Object Oriented Analysis and Design with Java

UE19CS353

Prof. Sindhu R Pai
Department of Computer Science and Engineering

Acknowledgements: Significant portions of the information in the slide sets presented through the course in the class, are extracted from the prescribed text books, information from the Internet. Since these are only intended for presentation for teaching within PESU, there was no explicit permission solicited. We would like to sincerely thank and acknowledge that the credit/rights remain with the original authors/creators only

UE19CS353: Object Oriented Analysis and Design with Java

Composition

Prof. Sindhu R Pai

Department of Computer Science and Engineering

Composition - Agenda

- Introduction
- Types
- Benefits
- Why Composition over Inheritance?
- Rules to choose Composition over Inheritance
- Coding Examples – Wrong and Right Design
- References

Introduction

- The way in which something is put together or arranged
- In OOP, class that references one or more objects of other classes in instance variables.
- If an object contains the other object and the **contained object cannot exist without the existence of first object** - Composition
- A way of describing **reference between two or more classes using instance variable** and an instance should be created before it is used.
- Allows you to model a **has-a association** between objects
- Real world examples:

Car has engine, Room has fan, library has books, computer has keyboard, mouse, speaker, modern coffee machines have an integrated grinder and a brewing unit.

Types

- Based on how you visualize the composed objects of the whole object

Association: When one object wants another object to perform a service for it. You normally see only the whole object

Aggregation: Complex object composed of other objects. You normally see the parts that build the whole object.

- Desktop computer system contains both associations and aggregation.

Interaction between the computer box, the monitor, the keyboard, and the mouse is association

The computer box itself represents aggregation. Computer box actually contains complex system made up of other objects, motherboards, hard drives, video cards, etc.

Benefits

- Provides better test-ability of a class to design clean APIs

- Allows to easily replace the composed class implementation with a better and improved version without adapting any external clients

- Provides code reusability

Why composition over inheritance?

- **Limitations of inheritance**

  Sub-class should support every method super class

  Sub-class objects get a single unnamed object of the superclass.

  Binding of subclass to superclass happens at compile time

- **Coding examples**

Rules to choose composition over inheritance

- Choose composition over inheritance when,

  Subclass cannot support every property/method of the superclass.

  A sub-class requires more than one object of the superclass.

- The rule of programming : “prefer composition over inheritance”

  When in doubt, prefer composition over inheritance.

Coding examples – Wrong and Right Design

- Implementation of Room having Fan
- Implementation of Line having two end points – start and stop

References

- OOP Series — What is Composition? | Geek Culture (medium.com)
- Association, Composition and Aggregation in Java – GeeksforGeeks
- Composition over inheritance – Wikipedia

PES
UNIVERSITY
ONLINE

THANK YOU

Prof. Sindhu R Pai

Department of Computer Science and Engineering

sindhurpai@pes.edu
+91 8277606459