

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Лабораторна робота №4

Тема: «Патерн проектування Адаптер»

Дисципліна: «Ефективність та якість архітектурних рішень інформаційних систем»

Виконав: студент групи ІКМ-М224а

Нестеренко Владислав Валентинович

Перевірів: Асистент кафедри

Хорошун Андрій Сергійович

Харків - 2024р.

Мета: Здобути навички з реалізацією паттерна проектування Адаптер.

Завдання:

1. Ознайомитися з патерном Адаптер.
2. Виконати всі пункти лабораторної роботи.
3. Надати звіт про виконану роботу.

Хід роботи:

1. Дано наступний код:

```
<?php

interface Notification
{
    public function send(string $title, string $message);
}

class EmailNotification implements Notification
{
    private $adminEmail;

    public function __construct(string $adminEmail)
    {
        $this->adminEmail = $adminEmail;
    }

    public function send(string $title, string $message): void
    {
        mail($this->adminEmail, $title, $message);
        echo "Sent email with title '$title' to '{$this->adminEmail}' that says '$message'.";
    }
}
```

2. Є система у якій є клас EmailNotification відповідальний за відправку email-сповіщень.
3. Змінити інтерфейс Notification немає можливості. Його слід залишити у тому вигляді у якому його наведено у завданні.
4. Необхідно додати можливість відправки сповіщень у месенджер Slack. Для цього ми повинні авторизуватися у Slack за допомогою login та apiKey, а також вказати chatId - у який саме чат ми хочемо надіслати сповіщення
5. Необхідно додати можливість відправки сповіщень SMS. Для відправки нам треба вказати phone та sender - ім'я від якого буде надіслано sms.

Необхідно створити структуру класів та методів яка буде демонструвати реалізацію патерну Адаптер і буде вирішувати описане завдання.

Також необхідно навести приклад клієнтського коду де буде реалізовано відправку email, повідомлення у Slack та sms.

У рамках виконання лабораторної роботи не потрібно описувати деталі реалізації самих методів! Достатньо вказати сам метод та параметри який він приймає та повертає.

Виконання:

Адаптер — це структурний патерн проектування, що дає змогу об'єктам із несумісними інтерфейсами працювати разом.

Реалізація на Java:

Інтерфейс Notification - визначає метод для відправки повідомлень

```
public interface Notification {  
    void send(String title, String message);  
}
```

Клас EmailNotification: Реалізує інтерфейс Notification для відправки email.

```
public class EmailNotification implements Notification {  
    private String adminEmail;  
  
    // Конструктор приймає email адміністратора  
    public EmailNotification(String adminEmail) {  
        this.adminEmail = adminEmail;  
    }  
  
    // Метод для відправки email-повідомлення  
    @Override  
    public void send(String title, String message) {  
        // Реалізація відправки email  
        System.out.println("Sent email with title '" + title + "' to '" + adminEmail +  
"'" that says '" + message + "'");  
    }  
}
```

Клас SlackNotification - для відправки повідомлень у Slack.

```
// Клас SlackNotification для відправки повідомлень у Slack  
public class SlackNotification {  
    private String login;  
    private String apiKey;  
    private String chatId;  
  
    // Конструктор приймає дані для авторизації у Slack  
    public SlackNotification(String login, String apiKey, String chatId) {  
        this.login = login;  
        this.apiKey = apiKey;  
        this.chatId = chatId;  
    }  
  
    // Метод для відправки повідомлення у Slack  
    public void sendSlackMessage(String title, String message) {  
        System.out.println("Sent Slack message with title '" + title + "' to chat '" +  
chatId + "' that says '" + message + "'");  
    }  
}
```

Клас SlackNotificationAdapter - адаптер, який реалізує Notification для використання SlackNotification.

```
public class SlackNotificationAdapter implements Notification {
    private SlackNotification slackNotification;

    // Конструктор приймає екземпляр SlackNotification
    public SlackNotificationAdapter(SlackNotification slackNotification) {
        this.slackNotification = slackNotification;
    }

    // Метод для відправки повідомлення у Slack через адаптер
    @Override
    public void send(String title, String message) {
        slackNotification.sendSlackMessage(title, message);
    }
}
```

Клас SMSNotification - для відправки SMS.

```
// Клас SMSNotification для відправки SMS
public class SMSNotification {
    private String phone;
    private String sender;

    // Конструктор приймає номер телефону та ім'я відправника
    public SMSNotification(String phone, String sender) {
        this.phone = phone;
        this.sender = sender;
    }

    // Метод для відправки SMS
    public void sendSMS(String title, String message) {
        // Реалізація відправки SMS
        System.out.println("Sent SMS with title '" + title + "' from '" + sender + "' to phone '" + phone + "' that says '" + message + "'");
    }
}
```

Клас SMSNotificationAdapter - адаптер, який реалізує Notification для використання SMSNotification.

```
public class SMSNotificationAdapter implements Notification {
    private SMSNotification smsNotification;

    // Конструктор приймає екземпляр SMSNotification
    public SMSNotificationAdapter(SMSNotification smsNotification) {
        this.smsNotification = smsNotification;
    }

    // Метод для відправки SMS через адаптер
    @Override
    public void send(String title, String message) {
        smsNotification.sendSMS(title, message);
    }
}
```

Клієнтський код який демонструє відправку email, повідомлення у Slack та SMS:

```
//Приклад клієнтського коду
public class Main {
    public static void main(String[] args) {
        // Створення та використання EmailNotification
        Notification emailNotification = new EmailNotification("admin@example.com");
        emailNotification.send("Email Title", "This is an email message.");
    }
}
```

```
// Створення SlackNotification та використання його через адаптер
SlackNotification slack = new SlackNotification("userLogin", "apiKey123",
"chat123");
Notification slackNotification = new SlackNotificationAdapter(slack);
slackNotification.send("Slack Title", "This is a Slack message.");

// Створення SMSNotification та використання його через адаптер
SMSNotification sms = new SMSNotification("1234567890", "MySender");
Notification smsNotification = new SMSNotificationAdapter(sms);
smsNotification.send("SMS Title", "This is an SMS message.");
}
}
```

Висновки

Під час виконання даної лабораторної роботи я здобув навички з реалізації патерна проектування Адаптер. Я створив структуру класів та методів, яка демонструє застосування цього патерна для інтеграції різних видів повідомлень у єдиний інтерфейс Notification. Зокрема, я написав адаптери для системи відправки повідомлень у Slack та SMS, які адаптують їх до існуючого інтерфейсу Notification.