Introduction

This application is a tool for parsing an eBook file or URL and a dictionary file, creating an index of the words in the eBook with their associated page numbers and definitions, then outputting the index to a specified output file. Aside from selecting the eBook file directory or URL, the user may also change the dictionary file directory as well as the file directory for words to omit from the index. As additional features, the user can then interact with the index through a menu-based interface, allowing them to search for a specific word, view the top frequent or infrequent words, view a list of the words sorted alphabetically in either ascending or descending order, and view the total number of unique words.

Running the Application

To run the application, navigate to the bin directory of the project and run the following command:

```
java --enable-preview -cp ./indexer.jar ie.atu.sw.Runner
```

Design Patterns

The following design patterns are used in this application:

Abstract class: The AbstractParser class is an abstract implementation of the Parser interface that provides a skeletal implementation of the parse and addToIndex methods. This allows for shared code between concrete implementations of the Parser interface, while still allowing for flexibility in the implementation of the parse and addToIndex methods.

Interface: The Parser interface defines methods for parsing a file and adding words to an index. This allows for multiple concrete implementations of the Parser interface, providing flexibility in the way that different types of files are parsed and added to the index.

Inheritance: The DictionaryParser, EBookParser, and StopWordsParser classes all extend the AbstractParser class, inheriting its shared code and implementing the parse and addToIndex methods specific to their respective purposes. This allows for reuse of code and a clear hierarchy of classes.