https://github.com/GamaCatalin/SymbolTable

Symbol table using binary search tree

A node is defined that holds the value and references to the left and right children.

- Add(String value):
    - Checks the given value against the value hold in the current node
        - If the value is equals, then it returns false, as it cannot add the value
        - If the value is less then it checks if the left child exists
            - If it exists it calls the add method on it
            - If it does not exists then it is added, returns true
        - If the value is bigger, then it checks if the right child exists
            - If it exists it the add method on it
            - If it does not exist then it is added, returns true
        - If it passes all the checks then something went wrong, returns false

- Exists (String value):
    - Checks the given value against the value hold in the current node
        - If the value is equals, then it returns true
        - If the value is less then it checks if the left child exists
            - If it exists it calls the exists method on it
            - If it does not exists then returns false
        - If the value is bigger, then it checks if the right child exists
            - If it exists it the exists method on it
            - If it does not exist then returns false
        - If it passes all the checks then something went wrong, returns false

- Walk(String value):
    - Walks the nodes performing checks, returns the path that it needs to follow from the given node to get to the value

The binary search tree is implemented as utilizing the node methods from a root node.

- Performs all the functionalities as above, calling them from the root,
    - If the root does not exist it creates one for the add, return false for the exit and return an empty string for the walk path