10 Object-Oriented Programming: Polymorphism and Interfaces

Objectives

In this chapter you’ll:

- Learn the concept of polymorphism and how it enables “programming in the general.”
- Use overridden methods to effect polymorphism.
- Distinguish between abstract and concrete classes.
- Declare abstract methods to create abstract classes.
- Learn how polymorphism makes systems extensible and maintainable.
- Determine an object’s type at execution time.
- Declare and implement interfaces, and become familiar with the Java SE 8 interface enhancements.

Outline

1. 10.1 Introduction

2. 10.2 Polymorphism Examples
3. 10.3 Demonstrating Polymorphic Behavior
4. 10.4 Abstract Classes and Methods
5. 10.5 Case Study: Payroll System Using Polymorphism
    1. 10.5.1 Abstract Superclass `Employee`
    2. 10.5.2 Concrete Subclass `SalariedEmployee`
    3. 10.5.3 Concrete Subclass `HourlyEmployee`
    4. 10.5.4 Concrete Subclass `CommissionEmployee`
    5. 10.5.5 Indirect Concrete Subclass `BasePlusCommissionEmployee`
    6. 10.5.6 Polymorphic Processing, Operator `instanceof` and Downcasting
6. 10.6 Allowed Assignments Between Superclass and Subclass Variables
7. 10.7 `final` Methods and Classes
8. 10.8 A Deeper Explanation of Issues with Calling Methods from Constructors
9. 10.9 Creating and Using Interfaces
    1. 10.9.1 Developing a `Payable` Hierarchy
    2. 10.9.2 Interface `Payable`
    3. 10.9.3 Class `Invoice`
    4. 10.9.4 Modifying Class `Employee` to Implement Interface `Payable`
    5. 10.9.5 Using Interface `Payable` to Process `Invoices` and `Employees` Polymorphically
    6. 10.9.6 Some Common Interfaces of the Java API
10. 10.10 Java SE 8 Interface Enhancements

    1. 10.10.1 `default` Interface Methods
    2. 10.10.2 `static` Interface Methods
    3. 10.10.3 Functional Interfaces

11. 10.11 Java SE 9 `private` Interface Methods
12. 10.12 `private` Constructors
13. 10.13 Program to an Interface, Not an Implementation

    1. 10.13.1 Implementation Inheritance Is Best for Small Numbers of Tightly Coupled Classes
    2. 10.13.2 Interface Inheritance Is Best for Flexibility
    3. 10.13.3 Rethinking the Employee Hierarchy

14. 10.14 (Optional) GUI and Graphics Case Study: Drawing with Polymorphism
15. 10.15 Wrap-Up

1. Summary
2. Self-Review Exercises
3. Answers to Self-Review Exercises
4. Exercises
5. Making a Difference