**Node**

**“Composite 패턴”**

**Author : Ganghee-Lee**

**Date : 2019-12-02**

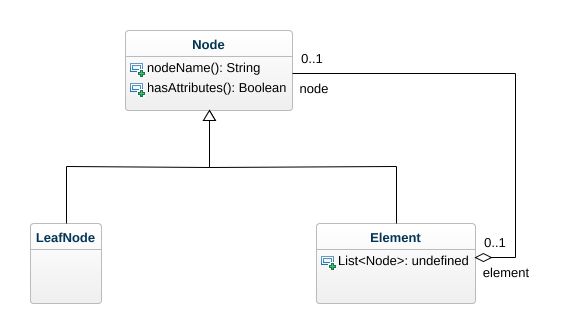
# Overview

- Node의 interface를 통해서 Element인지, LeafNode인지 모르게 이용할 수 있다. Abstract로 되어진 nodeName(), hasAttributes(), outerHtmlHead(), outerHtmlTail() 등의 interface를 공유하고 있다.

- abstract를 통해 공유하는 메소드들 중 Leaf에서 쓰이지 않을 것들은 throw IOException등을 통해 예외처리 되어있다.

- LeafNode의 경우 Empty List를 가지고 있어서 전체를 훑을 때 Empty인지만 검사하면 되도록 구현해 놓았다.

# Class Diagrams



### Class Node

Attributes

* nodeName() : String
* hasAttributes() : Boolean
* attribute : Undefined

Direct sub-classes

* Element
* LeafNode

Associations

* element : Element [0..1]

### Class Element

Attributes

* List<Node> : Undefined

Super classes

* Node

Associations

* node : Node [0..1]

### Class LeafNode

Super classes

* Node