****

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 №5  
**ШАБЛОНИ «ADAPTER»,**

**«COMMAND, «CHAIN OF RESPONSIBILITY», «PROTOTYPE»**

Виконав Перевірив:

студент групи ІА – 13: Мягкий М. Ю

Нестеренко Ростислав

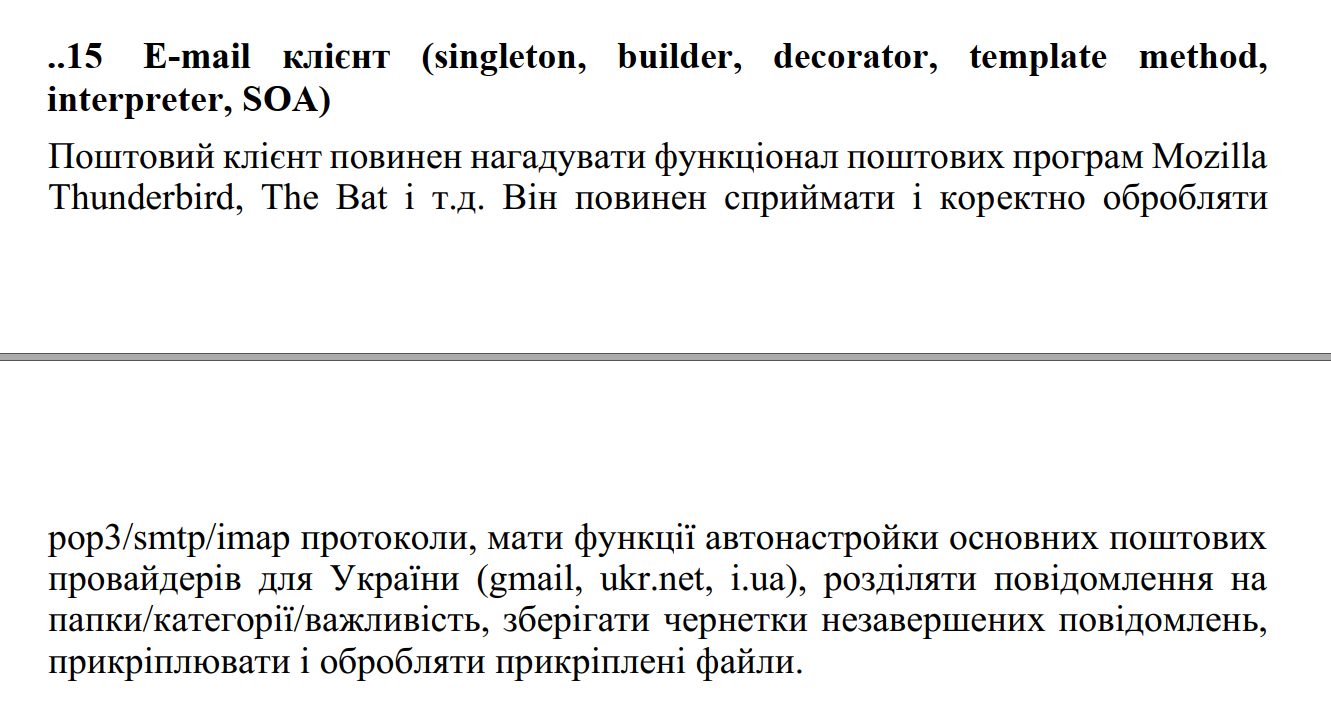
**Завдання:**

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 Builder використовується для створення складних об'єктів **EmailClient** та **EmailOrganizer** з допомогою побудови їх параметрів крок за кроком. Основна мета паттерна - розділити процес створення об'єкта від його представлення, дозволяючи створювати різні конфігурації об'єктів без необхідності створювати велику кількість конструкторів.

Основні елементи паттерна Builder в коді:

1. **EmailClientBuilder** - це клас-будівник, який використовується для побудови об'єктів **EmailClient** та **EmailOrganizer**. Він містить методи для налаштування параметрів об'єкта, такі як **set\_provider**, **set\_user\_email** та **set\_user\_password**.
2. **build** та **build\_organizer** - це методи класу **EmailClientBuilder**, які фактично створюють об'єкти **EmailClient** та **EmailOrganizer** відповідно, використовуючи задані параметри.
3. У головному коді (**EmailServer** та **EmailClient**) будівники використовуються для створення об'єктів **EmailClient** та **EmailOrganizer** з відповідними параметрами, використовуючи послідовний підхід до конфігурації об'єктів.

Цей підхід дозволяє зробити код більш гнучким і легко розширюваним, оскільки ви можете додавати нові параметри для об'єктів без необхідності змінювати всю логіку створення об'єктів.

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