



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 7  
З дисципліни: «Теорія розробки програмного  
забезпечення»  
Web-browser  
TemplateMethod

Виконав:

студент групи ІА-14  
Калінін Я.В

Перевірив:

Мягкий М.Ю

Київ 2023

**Мета:** Реалізувати шаблон проектування «Proxy» до заданої теми :

Web-browser (proxy, chain of responsibility, factory method, template method, visitor, p2p) Веб-браузер повинен мати можливість зробити наступне: мати адресний рядок для введення адреси сайту, переміщатися і відображати структуру html документа, переглядати підключений javascript та css файли, перегляд всіх підключених ресурсів (зображень), коректна обробка відповідей з сервера (коди відповідей HTTP) - переходи при перенаправленнях, відображення сторінок 404 і 502/503.

**Теоритичні відомості : Шаблонний метод** — це поведінковий патерн проектування, який визначає кістяк алгоритму, перекладаючи відповідальність за деякі його кроки на підкласи. Патерн дозволяє підкласам перевизначати кроки алгоритму, не змінюючи його загальної структури.

Код:

```
1 package com.example.demo;
2
3 // inheritor
4 public abstract class ContentRendererTemplate {
5     // Цей метод визначає скелет алгоритму
6     no usages
7     public final void renderContent() {
8         loadHTML();
9         applyCSS();
10        executeJavaScript();
11    }
12
13    // Абстрактні методи, які будуть реалізовані в конкретних підкласах
14    1 usage 1 implementation
15    protected abstract void loadHTML();
16
17    1 usage 1 implementation
18    protected abstract void applyCSS();
19
20    1 usage 1 implementation
21    protected abstract void executeJavaScript();
22
23 }
```

Шаблон проектування "Template Method" використовується для визначення скелету алгоритму у базовому класі, але дозволяє підкласам перевизначати певні кроки алгоритму без зміни його структури. Тут ми створюємо абстрактний базовий клас, який визначатиме скелет алгоритму, а конкретні підкласи будуть реалізовувати конкретні етапи алгоритму для відображення HTML, CSS та JS вмісту.

```

1 package com.example.demo;
2
3 public class HtmlCssJsRenderer extends ContentRendererTemplate {
4
5     1 usage
6     @Override
7     protected void loadHTML() {
8         // Логіка для завантаження HTML
9         System.out.println("HTML");
10    }
11
12    1 usage
13    @Override
14    protected void applyCSS() {
15        // Логіка для застосування CSS
16        System.out.println("CSS");
17    }
18
19    1 usage
20    @Override
21    protected void executeJavaScript() {
22        // Логіка для виконання JavaScript
23        System.out.println("Execute JS");
24    }
25 }

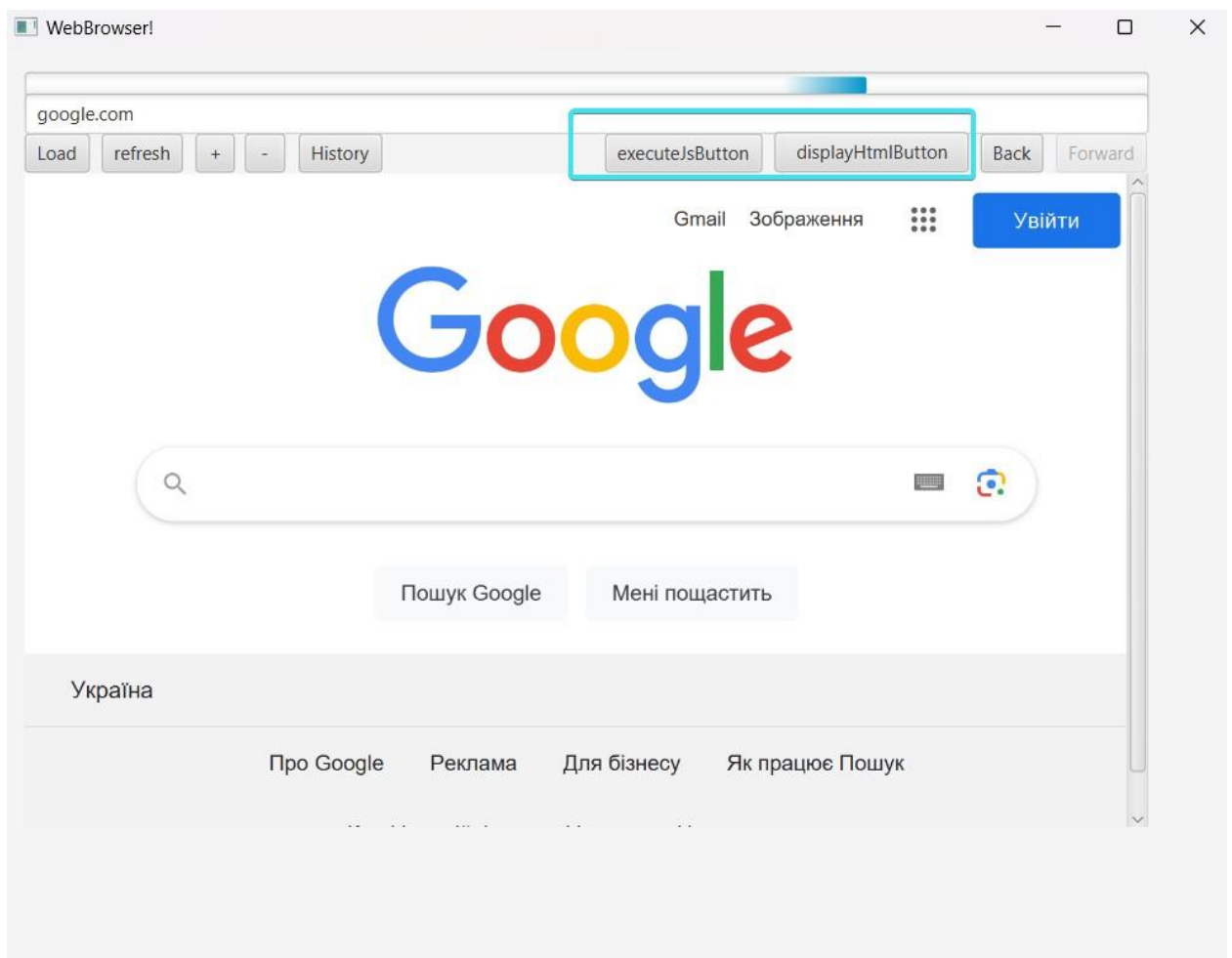
```

Тепер можна використовувати HtmlCssJsRenderer для виклику методу renderContent(), який визначає скелет алгоритму, а конкретні етапи будуть виконані відповідно до реалізації у підкласі.

Також я реалізував візуальне відображення кнопок відповідно до моєї логіки а саме : кнопка з відображення структури html/css та переглядати

## підключений javascript

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.control.Button?>
4 <?import javafx.scene.control.ProgressBar?>
5 <?import javafx.scene.control.TextField?>
6 <?import javafx.scene.layout.AnchorPane?>
7 <?import javafx.scene.web.WebView?>
8
9 <AnchorPane prefHeight="700.0" prefWidth="900.0" xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1" fx:controller="
10     <ProgressBar fx:id="progressBar" layoutX="14.0" layoutY="14.0" prefWidth="730.0" />
11     <WebView fx:id="webView" layoutX="14.0" layoutY="80.0" prefHeight="425.0" prefWidth="730.0" />
12     <Button layoutX="14.0" layoutY="54.0" mnemonicParsing="false" onAction="#loadPage" text="Load" />
13     <TextField fx:id="textField" layoutX="14.0" layoutY="28.0" onAction="#loadPage" prefHeight="26.0" prefWidth="730.0" />
14     <Button layoutX="64.0" layoutY="54.0" mnemonicParsing="false" onAction="#refreshPage" text="refresh" />
15     <Button layoutX="125.0" layoutY="54.0" mnemonicParsing="false" onAction="#zoomIn" prefHeight="18.0" prefWidth="25.0" text="+" />
16     <Button layoutX="157.0" layoutY="54.0" mnemonicParsing="false" onAction="#zoomOut" prefHeight="18.0" prefWidth="26.0" text="-" />
17     <Button layoutX="192.0" layoutY="54.0" mnemonicParsing="false" onAction="#displayHistory" prefHeight="18.0" prefWidth="54.0" text="History" />
18     <Button fx:id="backButton" layoutX="635.0" layoutY="54.0" mnemonicParsing="false" onAction="#goBack" text="Back" />
19     <Button fx:id="forwardButton" layoutX="684.0" layoutY="54.0" mnemonicParsing="false" onAction="#goForward" text="Forward" />
20     <Button fx:id="displayHtmlButton" layoutX="501.0" layoutY="53.0" mnemonicParsing="false" onAction="#displayHtmlStructureOnPage" prefHeight="18.0" prefWidth="100.0" text="displayHtmlButton" />
21     <Button fx:id="executeJsButton" layoutX="391.0" layoutY="54.0" mnemonicParsing="false" onAction="#executeJavaScript" text="executeJsButton" />
22     <!--<Button layoutX="253.0" layoutY="54.0" mnemonicParsing="false" onAction="#executeJS" text="ExecuteJS" />-->
23
24 </AnchorPane>
25
```



**Висновок :** у цій лабораторній роботі я ознайомився з шаблоном проектування «TemplateMethod» та реалізували базову структуру відповідно до тематики мого проекту



Київ 2023