|  |
| --- |
| **Name** |
| **Ahmed Metwally Ali** |

Meeting Planner Project

# Overview

Briefly the code implementation of the meeting planner project is consists of three files.

* Header file (BST.h) for implementing the declaration of the BST class with its *data* and *operations (methods)*.
* Source file (BST.cpp) for implementing the definitions of the operations which is declared into the class which is implemented in file BST.h.
* Source file (main.cpp) for getting inputs from terminal and reading input file and handling errors which may occur in the input file. Then printing the output into output file and on screen.

Now, let’s go inside the *BST.h* and review the class structure.

Class is divided into two main sections:

* Private section
* Public section

1. **In private section** -*where we implement data which user have no access on it*-
   1. Struct node; typedef node\* link;
      1. **Main Data:** *-which is used to store data of the node which user want to store it-.*

Data type: string int int

Title

Day\_key1

Hour\_key2

Variable name:

# Depending on the value of day\_key1 and hour\_key2 which we consider as keys of each node the decision is made to perform the majority of commands.

* + 1. **Linking data:** *-which is used to link the created node with other nodes in the tree-*

Data type: link link

Rightptr

Leftptr

Variable name:

Leftptr and rightptr are the pointers on the node which link the node into the tree with other nodes. And if the node was leaf then, their values would be **NULL.**

* 1. Link root;

This is a pointer to struct of type node which used to point on the head node of the tree.

* 1. Int size;

This variable is used to get size of the tree *-calculate the Num. of nodes which exist in the tree in-.*

1. **In Public section -***where the methods have been declared and the user have access on them***-**

Implementation of the methods which are used to perform commands that have been mentioned in the description which are (ADD – MOD – DEL – Find - Print) are divided into two sections.

* First one, the methods in public section to make it easy for user to access them.

And the user called them in main file and pass to them the data which are used to perform task correctly.

* Second one, the methods in the private section. Where the code implementation exist.

Whereas, methods in public section is called in main file and user pass to them data. Then, in these methods a private methods’ section is called for implementing this task. Because that methods in private section is passed to them from public section *the root pointer* which the user cannot access.

# Main Flows

After getting into the main function I check on the number of arguments which passed to main through terminal. Then check on the existence of the input file for making sure that everything will execute correctly.

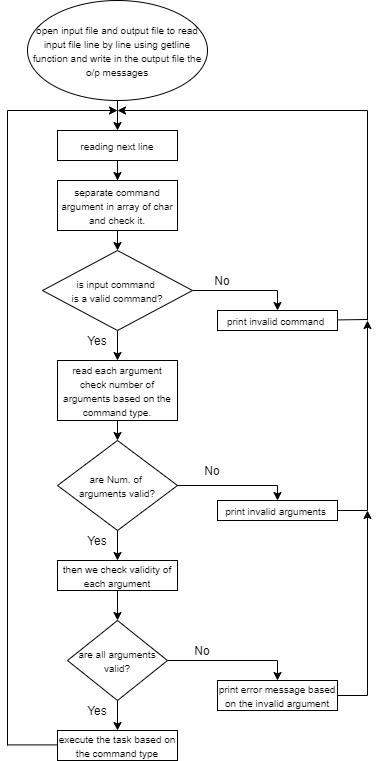
Here is the main flow inside main program.

Figure 1 main flow of the program

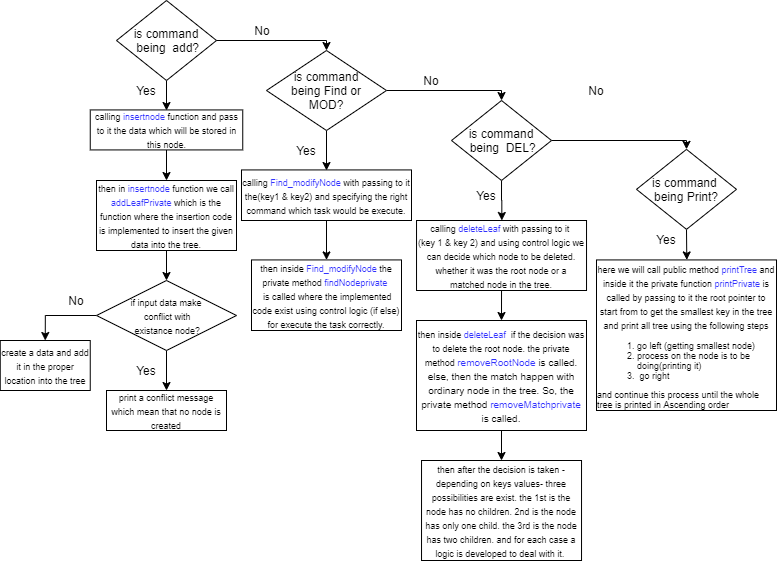
So, here it is the flow for executing command which has been read from the input file.

Figure 2 flow for executing command

References

1. Data Structures and Algorithms in C++: Adam Drozdek
2. Data Structures via C++: M. Berman.
3. Weekly lectures.