Jason Anrico

Lab 4

11/23/2019

**Vigenere Cipher Program**

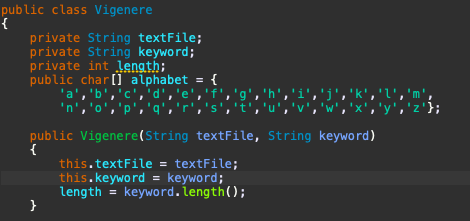
**Introduction:**

This assignment was super interesting to create. I decided to develop mine in Java since I am taking a Java course here at Southern. For starters and for someone who doesn’t know what a Vigenere cipher is, ill explain. The idea behind a Vigenere Cipher is to disguise plaintext so an intruder cant read plaintext of a message. The formula is P+K=C where P is the position of plaintext, C is the ciphertext sequence, and K is the key letter in the alphabet where the positions are numbered from 0 to 25 or modulo 26. This program I wrote allows you to choose to Encrypt a text file and Decrypt the encrypted text file.

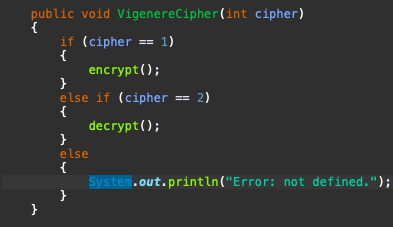
**Design and Implementation:**

1. From the start I new that I had to have two different classes

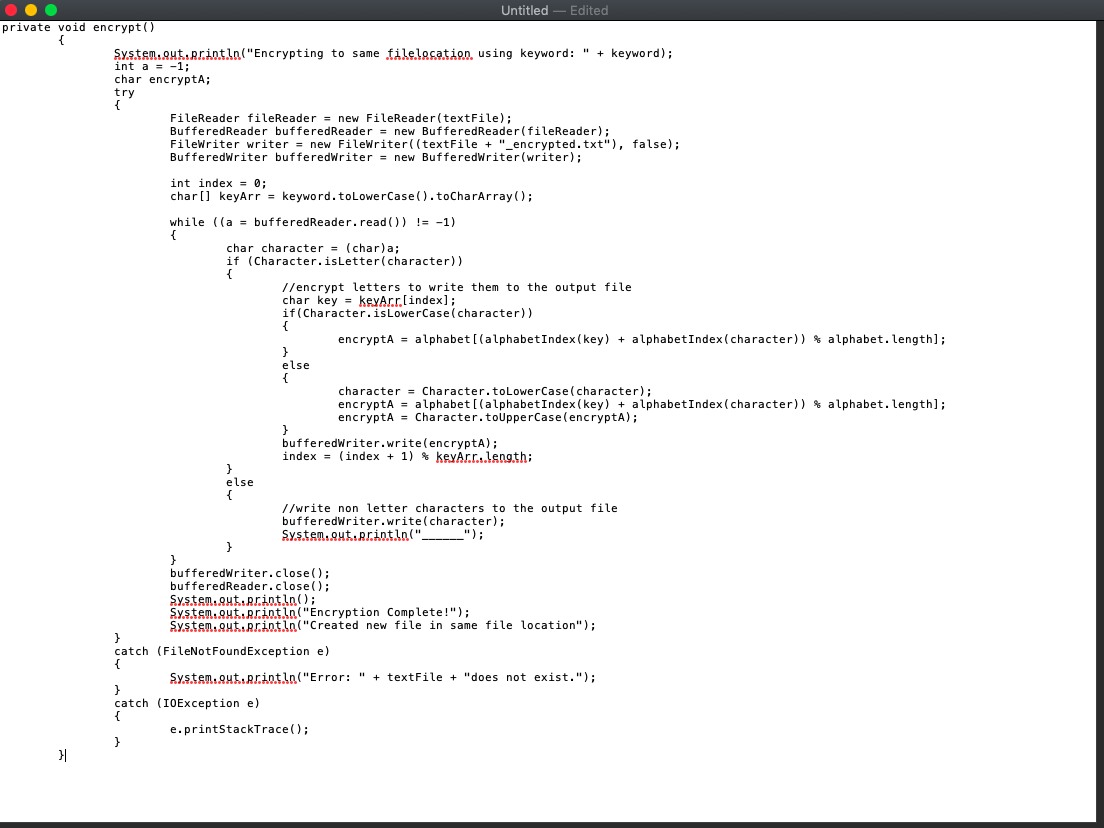
* Vigenere.java (constructer class)
* Execute.java (main function to run program)

1. 

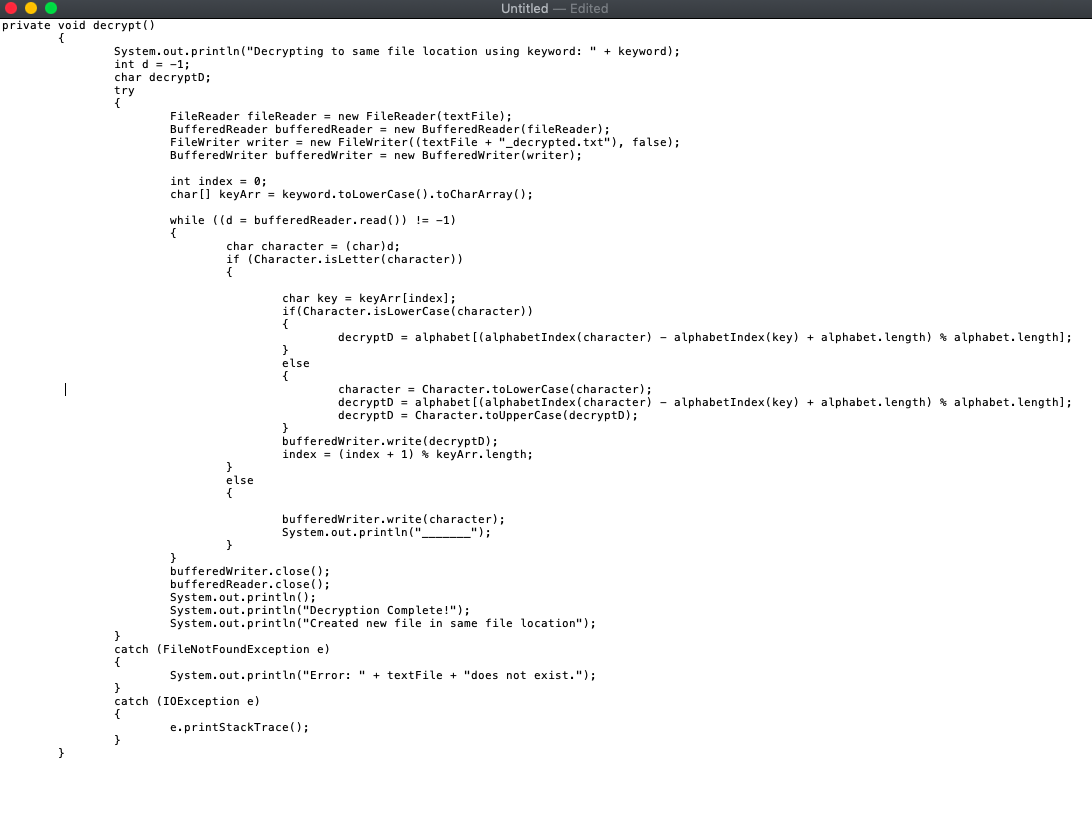
* So what I did here was declare my inputs and made a character (char) array [] so my program knew what the alphabet is. (Computers are smart right)

1. 

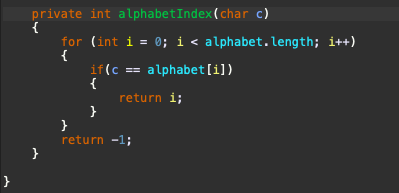
* Here I made a public void and passed the integer cipher stating that
  + If the cipher = 1 I am encrypting the message passing the encrypt() method.
  + If the cipher =2 I am decrypted the message passing the decrypt() method,
  + If neither of those are selected you’ll be given an error message

1. 

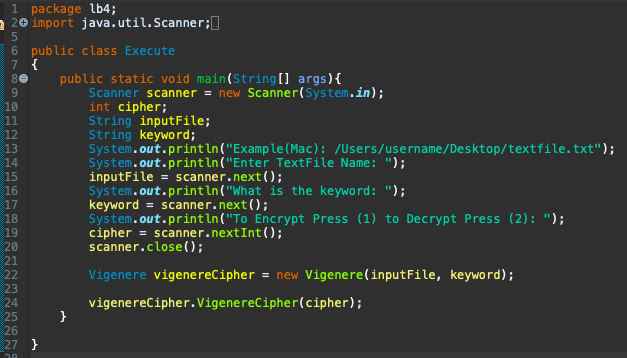
* Now the fun part! Encryption. I had to copy this into TextEdit since it couldn’t get a full screenshot of the method in Eclipse. But ill go through each part of it.
* As you see we made a private void called encrypt()
* We initialized the variable a to -1 and we read file name.
* We then use the FileWriter method in java to write our new file with the encrypted message and gave it the extention \_encrypted.txt.
* we need to start at index 0 which is why index is set to 0.
* our character array was named to keyArr.
* While our variable a = bufferedReader it is not equal (!=) -1
  + All the if else statement does is decides if characters are uppercase or lowercase and follows suit by encrypting the message to the file using the modulo of the alphabet. If it executes correctly you will see Encryption Complete! Created new file in same file location!
    - Also need to mention I had to close the bufferedWriter and reader.
  + If not I used a try catch to tell the user the file doesn’t exist and throws a IOException error with the stacktrace.

1. 

* Now onto Decrypting the encrypted file. (I had to do the same thing here with TextEdit)
* There isn’t much to really explain on this part, it essentially does all the same things as the encypt() method. Except you see that we had to subtract the alphabetIndex(key) method then add by the alphabets array length and take the modulo of the alphabet length. (I tried using simple variable names to make this as intuitive as possible). Which then gives us the hidden message in the encrypted file.
* We have the same try catch methods and also close the writer and reader.

1. 

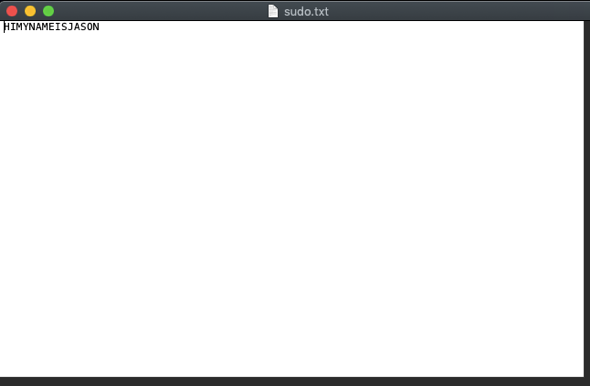
* Now our final method which is just the index of the alphabet. I passed char c to this function by if you were to make this on your own you can use any letter you wish. (Like I said I tried leaving things basic)
* We used a for loop to iterate through the alphabet using i as our index.
* We had just finished a project like this in my CSC 229 class so this made the most sense to me.

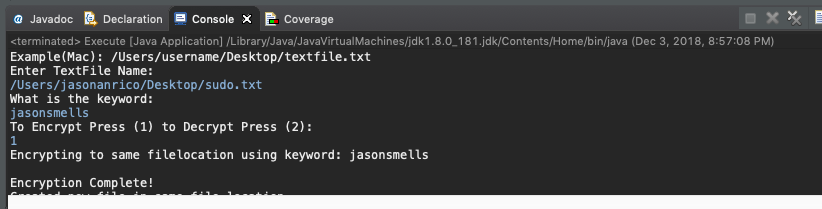
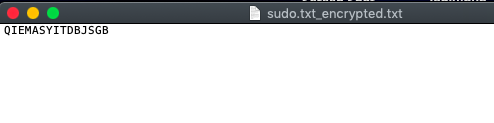
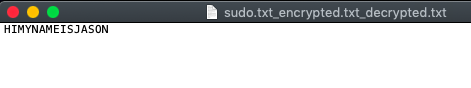
1. 

* Now we have our main method. (This essentially runs the program)
* I obliviously used Scanner to scan files
* Named my key variables
* And asked for input from the user (I figured it would be a good idea to show an example before the user types in textfile.txt and receive an error! Remember to put the file path! Also I couldn’t put a windows example since apparently you cant use \ with system.out.println(). )

**Test:** (I will be using the same file name and file path just editing the text inside)

* **Test1:**

****

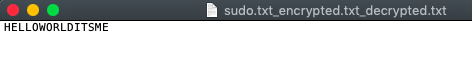
* As you can see here I have a text file named sudo.txt (I got out of my SysAdmin class and had the word sudo stuck in my head) and it reads “HIMYNAMEISJASON”
* we then do the same thing and run /Users/jasonanrico/Desktop/sudo.txt\_encrypted.txt and receive (Works like a charm)
* **Test #2:**

- Here I used the same text file I just edited it to say “HELLOWORLDITSME”

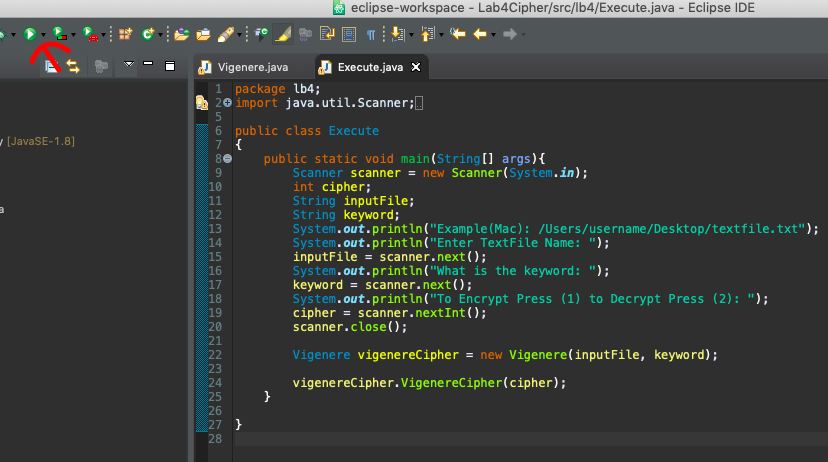
Encrypted file (keyword: hellomyname)

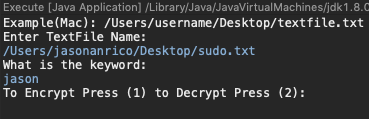


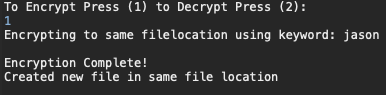
Decrypted file

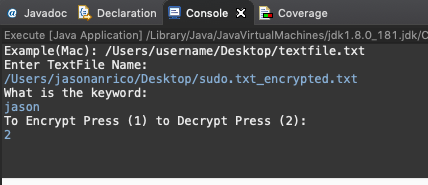


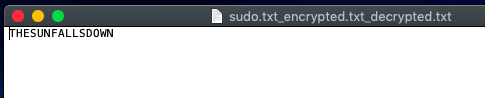
**User Manual:**

1. As a user all you need to do is execute or run the Exectue.java file. 
2. Know the name and location of your text file to be encrypted



* Since we are encrypting I’m going to press 1 and enter 
* Now im going to check my desktop for a file named sudo.txt\_encrypted.txt
* Look There it is! Now were going to open it to make sure our message got encrypted. Look at that it did!
* Now we will decrypt this file MAKE SURE YOU DECRYPT THE ENCRYPTED FILE NOT THE ORIGINAL. Select 2 then enter.



* Make sure the file is in the same location 
* And now we have our decrypted message!
* Beautiful.