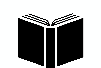
**Design Patterns Term Project**

# JSoup 기능 확장 및 설계 개선

2016 김지민

20172151 이성민

2016 하태윤

**JSoup 개요**

**Jsoup: Java HTML Parser**

JSoup은 HTML 문서에 저장된 데이터를 구문 분석, 추출 및 조작하도록 설계된 오픈 소스 Java 라이브러리다. Jsoup은 다음과 같은 기능을 한다.

1. URL, 파일, 문자열을 소스로 하여 HTML을 파싱한다.
2. DOM 구조를 추적하거나 CSS 선택자를 사용하여 데이터를 찾아 추출해낸다.
3. 문서 내의 HTML 요소, 속성, 텍스트를 조작할 수 있다.
4. 사용자가 입력한 데이터로부터 XSS 공격을 방지하기 위해서 안전한 화이트 리스트 방식으로 지정된 태그만 남기고 나머지는 제거할 수 있다.
5. 깔끔한 형태의 HTML을 출력할 수 있다.

**Open source**

JSoup은 MIT 라이선스에 따라 배포되는 오픈소스 라이브러리다.

**JSoup 사용법**

JSoup을 사용하기 위해서는 <http://jsoup.org/download>에서 다운로드 받은 후 해당 자바 프로젝트에 jar 파일을 추가해야한다. Jsoup을 사용할 파일에 org.jsoup.Jsoup, org.jsoup.nodes.Document, org.jsoup.select.Elements 등 필요한 패키지를 import 해준다.

****

**JSoup 설계 및 구현 조사**

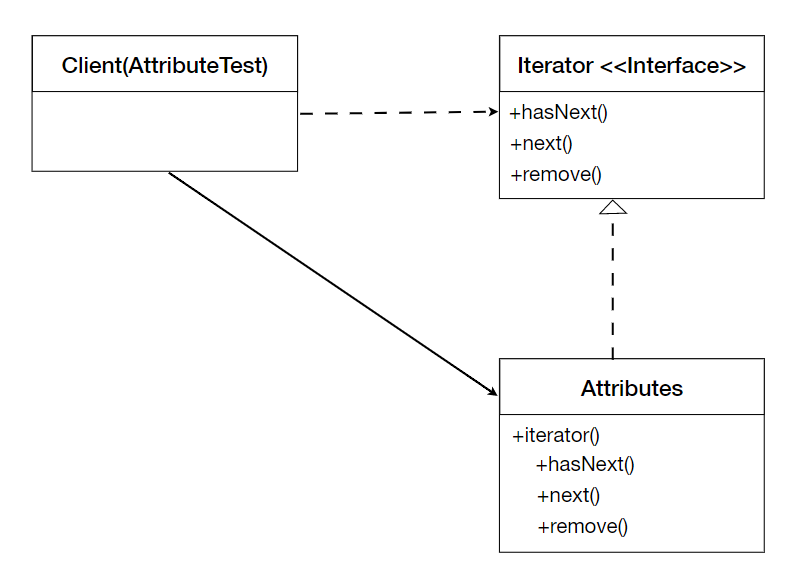
**설계 Overview**

//////////////////////////////////////////////////////////

**적용되어 있는 설계패턴 소개**

**- Iterator Pattern: 자료구조에 상관없이 객체 접근 방식을 통일시키고자 할 때 사용된다.**

1)

****

판단근거1: Attributes class가 Iterator interface를 implement하여 hasNext(), next(), remove()를 override한다.

판단근거2: Attributes class의 iterator()는 iterator의 method들을 override하는 동시에 create한다.

판단근거3: AttributeTest class에서 Attribute 객체를 사용하여 implement한 iterator를 사용한다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명2)

스크린샷이(가) 표시된 사진

자동 생성된 설명판단근거1: DatasetIterator class에서 iterator interface를 implement하여 hasNext(), next(), remove()를 override한다.

판단근거2: EntrySet class의 iterator()에서 DatasetIterator 객체를 생성한다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

**- Façade Pattern: 단순화된 인터페이스를 통해 서브시스템을 더 쉽게 사용할 수 있도록 하기위한 용도로 쓰인다.**

스크린샷이(가) 표시된 사진

자동 생성된 설명

판단근거1: Parser class는 Treebuilder, ParseErrorList, ParseSettings 객체를 선언하여 사용한다.

판단근거2: Connection interface는 Parser 객체를 선언하여 parser의 기능을 간편하게 사용한다.

스크린샷이(가) 표시된 사진

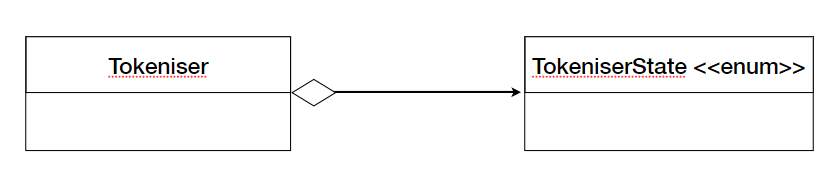
자동 생성된 설명

**스크린샷이(가) 표시된 사진

자동 생성된 설명**

**- State Pattern: 객체가 특정 상태에 따라 행위를 달리하는 상황에서 상태를 객체화하여 상태가 행동할 수 있도록 위임하는 패턴이다.**

1)

****

판단근거1: TokeniserState에 Tokeniser의 state들을 enum으로 정의하고 state마다 read()를 정의하였다.

판단근거2: Tokeniser class에서 TokeniserState 객체를 선언하여 상태가 변할 때마다 각 state의 read()를 사용한다.

판단근거3: HTMLTreeBuilder class에서 tagName에 따라 Tokeniser의 state를 변환시킨다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷, 모니터, 화면, 검은색이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

2)

스크린샷이(가) 표시된 사진

자동 생성된 설명판단근거1: HTMLTreeBuilderState에 HTMLTreeBuilder의 state들을 enum으로 정의하고 state마다 함수를 정의하였다.

판단근거2: HTMLTreeBuilder class에서 HTMLTreeBuilderState 객체를 선언하여 노드의 이름에 따라HTMLTreeBuilderState의 state를 변화시켜준다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

**- Adapter Pattern: 호환성이 없는 인터페이스 때문에 함께 동작할 수 없는 클래스들이 함께 작동하도록 해준다.**

스크린샷이(가) 표시된 사진

자동 생성된 설명

판단근거1: DatasetIterator class는 Iterator<E> interface를 implement한다.

판단근거2: DatasetIterator class는 Attribute 객체를 선언하여 override한 hasNext(), next(), remove()에서 Attribute의 함수 getKey(), isDataAttribute()를 사용한다.

스크린샷이(가) 표시된 사진

자동 생성된 설명스크린샷이(가) 표시된 사진

자동 생성된 설명

**팀이 수행한 기능 확장과 설계 개선**

**확장된 기능**

//////////////////////////

**설계 개선 내용**

설계 1: Visitor Pattern

/////////////////////////////

설계 2: Factory Pattern

///////////////////////////////

설계 3: Builder Pattern

사람, 하늘이(가) 표시된 사진

자동 생성된 설명

개체이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

Element class를 상속받은 FormElement, Document, PseudoTextElement와 LeafNode를 상속받은 TextNode, CDataNode, XmlDeclaration, DataNode, Comment, DocumentType에 대한 각각의 builder class를 생성하였다. 각각의 builder class는 ElementBuilder, LeafNodeBuilder를 상속받아 abstract method들을 구현한다. Builder class들은 constructor들의 parameter를 product로 생성하며 client인 MakeElement, MakeLeafNode class는 director인 ElementDirector와 LeafNodeDirect를 통해 product에 접근한다.

**상세한 변경 내용 설명 및 기존 설계/코드와의 비교**

- 설계 1

//////////////////////////////////

- 설계 2

//////////////////////////////////

- 설계 3

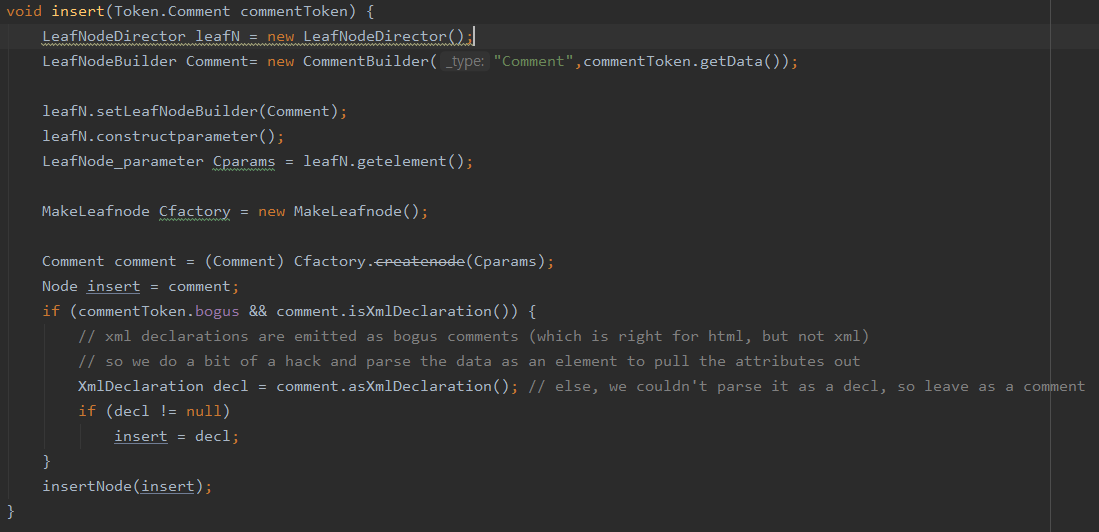
스크린샷, 모니터, 검은색이(가) 표시된 사진

자동 생성된 설명

변경된 HTMLTreeBuilder.java



변경된 XMLTreeBuilder.java



LeafNode\_parameter class



CommentBuilder class



LeafNodeBuilder와 ElementBuilder는 abstract class로 각각 자식 클래스에서 구현해야 할 method들을 abstract로 선언하였다. LeafNode\_paratmeter와 Element\_parameter class는 parameter를 상속받아 매개변수로 사용되는 변수들을 선언하고, 각 변수에 대한 get(), set()함수를 구현하였다. CommentBuilder 등의 9개 Builder class들은 각각의 부모 클래스의 builder 클래스를 상속받으며, 생성자의 매개변수 종류에 따라 parameter를 사용해 변수의 값을 셋팅한다.

기존의 HTMLTreeBuilder class와 XMLTreeBuilder class에서는 클래스 내에서 직접 Comment 객체를 생성하였다. 그러나 Builder Pattern을 적용한 후에는 builder와 director를 통해 comment 객체를 생성한다.

**도입된 설계 패턴 및 설계 원칙과 적용 이유**

- 설계 1

//////////////////////////////////////

- 설계 2

////////////////////////////////////////

- 설계 3

설계 3에는 Builder Pattern을 도입하였다. Builder Pattern은 Factory Pattern을 적용시키면서 클래스의 constructor가 오버로드 되어 있는 경우 factory에서 매개변수에 따라 구분하여 객체를 생성할 수 없는 상황을 해결하기 위해 적용하였다.

****

**테스트 내역**

****

**GitHub 프로젝트 활용 요약**

**팀의 github 프로젝트 주소**

https://github.com/hataeyunn/designpattern.git

**프로젝트 progress history 스크린샷**

**팀원별 기여를 잘 나타낼 수 있는 각종 자료**