Delroy Mathieson

Dr. Rivas

Project 2: Final Write-Up

12 December 2016

For the semester project, I chose to code an application based project and developed a Text to Binary converter. This program involves encoding binary data in a sequence of characters and using ASCII codes to translate and represent the numeric and alphabetic characters. ASCII will also verify that each character is defined by eight bytes. The UML diagram consist of the main methods and classes that I will use when coding the program. It includes the value of the binary text, the StringBuffer method and inputs needed for the program. So far working on my project I came up with the necessary classes I will need to use when programming the converter. My code needs to consist of a Scanner that will scan the user input, a printed function to notify the user to enter a word and a variety of string related classes and methods. The string class provides a method for dealing with character values. The String byte[] bytes constructor constructs a new string by decoding the specified array of bytes using the default character set. After doing some research I found that I would need to use the getByte and stringBuffer or builder to convert english text to binary numbers. The getByte method encodes the string into a sequence of bytes using the

character set resulting in a new byte array. This can be used to represent character strings and implement them to eight bytes of zero and ones. Stringbuffer supports mutable strings; because string objects are immutable they can be shared. String Buffer and builder are similar in the sense that the classes store object data that can be modified. Regardless of being slower in performance compared to stringBuilder I decided to use StringBuffer because of its multiple thread use and it being safer than StringBuilder. The append method is used when programming with the stringBuffer method and is used when the addition operator is used on a string object. Some of the code functions I intend to use are if, else and elseif which would be used to save information on the English alphabet and its binary supplement and return the correct value. My next step will be using a for loop that consist of if, else statements to compute the math logic to convert the string to binary numbers. In the end, when the program is complete, it will ask the user to input a word or string of characters and when entered the program will display the corresponding binary numeric code for the characters. To prevent errors the user may not enter binary code as a string, however the user is able to use special characters and keys.

After finishing my program, I was able to successfully code a text to binary conversion program. The program displays the user to enter a word then converts the entered word into a series of binary characters. After furthering my research on converting text to binary I decided to use if/else statements nested in a for loop to compute and compile the conversions needed to translate any character from english text to binary language of zero and ones. The for loop verifies that the outputted numbers are less than eight bytes but have a value up to 128 bytes. Then the if statement checks if the values of the operator are equal or not and uses the left shift operator to shift the value to the left by the number of bits specified. After all the computations are completed the System.out.print function is used to display the completed binary conversion after the user input was entered. The end product of my program came out how I wanted it but it could have been greatly improved. My next steps is to develop coding for the converter to accept full sentence strings and to be able to convert binary characters to text. This should make my program more efficient and a better overall useful program.