****

Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №7  
**ШАБЛОНИ «MEDIATOR»,**

**«FACAD, «BRIDGE», «TEMPLATE METHOD»**

Виконав Перевірив:

студент групи ІА – 13: Мягкий М.Ю

Нестеренко Ростислав

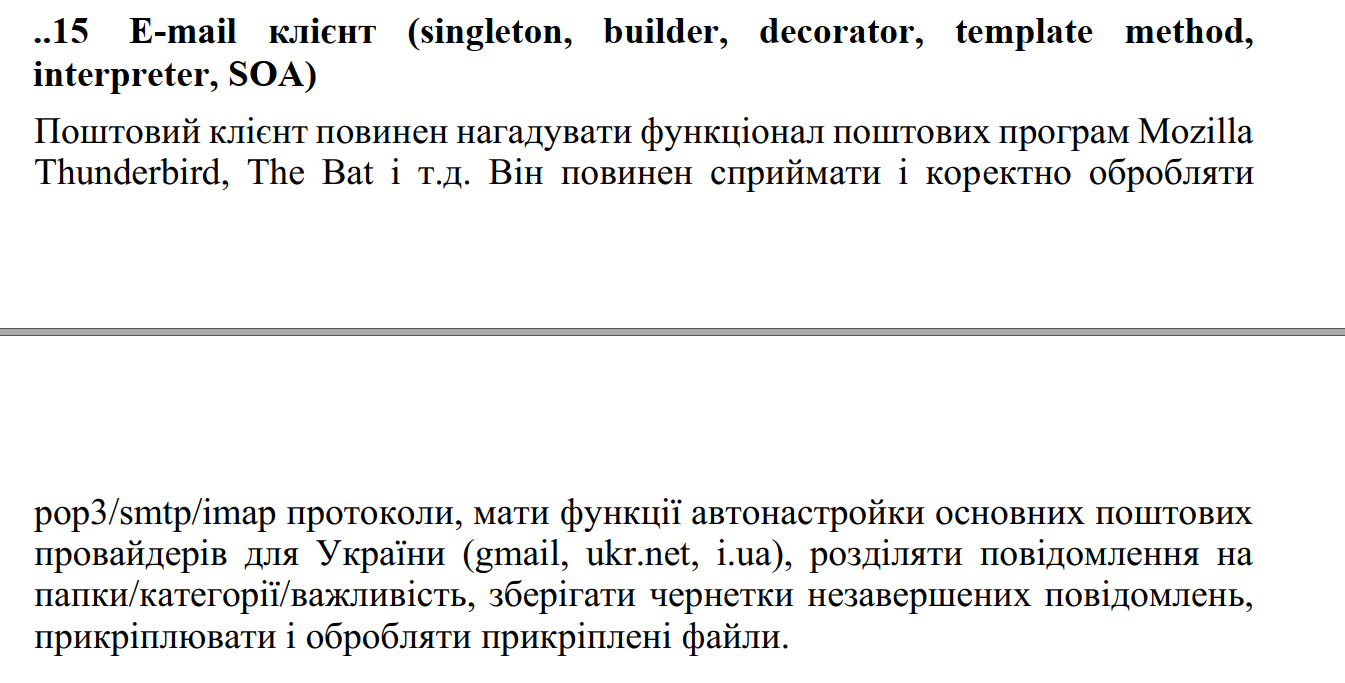
**Завдання:**

1. Ознайомитися з короткими теоретичними відомостями.

2. Реалізувати частину функціонала робочої програми у вигляді класів та їхньої взаємодії для досягнення конкретних функціональних можливостей.

3. Застосування одного з розглянутих шаблонів при реалізації програми

**Варіант:**



**Хід роботи**

Паттерн Template Method (шаблонний метод) використовується для визначення загальної структури алгоритму, але дозволяє підкласам перевизначити певні кроки цього алгоритму без зміни його загальної структури.

Основний шаблон Template Method реалізований в класі **EmailTemplate**. У цьому класі є загальний алгоритм для роботи з електронною поштою, який включає такі кроки, як підключення до сервера, підготовка повідомлення, відправлення повідомлення і відключення від сервера. Ці кроки визначені у вигляді абстрактних методів, що дозволяє підкласам **EmailTemplate** перевизначати їх, виконуючи конкретні дії відповідно до своїх потреб.

Наприклад, клас **EmailClient** є одним з підкласів **EmailTemplate**. У цьому класі конкретно реалізовані методи **connect\_to\_server**, **prepare\_message**, **send\_message** та **disconnect\_from\_server** для роботи з SMTP-сервером та відправлення листів.

Інший клас, **EmailOrganizer**, також є підкласом **EmailTemplate**, і в ньому реалізовані власні конкретні реалізації методів для роботи з IMAP-сервером та організації листів.

При цьому у класі **EmailServer** створюється екземпляр підкласу **EmailTemplate** (наприклад, **EmailClient** або **EmailOrganizer**) і виконуються операції над електронною поштою, використовуючи метод **perform\_email\_operation**. При цьому конкретний підклас, який вибирається при створенні **EmailServer**, визначає конкретну реалізацію кожного з методів.

Цей шаблон дозволяє легко додавати нові підкласи **EmailTemplate**, які можуть працювати з різними типами поштових серверів або виконувати різні операції з поштою, зберігаючи при цьому загальну структуру алгоритму.

Висновок: ознайомився з короткими теоретичними відомостями. Реалізував частину функціонала робочої програми у вигляді класів та їхньої взаємодії для досягнення конкретних функціональних можливостей. Застосував паттерн Template Method для реалізації програми.