**CMSC214 Project 3**

**Content: Project 3 | Part 1 - Ch 16 & Part 2 - Ch 17**

**Lessons Learned, Screenshots for test cases, Checklist**

**Class: CMSC214 - CRN: 22626**

**Instructor: Mark Estep**

**Description:**

**Part 1 - Ch 16: A program that uses scroll bars to select the foreground color for a label.**

**Part 2 - Ch 17: A program that prompts the user to enter an input file name and an output file name and saves the unencrypted version of the input file to the output file.**

**Due: 09/27/2020**

**Student name: Derya Ozdemir Kurin**

**MC ID: M21091005**

**Part 1 - Ch 16:**

**Lessons Learned:**

During this project I learned to work with scroll bars of Control class in JavaFx. I worked on the layout to give them horizontal orientation. After examining the class properties and methods of scroll bars, I made use of min, max values and set them to the max and min values of rgba color values which were 0.0 and 1.0. I realized that setting the block or unit increment values for scroll bars did not change the behavior, so I did not use these properties in my code. It became clear to me, in order to change the color of a node in my code, I needed to set only min max values, write an event handler method and finally add event listeners. I did not need to bind any properties but achieve the required feature by just adding event handlers.

I found out that in order to use the label and the color properties in methods, I needed to declare them at the top of my program in the data field, so that I could use them throughout my program especially in the handler method.

As the handler method I wrote a setColor method and call this method every time a scrollbar value is changed through event listeners.

To place the scroll bars, I utilized a Grid Pane and as required in the instructions I used a Border Pane and place all the children nodes of the program to this outer pane.

P.S Since the scrollbars are all set to 0.0 at the start of the program, changing one of the colors make the text disappear until the opacity value should be increased enough to see the text.

**Screen Shots:**

**A picture containing graphical user interface

Description automatically generated\A picture containing graphical user interface

Description automatically generated**

**A picture containing diagram

Description automatically generatedA picture containing diagram

Description automatically generated**

**Checklist:**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N** | **Comments** |
|  | Source java files |  |  |
|  | Files: |  |  |
|  | FirstInitialLastName\_Project03\_16.zip | **Y** |  |
|  | FirstInitialLastName\_Project03.doc | **Y** |  |
|  | Program compiles | **Y** |  |
|  | Program runs | **Y** |  |
|  | Checklist is completed and included in the Documentation | **Y** |  |
|  | Documentation file: |  |  |
|  | ~~Comprehensive Test Plan~~ |  |  |
|  | Screenshots based on Test Plan | **Y** |  |
|  | ~~UML diagram~~ |  |  |
|  | Lessons Learned | **Y** |  |
|  |  |  |  |
|  | Checklist | **Y** |  |

**Part 1 - Ch 17:**

**Lessons Learned:**

With this project I exercised to read from and write into a file. I also learned the ways to encrypt a file and save it with the read() method of BufferedInputStream and write() method of the BufferedInputStream. read() method returns the byte value of the characters and encrypting it is possible by simply applying the same value changing formula to all the returned byte values.

In order to read all the values in a file, we utilize a while loop and terminate it when the next byte value is -1 which indicates that the read method reached at the end of the file.

In order to decrypt a file we need to know the exact formula that crypted the file and can achieve it by reversing the value change.

It is also important that if the File is not found with the asked name, the program throws a FileNotFoundException.

**Screen Shot:**

Graphical user interface, application

Description automatically generated

Text.txt is the original file having human readable text in it.

EncryptedText.txt is the file that was created after running the given code in the instructions.

Decrypted.txt is the file which was saved after my program ran with the appropriate inputs.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N** | **Comments** |
|  | Source java files |  |  |
|  | Files: |  |  |
|  | FirstInitialLastName\_Project03\_17.zip | **Y** |  |
|  | FirstInitialLastName\_Project03.doc | **Y** |  |
|  | Program compiles | **Y** |  |
|  | Program runs | **Y** |  |
|  | Checklist is completed and included in the Documentation | **Y** |  |
|  | Documentation file: |  |  |
|  | ~~Comprehensive Test Plan~~ |  |  |
|  | Screenshots based on Test Plan | **Y** |  |
|  | ~~UML diagram~~ |  |  |
|  | Lessons Learned | **Y** |  |
|  |  |  |  |
|  | Checklist | **Y** |  |