Inheritance

2 – 4 pm Tuesday 7/22/2008 @JD2211

1

Agenda

- Review
  - Relationships
- Inheritance

2

Review

- Association
- Aggregation
- Composition
- 7-3-Lab

3

Inheritance

- Inheritance is the process of deriving a new class from an exiting one.
- One purpose of inheritance is to reuse existing software
- Inheritance is the is-a relationship.

4

Inheritance (cont'd)

- The original class that is used to derive a new one is called *parent* class, *super* class, or *base* class.
- The *derived* class is called a *child* class, or sub class.

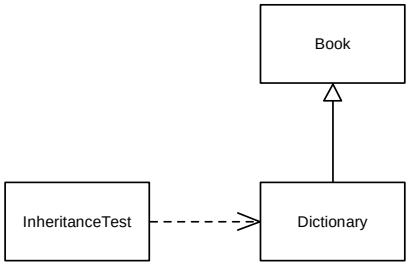
5

Inheritance - Java

- Three reserved words
  - `extends`
  - `protected`
  - `super`

6

Inheritance - UML



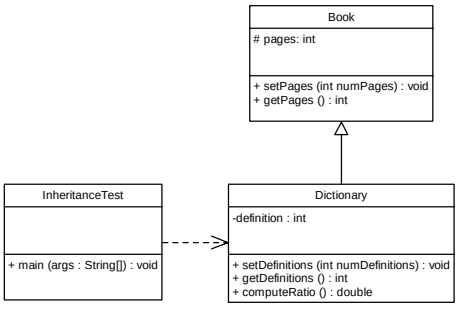
7

Inheritance – Java Code

```
Public class InheritanceTest {
       public static void main (String [] args)

       Dictionary webster = new Dictionary();
       ..........
}


public class Book {
..........
}


public class Dictionary extends Book {
..........
}
```

8

Inheritance - UML



9

```
public class Book {
   protected int pages = 1500;

   public void setPages (int numPages) {
      pages = numPages;
   }

   public int getPages (){
     return pages;
   }
}
```

10

```
public class Dictionary extends Book {
  private int definitions = 52500;

   public void setDefinitions (int numDefinitions){
      definitions = numDefinitions;
   }

   public int getDefinitions(){
      return definitions;
  }

   public double computeRatio(){
      return definitions/pages;
   }
}
```

11

```
public class InheritanceTest {
   public static void main (String[] args){
      Dictionary webster = new Dictionary ();

      System.out.println ("Number of pages: " +
                          webster.getPages());
      System.out.println ("Number of definitions: " +
                          webster.getDefinitions());
      System.out.println ("Definitions per page: " +
                          webster.computeRatio());
   }
}
```

12

The `protected` Visibility Modifier

- Protected visibility provides the best possible encapsulation that permits inheritance

13

The `super` Reference

- All methods and variables, even those declared with private visibility, are inherited by the child class. Constructors, however, are NOT inherited.
- The reserved word `super` can be used in a class to refer to its parent class. Using the super reference, we can access a parent's members.

14

```
public class Book2 {
   protected int pages;

   public Book2 (int numPages) {
      pages = numPages;
   }

   public void setPages (int numPages) {
     pages = numPages;
   }

   public int getPages (){
      return pages;
   }
}
```

15

```
public class Dictionary2 extends Book2 {
   private int definitions;

   public Dictionary2 (int numPages, int numDefinitions) {
      super (numPages);
     definitions = numDefinitions;
   }

   public void setDefinitions (int numDefinitions){
      definitions = numDefinitions;
   }

   public int getDefinitions(){
      return definitions;
   }

   public double computeRatio(){
      return definitions/pages;
   }
}
```

16

```
public class InheritanceTest2 {
   public static void main (String[] args){
      Dictionary2 webster = new Dictionary2 (1500, 52500);

      System.out.println ("Number of pages: " +
                  webster.getPages());
      System.out.println ("Number of definitions: " +
                  webster.getDefinitions());
      System.out.println ("Definitions per page: " +
                  webster.computeRatio());
   }
}
```

17

Lab Assignment

- http://www.csun.edu/~twang/Java/Lab/7-22-Lab.pdf

18