

Python Programming

Inheritance Homework 4

Mostafa S. Ibrahim

Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher

PhD from Simon Fraser University - Canada

Bachelor / Msc from Cairo University - Egypt

Ex-(Software Engineer / ICPC World Finalist)

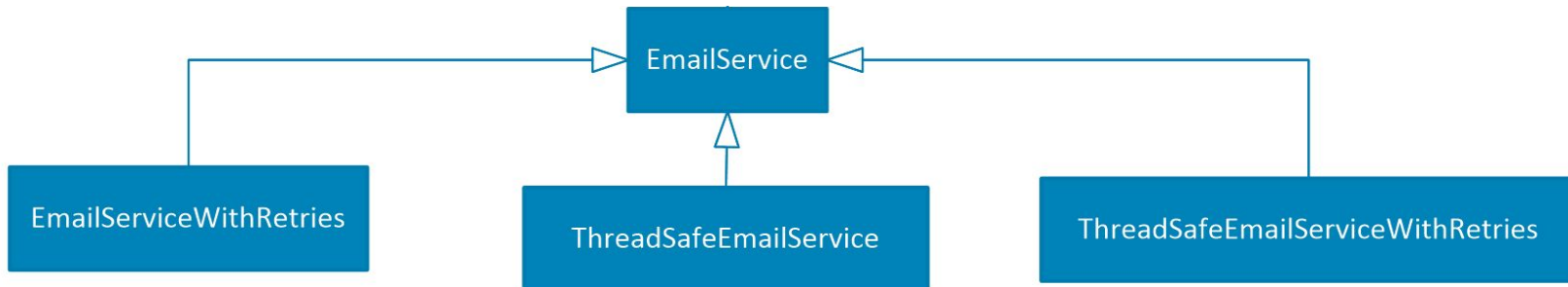


Problem #1: More Features

- Background.
 - When we write some service (e.g. Email service), we might have some **standard features**
 - 1) Pure service, just normal
 - 2) Logging support: means the service log info/issues to some file
 - 3) Caching support: means some information are in memory for fast processing
 - 4) Thread-Safe support: it means several threads access shared resources in proper way
 - There might be some software **guidelines to follow**
 - 1) There is only a single reason to change a class, otherwise split the features somehow
 - E.g. you don't create a single class with logging + caching + thread-safe support
 - You have now 4 reasons to do a change to this class!
 - 2) Don't play with others code. Consider it closed. Reuse/Extend what is available

Problem #1: More Features

- When you joined an email service company they had code for the following
 - A basic email service that just sends/downloads email once
 - An extension that support multiple trials if there is failure (using inheritance)
 - An extension that support thread safe
 - An extension that support multiple trials if there is failure + thread safe
 - Note that, there are several cases in the system that make use of any of the 4 classes



Problem #1: More Features

- A new module implemented in the code base: [Logger](#)
- A new feature is requested: Logging feature to the overall Email service
- There is a need to be able to use the current 4 classes:
 - Without logging feature (directly use available ones)
 - E.g. someone may need: basic service + retries (no logging or thread safe)
 - E.g. someone may need: basic service + logging (no retries or thread safe)
 - E.g. someone may need: basic service + logging + retries + thread safe
- Draw the new class diagram that supports logging feature
 - Remember, you shouldn't modify the current classes, but reuse them

Problem #2: More Features

- After a few months, there is a need to also support caching feature
- There is a need to be able to use the current classes (without caching):
 - E.g. with or without logging
- Now classes should support with or without caching
 - E.g. someone may need: basic service + caching + retrials (no logging or thread-safe)
 - E.g. someone may need: basic service + retrials + logging + caching (no thread-safe)
- Design a new UML
 - Use my solution for previous homework and modify it
 - There is a critical concern you should notice at the moment. What is it?

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”