation
- Assignment