

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота № 7

з дисципліни «Теорія розробки ПЗ»

Тема: ШАБЛОН «MEDIATOR», «FACADE», «BRIDGE», «TEMPLATE
METHOD»

Виконав:
студент групи ІА-14

Онуфрійчук Антон
Валерійович

Дата здачі _____

Захищено з балом _____

Перевірів: М'який Михайло
Юрійович

Тема: ШАБЛОН «MEDIATOR», «FACADE», «BRIDGE», «TEMPLATE METHOD»

Завдання.

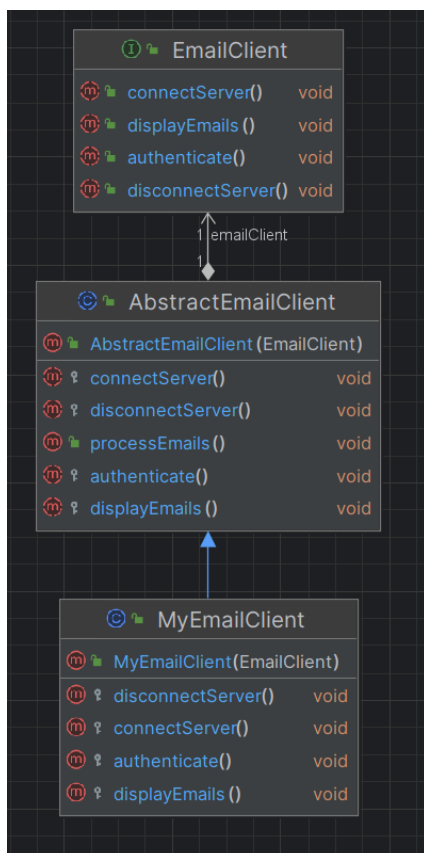
1. Ознайомитися з короткими теоретичними відомостями.
2. Проаналізуйте тему та намалюйте схему прецеденту, що відповідає обраній темі лабораторії.

..15 E-mail клієнт (singleton, builder, decorator, template method, interpreter, SOA)

Поштовий клієнт повинен нагадувати функціонал поштових програм Mozilla Thunderbird, The Bat і т.д. Він повинен сприймати і коректно обробляти

pop3/smtp/imap протоколи, мати функції автонастройки основних поштових провайдерів для України (gmail, ukr.net, i.ua), розділяти повідомлення на папки/категорії/важливість, зберігати чернетки незавершених повідомлень, прикріплювати і обробляти прикріплені файли.

Діаграма класів



```

public abstract class AbstractEmailClient {
    5 usages
    protected EmailClient emailClient;

    1 usage
    public AbstractEmailClient(EmailClient emailClient) { this.emailClient = emailClient; }
    // Template Method, який визначає основну послідовність дій

    no usages
    public final void processEmails() {
        connectServer();
        authenticate();
        // Можна додати інші кроки

        // Цей крок може бути перевизначений у підкласах
        displayEmails();

        disconnectServer();
    }

    // Абстрактні методи, які повинні бути реалізовані у підкласах

    1 usage 1 implementation
    protected abstract void connectServer();

    1 usage 1 implementation
    protected abstract void authenticate();

    1 usage 1 implementation
    protected abstract void displayEmails();

    1 usage 1 implementation
    protected abstract void disconnectServer();
}

```

```

public interface EmailClient {
    1 usage
    void connectServer();
    1 usage
    void authenticate();
    1 usage
    void displayEmails();
    1 usage
    void disconnectServer();
}

```

```

public class MyEmailClient extends AbstractEmailClient {
    no usages
    public MyEmailClient(EmailClient emailClient) {
        super(emailClient);
    }

    1 usage
    @Override
    protected void connectServer() {
        super.emailClient.connectServer();
        // Логіка підключення до сервера
    }

    1 usage
    @Override
    protected void authenticate() {
        super.emailClient.authenticate();
        // Логіка аутентифікації
    }

    1 usage
    @Override
    protected void displayEmails() {
        super.emailClient.displayEmails();
        // Логіка відображення повідомлень
    }

    1 usage
    @Override
    protected void disconnectServer() {
        super.emailClient.disconnectServer();
        // Логіка відключення від сервера
    }
}

```

Висновок: При виконанні лабораторної роботи реалізував патерн template method.