Overview

In this lab, we will write code that will take the contents of a text file, encode and write them to a binary file and the read from the binary file and decode the message. Once decoded, it will then print the result.

Specifics

This lab will contain two interfaces, IEncoder and IDecoder, and three classes: Encoder which will implement IEncoder, Decoder which will implemet IDecoder and a main class which is given to us and will run the program.

The encoder class will have a method that will accept two parameters, both of which are file names, one for the text file and another for the binary file. The purpose of the method is to:

1. The first task of this method is to open the text file and read it to a string using a try-catch block that will throw an error if a file not found exception is thrown
2. It will then print out the contents of the file and a message letting the user know that it is encoding the file
3. Next , it will establish a connection with the binary file using the FileOutputStream and DataOutputStream
4. Now, it will use a for loop, with the variable (i) and the condition while less than the length of the string, to go through the entire string that was created from the file in step 1 where it will pull a random integer (n) between 1 and 20, convert the character at (i) in the string to unicode. Then, it will write the unicode value of the char, the integer (n), and then it will write (n) empty bytes of data to the file.
5. Once out of the loop, it will write the integer -1 to the file to signify the end of the message.
6. It will then close the file and return to the main method.

The decoder class will have a method that will accept the binary file name as a parameter. The purpose of this method is to:

As this is a pair assignment, Gavin will write the entre encoder class and Jacob will write the entire decoder class.

Encoder Class

Use Scanner to store input .txt file into a String

Use a RandomAccessFile to write contents of coded message into a binary file (surround all RandomAccessFile code with try/catch for IOExceptions)

```
For Loop (for length of input String){

        Use a Random() object to get a random number between 1 and 20
        Create a Byte array the size of the random integer
        Then use the Random() object to fill the Byte array with random values

        Use RandomAccessFile to write 1 character from the String
        Then use the RandomAccessFile to write the random int we generated
                We should check if we are at the last location, because then instead of the
                 random int we should write -1 to indicate the end of the file
        For loop (for length of Byte array){
                Use RandomAccessFile to write 1 byte from the array into output file
        }
}
```

Decoder Class

Use RandomAccessFile again to read from the encoded file

Use a boolean to simply check if we’re at the end of the file or not

```
While loop (while we’re not at the end)

        Print a character
        Store the following integer in a variable
        Check if the integer is -1
                If it is, we set the end to true
        If it’s not -1, we go into another loop
                We will simply skip the cursor ahead to the next character, by doing readByte()
for             however many times the Int specifies
```

And that’s it. Use a try/catch around everything for IOExceptions

Main

Set up an Encoder and Decoder
Specify file path for the input
Then just run the encode() and decode() methods
Done