Object oriented programming principles

Nguyen Manh Duc ID: 103792724

# Introduction

Of all famous paradigms through out the history of computer programming none can be compare of term of iconic and popular like Object-Oriented (OO) Programming--and for good reason.

OOP is the practice of composing programs from objects and their data as opposed to a grocery-list styled procedural program where they are executed from top to bottom. This way of program makes the code be more natural in some other way and in some other way it encourages modularity and code reuse

# What is OOP basically?

OPP basically is group related data and functions together for the object. And object has fields and method. Fields which to recognize a thing and method which do this thing. An object has also physical object. Like example the television which has the colour, shape, materials and etc which does things like show movie, video or can play a game in this. So the colour and shape is fields and show tv show is method. And like physic world to design a car we need a blueprint of car. The blueprint in OPP and the object can be a class. And using this method we can create many copies of object which we want

So, the term of good design software, we need understand 4 things about the objects:

* Classifications
* Roles
* Responsibilities
* Collaboration

A role is a set of responsibilities and responsibility is a task which an object should do. So, an object may have many responsibilities, such as read, calculate, or print.

And collaboration is the relationship between different classes. It is very

important to identify clear collaboration and classification of objects.

And collaboration is the relationship between different classes. It is very

important to identify clear collaboration and classification of objects.

And collaboration is the relationship between different classes. It is very important to identify clear collaboration and classification of objects.

A well program need have high cohesion and low coupling. A cohesion is the method and fields need to be related to the task. And the coupling is the connection between the objects. Less the number of the link the better the code is.

Another important concept is the interface which help to avoid the overuse inheritance and prevent of making errors.

To master of OOP, we have to understand of 4 principles concepts abstraction, encapsulation, inheritance and polymorphism

# Abstraction

In a nutshell, software complexity is concealed through abstraction. Abstraction is used in the programming process throughout the analysis and design phases. It helps with the identification of product categories, roles, duties, and collaboration arrangements.

# Encapsulation

Data is protected from prying eyes through encapsulation. Encapsulation is used for implementing and developing software. Properties and access specifiers are used to restrict unauthorized access to or alteration of data or code. Encapsulation is used to achieve the necessary level of abstraction. Through abstraction and encapsulation, the programs are also designed to have high cohesive and low coupling.

# Inheritance

Building a new class from an existing one is a technique known as inheritance. It helps programmers write code more quickly and with less complexity. An is-a relationship is a particular kind of class link. A child class is therefore a subset of a parent/base class.

# Polymorphism

Polymorphism refers to the existence of several types or forms. OOP polymorphism enables increased program flexibility. Polymorphism is used in several ways in programming. You may enable polymorphism in your software by utilizing inheritance, which allows you to use any child classes by identifying an object as the parent class.

To get father more I have acquire 4 principles of my programming

By limiting field access and relying on attributes, I use abstraction and encapsulation in the drawing program. The data is shielded from access or modification through encapsulation. Avoiding duplication and shortening the writing process by inheriting the rectangle, circle, and line child classes from the parent class shape. In my program, inheritance also contributes to polymorphism. For instance, we might designate a rectangle or circle as a Shape Class rather of defining a class for it, adding versatility.

Diagram

Description automatically generated

The Maze Game is another example of OOP programming. On consoles only, the labyrinth game is accessible. In this program, each of the four concepts is discussed. The name and description of the game item were specified as private access to keep others out. The code is also easy to maintain because both the Item and Player classes originate from the Game Object class. In order to increase adaptability, the bag class was developed from the item class.

## Summary

OPP is useful in modern software development and technology is change day by day, but the OPP is remaining the same technique in different era. To order to understand deeply about OPP the four main principles need to learn more about

**Concept map**

Diagram

Description automatically generated