Laura Sofía Vallejo Bermúdez — 202312975

TALLER # 5 — Patrones de Diseño

Análisis Crítico de Uso del Patrón de Diseño "Estrategia" en el Proyecto jsoup

Introducción:

Este estudio se adentra en el análisis de los patrones de diseño, centrándose en la aplicación y eficacia del patrón de comportamiento conocido como "Estrategia" en el contexto del proyecto jsoup. Se destaca la importancia de los patrones de diseño en el desarrollo de software antes de abordar específicamente el patrón "Estrategia".

Información General del Proyecto:

El proyecto jsoup, disponible en el repositorio https://github.com/jhy/jsoup , se presenta como una biblioteca especializada en el análisis de HTML. Su propósito abarca desde la edición y limpieza hasta la extracción de información de documentos HTML, incluyendo medidas de seguridad contra ataques XSS (Cross-Site Scripting) [2]. La estructura del diseño se compone de una clase principal llamada Jsoup, que actúa como la puerta de acceso a las diversas funcionalidades de la librería. Un método crucial, parse, utiliza la clase Parser para analizar documentos HTML y construir un árbol de nodos mediante la interacción con un TreeBuilder que arma un árbol de LeafNodes. El documento HTML también se representa como un nodo que contiene a otros nodos, el cual es de clase Document y hereda de Element (ver Figura 1.). El proyecto enfrenta desafíos significativos, como la variabilidad en los datos de documentos HTML y la gestión de documentos dinámicos generados con JavaScript.

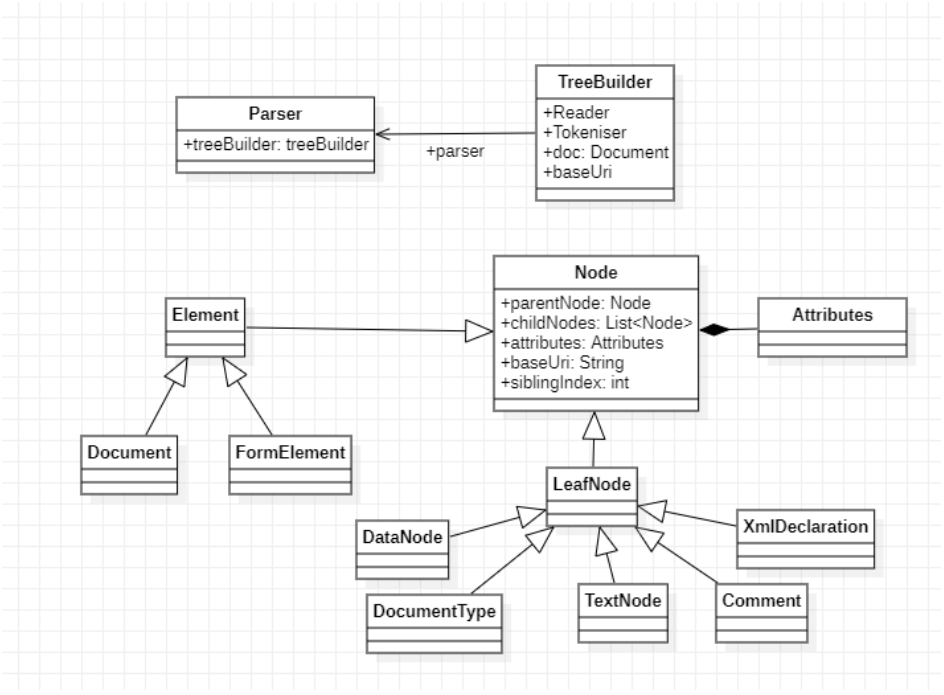

Figura 1. Representación de un nodo en el árbol y la clase principal y la relación entre *Parser* y *TreeBuilder*.

Retos:

Entre los retos de diseño que enfrenta el proyecto se encuentra la variabilidad que presenta un documento HTML en los datos que contiene. Se debe pensar en la manera ideal de almacenar la información del documento, manteniendo la jerarquía y el orden. Otro desafío es manejar documentos de HTML dinámicos generados con JavaScript, lo que implica la posibilidad de que un elemento cambie de estado después de su revisión. Consecuentemente, el diseño debe tener en cuenta la posibilidad de que los elementos ya leídos cambien de estado.

Información y Estructura del Fragmento del Proyecto con el Patrón de "Estrategia":

El fragmento de interés reside en la clase Parser y su conexión con el TreeBuilder, como se representa en la Figura 2 del documento. Este fragmento es esencial para comprender la implementación del patrón de diseño "Estrategia" en jsoup, derivado de la estructura descrita en el libro guía [1].

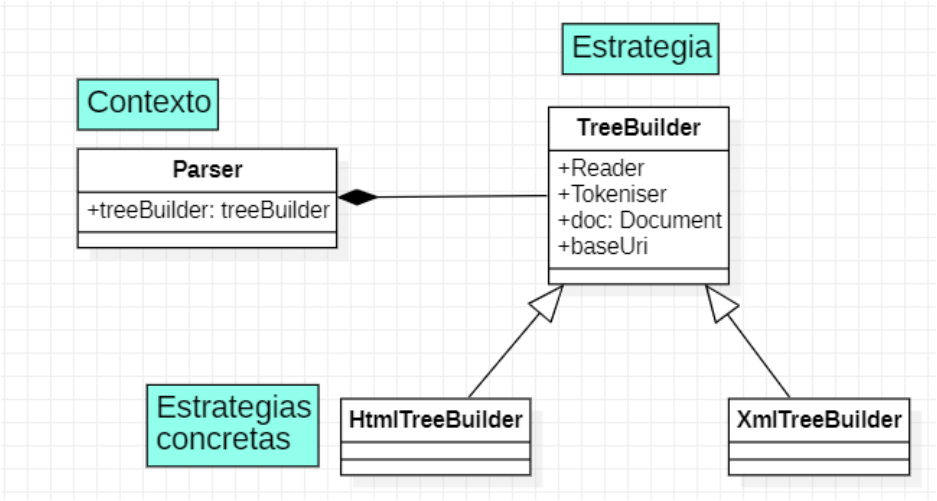

Figura 2. Fragmento de interés: implementación del patrón de diseño Estrategia en jsoup.

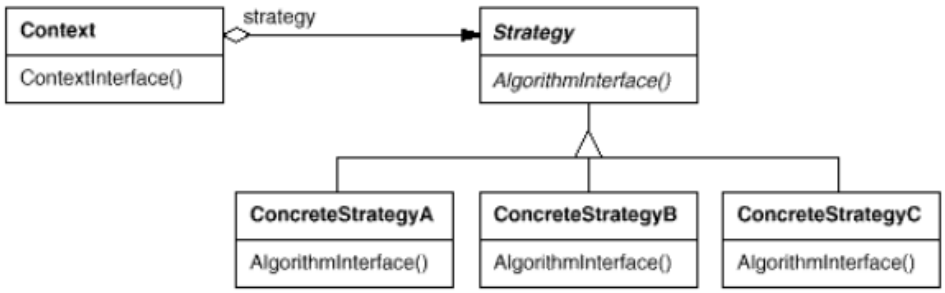

Figura 3. Estructura del patrón Estrategia [1].

Información General sobre el Patrón "Estrategia":

El patrón de diseño "Estrategia" pertenece a la categoría de comportamiento según el libro de GoF. Su función principal radica en caracterizar las interacciones y la distribución de responsabilidades entre diferentes clases y objetos. En el contexto de jsoup, este patrón se emplea para intercambiar algoritmos de construcción del árbol que representa el documento HTML, brindando flexibilidad y extensibilidad al proceso de análisis.

Información del Patrón Aplicado al Proyecto:

El patrón "Estrategia" se aplica en el proyecto jsoup para permitir el intercambio dinámico de algoritmos de construcción del árbol. La clase Parser actúa como el Contexto, mientras que diferentes implementaciones de TreeBuilder sirven como Estrategias concretas. La clase principal, Jsoup, utiliza este enfoque para analizar documentos HTML y construir un árbol de nodos, permitiendo la adaptabilidad a diferentes tipos de documentos y la simplificación del código.

Ventajas y Desventajas de la Implementación:

La ejecución del patrón Estrategia en jsoup conlleva múltiples ventajas, como la capacidad de realizar análisis independientemente del tipo de documento, la flexibilidad para intercambiar algoritmos y la simplificación del código, especialmente en la gestión de diferentes tipos de documentos o de lectura. No obstante, se presentan desafíos, como la restricción a estructuras de árbol, que, aunque eficiente, puede limitar la diversidad de documentos que jsoup puede manejar. Además, la posibilidad de añadir una capa adicional de delegación plantea interrogantes sobre el tiempo de ejecución y la eficiencia operativa.

Alternativas de Solución:

Algunas alternativas para abordar los desafíos identificados incluyen la creación de dos tipos de Parser diferentes o la inclusión de condicionales para gestionar diversos tipos de documentos. Sin embargo, estas opciones se descartaron a favor de la implementación del patrón de Estrategia debido a su capacidad para manejar cambios en los algoritmos de manera más eficiente y sin la necesidad de extender excesivamente el código. Se sugiere, sin embargo, la continua exploración de enfoques adicionales para mitigar las desventajas identificadas.

Conclusión:

En conclusión, esta investigación ofrece una visión detallada de la aplicación del patrón de Estrategia en jsoup, abordando los desafíos específicos que enfrenta y sugiriendo perspectivas para futuras investigaciones. La elección estratégica de este patrón en el marco del análisis de HTML ha demostrado ser efectiva, brindando una estructura robusta y adaptable para el desarrollo de la librería. La optimización continua y la exploración de nuevas posibilidades son esenciales para mantener la relevancia y eficacia de jsoup en un entorno tecnológico en constante evolución.

Referencias:

[1]E. Gamma, R. Helm, R. Johnson, J. Vlissides, y G. Booch, *Design Patterns: Elements of Reusable Object-Oriented Software*, 1.ª ed. Addison-Wesley Professional, 1994.

[2]J. Hedley, «jsoup». [En línea]. Disponible en: https://github.com/jhy/jsoup