

Software Engineering Term Project #2

Team J

21000004 강민규 / 21100673 조선웅 / 21200401 송지웅 / 21400451 오승준 / 21400772 최인회

We improved project from the last submitted code by changing the code into visitor pattern. We implemented parser and plain html code writer. Parser is implemented in the htmlWriter class. We use a function named startParse() to get a character from the code. After parsing, the code checks if there is a header by checking if there exist special tokens such as "#" or "*" in the sequence. Until now, we only implemented h1 to h6 header conversion and paragraph. The pure HTML generator first makes the html file in given input, and when closing, they also writes predefined values.

After constructing code for generating plain HTML code, we used Jtidy library to check if there are any errors in the generated HTML file. Our code gives output in the form of output.html In the java path, to test output.html, we implemented Jtidy such as:

1. We putted jtidy.jar in the java path.
2. Java -jar jtidy.jar output.html

Then, the jtidy library tells us where we have missed, like this:

```
C:\WUsers\W\>java -jar jtidy.jar output.html
line 1 column 28 - Warning: <meta> lacks "content" attribute
line 1 column 80 - Warning: <style> lacks "type" attribute
line 2 column 4 - Warning: trimming empty <p>
line 2 column 50 - Warning: replacing element </p> by <br>
line 2 column 50 - Warning: inserting implicit <br>
output.html: Doctype given is ""
output.html: Document content looks like HTML 4.01 Transitional
5 warnings, no errors were found!
<!DOCTYPE html>
<html>
<head>
<meta name="generator" content=
"HTML Tidy for Java (vers. 2009-12-01), see jtidy.sourceforge.net">
<meta charset="utf-8">
<title>Expected Output</title>
<style type="text/css">
</style>
</head>
<body id="Preview">
<h4>Paragraphs are separated by a blank line.</h4>
<br>
<br>
```

From now, we will implement emphasis (*String*), (**String**), (~~String~~), list, link, table, blockquote, and other headers and line break.

[ANT Apache]

```
C:\WUsers\W\>ant
Buildfile: C:\WUsers\W\build.xml

clean:

create:
    [mkdir] Created dir: C:\WUsers\W\bin
    [mkdir] Created dir: C:\WUsers\W\src
    [mkdir] Created dir: C:\WUsers\W\libs
    [mkdir] Created dir: C:\WUsers\W\docs
    [mkdir] Created dir: C:\WUsers\W\build

compile:

BUILD SUCCESSFUL
Total time: 5 seconds

C:\WUsers\W\>
```

Default target is "compile". Before compile is executed, "clean", "create" and "copy" are all executed in order. "Clean" erases all the originally existing files in the directory, "create" creates a new directory, and "copy" copies resources from target directory. Then compile target is executed java sources.

In the example to the left, docs is where the new HTML files will be added.