Design Pattern

[Team 6]

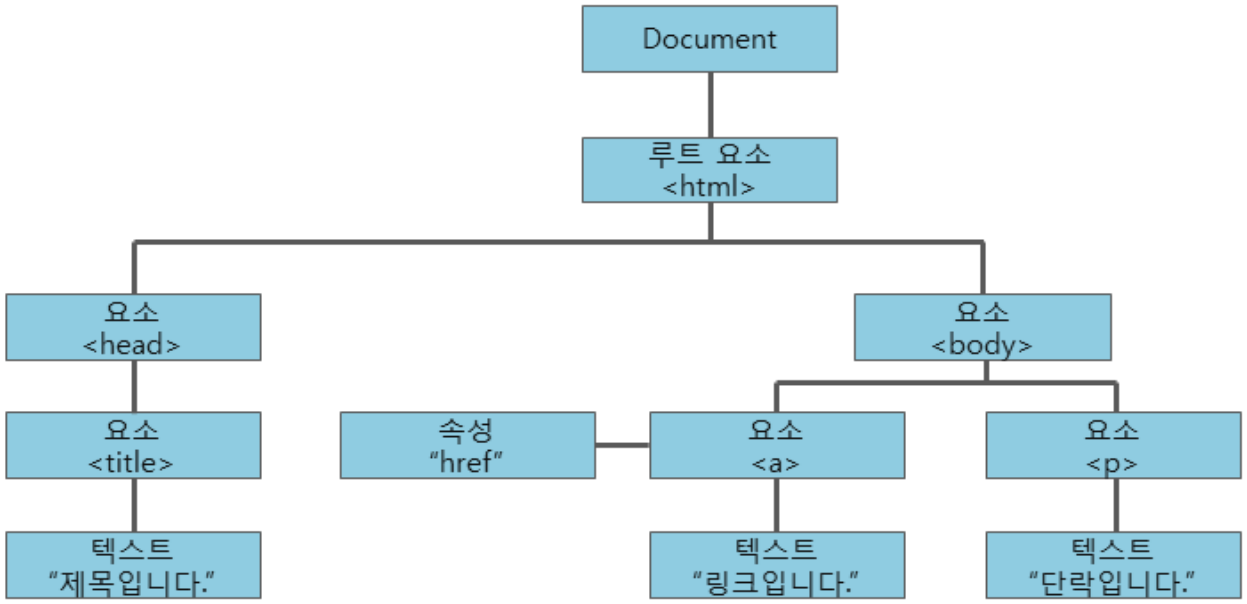
20165417 김소연
20153280 유승곤
20161864 이서라
20154686 채훈기

CONTENTS

01 Jsoup 소개

Jsoup 개요

- HTML 문서에 저장된 데이터를 파싱, 추출 및 조작하도록 처리된 오픈소소 라이브러리
- Jsoup은 웹페이지를 파싱한 후 DOM 트리 형식으로 저장



-〉 DOM(Document Object Model)
: 문서 객체 모델)

01 Jsoup 소개

Jsoup 주요 기능

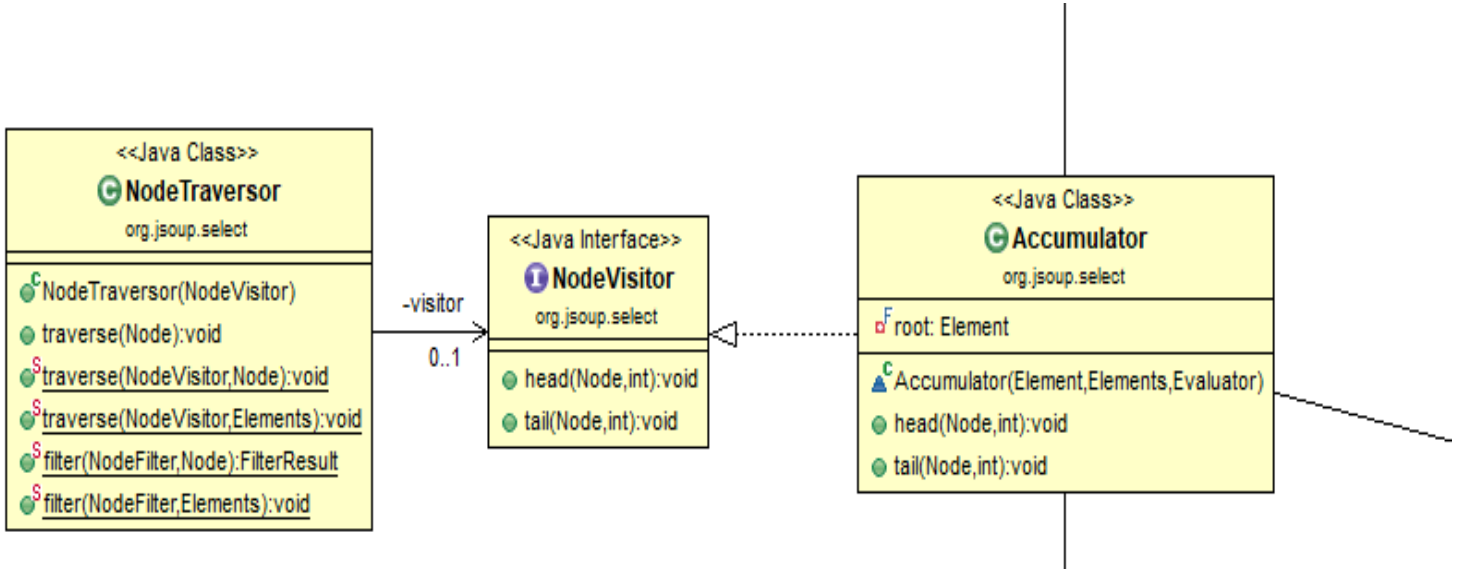
- URL에 접속해 HTML 코드를 받아온 후 해당 HTML코드를 Parsing하여 DOM tree로 만든다.

- Select Query문을 Parsing하여 사용자가 원하는 요소를 확인한다.

- DOM tree를 순회하면서 사용자가 원하는 요소인지 판별하고 출력한다.

02 Jsoup 분석

1. org.jsoup.select - Visitor pattern

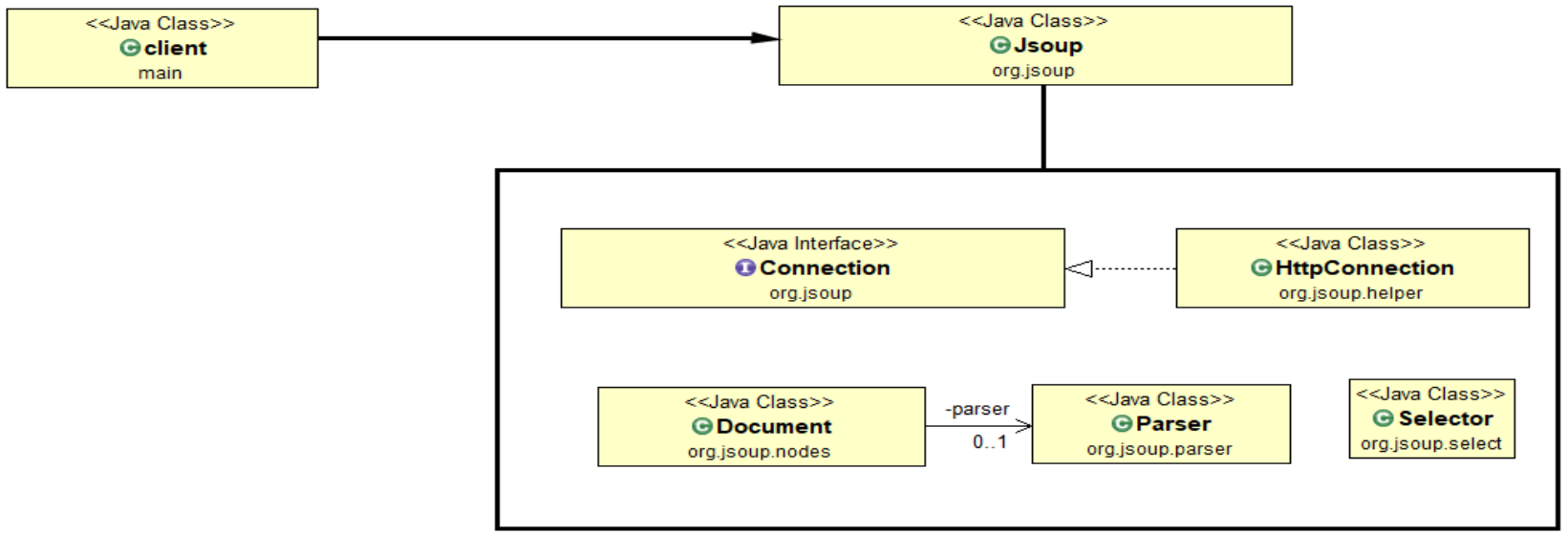


-〉 head tail Method

-〉 Tree를 순회할 경우 노드를 방문할 때 Head method 호출
노드를 나갈 때 Tail method 호출

02 Jsoup 분석

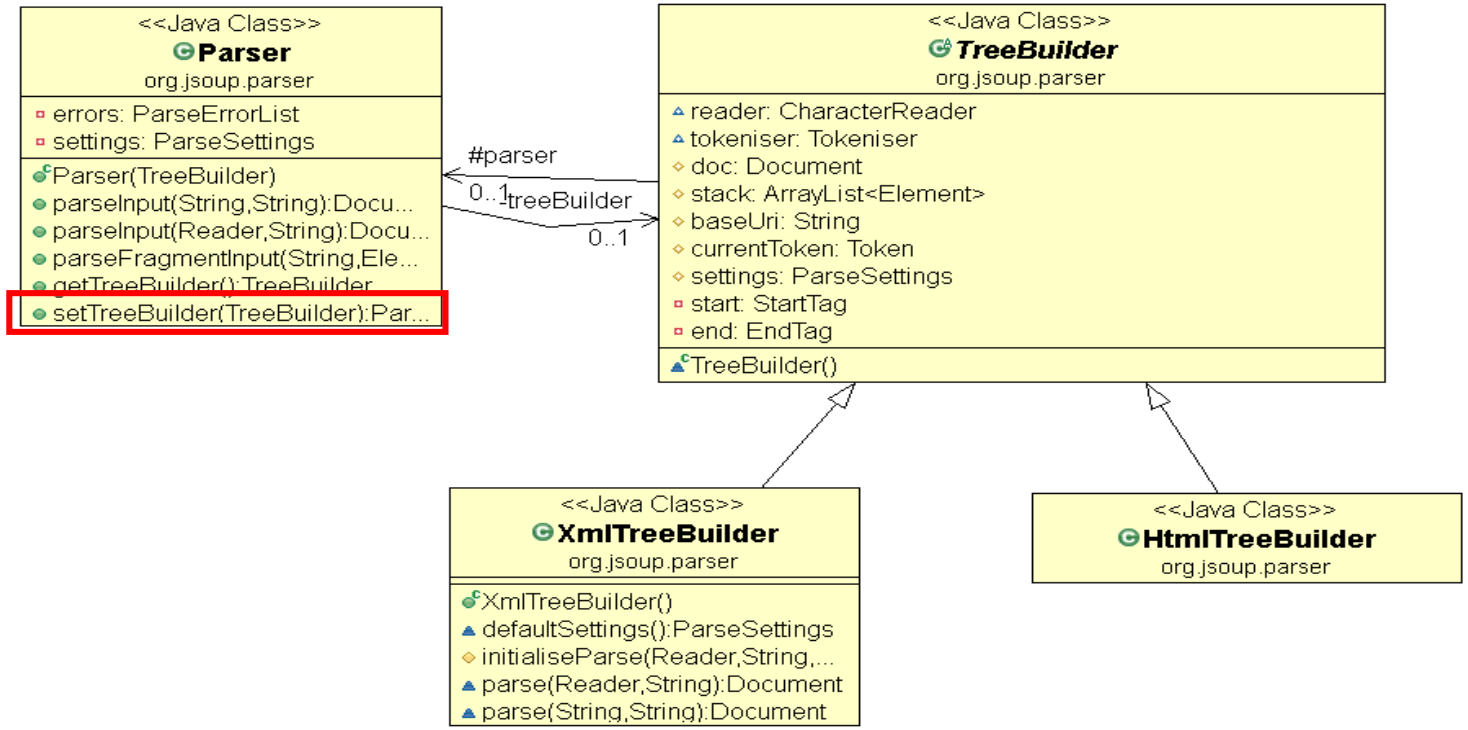
2. org.jsoup - Façade pattern



-〉 Jsoup 클래스는 사용자가 가장 먼저 접근하는 클래스.
Jsoup안에는 사용자가 원하는 복잡한 일련의 과정들이 하나의 메소드에 함축되어 있다.

02 Jsoup 분석

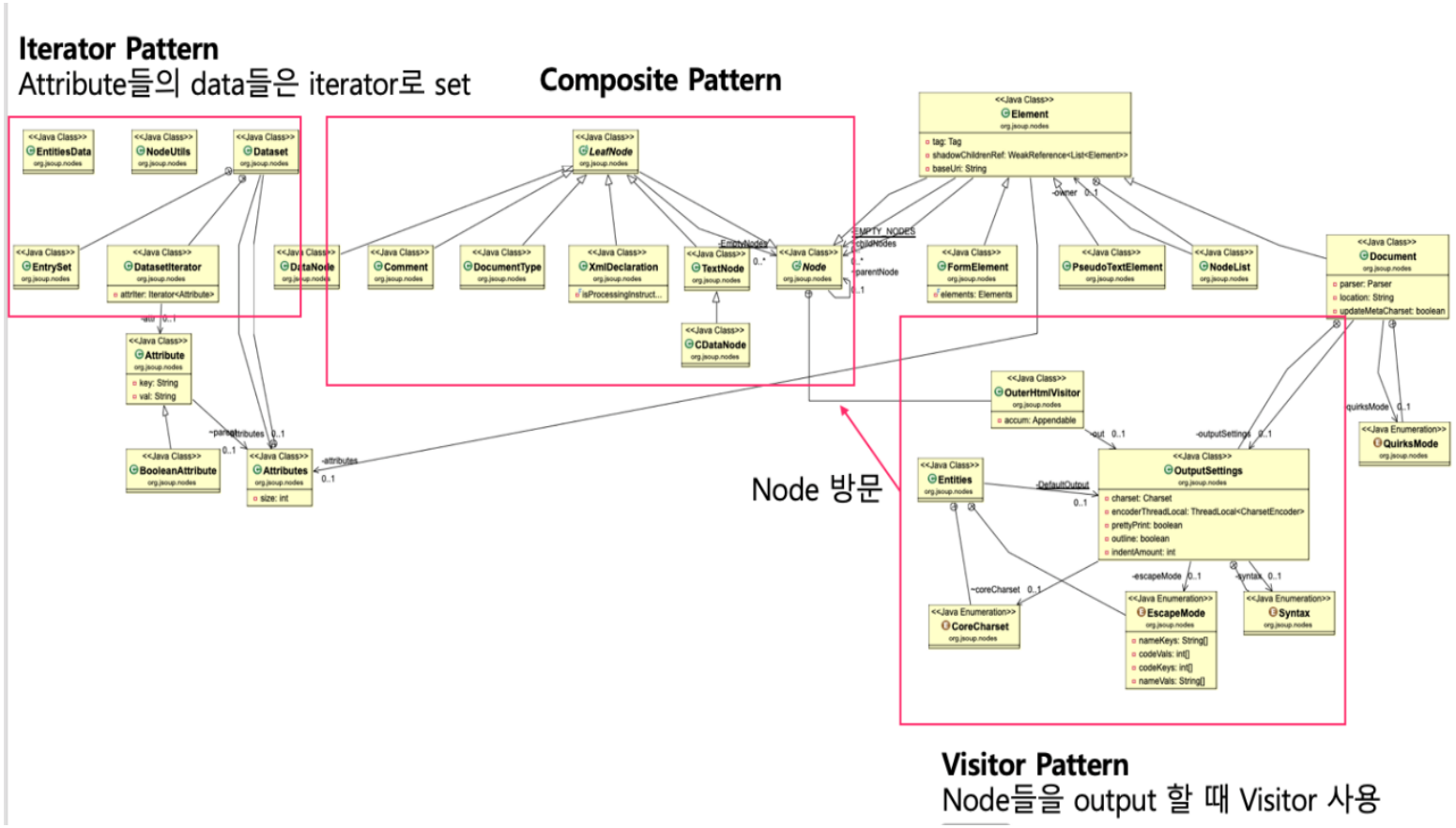
3. org.jsoup.parser - Builder pattern



-〉 Parser에서 Builder를 정의한다. Xml 또는 HTML파일을 Parsing할 때
tree builder를 이용하여 쉽게 객체를 만들어 낼 수 잇고 명확하게 표현할 수 있다.

02 Jsoup 분석

4. org.jsoup.node - Composite pattern



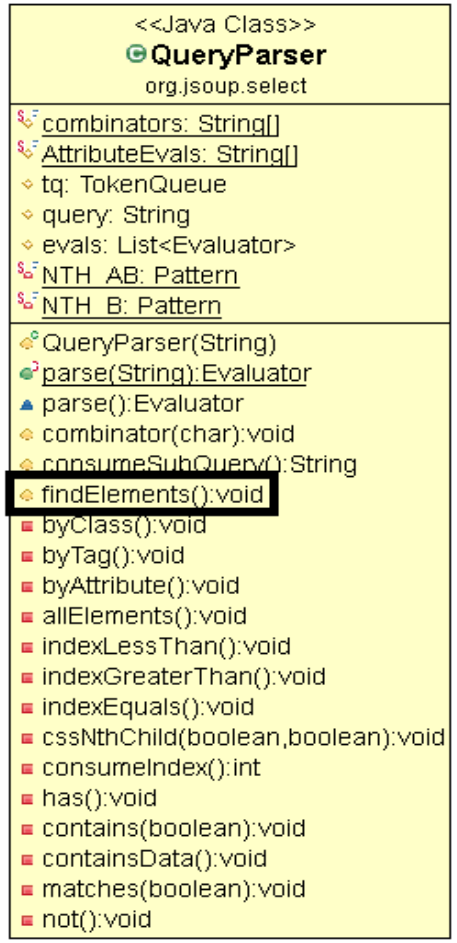
-〉 Node와 Elements로 구성되어 있는 클래스.

03 Jsoup 확장

〉설계 개선

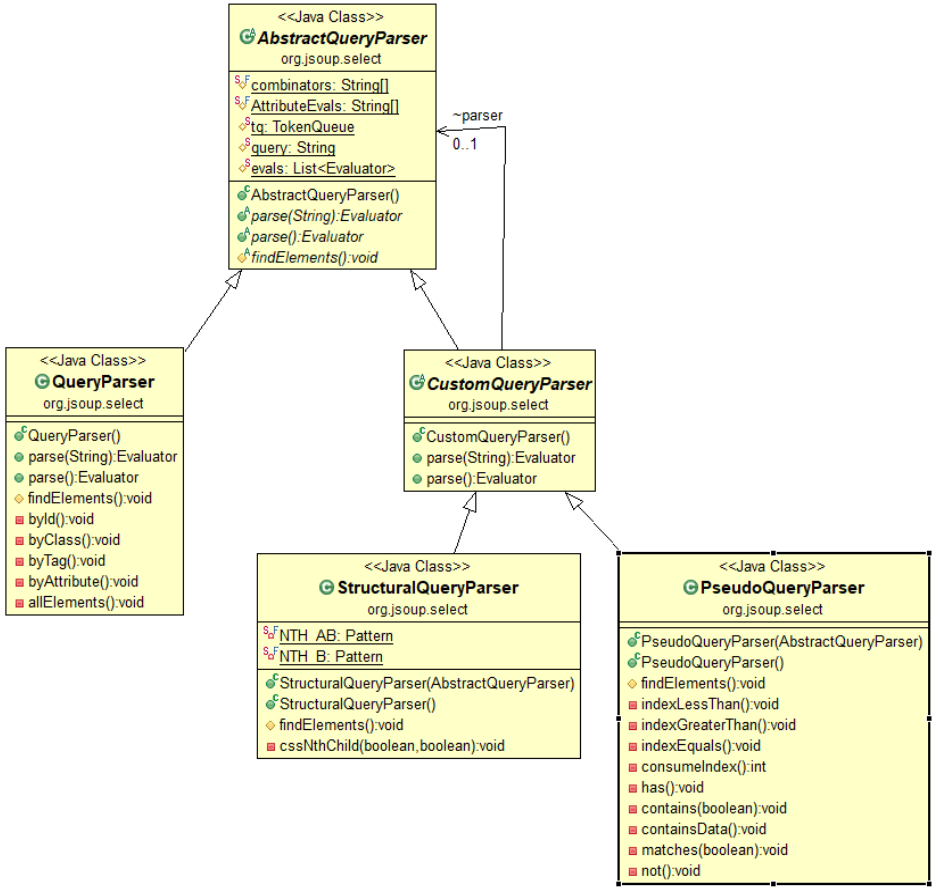
: QueryParser class를 Decorator Pattern을 이용하여 분리

03 Jsoup 확장




```
private void findElements() {
    if (tq.matchChomp("#"))
        byId();
    else if (tq.matchChomp("."))
        byClass();
    else if (tq.matchesWord() || tq.matches("*|"))
        byTag();
    else if (tq.matches("["))
        byAttribute();
    else if (tq.matchChomp("*"))
        allElements();
    else if (tq.matchChomp(":lt("))
        indexLessThan();
    else if (tq.matchChomp(":gt("))
        indexGreaterThan();
    else if (tq.matchChomp(":eq("))
        indexEquals();
    else if (tq.matches(":has("))
        has();
    else if (tq.matches(":contains("))
        contains(false);
    else if (tq.matches(":containsOwn("))
        contains(true);
    else if (tq.matches(":containsData("))
        containsData();
    else if (tq.matches(":matches("))
        matches(false);
    else if (tq.matches(":matchesOwn("))
        matches(true);
    else if (tq.matches(":not("))
        not();
    else if (tq.matchChomp(":nth-child("))
        cssNthChild(false, false);
    else if (tq.matchChomp(":nth-last-child("))
        cssNthChild(true, false);
    else if (tq.matchChomp(":nth-of-type("))
        cssNthChild(false, true);
    else if (tq.matchChomp(":nth-last-of-type("))
        cssNthChild(true, true);
    else if (tq.matchChomp(":first-child"))
        evals.add(new Evaluator.IsFirstChild());
    else if (tq.matchChomp(":last-child"))
        evals.add(new Evaluator.IsLastChild());
    else if (tq.matchChomp(":first-of-type"))
        evals.add(new Evaluator.IsFirstOfType());
    else if (tq.matchChomp(":last-of-type"))
        evals.add(new Evaluator.IsLastOfType());
```

03 Jsoup 확장




```
protected void findElements() {

    if (tq.matchChomp("#"))
        byId();
    else if (tq.matchChomp("."))
        byClass();
    else if (tq.matchesWord() || tq.matches("*|"))
        byTag();
    else if (tq.matches("["))
        byAttribute();
    else if (tq.matchChomp("*"))
        allElements();
    else // unhandled
        throw new Selector.SelectorParseException(

}
```

03 Jsoup 확장

```
protected void findElements() {

    if (tq.matchChomp("#"))
        byId();
    else if (tq.matchChomp("."))
        byClass();
    else if (tq.matchesWord() || tq.matches("*|"))
        byTag();
    else if (tq.matches("["))
        byAttribute();
    else if (tq.matchChomp("*"))
        allElements();
    else // unhandled
        throw new Selector.SelectorParseException(

}
```

default

```
protected void findElements() {

    if (tq.matchChomp(":lt("))
        indexLessThan();
    else if (tq.matchChomp(":gt("))
        indexGreaterThan();
    else if (tq.matchChomp(":eq("))
        indexEquals();
    else if (tq.matches(":has("))
        has();
    else if (tq.matches(":contains("))
        contains(false);
    else if (tq.matches(":containsOwn("))
        contains(true);
    else if (tq.matches(":containsData("))
        containsData();
    else if (tq.matches(":matches("))
        matches(false);
    else if (tq.matches(":matchesOwn("))
        matches(true);
    else if (tq.matches(":not("))
        not();
    else if (tq.matchChomp(":matchText"))
        evals.add(new Evaluator.MatchText());
    else // unhandled
        parser.findElements();
}
```

Pseudo

```
protected void findElements() {
    if (tq.matchChomp(":nth-child("))
        cssNthChild(false, false);
    else if (tq.matchChomp(":nth-last-child("))
        cssNthChild(true, false);
    else if (tq.matchChomp(":nth-of-type("))
        cssNthChild(false, true);
    else if (tq.matchChomp(":nth-last-of-type("))
        cssNthChild(true, true);
    else if (tq.matchChomp(":first-child"))
        evals.add(new Evaluator.IsFirstChild());
    else if (tq.matchChomp(":last-child"))
        evals.add(new Evaluator.IsLastChild());
    else if (tq.matchChomp(":first-of-type"))
        evals.add(new Evaluator.IsFirstOfType());
    else if (tq.matchChomp(":last-of-type"))
        evals.add(new Evaluator.IsLastOfType());
    else if (tq.matchChomp(":only-child"))
        evals.add(new Evaluator.IsOnlyChild());
    else if (tq.matchChomp(":only-of-type"))
        evals.add(new Evaluator.IsOnlyOfType());
    else if (tq.matchChomp(":empty"))
        evals.add(new Evaluator.IsEmpty());
    else if (tq.matchChomp(":root"))
        evals.add(new Evaluator.IsRoot());
    else // unhandled
        parser.findElements();
`
```

structural

03 Jsoup 확장

```java
/**
 * Parse a CSS query into an Evaluator.
 * @param query CSS query
 * @return Evaluator
 */
public static Evaluator parse(String query) {
    try {
        QueryParser p = new QueryParser(query);
        return p.parse();
    } catch (IllegalArgumentException e) {
        throw new Selector.SelectorParseException(e.getMessage());
    }
}
```

기존

```java
/**
 * Parse a CSS query into an Evaluator.
 * @param query CSS query
 * @return Evaluator
 */
public Evaluator parse(String query) {
    this.query = query;
    this.tq = new TokenQueue(query);
    try {
        return parse();
    } catch (IllegalArgumentException e) {
        throw new Selector.SelectorParseException(e.getMessage());
    }
}
```

변경

03 Jsoup 확장

```
public class Jsoup {
    private Jsoup() {}

     Parse HTML into a Document. The parser will make a sensible, balanced document tree out of any HTML.
    public static Document parse(String html, String baseUri) {

    public static void setQueryParser(AbstractQueryParser parser) {
        Element.setQueryParser(parser);
        Elements.setQueryParser(parser);
        Selector.setQueryParser(parser);
    }

public static Connection connect(String url) {
     AbstractQueryParser dparser = new QueryParser();
     setQueryParser(dparser);
     return HttpConnection.connect(url);
}
```

03 Jsoup 확장

```
public abstract class AbstractQueryParser {

    protected final static String[] combinators = {",", ">", "+", "~", " "};
    protected static final String[] AttributeEvals = new String[]{"=", "!=", "^=", "$=", "*=", "~="};

    protected static TokenQueue tq;
    protected static String query;
    protected static List<Evaluator> evals = new ArrayList<>();

    public abstract Evaluator parse(String query);

    public abstract Evaluator parse();

    protected abstract void findElements();
```

```
public abstract class CustomQueryParser extends AbstractQueryParser {
    AbstractQueryParser parser;

    public Evaluator parse(String query) {
        this.query = query;
        this.tq = new TokenQueue(query);
        try {
            return parse();
        } catch (IllegalArgumentException e) {
            throw new Selector.SelectorParseException(e.getMessage());
        }
    }
```

03 Jsoup 확장

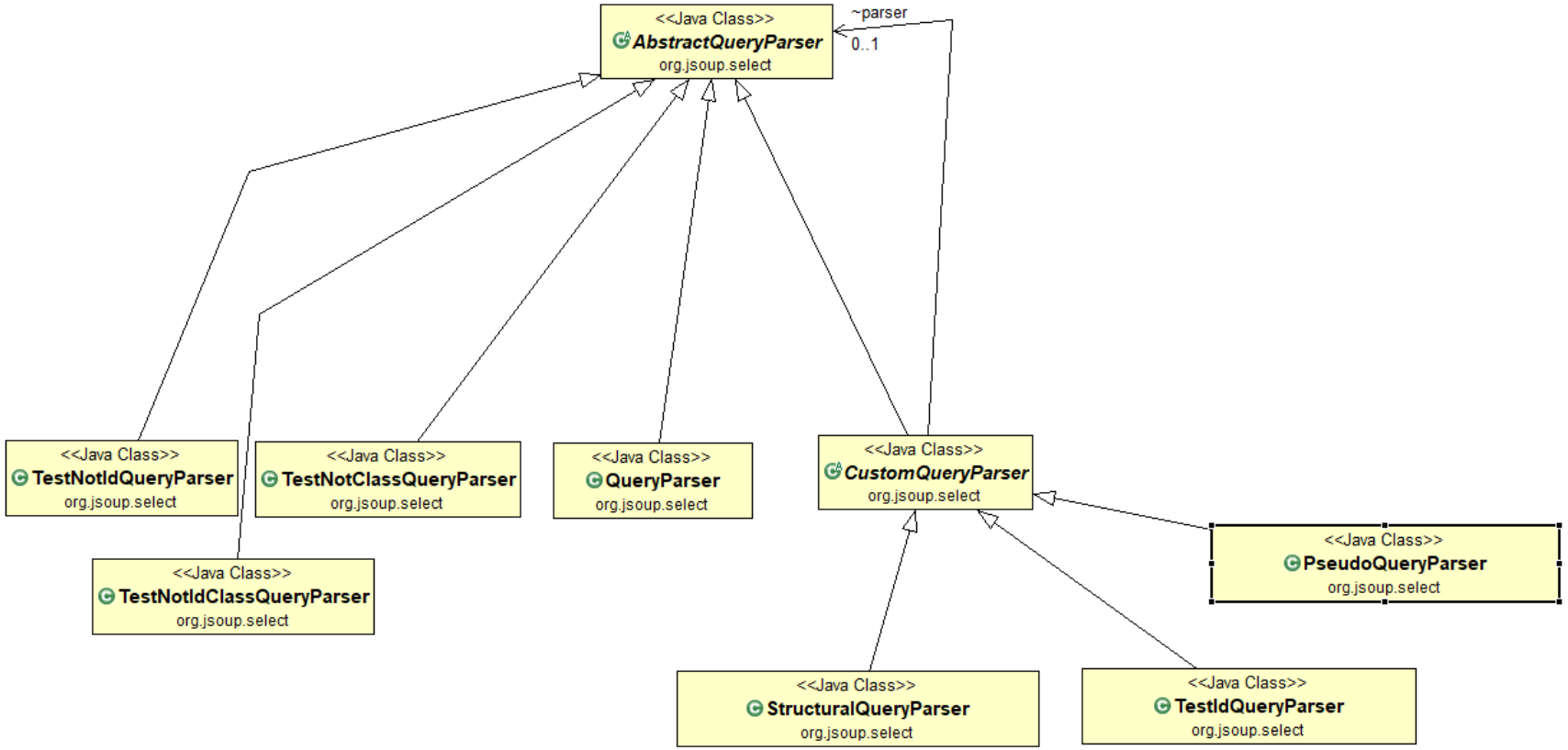
〉기능 확장
: DOM tree 구조 개선 및 시각화

```
doc.traverse(new NodeVisitor() {
    public void head(Node node, int depth) {
        if(node.nodeName()=="html"|| node.nodeName()=="head" || node.nodeName() == "title"
                ||node.nodeName()=="body"||node.nodeName()=="p"|| node.nodeName() == "h1"
                ||node.nodeName()=="h2"||node.nodeName()=="h3"||node.nodeName()=="h4"
                ||node.nodeName()=="h5"||node.nodeName()=="h6"||node.nodeName()=="div" ||node.nodeName()=="a"){

            try {
                for(int i=0; i<depth-1; i++) {
                    bw.write(_space);
                }
                if(depth > 1) {
                    bw.write(_cross);
                }

                if(node.hasAttr("class")) {
                    bw.write(node.nodeName()+ "\t" + "class=" + node.attr("class") +"\n");
                }else if(node.hasAttr("id")){
                    bw.write(node.nodeName()+ "\t" + "id=" + node.attr("id")+"\n");
                }else {
                    bw.write(node.nodeName()+"\n");
                }
```

```
├──div        class=u_skip
  ├──a
  ├──a
  ├──a
  ├──a
  ├──a
├──div        class=banner_area
  ├──div      class=area_flex
    ├──a      class=ba_link
      ├──div          class=ba_logo
      ├──div          class=ba_tw
        ├──p          class=ba_t
├──a          id=whale_promotion_download_file
├──div        class=wrap
  ├──div      class=header
    ├──div    class=special_bg
      ├──div          class=area_flex
        ├──div        class=area_logo
          ├──h1
            ├──a
        ├──div        class=area_links
          ├──a        class=al_favorite
          ├──a        class=al_jr
          ├──a        class=al_happybean
        ├──div        class=search
              ├──div          class=autocomplete
                ├──a          class=btn_arw _btn_arw fold
          ├──a        class=btn_keyboard
          ├──div      id=_nx_kbd
          ├──div      class=reatcmp
            ├──div    class=api_atcmp_wrap _atcmp
              ├──div          class=words _words
                ├──div        class=_atcmp_result_wrap
                ├──div        class=add_group _atcmp_answer_wrap
              ├──div          class=atcmp_plus _plus
                  ├──a        class=spat ico_info _plusHelp
                  ├──a        class=btn_turnon active
                  ├──a        class=btn_turnoff
              ├──div          class=layer_plus _plusAlert
                ├──div        class=_logout
                    ├──p      class=dsc
                    ├──div    class=btn_area
```

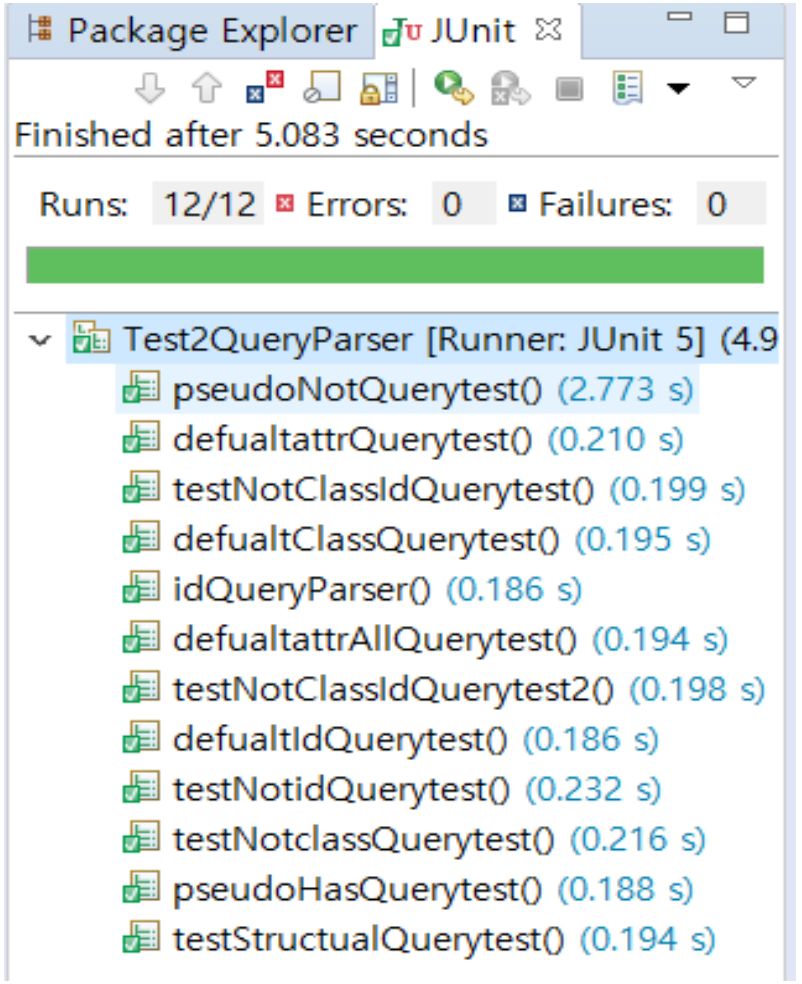
04 테스트 수행

Test result



04 테스트 수행

Test result



04 테스트 수행

Setup & Teardown

```
1  package org.jsoup.test;
2  import static org.junit.jupiter.api.Assertions.*;
13
14 class Test2QueryParser {
15     String url = "https://dhlottery.co.kr/gameResult.do?method=byWin";
16     Document doc = null;
17
18     @BeforeEach
19     void setUp() throws Exception {
20         try {
21             doc = Jsoup.connect(url).get(); // -- 1. get방식의 URL에 연결해서 가져온 값을 doc에 담는다.
22 //          ViewTree.makeViewTreeFile(doc, "test.txt");
23         } catch (IOException e) {
24             System.out.println(e.getMessage());
25         }
26     }
27
28     @AfterEach
29     void tearDown() throws Exception {
30         doc = Jsoup.connect(url).get(); // -- 1. get방식의 URL에 연결해서 가져온 값을 doc에 담는다.zz
31
32     }
```

04 테스트 수행

PesudoQuerytest

```
@Test
void pseudoHasQuerytest() {
    /*set query*/
    String selector = "div:has(div)";
    /*set parser*/
    AbstractQueryParser parser = new TestNotIdQueryParser();
    parser = new PseudoQueryParser(parser);
    Jsoup.setQueryParser(parser);
    /*running*/
    Elements titles = doc.select(selector); // -- 2. doc에서 selector의 내용을 가져와 Elemntes 클래스에 담는다.
    assertNotNull(titles);

}
@Test
void pseudoNotQuerytest() {
    /*set query*/
    String selector = "div:not(h2)";
    /*set parser*/
    AbstractQueryParser parser = new TestNotIdQueryParser();
    parser = new PseudoQueryParser(parser);
    Jsoup.setQueryParser(parser);
    /*running*/
    Elements titles = doc.select(selector); // -- 2. doc에서 selector의 내용을 가져와 Elemntes 클래스에 담는다.
    assertNotNull(titles);

}
```

04 테스트 수행

StructuralQuerytest

```
@Test
void testStructualQuerytest() {
    /*set query*/
    String selector = "div:first-child";
    /*set parser*/
    AbstractQueryParser parser = new QueryParser();
    parser = new StructuralQueryParser(parser);
    Jsoup.setQueryParser(parser);
    /*running*/
    Elements titles = doc.select(selector); // -- 2. doc에서 selector의 내용을 가져와 Elemntes 클래스에 담는다.
    assertNotNull(titles);

}
@Test
void testNotClassIdQuerytest() {
    /*set query*/
    String selector = ".footer";
    /*set parser*/
    AbstractQueryParser parser = new TestNotIdClassQueryParser();
    parser = new TestIdQueryParser(parser);
    parser = new TestClassQueryParser(parser);

    Jsoup.setQueryParser(parser);
    /*running*/
    Elements titles = doc.select(selector); // -- 2. doc에서 selector의 내용을 가져와 Elemntes 클래스에 담는다.
    assertNotNull(titles);

}
```

04 테스트 수행

DefaultQuerytest

```
33    @Test
34    void defualtIdQuerytest() {
35        /*set query*/
36        String selector = "#footer";
37        /*set parser*/
38        /*running*/
39        Elements titles = doc.select(selector); // -- 2. doc에서 selector의 내용을 가져와 Elemntes 클래스에 담는다.
40        assertNotNull(titles);
41
42    }
43    @Test
44    void defualtClassQuerytest() {
45        /*set query*/
46        String selector = ".footer";
47        /*set parser*/
48        /*running*/
49        Elements titles = doc.select(selector); // -- 2. doc에서 selector의 내용을 가져와 Elemntes 클래스에 담는다.
50        assertNotNull(titles);
51
52    }
53    @Test
54    void defualtattrQuerytest() {
55        /*set query*/
56        String selector = "[src]";
57        /*set parser*/
58        /*running*/
59        Elements titles = doc.select(selector); // -- 2. doc에서 selector의 내용을 가져와 Elemntes 클래스에 담는다.
60        assertNotNull(titles);
61
62    }
```

THANK YOU