Pseudocode

Morse Code Converter encode(m), decode(c), visit() and MorseCodeTree() constructor methods

MorseCodeTree() constructor

1. Read in from file “codes.txt”
2. While the file has more lines

2.2 set a variable to the next line

2.3 find the character at the beginning of the line.

2.4 Find the code within the line.

2.5 Make a new node as the root of the tree.

2.6 loop through the length of the code at the end of the line

2.6.1 Determine the first character in the code.

2.6.2 Check if the first character is a ‘.’

2.6.2.1 If the first character is a dot, set the parent as the left child.

2.6.2.2 If the left child doesn’t exist, make one with the value of the code.

2.6.3 Check if the first character is a ‘-’

2.6.3.1 If the first character is a dash, set the parent as the right child

2.6.3.2 If the right child doesn’t exist, make one with the value of the code.

2.7 Stop looping through the code

3. Stop looping through the file

Decode(String c) Method

1. Create an array of the words passed in, in Morse code
2. Create a blank message
3. Loop through each word in the array of words
   1. Split the current word up into each code
   2. Loop through the letters in the current code
      1. Set the current letter to a variable
      2. Create the parent node as the root.
      3. Loop through the characters in the current letter
         1. Set the current symbol to a variable.
         2. Check if the current character is a ‘.’
            1. If it is, move left down the tree.
            2. If you can’t, append to the message the current character
         3. Check if the current character is a ‘-‘
            1. If it is, move right down the tree.
            2. If you can’t, append to the message the current character
   3. Append a space to the end of the message to separate each word
4. Return the message created

visit(BinaryTreeNode node) in the CodeVisitor class

1. Check if the current node is the root of the tree 1.1 if it isn’t, create a variable with the parent of the current node 1.2 if the current node is the left child of its parent, push onto the stack a ‘.’ 1.3 if the current node is the right child of its parent, push onto the stack a ‘-‘ 1.4 Recall the method recursively with the parent of the current node as the parameter 2. If the current node was the root: 2.1 push onto the stack a newline character 2.2 loop until the stack is empty 2.2.1 pop of the top of the stack and append the character onto a string of codes.

Encode(String m)

1. Convert the parameter to all lowercase
2. Instantiate a code visitor object
3. Do a levelOrderTraversal through the tree.
4. Set a code variable to the getCode() generated by the tree traversal
5. Set a letter variable to the getLetters() generated by the tree traversal
6. Split the parameter into an array of words.
7. Create a blank message
8. Split the code variable into an array of all the codes
9. Split the letter variable into an array of all the letters
10. Loop through each word in the array of words from the parameter
    1. Loop through the length of the current word
       1. Loop through the array of all the English characters
          1. Check if the current letter in the word is equal to the current letter in the array of letters
             1. If it is, append to the message the code from the array of Morse code values that equivalent to the current index in array of letters.
       2. Append to the message a space to separate each Morse code letter
    2. Append to the message two spaces to separate each Morse code word.
11. Return the Morse code string built up