Project 2 Report

Author: HongFei Huang
Student ID: h338huan
Time: Mar 27th, 2021

1) Overview of classes

a) What class(es) did you design?

I designed four classes which are:

***Input_Processing, *illegal_exception, *Trie, *Node**

b) Describe the role of each class in your design?

i. **Input_Processing:** It will take user's input as its parameter and break it down into "command" and "parameter". A function will be responsible to plug these two strings into the corresponding member functions in Trie.

ii. **illegal_exception:** It will be thrown when an input is not valid. Used by Trie.

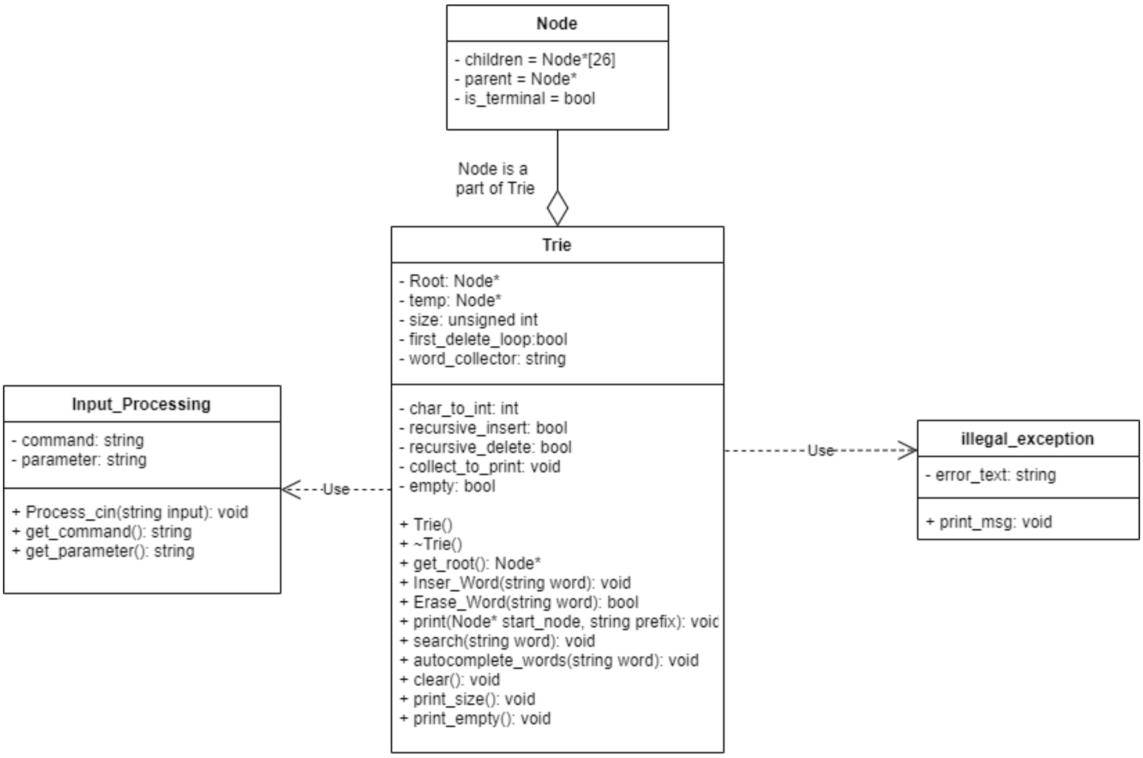
iii. **Node:** A node storing the addresses of the next 26 sub-nodes and its parent node. It also contains a Boolean expression to tell the programmer if it is a terminal node.

iv. **Trie:** Contains a root node which has 26 sub-nodes and all the other member functions to achieve desired functionalities.

c) How do classes relate to one another?

Node is a part of **Trie**. According to different user command, the **Trie** class either insert or delete certain nodes recursively to store the information. Trie will fetch commands and parameters processed by **Input_Processing** and use them in its member functions. **illegal_exception** works as an input validator in Trie; hence, when an error occurs, it terminates the current process and prints out error message.

2) UML Class diagram



3) Details on design decisions

i. **Constructor:** Trie – Using constructor to initialize the root node. Others are using default constructor.

ii. **Destructor:** Trie – Calling clear() and using "delete" to free all memories saved by pointers, Others are using default destructor.

iii. **No overriding done and no "const" used in this project.**

4) Test Cases

To start off, I used the provided test cases on GitHub for functionality debugging and Valgrind on ECE server for memory-leak check. I spotted two errors: 1 for deleting the wrong address in a pointer and 1 for autocomplete not printing when input was "*". Then, I tried several abnormal cases such as extremely long words(less than the maximum number of unsigned int) and kept calling clear() when the trie is empty. All cases were passed.

5) Performance considerations

Inserting, erasing, and searching will yield the same running time: n, because they are only required to traverse one single "branch" of the trie and focusing on one single word. The running time depends on the length of the word.

Printing, autocompleting, and clearing will yield the same running time: N, since they are all required to traverse every single terminal node during execution(they are calling the same sub-function that finds every single word recursively).

Size() and empty() are very simple functions that either returns one unsigned integer or call a function to check if any child node(s) in root is(are) non-empty.