Design Pattern Term Project

Team#15

정기민

20135286 조건상

20125148 황현제

1. **개요**

* **Jsoup 소개**

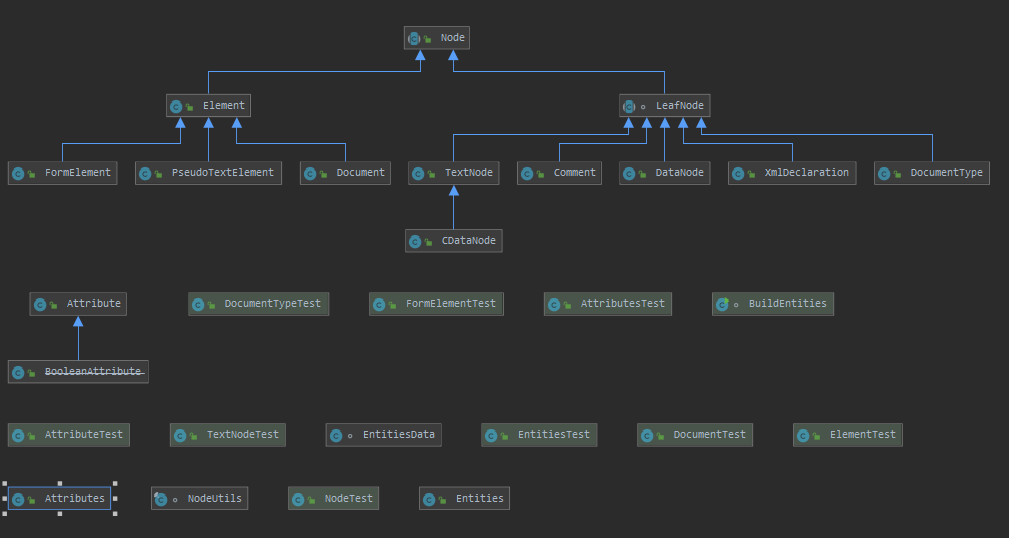
HTML을 작업할 수 있는 자바 라이브러리로써

DOM traversal 나 CSS selector를 이용하여 정보 찾기, 추출 가능

URL, 파일, 문자열로부터 HTML 파싱, 스크랩 가능

HTML 요소, 특성, 텍스트 조작 가능

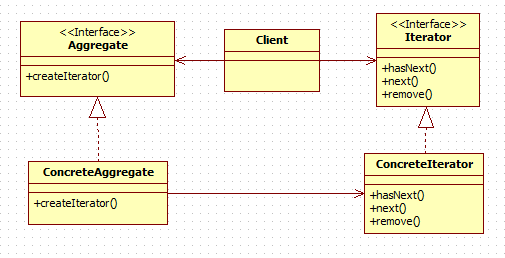
1. **Jsoup 설계 및 구현 조사**

* **설계 overview 조사**

Jsoup에 포함 되어있는 각 클래스들의 기능 및 사용 방법에 대해서 조사하고, 어떤 설계 패턴이 적용되어 있는지 조사하였다.

* **이미 적용되어 있는 설계패턴 조사**

1) Iterator Pattern



Aggregate: Iterable 인터페이스

Iterator: java.util.Iterator 인터페이스

ConcreteAggregate: Attributes 클래스

Attribute 클래스에 java.util.Iterator을 활용한 Iterator 패턴이 있다. Attribute 클래스에 Iterable 클래스를 implements 하고 ConcreteIterator를 생성하는 iterator() 메소드가 구현 되어있다. iterator() 메소드에는 iterator 패턴을 구현하기 위해 다음 데이터가 있는지 체크하는 hasNext() 메소드와 다음 데이터를 가리키는 next() 메소드, 데이터를 삭제하는 remove() 메소드가 구현 되어있다.

2) Fasad Pattern

-설명

1. **팀이 수행한 기능 확장과 설계 개선**

* **확장된 기능 요약**

1) 사용자가 하나의 클래스를 이용하여 원하는 태그의 정보를 추출할 수 있도록 한다

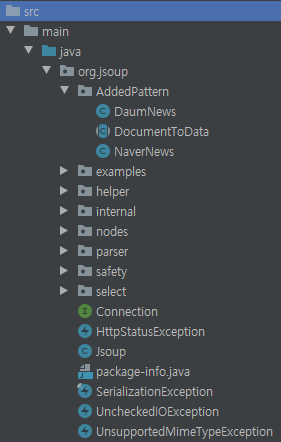
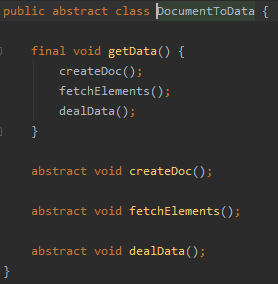
2) 태그 찾기: elements 클래스에 getAllTags() 메소드를 추가해 elements에 존재하는 모든 태그를 중복을 제외하여 ArrayList로 반환하는 메소드를 구현하였다.

3) GUI를 이용하여 쉽게 정보를 확인할 수 있다.

* **설계 개선 내용 요약**

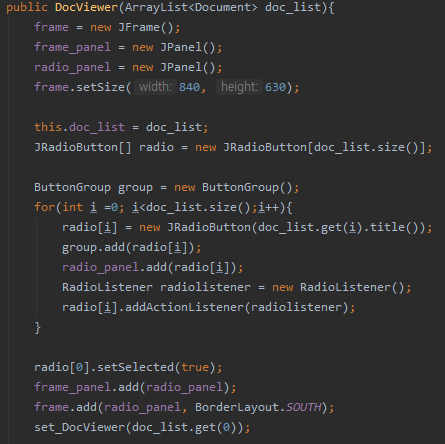
1) Template Method Pattern – 사용자 입장에서 파싱을 할 때 여러 함수를 이용해야 하는데, 과정이 비슷하다. 따라서 하나의 abstract 클래스로 파싱 하는 법을 정의하고, 사용자는 서브클래스를 재정의 하여 문서를 파싱 할 수 있도록 한다.

2) MVC Pattern – 여러 개의 문서(클래스 오브젝트)를 하나의 GUI 뷰를 통해서 정보를 확인할 수 있게 한다. 사용자가 클릭하는 것에 따라서 다른 정보를 표현한다.

* **상세한 변경 내용 설명 및 기존 설계/코드와의 비교**

1)

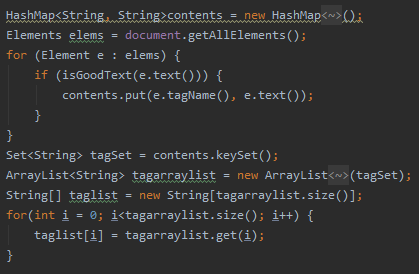
새롭게 패턴을 추가하기 위해서 Abstract 클래스를 추가하고, 사용자가 그것을 바탕으로 자신에 맞게 수정할 수 있도록 하였다. 사용자가 정의하는 각 concrete 클래스에서는 목적에 맞게 사용자가 정해진 함수들을 구현하여 전체적인 구조를 몰라도 어플리케이션 활용에 문제가 없도록 한다.

2) GUI 설명

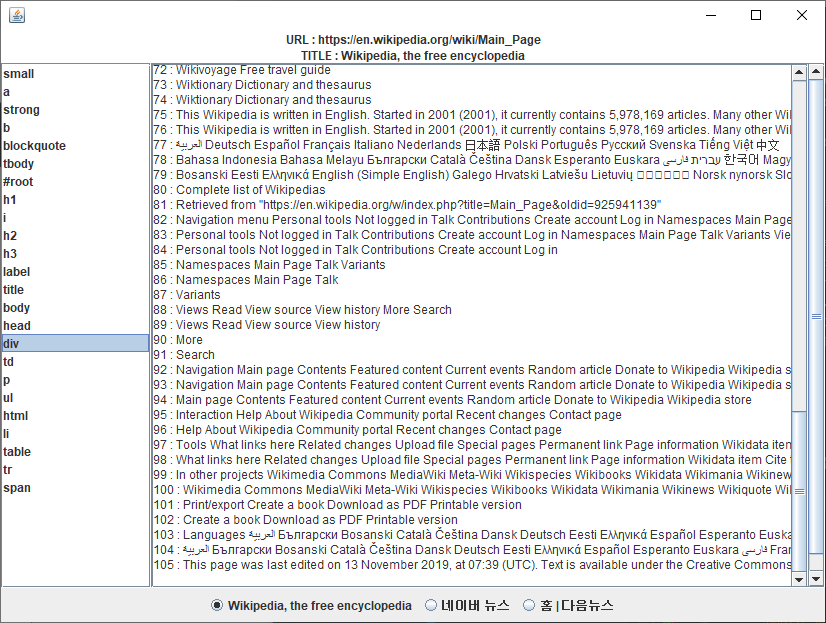
문서를 파싱 하더라도, 어떤 태그를 이용해야 어떤 내용을 추출할 수 있을지 모를 사용자를 위해 GUI로 기능을 제공한다.

주어진 문서 객체를 넘겨주게 되면 그 문서가 포함하고 있는 HTML 태그들을 검색하여 가져오게 된다. 그 후 그 태그 안에 무엇이 들어 있는지를 사용자에게 표시해준다.

여러 개의 문서를 다루는 경우, 하단에 부착되는 라디오 버튼을 통해서 서로 다른 문서에 대한 정보를 독립적으로 확인하는 것이 가능하다.



문서에 포함된 태그를 찾는 코드이다. 태그를 찾아오되, 텍스트로 표현되는 정보를 가지고 있는 태그만을 가져와서 사용자에게 표시해준다.



사용자는 원하는 문서에 해당하는 버튼을 클릭하여 문서에 포함된 태그들과 그 태그들이 포함하고 있는 텍스트를 시각적으로 확인할 수 있다.

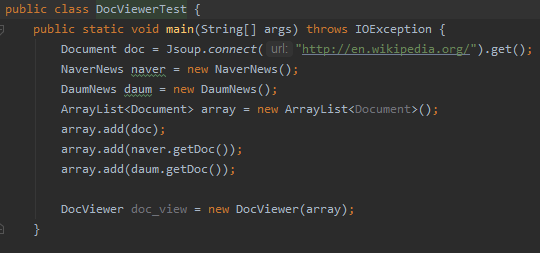
* **도입된 설계 패턴 및 설계 원칙과 적용 이유**

1) 개발자 입장에서 보다는, 이 응용프로그램을 사용하는 사용자 입장에서 설계 패턴을 도입하였다. Jsoup에 대한 사전 지식이 없는 사람이 응용프로그램을 사용한다고 가정하고, 그 사람이 최대한 클래스 구조나 내용에 대해서 몰라도 프로그램을 실행하는데 문제가 없도록 하기위해서, 또한 기본적인 기능을 사용할 수 있도록 하기 위해서 설계 패턴을 도입하였다.

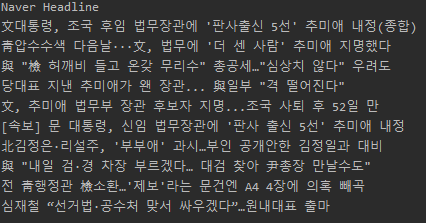
2) Template Method Pattern – 어느 문서를 파싱 하던 간에 1. Connection 클래스를 이용하여 연결한다. 2.원하는 elements를 태그를 이용하여 가져온다. 3.가져온 elements를 적절히 사용한다. 이 3단계로 문서를 파싱 하게 된다. 사용자 입장에서는 문서를 파싱 하기 위해서 여러가지 클래스에 대해 알아야 하고, 이는 사용자의 부담이 커지게 된다. 따라서 고정된 템플릿을 제공함으로써 사용자가 각 클래스를 알게 하는 부담을 줄이고, 원하는 대로 수정을 해서 자신에게 맞게 사용할 수 있게 하였다.

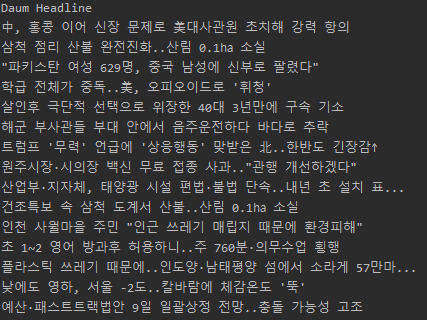
3) MVC Pattern – 문서 객체를 만들어서 문서에 있는 정보를 확인하고 싶은데, 어떤 태그에 어떤 내용이 담겨 있는지 확인할 수 없는 사용자들을 위하여 문서 객체(Model)가 포함하고 있는 정보를 GUI(View)를 통해서 보여준다. 사용자는 화면을 클릭하면서 자신이 보고 싶은 정보를 표시할 수 있게 한다.

1. **테스트 수행 내역**

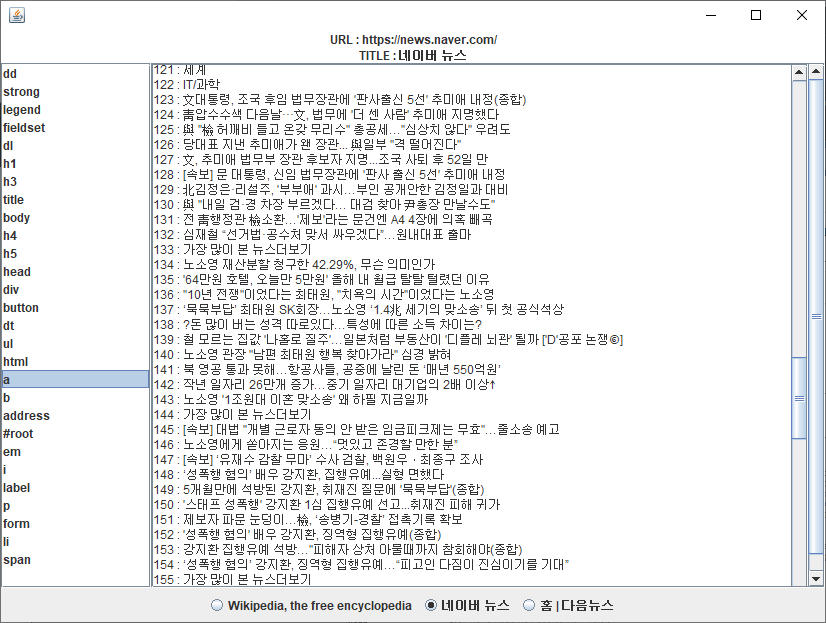
* **테스트 케이스**

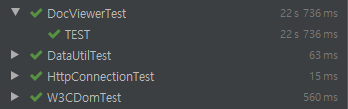
기존에 있던 Test 폴더에 DocViewerTest 클래스를 추가하고, Connection 클래스의 메소드를 이용하여 문서를 만들었다. 그 이후 우리가 만든 새로운 클래스를 통해 문서를 더 추가했다. 그리고 미리 정의된 함수들이 잘 작동하는지 확인하고, 문서들을 리스트로 묶어서 GUI를 생성한 뒤 의도대로 잘 작동하는지 테스트하였다.

* **결과**

NaverNews 클래스를 이용하여 헤드라인 기사를 뽑아낸 경우

DaumNews 클래스를 이용하여 헤드라인 기사를 뽑아낸 경우

GUI를 활용하여 웹페이지에 포함된 태그 정보를 확인하는 경우

Junit 테스트를 실행한 결과

1. **GitHub 프로젝트 활용 요약**

* github 주소: https://github.com/HyeonJe/DesignPattern\_Project
* Progress History & 팀원 별 기여

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1주차(11.8~14) | 2주차(11.15~21) | 3주차(11.22~28) | 4주차(11.29~12.6) |
| 정기민 | Jsoup 코드 분석 및 패턴 연구 |  |  |  |
| 조건상 | 설계 패턴분석(Iterator Pattern) | 중간 통합 테스트 진행 / | 최종 통합 테스트 |
| 황현제 | Template Method Pattern 적용 | 중간 통합 테스트 진행 | GUI에 Mediator 패턴 적용 / 최종 통합 테스트 |

(git history 별첨)