[Software Engineering] #Term Project Requirement Analysis Report

21100380 신현웅, 21300466 유병헌, 21300653 장해빈, 21400125 김시내

1. Class Design

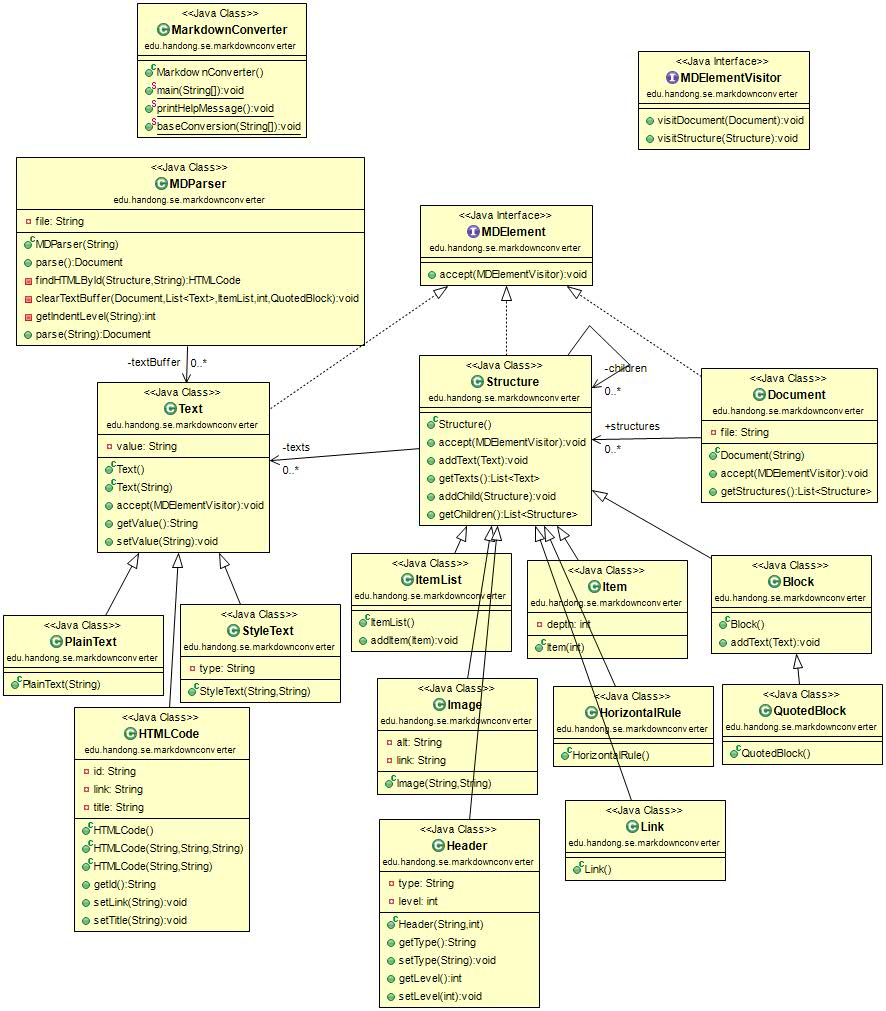
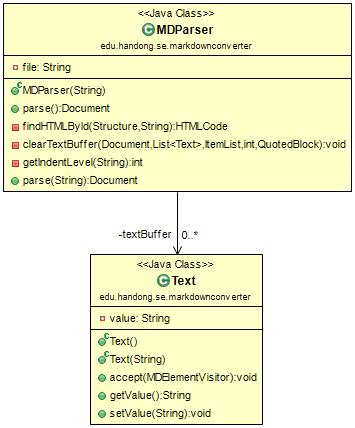


Figure UML from src code

This is the Markdown AST we ultimately made. The MD Parser fetches the md file via the Document type method parse. And we use textBuffer of Text class to distinguish text. The texts are divided into Plain Text, HTML Code, and Style Text. Text is then referred to by Structure. The structure is then inherited by Item List, Image, Header, Item, Horizontal Rule, Link, Block, and Quoted Block. The Structure refers to the Structure. In this way, classes that implement MD Element are formed.

First, let's take a closer look at the MDParser class. As mentioned above, MDParser classifies files read through the document type parse function. It then uses the findHTMLByid, clearTextBuffer, and getIndentLevel methods to handle the html-related work or to delete the textbuffer. If you look at the clearTextBuffer method, you can see that the parameter receives Document, List <Text>, ItemList, int, and QuotedBlock values. This also shows that MDParser is doing a lot of things.

Second, let's look at the Text class. The Text class is divided into three categories: Plain Text, Style Text, and HTML Code. When text represents bold or code, it is classified as Style Text. Likewise, if the text is part of the HTML code, it is classified as HTML code.

The third is Structure. The Structure includes Item List, Header, Item, Horizontal Rule, Link, Block, and Quoted Block that were not included in Text. Each inherits a structure that is not text, and the structure uses that information to get the text of the Text class. We also formed a visitor pattern by throwing itself to the MDElementVisitor. Header has type and level variables, so you can get and export each value with getType, setType, getLevel, setLevel method, and get type and level value with Header function.

The fourth is the Document class. Document creates a file and makes the file received by the Document function its own file. Then throw yourself to the MDElemntVisitor to use the vistor pattern like Structure. As shown in the MDParser section above, parse is used at that time by using the Document type.

The last is the MDElementVisitor class. The MDElementVisitor takes a Structure object and a Document object as shown above and forms a visitor pattern. The visitor pattern is a design pattern that separates the algorithm from the object structure. This separation allows you to add new actions without modifying the structure.

1. Design Issue

The design issue was how to design the Markdown syntax, clarify the ambiguities in the Markdown syntax, and more.

Let’s talk about design issues related to Markdown syntax. At first, we read the text about Markdown syntax. There are many items. So I did not know how to distinguish. We were worried about whether we had to divide it into a Block Element and a Span Element, as shown in the text, or to bundle the code and the code block together. In the meantime, we have found a site that has Markdown as AST. There were contents that were not listed in the text given by the professor. So we refer to that site and we were worried about what to choose because of the different classification method. The text includes Paragraph and Line Breaks, Headers, Block quotes, Lists, Code Blocks, Horizontal Rules, Links, Emphasis, Code, Images. The site contained elements that were not in the text. The elements include root, inline code, YAML, HTML, List Item, Table-Row, Table-Cell, Thematic Break, Break, Strong, Delete, Footnote, Link Reference, Footnote Reference, Definition, Footnote Definition, have. When we looked at the site, we thought that there was an infinite way to make structure. So, finally, the structure was selected by selecting the minimum element. And those that consist of text, such as HTML code, are classified in other classes which inherited Text class.