

Morse Code Translator

----- .-. /-.-. --- -.. . /- .-. .-.-. -... .- --- .-. .

Project 1 Documentation

By:

John Sapp

for:

Introduction to Java - COP 2800c - 22196

Submitted to:

Professor Ian O'Toole

2/11/24

This program was designed to complete several tasks surrounding Morse Code Translation. Morse Code is a simple binary language using dots “.” and dashes “-” to represent English characters and numbers.

This program uses a “Menu” to allow users to select one of 5 options

- Translate English to Morse code
- Translate Morse code to English
- Print a conversion chart
- Quit the program
- Reload menu.

User Experience:

Once the program has started to run, the user is greeted with a welcome message and a menu. The user selects one of the five options by typing in the corresponding number and pressing enter.

1 - English to Morse Code:

This option asks the user to enter a message in English and then press the enter key. The corresponding message is then printed to the terminal with “/” between individual words. This option then asks the user if they want to repeat the current option to translate a new message or if they want to quit or return to the menu.

2 - Morse Code to English

This option is very similar to option 1. The option asks the user to input a message in Morse code, then the message is displayed to the terminal with a space between every character. Like before this option then asks the user if they want to repeat the current option to translate a new message or if they want to quit or return to the menu.

3 - Print conversion chart

This option simply displays a conversion chart of English to Morse code with the English characters on the left and their corresponding Morse code translation. Like before this option then asks the user if they want to repeat the current option to translate a new message or if they want to quit or return to the menu.

4 - Quit program

This option will cause the program to stop running.

5 - Reload menu

This option will write the menu to the terminal again.

Program method IPO:

public static int mainMenu()		
Input	Process	Output
Gets user's menu selection as an integer	Check to see if this is the first time the mainMenu has been called. Parse the integer to String.	If this is the first time the menu has been called - display a welcome message. Displays the menu to the terminal Returns the selected menu number

public static void selection(int)		
Input	Process	Output
Selected menu number	Switch case to call the correct method	If the selected menu number is larger than 5 an error message is displayed and the menu is re-printed

public static void engToMorse()		
Input	Process	Output
Get a message from the user in English	The message is stored in a variable	The String variable is passed to the "getChar()" method. Once the "getChar()" method returns the userOptionSelection() method is called.

public static void morseToEng()		
Input	Process	Output
Get a message from the user in Morse code	The message is stored in a variable	The String variable is passed to the "parseMorseCode" method. Once the "parseMorseCode()" method returns the userOptionSelection() method is called.

public static void parseMorseCode(String)		
Input	Process	Output
Takes the String passed	Loop through the Sting. If a space character is not found the character is concatenated to a variable. If a space character is found the Morse code letter is completed.	The Morse code letter is sent to the translate() method

public static void printChart(String[][])		
Input	Process	Output
Takes in a 2d Array	Loops through the 2d Array	Prints the Array to the terminal.

public static void userOptionSelection(int)		
Input	Process	Output
<p>Takes an Integer that is passed into the method.</p> <p>Also asks the user for an integer</p>	<p>The passed int value is used to display the correct option for the user to select to repeat the last menu selection</p> <p>The int value collected is stored if the value is equal to the passed int or = 4 5. If the collected number is not one of the values above an error message is displayed and the method is called again.</p>	<p>Selection message for the user to choose an option from.</p> <p>Error messages if necessary.</p>

public static void getChar(String)		
Input	Process	Output
String	<p>Coverts the string so all characters in the string are upper case.</p> <p>Loop through the Sting to get individual characters stored in a char variable</p>	<p>The char variable is sent to the translate method passing the morseLookUp array, the char variable, index 1 value, index 2 value</p>

public static void translate(String[], String, int 1, int 2)		
Input	Process	Output
<p>Takes in 4 parameters; the lookup array, the string to reference to the lookup array, index 1, index 2</p> <p>The index values set the order in which the corresponding English to Morse code is read</p> <p>If index1 =0 && index2 =1 The loop will return English to Morse</p> <p>If index1 =1 && index2 =0 The loop will return Morse to English</p>	<p>Boolean value set to false, if true then no match has been found in the lookup array.</p> <p>For loop to iterate through the lookup array. If a space character is found the "/" character is printed to the terminal.</p> <p>Else If the string value equals a value in the 2d array the corresponding translation will be printed to the terminal</p>	<p>If the character is not found an error message will display</p> <p>Otherwise the translated string will be displayed</p>

public static void main(String[] args)		
Input	Process	Output
No input	<p>While loop runs while the variable "programRunning" does not equal 4</p> <p>The while loop calls the selection method while passing the mainMenu method.</p>	When the program has be "quit" by the user an exit message is displayed.