ECOTE - preliminary project
Translator of a LaTeX subset to HTML

Krzysztof Rudnicki, 307585
Semester: 2023L

April 26, 2023

1 General overview and assumptions

My task is to create a translator of LaTeX subset to selected text format with focus on LaTeX tables
I decided to change to translator of LaTeX subset to HTML since I know LaTeX very well and HTML relatively well, I decide to translate LaTeXinto HTML since HTML is easy, a little bit different than LaTeXand popular which makes this translator a practical tool.

1.1 Assumptions

- No LaTeX (%) comments in the script
- There are no extra packages in LaTeX script (provided with \usepackage keyword) besides ones distributed with LaTeX
- There are no extra classes in LaTeX script besides ones distributed with LaTeX
- There is nothing between \documentclass keyword and \begin{document} keyword
- No standard LaTeX instructions are modified in the script
- ”Tables” will be represented using LaTeX *table* environment

2 Functional requirements

The goal of the project is to transform .tex file to (working) .html file if the subset of .tex file is within project scope or output error message explaining why the html could not be outputed

2.1 LaTeX subset

This project will focus almost exclusively on *tabular* environment

- \documentclass{class}: Defines what layout standard LaTeXwill use
- \begin{document}: Ends (in our case empty) preamble
- \end{document}: Ends LaTeX document
- \begin{tabular}[pos]{table spec}: Opens environment used to typeset tables
- \end{tabular}: Closes environment used to typeset tables

Supported tabular arguments:

- l, c, r - respectively left-justified, centered and right-justified column
- | - single vertical line
- || - double vertical line
- p{'width'} - paragraph column (that wraps), aligned at the top
- m{'width'} - paragraph column (that wraps), aligned at the middle
- b{'width'} - paragraph column (that wraps), aligned at the bottom

Supported commands inside tabular environment

- & - separate columns
- \\ - start new row
- \hline - horizontal line

- \cline{i-j} horizontal line beginning in column i and ending in column j
- \newline - new line *inside* the cell

3 Implementation

I decided to use Python as a language in which I will implement my solution The reasons for using python are as follow:

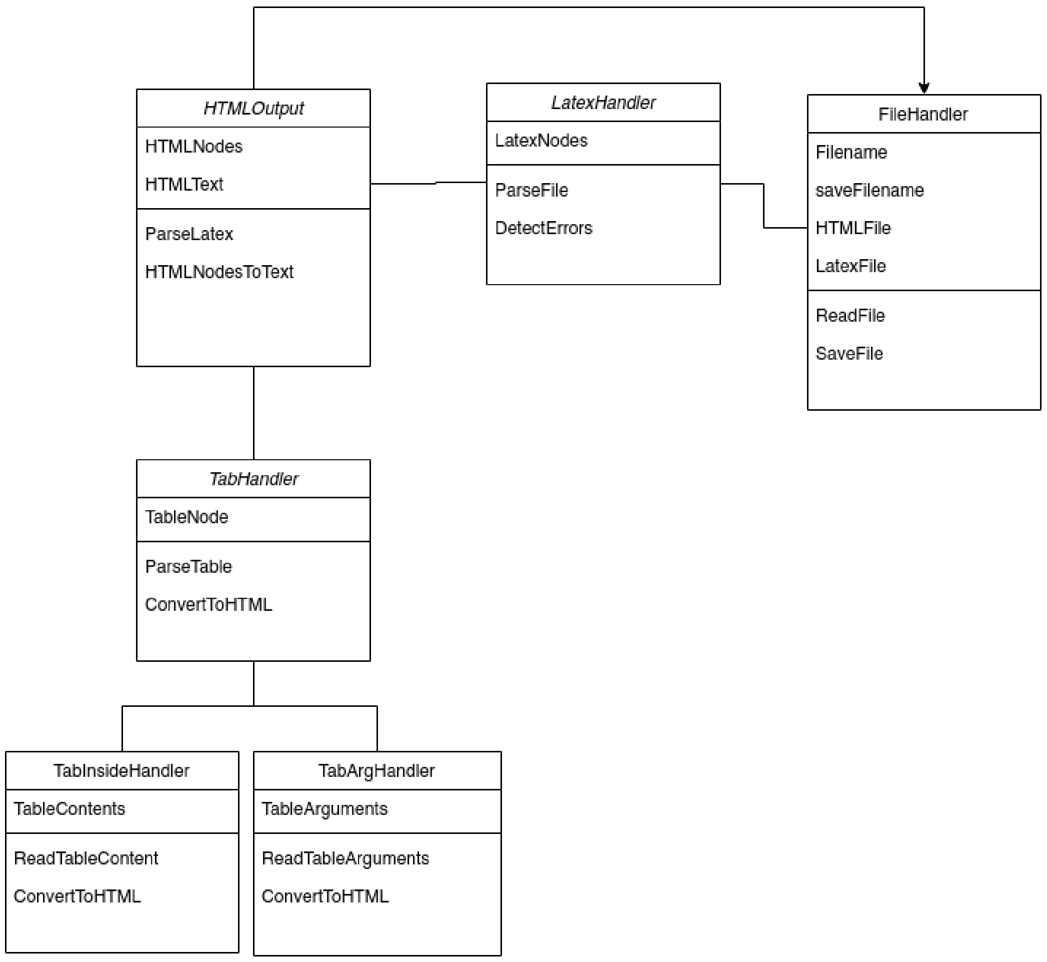
1. It is the easiest language among those that I know
2. I know it enough to be confident in my ability to implement this solution in python
3. I want to learn python more through this project

Negative aspects of python which is that it is very slow language do not bother me as I believe the project scope will not be big enough for this to become an issue

3.1 General architecture

Module	Description
Main	Handles parameters inputed as program arguments and interaction between modules
FileHandler	Handles file reading
LatexHandler	reads LaTeX file content, parses it using LatexWalker class of pylatexenc library and detects errors
TabHandler	Transforms tabular environment into html table
TabArgHandler	Handles positional and table spec arguments of tabular environment and translates them to html
TabInsideHandler	Handles actual content of table and translates them to html
HTMLOutput class	Transforms LaTeX code into html with the assumptions that no errors were found by LatexHandler and if there were any they were dealt with

Figure 1: Module class diagram



3.2 Data structures

- File entered by user is represented by python File class
- Parsed LaTeX code is represented by data structure based on node classes
- Tabular environment parameters are stored in an array, if the parameter contains additional optional parameters they are stored in a pair of the parent argument and array of all optional arguments
- Generated HTML code is stored in node classes

- Final HTML code is stored in plain text and then written to File

3.3 Module descriptions

3.3.1 Main

Handles user arguments and communication between modules
Input:
args[] - Array of user inputed arguments
Functions:

- Handle arguments - Checks if arguments are correct and adjust program to their contents, in particular saves tex file path
- Invoke FileHandler - starts file handler class and gets returned file or errors
- Invoke Latex Handler - Sends file from file handler to LatexHandler

3.3.2 FileHandler

Reads file from filename provided by user and at the end saves HTML file
Parameters:

- filename - user entered tex filename
- saveFilename - (optional) filename for output file
- HTMLFile - converted final html file in File format
- LatexFile - latex file in File format

Functions:

- ReadFile - Using user filename as argument reads file and saves it to LatexFile
- SaveFile - Using filename provided by user or tex filename as argument converts HTML string to file and saves it

3.3.3 LatexHandler

Reads file from filename provided by user and at the end saves HTML file
Parameters:

- filename - user entered tex filename
- saveFilename - (optional) filename for output file
- HTMLFile - converted final html file in File format
- LatexFile - latex file in File format

Functions:

- ReadFile - Using user filename as argument reads file and saves it to LatexFile
- SaveFile - Using filename provided by user or tex filename as argument converts HTML string to file and saves it

3.3.4 HTMLOutput

Reads file from filename provided by user and at the end saves HTML file
Parameters:

- filename - user entered tex filename
- saveFilename - (optional) filename for output file
- HTMLFile - converted final html file in File format
- LatexFile - latex file in File format

Functions:

- ReadFile - Using user filename as argument reads file and saves it to LatexFile
- SaveFile - Using filename provided by user or tex filename as argument converts HTML string to file and saves it

3.3.5 TabHandler

Handles tables and converts them to html nodes
Parameters:

- TableNode - node containing only the table

Functions:

- ParseTable - Parses Table node and checks for errors
- ConvertToHTML - Converts table node to html using TabInsideHandler and TabArgHandler

3.3.6 TabInsideHandler

Handles inside of tabular environment and converts them to html nodes
Parameters:

- TableContents - Latex node containing only inside of table

Functions:

- ReadTableContent - Parses table content and checks for errors
- ConvertToHTML - Converts table inside node to html

3.3.7 TabArgHandler

Handles arguments of tabular environment and converts them to html nodes
Parameters:

- TableArguments - Arguments provided with \begin{tabular} environment

Functions:

- ReadTableARguments - Parses table arguments and checks for errors
- ConvertToHTML - Converts table arguments to html

3.4 Input/output description

Input is a .tex file (LaTeX file)
Outpus is an .html file
In case of errors error message will be outputed on the terminal
Input File path is entered as an argument to terminal with ”-i” or ”–input” flag for example:

```
python main.py -i texFile.tex
```

Output file path can be named by user by using ”-o” or ”–output” flag:

```
python main.py -i texFile.tex -o htmlFile.html
```

If no ”-o” flag is issued the output file will have the same name as input file with changed extension to html (so in this example texFile.tex will become texFile.html)
If the path to file name consists of spaces, path name needs to be but in ””

```
python main.py -i "My␣Folder/input.tex"
```

3.5 Others

4 Functional test cases

Title	Input (LaTeX)	Output
empty file		`Error! expected \documentclass at the begining of LaTeX file`
Document class	`\documentclass[options]{class}`	`Error! expected \begin{document} after document class`
`Extra text between document class and begin document`	`\documentclass[options]{class}` `"extra text"` `\begin{document}`	`Error! unexpected text between document class and begin document`
`Just document class and begin document`	`\documentclass[options]{class}` `\begin{document}`	`Error! no \end{document} at the end of LaTeX code`

Title	Input (LaTeX)	Output

Just document class and begin/ end document	\documentclass[options]{class}
	\begin{document}
	\end{document}
		<html>
		</html>

Plain text inside	\documentclass[options]{class}
	\begin{document}
	Lorem ipsum dolor sit amet.
	\end{document}
		<html>
		Lorem ipsum dolor sit amet.
		</html>

Reduntant \end{document} (ignored)	\documentclass[options]{class}
	\begin{document}
	Lorem ipsum dolor sit amet.
	\end{document}
	\end{document}
		<html>
		Lorem ipsum dolor sit amet.
		</html>

LaTeX comments	\documentclass[options]{class}
	\begin{document}
	Lorem ipsum dolor sit amet.
	% some comment
	\end{document}
	\end{document}
		Error! LaTeX comment detected at line 3

Title	Input (LaTeX)	Output

Table with vertical lines	`\documentclass[options]{class}`
	`\begin{document}`
	`\begin{tabular}{ l | c | r }`
	`test & 2 & test \\`
	`4 & 5 & 6 \\`
	`\end{tabular}`
	`\end{document}`
		`<html>`
		`<table>`
		`<tr>`
		`<td align='left'>test</td>`
		`<td align='center' style="border-left: 1px solid black;">2</td>`
		`<td align='right'>test</td>`
		`</tr>`
		`<tr>`
		`<td align='left'>4</td>`
		`<td align='center' style="border-left: 1px solid black;">5</td>`
		`<td align='right'>6</td>`
		`</tr>`
		`</table>`
		`</html>`

Missing &	`\documentclass[options]{class}`
	`\begin{document}`
	`\begin{tabular}{ l c r }`
	`1 & 2 & 3 \\`
	`4 & 5 6 \\`
	`\end{tabular}`
	`\end{document}`
		`Error! Missing third column in second row`

Title	Input (LaTeX)	Output

Too much columns	`\documentclass[options]{class}`
	`\begin{document}`
	`\begin{tabular}{ l c r }`
	`1 & 2 & 3 & 4 & 5 \\`
	`4 & 5 6 \\`
	`\end{tabular}`
	`\end{document}`
		`Error! Too much columns in row 1, expected 3, got 5`