Group: Gabriella Willis, Shelby Mohar, Alivia Dutcher

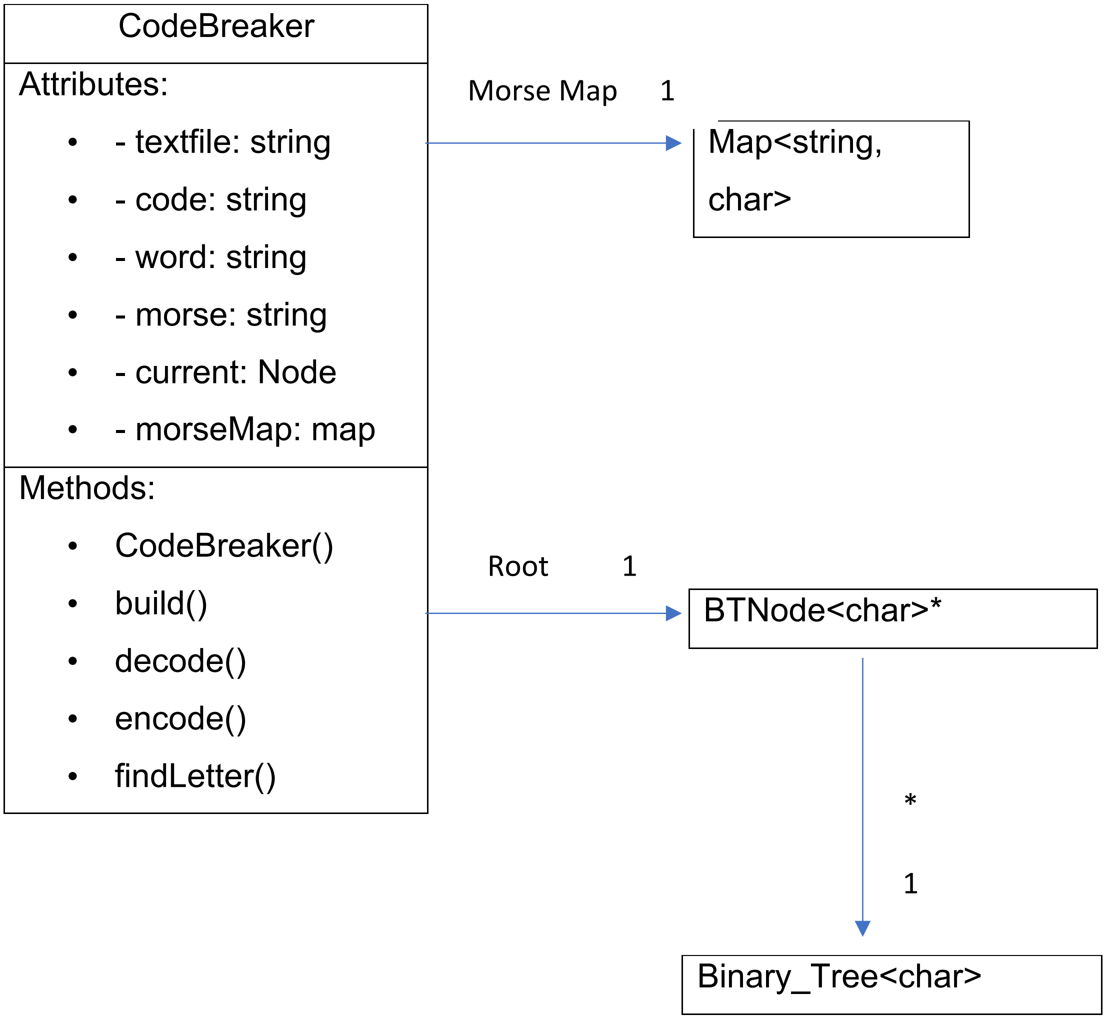
CS303 – Project 2B

Due: July, 23, 2019

Project Assumptions:

- Symbols going into the encoder would be alphas/no unexpected variables
- Letters would be lowercase
- Each alpha letter or Morse letter would be separated by a space

UML:



CodeBreaker

Attributes:

- - textfile: string
- - code: string
- - word: string
- - morse: string
- - current: Node
- - morseMap: map

Methods:

- CodeBreaker()
- build()
- decode()
- encode()
- findLetter()

Group: Gabriella Willis, Shelby Mohar, Alivia Dutcher

CS303 – Project 2B

Due: July, 23, 2019

Big (O):

- CodeBreaker():
    - Build(ifstream &textfile) – O(n)
    - buildBT(char alpha, string element, BTNode *newRoot) – O(n)
    - findletter(string morse, BTNode *current) – O(n)
    - decode(string code) – O(1)
    - encode(string word) – O(1)

The implementation of this could perhaps be better if some of the functions were made into smaller functions. However, almost no time would have been saved and it's a pretty efficient program.

References:

- https://stackoverflow.com/questions/23484850/inserting-morse-code-into-binary-tree/23484930#23484930

  Helped in understanding where to begin binary tree.
- http://math.hws.edu/eck/cs225/s03/binary_trees/

  Refresher on how things moved within trees

Group: Gabriella Willis, Shelby Mohar, Alivia Dutcher

CS303 – Project 2B

Due: July, 23, 2019

- https://www.cprogramming.com/tutorial/lesson18.html

  Break down of trees and gave us ideas on what needed to be implemented
- https://www.geeksforgeeks.org/morse-code-implementation/

  Basic understanding of morse code within trees
- https://stackoverflow.com/questions/45386858/segmentation-fault-in-morse-code-binary-search-tree

  Helped a lot with understanding how to decode. We needed to establish the root as well as making sure there was a node to be accessed.
- CS303 Notes:
  - Huffman Trees, Binary Search Trees, Sets and Maps, Efficiency of Algorithms