8 - 9

Classes and Objects: A Deeper Look (Part III)

Last time

- inheritance and composition examples
- keyword `this` and `super`
- `protected` keyword
- `@Override` annotation
- `java.lang.Object` class
- Examples: Time and Clock classes and inheritance

Objectives

- composition review
- inheritance review
- intro to polymorphism and abstract classes

OOP Concepts

- **Composition** (compose using existing classes)
- **Inheritance** (inherit from existing classes)

Inheritance

- **Subclass extends superclass**
  - **Subclass**
    - **More specialized group of objects**
    - **Behaviors inherited from superclass (can customize)**
    - **Additional behaviors**
- **Sometimes referred to as a *is-a* relationship**

Superclass	Subclasses
Student	GraduateStudent, UndergraduateStudent
Shape	Circle, Triangle, Rectangle
Loan	CarLoan, HomeImprovementLoan, MortgageLoan
Employee	Faculty, Staff
BankAccount	CheckingAccount, SavingsAccount

Inheritance examples.

Inheritance example

- MyPanel class extends `JPanel`

```
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JPanel;

// myPanel class that extends JPanel and setups up painting canvas
public class MyPanel extends JPanel {

    @Override
    public void paintComponent( Graphics g ) {
        super.paintComponent(g);

        g.setColor(Color.BLUE);
        g.fillRect(0, 0, 100, 100);
    }
}
```

Class hierarchy

- **Direct superclass**
  - **Inherited explicitly (one level up hierarchy)**
- **Indirect superclass**
  - **Inherited two or more levels up hierarchy**
- **Single inheritance**
  - **Inherits from only one superclass**
  - **Java only allows single inheritance (no multiple parents)**

Inheritance hierarchy

- **Inheritance relationships: tree-like structure**

- **Each class becomes**
  - **Superclass: supply members to other classes**

  OR
  - **Subclass: inherit members from other classes**

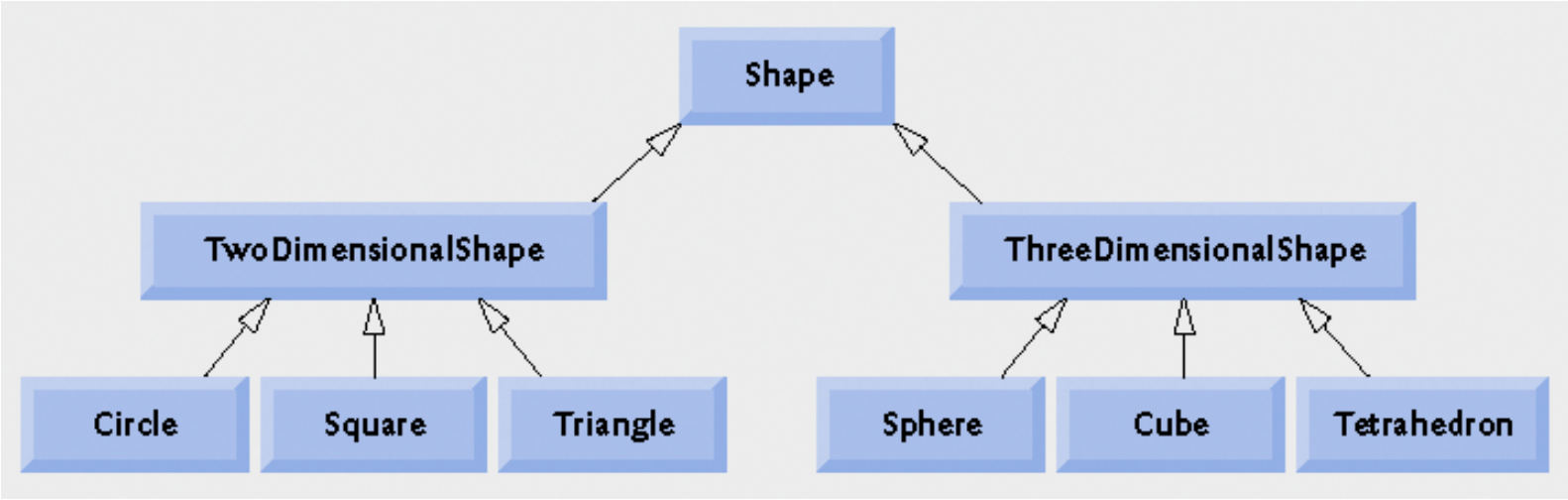

Fig. 9.3 | Inheritance hierarchy for Shapes.

Example: Instrument class

- Instrument class with `play()` method
  - `Guitar`
  - `Piano`
  - `Drums`

- Create array of Instrument objects
  - Iterate over objects and invoke `play()` method

Polymorphism: Upcasting

- **A superclass reference can store a subclass object**
  - **Possible since a subclass object *is a* superclass object as well**
- **This is often called upcasting**

Polymorphism: Downcasting

- **On the other hand, a subclass reference can be aimed at a superclass object only if the object is downcasted**

- **Usually this is not safe (why?)**

Abstract Classes

- **Abstract classes**
  - **Classes that are too general to create real objects**
  - **Used only as abstract superclasses for concrete subclasses and to declare reference variables**
  - **Many inheritance hierarchies have abstract superclasses occupying the top few levels**

Abstract Classes and Methods

- Keyword `abstract`
  - Use to declare a class `abstract`
  - Also use to declare a method `abstract`
    - Abstract classes normally contain one or more abstract methods
    - All *concrete* subclasses must override all inherited abstract methods

protected Members

- protected access
  - Intermediate level of access protection between public and private
  - protected members accessible by
    - superclass members
    - subclass members
    - Class members in the same package

Composition

- **Composition**
  - **A class can have references to objects of other classes as members**
  - **Sometimes referred to as a *has-a* relationship**

Composition example

- **DialPadPanel class assembled using composition**

```
import javax.swing.JPanel;
import javax.swing.JButton;
import javax.swing.JLabel;
import java.awt.GridLayout;

public class DialPadPanel extends JPanel {

    public DialPadPanel() {

        setLayout(new GridLayout(5,3));

        add(new JLabel());
        add(new JLabel("Dial me!", JLabel.CENTER));
        add(new JLabel());

        add(new JButton("*"));
        ...
    }
```

GUI: Displaying Text and Images Using Labels

- **Labels**
  - **Display information and instructions**
  - `JLabel`
    - **Display a single line of text**
    - **Display an image**
    - **Display both text and image**