****

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 №4  
**ШАБЛОНИ «SINGLETON»,**

**«ITERATOR», «PROXY», «STATE»,**

**«STRATEGY»**

Виконала Перевірив:

студентка групи ІА – 13: Мягкий М. Ю.

Нестеренко Ростислав

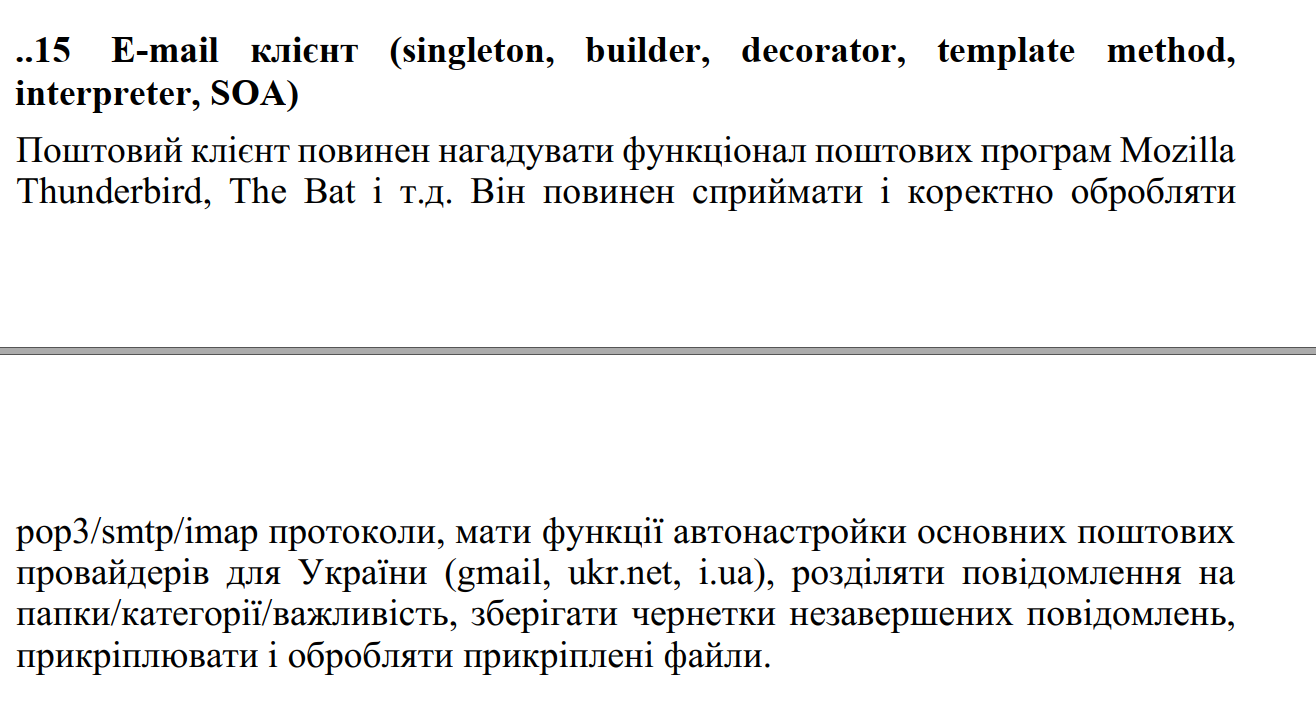
**Завдання:**

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**

**Одинак** — це породжувальний патерн проектування, який гарантує, що клас має лише один екземпляр, та надає глобальну точку доступу до нього.

1. **Зберігання Інстансів**: У класі **EmailProvider** використовується словник **\_instances** для зберігання інстансів. Ключем у цьому словнику є кортеж, що містить параметри конфігурації (SMTP сервер, порт SMTP, сервер IMAP, сервер POP3).
2. **Перевизначення методу \_\_new\_\_**: Метод **\_\_new\_\_** перевизначений для контролю створення інстансів. Перед створенням нового інстансу перевіряється, чи інстанс з такою ж конфігурацією вже існує у словнику **\_instances**. Якщо так, то повертається існуючий інстанс, інакше створюється новий.
3. **Ініціалізація Інстансу**: У методі **\_\_init\_\_** використовується атрибут **\_initialized** для перевірки, чи інстанс вже був ініціалізований. Це запобігає повторній ініціалізації інстансу при кожному виклику **\_\_init\_\_**, оскільки **\_\_new\_\_** може повертати вже існуючий інстанс.

Цей підхід забезпечує, що для кожної унікальної конфігурації сервера електронної пошти існує лише один інстанс **EmailProvider**, що дозволяє ефективно використовувати ресурси і забезпечує консистентність даних.

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