Swinburne University of Technology

COS20007 Object Oriented Programming

Key Object Oriented Concepts

PDF generated at 11:36 on Thursday 12th October, 2023

OOP Principles

Introduction

The object-oriented programming (OOP) paradigm places more focus on the objects than on the logic and function that go into software development. This type of programming is suitable for since programmers can change and rearrange objects inside of programs. large-scale, intricate, and frequently updated applications. OOP is also reusable and modular, it is commonly utilized as a powerful way for arranging and managing complex programes.

There are 4 key Principals: Abstraction, Encapsulation, Inheritance and Polymorphism

Abstraction: Abstraction is a fundamental concept in object-oriented programming, which entails focusing on an object's essential characteristics while ignoring the specifics that are unimportant in the context at hand. It allows for the brief and straightforward representation of complex systems.

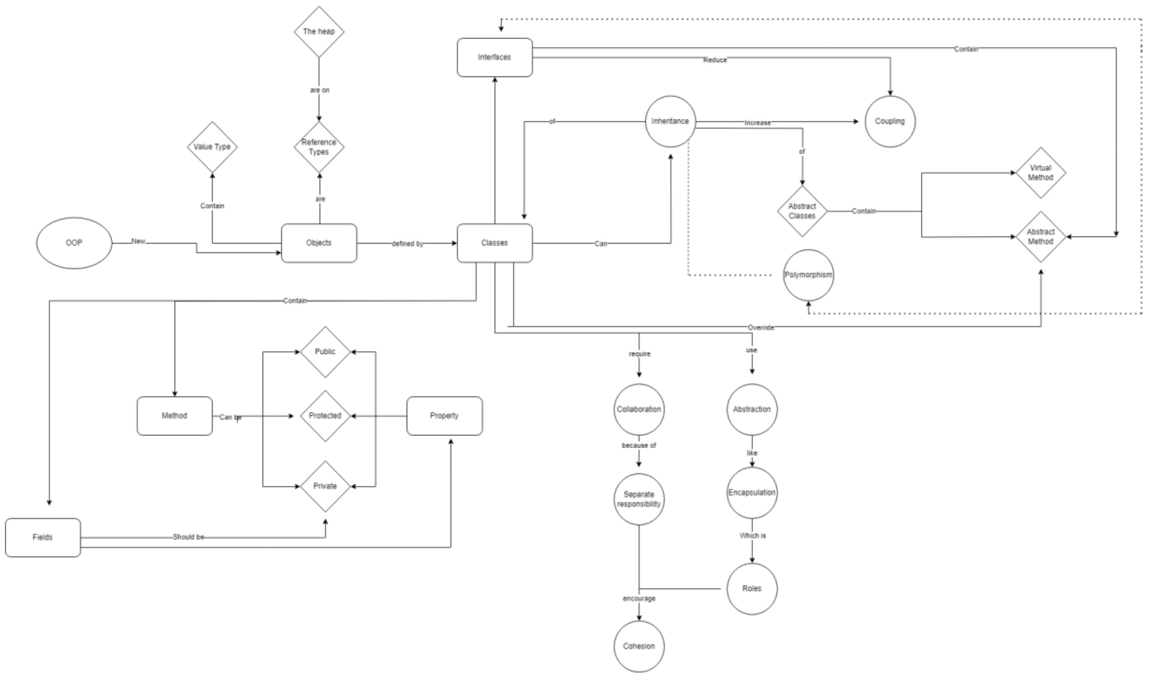
Encapsulation: Encapsulation is the idea of controlling how things can access information by enclosing information and functionality inside of those items. Although objects can be used thanks to methods and properties, their internal workings still need to be protected. This is due to the fact that using things only calls for a comprehension of how they work, and interfering with them mistakenly might have disastrous effects. By employing access modifiers, access to classes that have been programmed can be altered.

Inheritance: Inheritance is a way to create new classes based on existing claases.Typically, the child class specializes or customizes certain behaviors that are generalized in the parent class.The child class can inherit all the characteristics of the parent class. Developers use inheritance to create a family of related classes, promoting the reuse of code.

Abstract classes and interfaces are essential in the context of inheritance. Inheritable abstract classes offer a framework with certain members that don't have specific logic. Although classes can derive from interfaces, an interface's main use is to offer functionality to classes that might not otherwise share the class family's features. It's crucial to remember that an interface can have multiple base classes, whereas a class can only have one.

Polymorphism: The word “polymorphism” stands for having multiple forms. Polymorphism, which enables objects of diverse types to be seen as belonging to the same base type, is a fundamental concept in object-oriented programming. This indicates that you don't need to be aware of any particular derived types of various object types in order to represent them using a base type and call methods on them

Concept Map


The heap
are on
Reference Types
are
Value Type
Contain
OOP
New
Objects
defined by
Classes
Interfaces
Reduce
Contain
Inheritance
of
Increase
Coupling
of
Abstract Classes
Contain
Virtual Method
Abstract Method
Can
Polymorphism
Override
Contain
Method
Can be
Public
Protected
Private
Property
Fields
Should be
require
Collaboration
because of
Separate responsibility
encourage
Cohesion
use
Abstraction
like
Encapsulation
Which is
Roles