**REPORT**

**-Jsoup Project (6 team)**

**과 목 명** : 설계패턴

**담당교수님** : 이찬근

**이 름** :20165417 김소연

20153280 유승곤

20161864 이서라

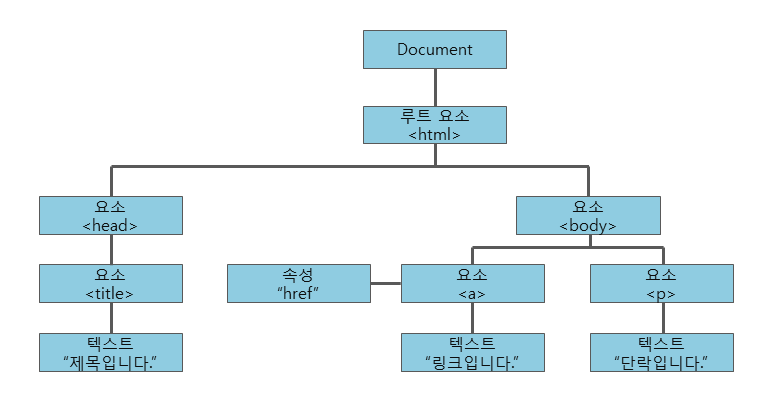
20154686 채훈기

**제 출 일** : 2019.11.??

1. **Jsoup 개요**

Jsoup : HTML 문서에 저장된 데이터를 구문 분석, 추출 및 조작하도록 설계된 오픈 소스 Java 라이브러리.

Jsoup은 DOM 방식으로 웹페이지를 파싱 & 트리 형식으로 처리



DOM

-Document Object Model(문서 객체 모델)

API를 이용하여 특정 element에 접근 후 정보 읽기, 수정 가능

웹사이트 내용을 파싱할 때 사용, 원하는 정보만 추출하는 것이 가능

Jsoup의 주요 클래스

**Document : Jsoup으로 얻어온 결과. html 전체 문서**

**Element : Document의 html 요소**

**Elements : element가 모인 자료형**

**Connection : Jsoup의 connect method를 이용한 객체**

**Response : Jsoup이 URL에 접속해 얻어온 결과**

**1. url받아서 html코드 받아오기**

**2. 받은 html 파싱해서 DOM tree 만들기**

**3. select query문 파싱**

**4. DOM tree 순회하면서 원하는거 뽑아내기**

1. **Jsoup 설계 구현 및 조사**

* **설계 overview**
* **이미 적용되어 있는 설계패턴 소개**

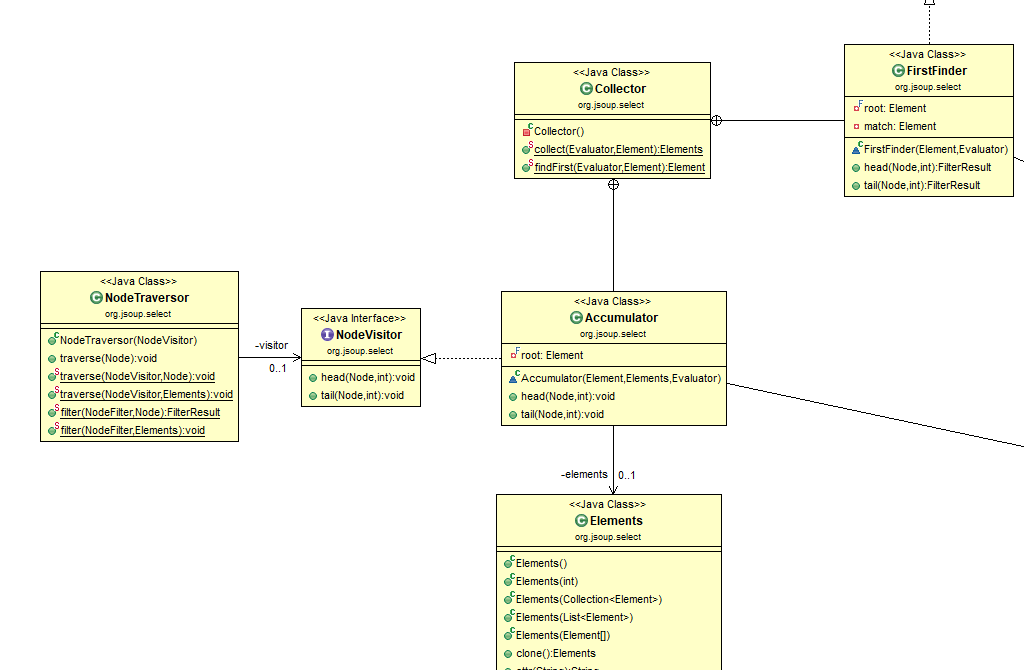
**(설계패턴의 이름, 패턴에 대한 클래스 다이어그램(전체적으로 할 필요는 없음) 판단 근거, 소스코드와 매핑 등)**

**클래스 다이어그램 그리기 :** <https://niceman.tistory.com/134> 참고!

**helper, node-composite, parser-builder, select-visitor, jsoup.java-façade?**

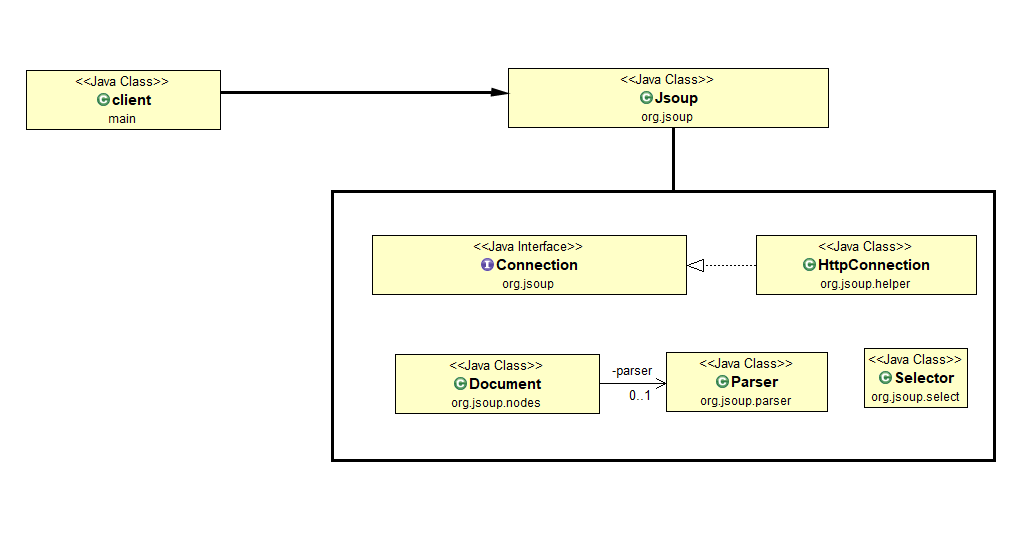
**org.jsoup.select**

**visitor pattern :** This interface provides two methods, {@code head} and {@code tail}. The head method is called when the node is first seen, and the tail method when all of the node's children have been visited.

****

**org.jsoup**

**façade pattern :** first, client use jsoup class.In jsoup class, connect method does a lot of work. It sends GET and POST methods to browser and parse the html codes



1. **Jsoup 기능 확장(2명) 및 설계 개선(2명)**

**DOM tree 구조를 시각화해서 보여주기**

1. **테스트 수행 내역**

* **Test case (Junit test code)**
* **Test case screenshot**

1. **GitHub 프로젝트 활용**

* 팀의 Github 프로젝트 주소
* 프로젝트 progress history 스크린샷
* 팀원별 기여를 잘 나타낼 수 있는 각종 자료

이미 적용되어 있는 설계패턴

: composite / Iterator / Decorator pattern 적용

jsoup의 주요 클래스

* Document : jsoup으로 얻어온 결과. html 전체 문서
* Element : Document의 html 요소
* Elements : element가 모인 자료형
* Connection : jsoup의 connect method를 이용한 객체
* Response : jsoup이 URL에 접속해 얻어온 결과

jsoup의 기본 기능 순서

* 가져올 html url을 지정
* Connect 객체를 통해 URL에 접속
* Document 객체에 페이지의 전체 소스를 저장
* Document의 Element들을 parsing
* parsing한 요소 중 원하는 TAG(ex. title, id, image, attribute 등) 에 따라 값 불러오기

이때 불러오는 방식은 Iterator를 사용해서 값 하나씩 hasnext() 메소드를 이용해서 하나씩 가져오거나 html의 속성(<a>, <p> 등)을 한꺼번에 불러올 수 있음.

기능확장 -> 웹크롤링할 때, 원하는 태그값들을 모두 불러올 수 있으나, 그 중 필요한 값만 설정하여 불러올 수 있게

ex) 영화차트를 볼 수 있는 html 페이지에서 장르에 따라 리스트에서 제거하거나 추가할 수 있는 기능?

쇼핑몰에서 보고싶은 요소를 위해 리스트에서 제거/추가/분류