**CIS 163**

**Project 4 – Mix and UN-mix**

**Use a Double Linked list with no tail pointer.**

**Purpose:**

You have been hired by the NSA (National Security Agency) and you are a team member of the Java Spy team. The NSA wants the ability to encrypt messages (mix up a message) and decrypt (unMix up a message) so they can communicate with other NSA spies in secret. However, all the standard approaches have been compromised (watch the 1992 movie called Sneakers, or quantum computing). So, they have created a way to mix up a message and un-mix the message using the following approach described below. Good luck, this project will self-destruct in 5 minutes ☺.

**Due Date**

* At the beginning of the lab; see the schedule, last page of the syllabus.

**Before Starting the Project**

* Review all the Chapters that were covered in the CIS163 book
* Read this entire project description before starting

**Learning Objectives**

After completing this project you should be able to:

* Create and manipulate linked lists and structures
* Create your own OO design
* Write to and read from text files

**Program Description**

Your assignment is to develop two programs (named: Mix.java and UnMix.java) that allow users to "mix" a secret message and un-mix the message, so to retrieve the original message. A simple demonstration of the program will be done in class. In more detail, first create a program called "Mix" that will:

1. Accept a string as input from the command line.
2. Place that string into a linked list. **Use a double Linked list with NO tail pointer where each node represents a single character.**
3. Preform the commands given by the user
4. run UnMix progam that retrieves the original message.
5. Use the following to guide your program’s users experience.

**A Very Simple example using the command above and executing the "Mix" program:**

**(a more full and complex example is below)**

java mix This is a secret message (in IntelliJ, you will have to configure the run/arguments)

Message:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

T h i s i s a s e c r e t m e s s a g e

Command:

b a 0

Message:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

a T h i s i s a s e c r e t m e s s a g e

Command:

Q Key

Final mixed up message: “aThis is a secret message “

**(Notice: the mixed-up message should be in double quotes because of spaces in the message)**

**Here is the initial set of commands:**

**Q filename** means, quit! (Important, please print to the screen the final mixed up

message when the program quits.) Also it means, to save off the set of undo commands into text file named "filename".

**b s #**  means, insert String “s”' at position # (this is a insert before command)

(For example, b abc 3 would insert abc starting at position 3)

**r # \***  means, remove all the characters within the message, range # \*

(for example: r 3 5 would start at position 3 and remove 3,4,5)

H Display a help page.

d # Delete all the # in the message (# is one character)

r # \* Replace all the # with \* in the message (#, \* are each one character)

z Randomize it, runs a series of b, r, d, and r commands (see above) randomly, multiple number of times.

The exact contents of what to save off into a text file are up to you. For the example below, the text file could contain what the user entered: "b a 0". **Also note no other text files (or any other type of data file) can be created.**

**Now, the second part of the assignment, is to create a simple unMix program. Here is an example using the example above and executing the "UnMix" program:**

The second program will "Unmix" the message, so to retrieve the original message. This program will be called "UnMix" and only needs two inputs to run from the command line:

java Key “Tahis is a secret message “

**(Notice: the mixed-up message should be in double quotes because of spaces in the message)**

output:

The Original message was: This is a secret message

**Finally, here are some more mix up commands that are required for this project:**

**p # &**  means, paste from clipboard &, start at #

**c # % &** means, copy to clipboard &, starting at # to % (inclusive)

**x # % &** means, cut to clipboard &, starting at # to % (inclusive)

***IMPORTANT Requirement 1:*** There are multi-clipboards, i.e., starting at 0. **You must use a single linkedList with a tail to handle multi-clipboards.** Create a linked list of "New Nodes" that point to each clipboard, where each clipboard is a linked list. Your instructor will explain more about this during class. This will require creating a "New Node" class that looks something like this:

public class ClipBoardData {

private int clipboardNumber;

private ClipBoardData next;

private LinkedList topOfClipBoard;

…

}

A visual representation of using a ClipBoardData class:

Top

Null

9

1

3

**Create a new single LinkedList class with a tail (clipBdLinkedList)**

private ClipBoardData top;

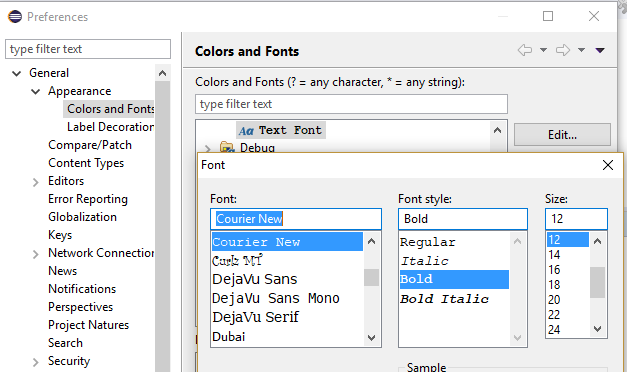
private ClipBoardData tail;

create all needed methods.

**Note:** Use the linked list above wisely and only store the needed clipboards. In other words, if the user has created 3 clipboards (e.g., 3, 1, 9), then only have 3 ClipBoardDatas in your list.

A simple solution for multi-clipboards is to use a hashtable. You can use a hasbtable <Integer, LinkList> if you can not solve the above requirement. -10 reduction in points.

***IMPORTANT Requirement 2:*** Some help for printing out the message, this can be tricky.



System.***out***.print ("Message:\n");

**for** (**int** i = 0; i < userMessage.length(); i++)

System.***out***.format ("%3d", i);

System.***out***.format ("\n");

**for** (**char** c : userMessage.toCharArray())

System.***out***.format("%3c",c);

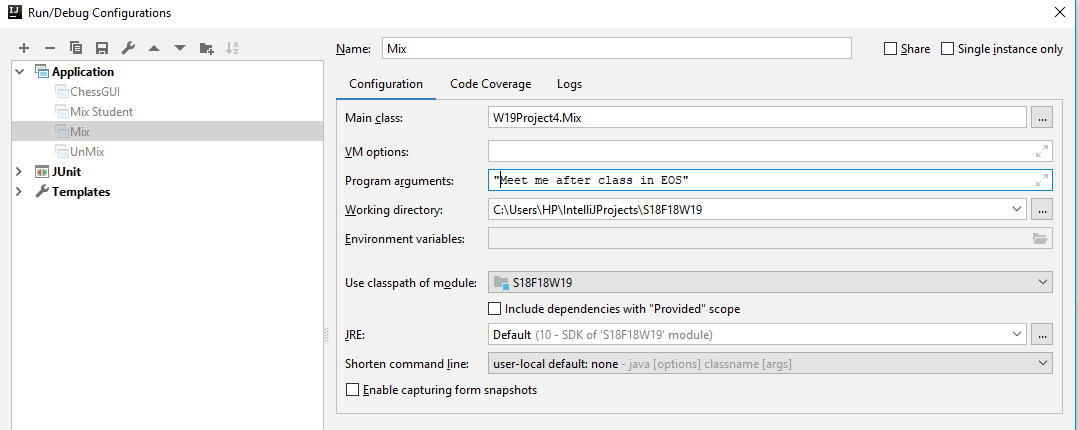
System.***out***.format ("\n");

***IMPORTANT Requirement 3:*** **Do not import the API’s LinkedList.**

***IMPORTANT Requirement 4:*** **Be able to handle spaces in all cases. Input/output, cutting, etc…**

***IMPORTANT Requirement 5:*** **Full Error checking. See demo in class for more details.**

Here is a sample full run of the project:



Meet me after class in EOS

Message:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

M e e t m e a f t e r c l a s s i n E O S

Command: r 11 14

Message:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

M e e t m e a f t l a s s i n E O S

Command: r 5 7

Message:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

M e e t a f t l a s s i n E O S

Command: r 16 16

Message:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

M e e t a f t l a s s i n O S

Command: x 1 5 1

Clip board => 'eet a'

Message:

0 1 2 3 4 5 6 7 8 9 10 11 12

M f t l a s s i n O S

Command: p 3 1

Message:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

M f t e e t a l a s s i n O S

Command: b abc 17

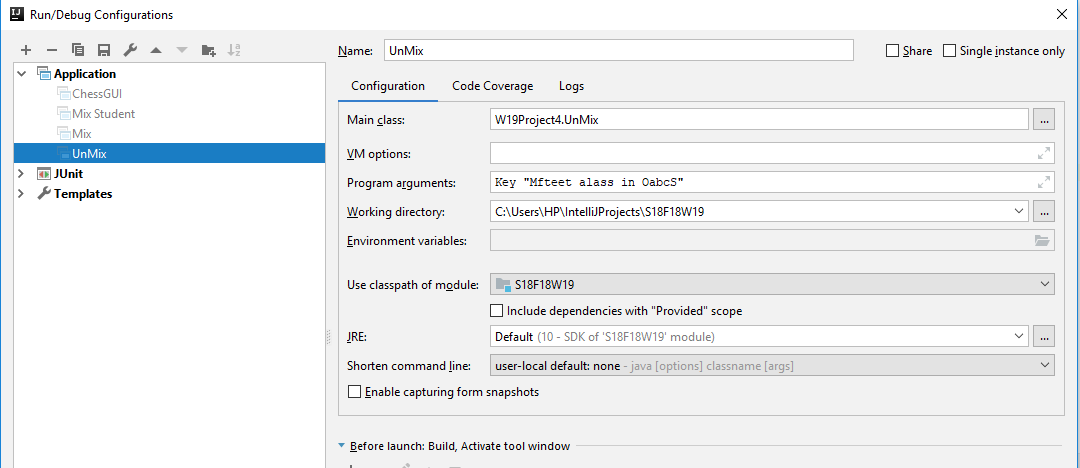
Message:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

M f t e e t a l a s s i n O a b c S

Command: Q Key

Final mixed up message: "Mfteet alass in OabcS"



The Original message was: Meet me after class in EOS

**CIS 163 – Computer Science II**

**Project 4: Mix / UnMix**

|  |  |
| --- | --- |
| Student Name |  |
| Date Submitted, Days Late, Late Penalty |  |

|  |  |  |
| --- | --- | --- |
| **Graded Item** | **Points Assigned** | **Comments** |
| Javadoc Comments and Coding Style/Technique  (<http://www.cis.gvsu.edu/studentsupport/javaguide>)  Here are some examples but not limited to:   * Code Indentation (auto format source code in IDE) * Naming Conventions (see Java style guide) * Proper access modifiers for fields and methods * Use of helper (private) methods * Using good variable names * Header/class comments * Every method uses @param and @return * Every method uses a /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* separator * Overall layout, readability, No text wrap * Using /\*\* … / for each Instance variable * Has many inner “inner” comments | **10** |  |
| **Basic mix Functionality**   * **Q filename means, quit! (Important, please print to the screen the final mixed up message when the program quits.) Also it means, to save off the set of undo commands into text file named "filename".** * b c # means, insert String “s”' at position #   + b abc 3 * r # \*means, remove all the characters within the message, range # \*   + r 3 5 would start at position 3 and remove 3,4,5 * All the rest of the commands explained above | **25** |  |
| **Multi Clipboards Functionality**   * p # & means, paste from clipboard &, start at # * c # % & means, copy to clipboard &, starting at # to % (inclusive) * x # % & means, cut to clipboard &, starting at # to % (inclusive)   Clipboard -10 pts for using hashtable for clipboards | **20** |  |
| **New commands** | **10** |  |
| **The file that contains the key undo commands** | **10** |  |
| **unMix Functionality -** Partial credit will be given for a partially functioning unMix. | **25** |  |
| **Max Possible Points** | **100** |  |
| **Extra credit, run from the command line** | **4** | **See instructor** |

**Comments:**