Swinburne University of Technology

COS20007 Object Oriented Programming

Key Object Oriented Concepts

PDF generated at 12:05 on Tuesday 10th October, 2023

Produced by OOP Submission Formatter

I. Introduction

Object-Oriented Programming (OOP) stands as a foundational programming paradigm. It centres on the organization of code into objects, each an instance of a class. These objects encapsulate both data, presented as attributes, and behaviour, delivered through methods. OOP offers a robust framework for code structuring, enhancing maintainability, extensibility, and reusability. This report delves into the four pivotal principles of Object-Oriented Programming.

II. Key Principles

A. Encapsulation: Encapsulation is the practice of concealing an object's internal mechanics from external access. It promotes the exposure of a controlled public interface for interaction with other objects. The advantage lies in safeguarding an object's internal state, simplifying maintenance.

Unit’s program: The "MyRectangle" class in the ShapeDrawing program preserves this concept by maintaining private float values height and width, being accessible and modifiable solely through properties.

B. Inheritance: Inheritance is the mechanism through which one class can inherit the attributes and behaviours of another class. It is a cornerstone of OOP, promoting code reuse and simplifying the development and maintenance of complex software systems.

Unit’s program: The "MyCycle" class within the ShapeDrawing program is a notable example of inheritance. It inherits essential attributes and methods from the "Shape" class, eliminating the need to redefine them for rectangles.

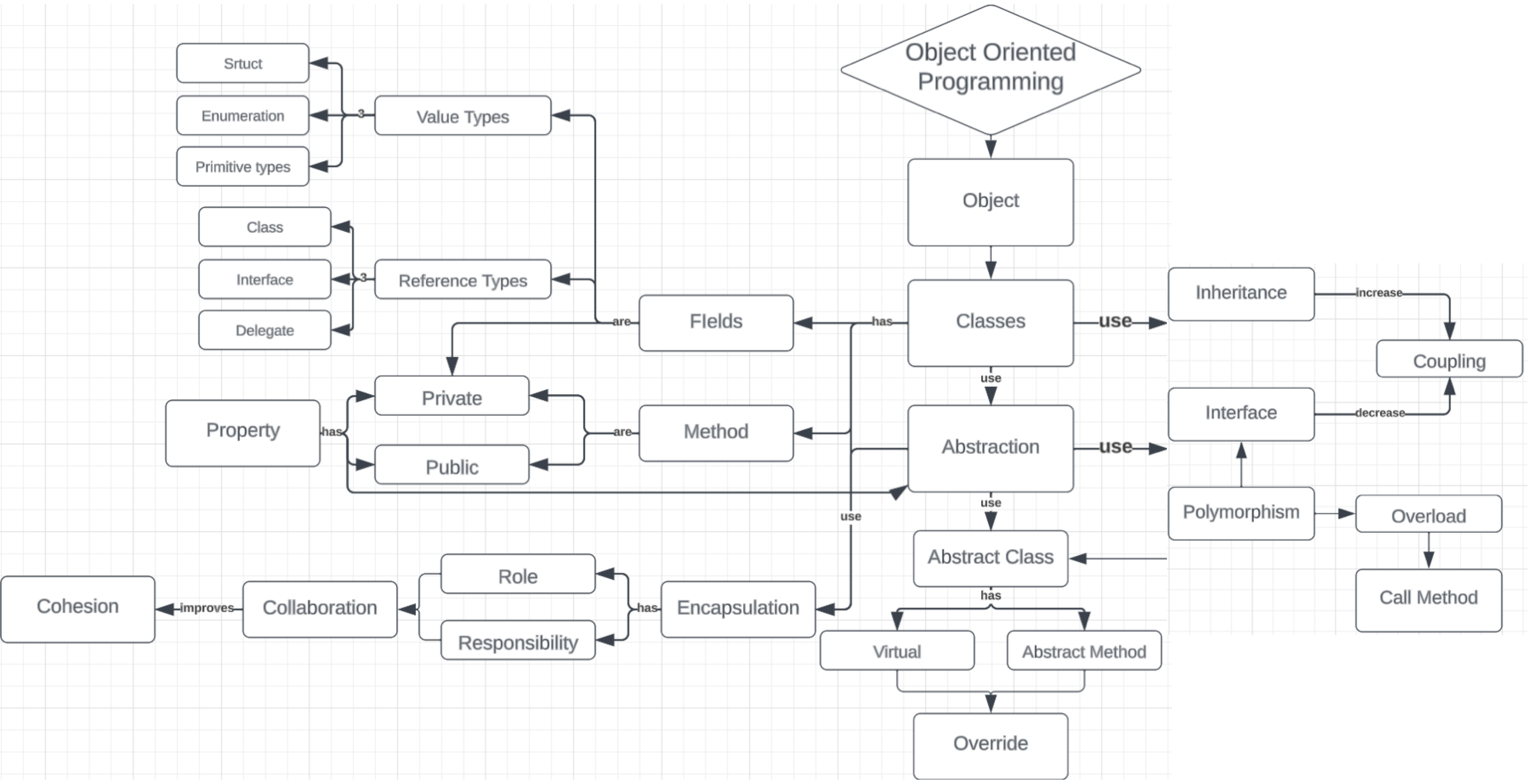
C. Polymorphism: Polymorphism is the concept of objects assuming various forms or, in OOP terms, different objects responding to the same message in distinct ways. It promotes code flexibility and extension.

Unit’s program: In the SwinAdventure program, the concept of polymorphism is evident through the "IHaveInventory" interface, where various game objects, such as rooms and items, implement their unique "Locate" method in response to a common message.

D. Abstraction: In programming, abstraction focuses on defining the essential characteristics and behavior of an object or class while hiding intricate implementation details. It simplifies complex systems, enhances code comprehensibility, and helps identify roles and responsibilities of entities (classes) within a program. Thus, abstraction is a fundamental concept in object-oriented programming which helps in managing code complexity, improving maintainability, and promoting reusability.

Unit’s program In the ShapeDrawing, the "Shape" abstract class embodies abstraction by defining core methods like "Draw," "IsAt," and "DrawOutline." These methods capture the essential operations for all shapes while leaving the implementation details to the individual shape classes.

III. Concept Map


Object Oriented Programming
Object
Classes
has
FIelds
are
Value Types
3
Srtuct
Enumeration
Primitive types
Reference Types
3
Class
Interface
Delegate
Private
Public
are
Method
Property
has
use
Abstraction
use
Abstract Class
has
Virtual
Abstract Method
Override
use
Encapsulation
has
Role
Responsibility
Collaboration
improves
Cohesion
use
Inheritance
increase
Coupling
use
Interface
decrease
Polymorphism
Overload
Call Method