Lab 12 -- Document Visitors

The goal of this lab is to design and implement a Visitor design pattern.

Rich-Text Documents

Rich text documents combine text with formatting information in a way that allows the text in those documents to use a mixture of fonts, font sizes, font styles (bold, italics, etc), and paragraph styles. Rich text documents may also contain non-text content such as images. This is contrast to plain text documents that contain only plain, unformatted text.

Different programs are designed to display a text document correctly depending on the type of formatting that is contained in the file. The type of a file is usually indicated by the file extension. For example, an ".html" document contains markup that can be interpreted by your web browser; a ".md" document can be displayed by a markdown editor; a ".docx" document contains markup that can be interpreted by a word processor, etc. The list goes on ..

Many forms of rich text documents have similar types of formatting:

* **Basic text** refers to the regular text found in the document.
* **Bold text** refers to text that is emphasized through darkening.
* **Italic text** refers to text that is emphasized through using a slightly slanted font.
* **Hypertext** represents hyperlinks that usually include a "url" in addition to the describing text.
* Groups of text are usually organized into **paragraphs** consisting of one or more text elements.
* **Headings** are used to represent titles in our text and can usually be created at several levels.

What to do

**Package:** document

To start this assignment, download this [provided starter code zip file](https://northeastern.instructure.com/courses/63372/files/8872215?wrap=1)[download](https://northeastern.instructure.com/courses/63372/files/8872215/download?download_frd=1)that contains several different types of text elements. Take a moment to familiarize yourself with the classes in the document.element package. Each class, implementing the TextElement interface, represents a different type of text formatting. Finally, the Document class in the document package is used to represent a generic document.

In this lab, we will be using the visitor design pattern to add new capabilities to the Document class.

To do this you need to add a method to the TextElement interface. Start by making the elements of the document "visitable" by adding the following method to the TextElement interface, and implementing it in the implementing classes:

public <R> R accept(TextElementVisitor<R> visitor)

Next, implement each of the following visitors, in the document package:

1. WordCountVisitor: counts the number of words that are in a document.
2. BasicStringVisitor generates a simple string representation of the document. In a simple string representation, the text of each element is added one at a time using a space between each element. Additional information (e.g., the level of a heading) is ignored.
3. HtmlStringVisitor generates an HTML version of the document. In HTML, white space is not important so each element of the document should be separated by a newline ("\n"). In addition, formatting tags should be added according to [basic html syntax (Links to an external site.)](https://www.arubanetworks.com/techdocs/ClearPass/6.6/Guest/Content/Reference/BasicHTMLSyntax.htm).
4. MarkdownStringVisitor generates a Markdown version of the document. Similar to HTML, white space is less important to the document so each element of the document should be separated by a newline ("\n"). In addition, formatting tags should be added according to [basic markdown syntax (Links to an external site.)](https://www.markdownguide.org/cheat-sheet/).

Now, you should add two methods to the Document class that use these visitors:

1. countWords that returns the number of words in the document.
2. toText that takes one of these "string visitors" as a parameter and returns a String as per the above description. String visitors should accumulate their result and return it when toString is called on the visitor.