|  |  |
| --- | --- |
| **Name**:{Khairul Rizqi Bin Mohd Shariff}  **Tutorial Group ID**: {W15.} | NUS Ez-Link Card Photo |

# Code

/\*\*

\* This class is used to store the main function

\* **@author** Khairul Rizqi Bin Mohd Shariff

\*/

**import** java.io.IOException;

**public** **class** TextBuddy{

**public** **static** **void** main(String[] args) **throws** IOException {

TextBuddyDriver instance = **new** TextBuddyDriver();

instance.bootUp(args[0]);

}

}

/\*\*

\* This class is used as the driver function for the TextBuddy

\* Implemented all the 5 operations which is to add, display,

\* delete, clear and exit

\* @author Khairul Rizqi Bin Mohd Shariff

\*/

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class TextBuddyDriver {

private final static String WELCOME\_MESSAGE\_PART\_1 = "Welcome to TextBuddy. ";

private final static String WELCOME\_MESSAGE\_PART\_2 = " is ready for use";

private final static String ADD\_COMMAND = "add";

private final static String DISPLAY\_COMMAND = "display";

private final static String DELETE\_COMMAND = "delete";

private final static String CLEAR\_COMMAND = "clear";

private final static String EXIT\_COMMAND = "exit";

private static String filename;

static int lineCounter = 1;

static FileWriter newFile;

static BufferedWriter fileWriter;

static Scanner commandScanner = new Scanner(System.in);

static FileReader fileReader;

static BufferedReader textReader;

static String emptyMessageDone = " is empty";

static String clearMessageDone = "all content deleted from ";

public void bootUp(String filename) throws IOException {

fileInitializer(filename);

TextBuddyDriver.printWelcomeMessage(filename);

commandHandler();

}

private static void commandHandler() throws IOException {

while (true) {

String commandInput = commandScanner.next();

if (commandInput.equals(ADD\_COMMAND)) {

addCommandExecution(commandScanner.nextLine());

} else if (commandInput.equals(CLEAR\_COMMAND)) {

clearCommandExecution();

} else if (commandInput.equals(DELETE\_COMMAND)) {

deleteCommandExecution(commandScanner.nextLine());

} else if (commandInput.equals(DISPLAY\_COMMAND)) {

displayCommandExecution();

} else if (commandInput.equals(EXIT\_COMMAND)) {

exitCommandExecution();

}

}

}

private static void exitCommandExecution() throws IOException {

fileWriter.close();

commandScanner.close();

System.exit(0);

}

private static void displayCommandExecution() throws IOException {

fileReader = new FileReader(filename);

textReader = new BufferedReader(fileReader);

String line;

int numberOfLines = 0;

while ((line = textReader.readLine()) != null) {

System.out.println(line);

numberOfLines++;

}

if (numberOfLines == 0) {

System.out.println(filename + emptyMessageDone);

}

textReader.close();

fileReader.close();

}

private static void deleteCommandExecution(String commandInput) throws IOException {

fileReader = new FileReader(filename);

textReader = new BufferedReader(fileReader);

int numberOfLinesToBeCopied = 0;

String line;

List<String> bufferList = new ArrayList<String>();

/\*\* Used to store all the inputs from the file to be

\* transferred to a new file without having the line to be deleted

\*/

String messageToBeDeleted = "";

while ((line = textReader.readLine()) != null) {

if (!line.substring(0, 1).equals(commandInput.substring(1))) {

bufferList.add(line.substring(3));

numberOfLinesToBeCopied++;

} else {

messageToBeDeleted = line.substring(3);

}

}

lineCounter = 1;

FileWriter afterDeleteFile = new FileWriter(filename);

BufferedWriter newFileWritter = new BufferedWriter(afterDeleteFile);

for (int i = 0; i < numberOfLinesToBeCopied; i++) {

newFileWritter.write(lineCounter + ". " + bufferList.get(i) + "\n");

lineCounter++;

}

newFileWritter.flush();

newFileWritter.close();

System.out.println("deleted from " + filename + ": \"" + messageToBeDeleted + "\"");

}

private static void clearCommandExecution() throws IOException {

newFile = new FileWriter(filename);

lineCounter = 1;

System.out.println(clearMessageDone + filename);

}

private static void addCommandExecution(String textInput) throws IOException {

String toBePrinted = lineCounter + "." + textInput;

/\*\*to add the numbering in front of the text \*/

fileWriter.write(toBePrinted + "\n");

fileWriter.flush();

lineCounter++;

/\*\*to remove the empty space before the textInput \*/

System.out.println("added to " + filename + " : " + "\"" + textInput.substring(1) + "\"");

}

private static void printWelcomeMessage(String filename) {

String printWelcomeMessage = WELCOME\_MESSAGE\_PART\_1 + filename + WELCOME\_MESSAGE\_PART\_2;

System.out.println(printWelcomeMessage);

return;

}

private static void fileInitializer(String file) throws IOException {

filename = file;

newFile = new FileWriter(new File(filename), true);

fileWriter = new BufferedWriter(newFile);

return;

}

}

# TestInput.txt

add little brown fox

display

add jumped over the moon

display

delete 2

display

clear

display

exit

# ExpectedOutput.txt

Welcome to TextBuddy. mytextfile.txt is ready for use

added to mytextfile.txt : "little brown fox"

1. little brown fox

added to mytextfile.txt : "jumped over the moon"

1. little brown fox

2. jumped over the moon

deleted from mytextfile.txt: "jumped over the moon"

1. little brown fox

all content deleted from mytextfile.txt

mytextfile.txt is empty