

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

Лабораторна робота № 6

з дисципліни «Технології розробки програмного забезпечення»

Тема: шаблони: «ABSTRACT FACTORY», «FACTORY METHOD»,
«MEMENTO», «OBSERVER», «DECORATOR»

Виконав:

студент групи ІА-14

Захарчук Захар Сергійович

Дата здачі

Захищено з балом _____

Перевірив:

Мягкий Михайло Юрійович

Київ, 2023

Завдання:

1. Ознайомитися з короткими теоретичними відомостями.
2. Реалізувати частину функціоналу робочої програми у вигляді класів та їхньої взаємодії для досягнення конкретних функціональних можливостей.
3. Застосування одного з розглянутих шаблонів при реалізації програми.

Хід роботи

Відповідно до теми завдання “Текстовий редактор” було реалізовано паттерн проектування “Observer”(Спостерігач). Відповідно до варіантів використання було вирішено імплементувати даний паттерн як частину логіки підсвічування синтаксису мови Java. Класи, які відносяться до даного паттерну зображені на рисунку 1:

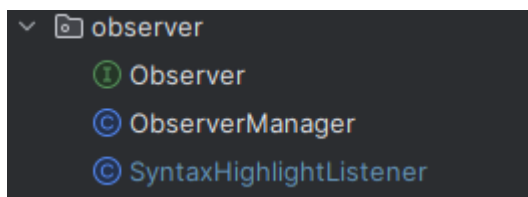


Рисунок 1

На рисунку 2 зображено інтерфейс Observer, який на даний момент має одну імплементацію, а саме SyntaxHighlightListener. Інтерфейс має єдиний метод update():

```
4 usages 1 implementation Zakhar Zakharchuk
public interface Observer {
    1 usage 1 implementation Zakhar Zakharchuk
    void update();
}
```

Рисунок 2

На рисунку 3 зображено клас SyntaxHighlightListener, повний код знаходиться за посиланням <https://github.com/ZakharZakharchuk/trpz-zakharchuk-zakhar>, який реалізує метод update, а саме викликає різні методи підсвічування синксису, які реалізовані через “Templathe method” паттерн який буде розглянутий в наступній роботі.

```

2 usages
public class SyntaxHighlightListener implements Observer {

    2 usages
    private final SyntaxHighlightTemplate reservedWordHighlight;
    2 usages
    private final SyntaxHighlightTemplate stringHighlight;
    2 usages
    private final SyntaxHighlightTemplate numberHighlight;
    2 usages
    private final SyntaxHighlightTemplate annotationHighlight;

    1 usage  Zakhar Zakharchuk
    public SyntaxHighlightListener(Editor editor, ObserverManager observerManager) {
        reservedWordHighlight = new ReservedWordSyntaxHighlight(editor);
        stringHighlight = new StringSyntaxHighlight(editor);
        numberHighlight = new NumberSyntaxHighlight(editor);
        annotationHighlight = new AnnotationSyntaxHighlight(editor);
        observerManager.addObserver(this);
    }

    1 usage  Zakhar Zakharchuk
    @Override
    public void update() {
        reservedWordHighlight.update();
        stringHighlight.update();
        numberHighlight.update();
        annotationHighlight.update();
    }
}

```

Рисунок 3

Оповіщення слухачів відбувається через виклик методу notify() в класі ObserverManager. Приклад на рисунку 4:

```

public class ObserverManager {

    2 usages
    private final List<Observer> observers = new ArrayList<>();

    1 usage  Zakhar Zakharchuk
    public void addObserver(Observer observer) { observers.add(observer); }

    2 usages  Zakhar Zakharchuk
    public void notifyObservers() {
        for (Observer observer : observers) {
            observer.update();
        }
    }
}

```

Рисунок 4

Висновок: виконуючи дану лабораторну роботу було імплементовано паттерн “Спостерігач”. Було створено один клас спостерігача, який відповідає за підсвічування синтаксису. Також були розглянуті теоретичні відомості та реалізації інших паттернів проектування з даної лабораторної роботи.