COS20007 Object Oriented Programming
Hurdle Task 1: Semester Test

Dang Khoa Le
Student ID: 103844421
Major: Bachelor of Software Engineering

Abstract— **This document presenting Task 2 of the Semester Test, demonstrating the understanding of object-oriented programming and the core concepts of object-oriented design.**

I. Polymorphism:

Q: Describe the principle of polymorphism and how it was used in Task 1.

Polymorphism Principle: "Polymorphism comes from the Greek words 'polys' meaning much or many and 'morphe' (or morphism) meaning form or shape. Polymorphism is one of the core principles of Object-Oriented Programming (OOP). It 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.

Within this task, by introducing the Thing abstract class, both File and Folder classes are derived from it. This allows instances of file and folder to be treated as instances of Thing (class). A collection of Thing objects can be able to add both Files and Folders to it. Polymorphism principal enables code design that operates on the common attributes and methods of Thing within any specific derived types (whether it's a file or a folder).

II. FileSystem and Folder classes:

Q: Consider the FileSystem and Folder classes from the updated design in Task 1. Do we need both of these classes? Explain why or why not.

Both of the FileSystem and Folder classes are compulsory and have to be implemented to satisfy Task 1 requirements.

The FileSystem class is the overall file system that works as a root directory (similar to a parent folder at high-level interfacing). It manages an entire file system with any folders and files to be stored within. The Folder class represents lower-directories or folders within the file system (similar to a child folder at low-level and individual interfacing). The two classes can be distincted that FileSystem manages the entire structure as a root directory, while Folder class has individual folders and their contents (folders or files) within the file system.

III Thing class:

Q: What is wrong with the class name Thing? Suggest a better name for the class, and explain the reasoning behind your answer.

The name (Thing) doesn't necessarily wrong, however, they don't generally have any meaning related to the topic nor could be able to represent or describe any concept. In OOP, it is also better to use class-name that can be descriptive and informative for the concept design, which support developers' cognition towards the code base, in terms of the readability, understandability, locatability and documentation.

Some such names that can be used instead of 'Thing' are 'Object', 'Item', 'Resource', 'Component' and as this class represent a direct - general implementation object to the FileSystem class, they could be named as specifically as 'FileSysyemObject", 'FileSysyem_Object" or in short, 'FSObject.

IV. Abstract:

Q: Define the principle of abstraction, and explain how you would use it to design a class to represent a Book.

Abstraction is a fundamental concept of OOP that simplifies complex systems by ignoring irrelevant details from users. Rather than delving into the inner workings, abstraction permits users to interact with a system through well-defined inputs. Additionally, abstraction involves representing an entity by filtering out unimportant specifics, emphasizing the essential aspects. In OOP, abstraction enables users to define classes or entities without the need to focus on excessive details.

To design a class representing a book, we need to implement a code structure that is similar to this:

+ Implement properties: relevant attributes such as the title, author, publish-date with the book types (audio book or reading book), genres (action, adventure, novel, documentary), and their target audience (for children, teenagers or adults).
+ Implement constructor: relevant constructors to the mentioned properties.
+ Implement classes: a class named as Book and additional classes that responsible to demonstrate the book structure's behaviours/functions, and storing the book's content itself.
+ Implement abstraction: neglect unnecessary specifics and focus on direct attributes to the book.

V. References:

[1]Wikipedia Contributors, "Polymorphism (computer science)," Wikipedia, Nov. 06, 2019.
https://en.wikipedia.org/wiki/Polymorphism_(computer_science)

[2]"How to Write Meaningful Variable Names? | Writing Clean Code," workat.tech.
https://workat.tech/machine-coding/tutorial/writing-meaningful-variable-names-clean-code-za4m83tiesy0#:~:text=Classes%20should%20have%20descriptive%20names (accessed Sep. 27, 2023).

[3]Packtpub.com, 2022.
https://subscription.packtpub.com/book/programming/9781788296540/47/ch47lvl1sec55/setting-up-the-book-class#:~:text=Let (accessed Sep. 27, 2023).