Drew Sweeney

Dr. Hu

CSC341

Thursday, April 7, 2022

Homework 8 Reflection:

I felt like after I emailed you for help on this assignment, I was able to really fly through it. I think that I really understood this assignment and am happy that I was able to make everything work.

```
package homework8;

import javax.swing.JFrame;
import javax.swing.*;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Scanner;

public class WordCounter
{

	//initialize text field panel1
	JTextField textFieldDoc = new JTextField("Enter name of text file here:", 40);

	//initialize text field panel3 west
	JTextField pickAWord = new JTextField("Type word here hit enter: ", 40);

	//initialize text area
	JTextArea frequencyOfWord = new JTextArea("Frequency of your word: ", 20,20);
	JTextArea displayOccurences = new JTextArea(40, 40);
	JTextArea displayText = new JTextArea(40,40);

	//initialize button
	JButton analyze = new JButton("Get File & Analyze");
	JButton sorter = new JButton("Sort words based on frequency");
```

```
	//create ArrayLists
	ArrayList<String> words = new ArrayList<String>();
	ArrayList<Integer> count = new ArrayList<Integer>();

	public static void main(String[] args) throws IOException
	{
		new WordCounter("Word Counter");
	}

	//------------------------------------------------------------------------------
-----------------------------------------------

	public WordCounter(String title)
	{
		//setting width and height
		int width = 1000;
		int height = width * 9/16;

		//creates new instance of JFrame
		JFrame frame = new JFrame(title);

		//creating the frame
		//make sure it quits when x is clicked
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		//make sure frame size cannot be changed
		frame.setResizable(false);
		//makes the frame visible
		frame.setVisible(true);
		//makes the frame focusable
		frame.setFocusable(true);
		//packs all elements into frame
		frame.pack();
		//sets size of the frame
		frame.setSize(width,height);
		//sets the frame to the middle of the screen
		frame.setLocationRelativeTo(null);
		//layout
		frame.setLayout(new BorderLayout(15,15));

		//creates new instance of JPanel
		JPanel panel1 = new JPanel();
		panel1.setPreferredSize(new Dimension(300,100));
	  //panel1.setBackground(Color.RED);

		JPanel panel2 = new JPanel();
		panel2.setPreferredSize(new Dimension(300,40));
	  //panel2.setBackground(Color.ORANGE);

		JPanel panel3 = new JPanel();
		panel3.setPreferredSize(new Dimension(300,110));
	  //panel3.setBackground(Color.YELLOW);

	  //setting border layout for all panels
	  panel1.setLayout(new BorderLayout());
```

```
panel2.setLayout(new BorderLayout());
panel3.setLayout(new BorderLayout());

	//initialize text field panel1
	//JTextField textFieldDoc = new JTextField(40);
	//textFieldDoc.setBackground(Color.GREEN);
	panel1.add(textFieldDoc, BorderLayout.WEST);

	//initialize button panel1
	analyze.setBackground(Color.CYAN);
	panel1.add(analyze, BorderLayout.CENTER);

	//action of button analyze
	analyze.addActionListener(new ActionListener()
	{
		@Override
		public void actionPerformed(ActionEvent e)
		{
			String fileName = getDocName();
			try
			{
				sortThroughTextFile(fileName);
			} catch (IOException e1)
			{
				// TODO Auto-generated catch block
				System.out.println("Could not find file");
				e1.printStackTrace();
			}
		}
	});

	//action of enter button
	pickAWord.addKeyListener(new KeyAdapter()
	{
		@Override
	  public void keyPressed(KeyEvent e) {
		if(e.getKeyCode() == KeyEvent.VK_ENTER)
		{
		 String textFile = getDocName();
		 String userText = getUserEnteredWord();
		 try
		 {
				getWordFreq(textFile, userText);
			} catch (IOException e1)
		 {
				e1.printStackTrace();
			}
		}
	}

	 });

	 sorter.addActionListener(new ActionListener() {

		@Override
```

```
                public void actionPerformed(ActionEvent e)
                {
                        String textFile = getDocName();
                        try
                        {
                                sortFromHighestToLowest(textFile);
                        } catch (IOException e1)
                        {
                                System.out.println("Third one");
                                e1.printStackTrace();
                        }
                }

        });


        //initialize text field panel2 west
        panel2.add(displayOccurences, BorderLayout.WEST);

        //initialize text field panel2 east
        panel2.add(displayText, BorderLayout.EAST);

        //initialize text field panel3 west
        //JTextField pickAWord = new JTextField(40);
        panel3.add(pickAWord, BorderLayout.WEST);

        //initialize text field panel3 east
        panel3.add(frequencyOfWord, BorderLayout.EAST);

        //initialize button panel1
        sorter.setBackground(Color.CYAN);
        panel3.add(sorter, BorderLayout.CENTER);

        //add panel to frame
        frame.add(panel1, BorderLayout.NORTH);
        frame.add(panel2, BorderLayout.CENTER);
        frame.add(panel3, BorderLayout.SOUTH);
}

//gets the text within the box
public String getDocName()
{
        String userText = textFieldDoc.getText();
        return userText;
}

public String getUserEnteredWord()
{
        String userText = pickAWord.getText();
        return userText;
}

public void getWordFreq(String textFile, String userText) throws IOException
{
        //create input stream and use scanner
```

```
            FileInputStream fileIn = new FileInputStream(textFile);
            Scanner fileInput = new Scanner(fileIn);
            int counter = 0;

            while (fileInput.hasNext())
            {

                  //get the next word from file
                  String nextWord = fileInput.next();
                  //make all letters lower case and remove all unneeded letters
                  String revisedWord =
nextWord.toLowerCase().replaceAll("\\p{Punct}", "");
                  if (revisedWord.equals(userText))
                  {
                        counter++;
                  }
                  //determine if word is in the array list
                  //get index if word is in there
                  //add word to count array list
                  if (words.contains(revisedWord))
                  {
                        //get the #index of the word in words array
                        int index = words.indexOf(revisedWord);
                        //add to the place the word is stored
                        count.set(index, count.get(index) + 1);
                  }
                  else
                  {
                        //if word not found then add the word
                        words.add(revisedWord);
                        //add another to the count
                        count.add(1);
                  }
            }

            //close

            fileInput.close();
            fileIn.close();

            if (words.contains(userText))
            {
                  frequencyOfWord.setText(userText + " is used " + counter + "
time(s)");
            }

      }



      //-----------------------------------------------------------------


      public void sortThroughTextFile (String textFile) throws IOException
      {
```

```
		//create input stream and use scanner
		FileInputStream fileIn = new FileInputStream(textFile);
		Scanner fileInput = new Scanner(fileIn);

		while (fileInput.hasNext())
		{
			//get the next word from file
			String nextWord = fileInput.next();
			//put all words into displayTextArea location
			//make sure text wraps around the area
			displayText.setLineWrap(true);
			displayText.append(nextWord + " ");
			//make all letters lower case and remove all unneeded letters
			String revisedWord =
nextWord.toLowerCase().replaceAll("\\p{Punct}", "");

			//determine if word is in the array list
			//get index if word is in there
			//add word to count array list
			if (words.contains(revisedWord))
			{
				//get the #index of the word in words array
				int index = words.indexOf(revisedWord);
				//add to the place the word is stored
				count.set(index, count.get(index) + 1);
			}
			else
			{
				//if word not found then add the word
				words.add(revisedWord);
				//add another to the count
				count.add(1);
			}
		}

		//close
		fileInput.close();
		fileIn.close();

		int numberOfWords = words.size();
		displayOccurences.setText(" Number of unqiue words is: " +
numberOfWords + "\n");
		//print the results
		for (int i = 0; i < words.size(); i++)
		{
			displayOccurences.append(" " + words.get(i) + " occurred " +
count.get(i) + " time(s)\n");
			System.out.print(words.get(i) + " occurred " + count.get(i) + "
time(s)\n");
		}
	}

	public void sortFromHighestToLowest(String textFile) throws IOException
	{
```

```
            LinkedHashMap <String, Integer> map = new LinkedHashMap <String,
Integer>();

            try (BufferedReader br = new BufferedReader(new FileReader(textFile))) {
                  StringBuilder sb = new StringBuilder();
                  String line = br.readLine();

                  while (line != null)
                  {
                        String [] words = line.split(" ");
                        for (int i = 0; i < words.length; i++)
                        {
                              if (map.get(words[i]) == null)
                              {
                        map.put(words[i], 1);
                              }
                              else
                              {
                        int newValue =
Integer.valueOf(String.valueOf(map.get(words[i])));
                        newValue++;
                        map.put(words[i], newValue);
                     }
                        }
                        sb.append(System.lineSeparator());
                  line = br.readLine();
                  }
            }

            displayOccurences.setText("");

            List<Map.Entry<String, Integer>> list = new ArrayList<>(map.entrySet());

            Collections.sort(list, new Comparator<Map.Entry<String, Integer>>(){

                  public int compare(Map.Entry<String, Integer> o1,
Map.Entry<String, Integer> o2)
                  {
                        return o2.getValue()-o1.getValue();
                  }
            });

            for (Map.Entry<String, Integer> entry : list)
            {
                  displayOccurences.append(entry.getKey() + " occurred " +
entry.getValue() + " time(s)\n");
            }

      }
}
```

Output:

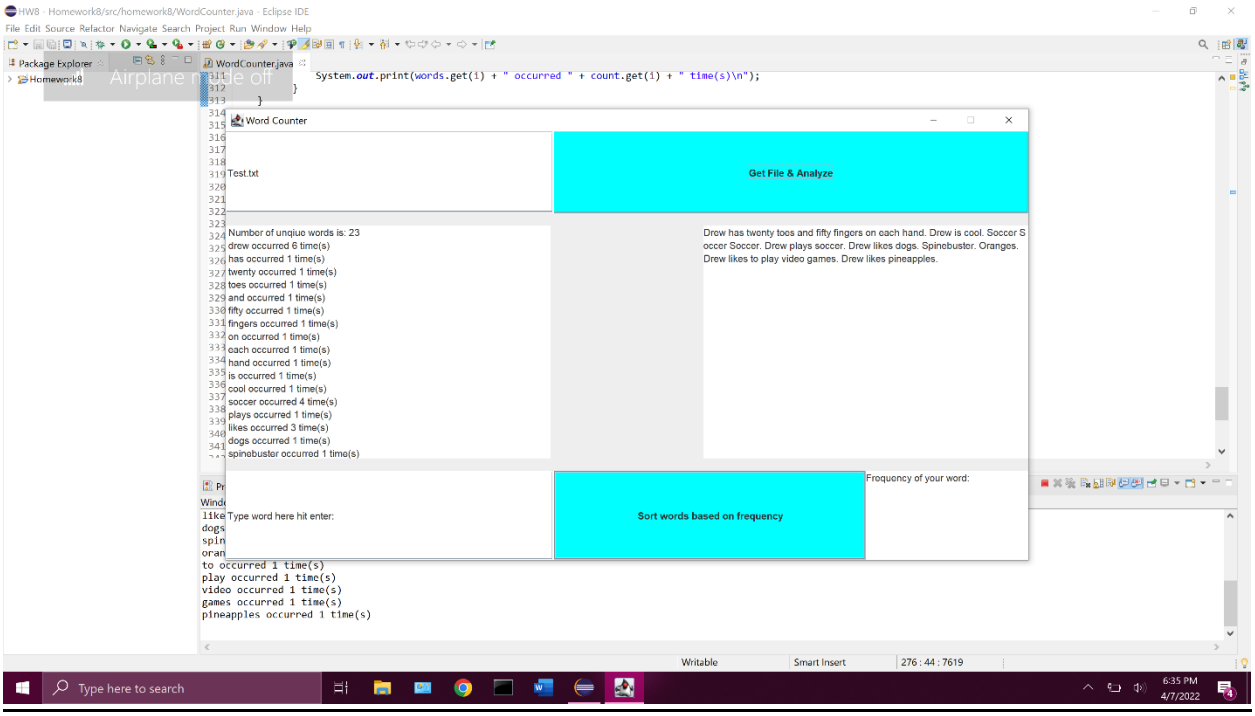

HW8 - Homework8/src/homework8/WordCounter.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Homework8
WordCounter.java
System.out.print(words.get(i) + " occurred " + count.get(i) + " time(s)\n");
}
}
Word Counter
Test.txt
Get File & Analyze
Number of unqiue words is: 23
drew occurred 6 time(s)
has occurred 1 time(s)
twenty occurred 1 time(s)
toes occurred 1 time(s)
and occurred 1 time(s)
fifty occurred 1 time(s)
fingers occurred 1 time(s)
on occurred 1 time(s)
each occurred 1 time(s)
hand occurred 1 time(s)
is occurred 1 time(s)
cool occurred 1 time(s)
soccer occurred 4 time(s)
plays occurred 1 time(s)
likes occurred 3 time(s)
dogs occurred 1 time(s)
spinebuster occurred 1 time(s)
Drew has twenty toes and fifty fingers on each hand. Drew is cool. Soccer S
occer Soccer. Drew plays soccer. Drew likes dogs. Spinebuster. Oranges.
Drew likes to play video games. Drew likes pineapples.
Type word here hit enter:
Sort words based on frequency
Froquency of your word:
to occurred 1 time(s)
play occurred 1 time(s)
video occurred 1 time(s)
games occurred 1 time(s)
pineapples occurred 1 time(s)
Writable
Smart Insert
276 : 44 : 7619
Type here to search
6:35 PM
4/7/2022

Diagram 1:

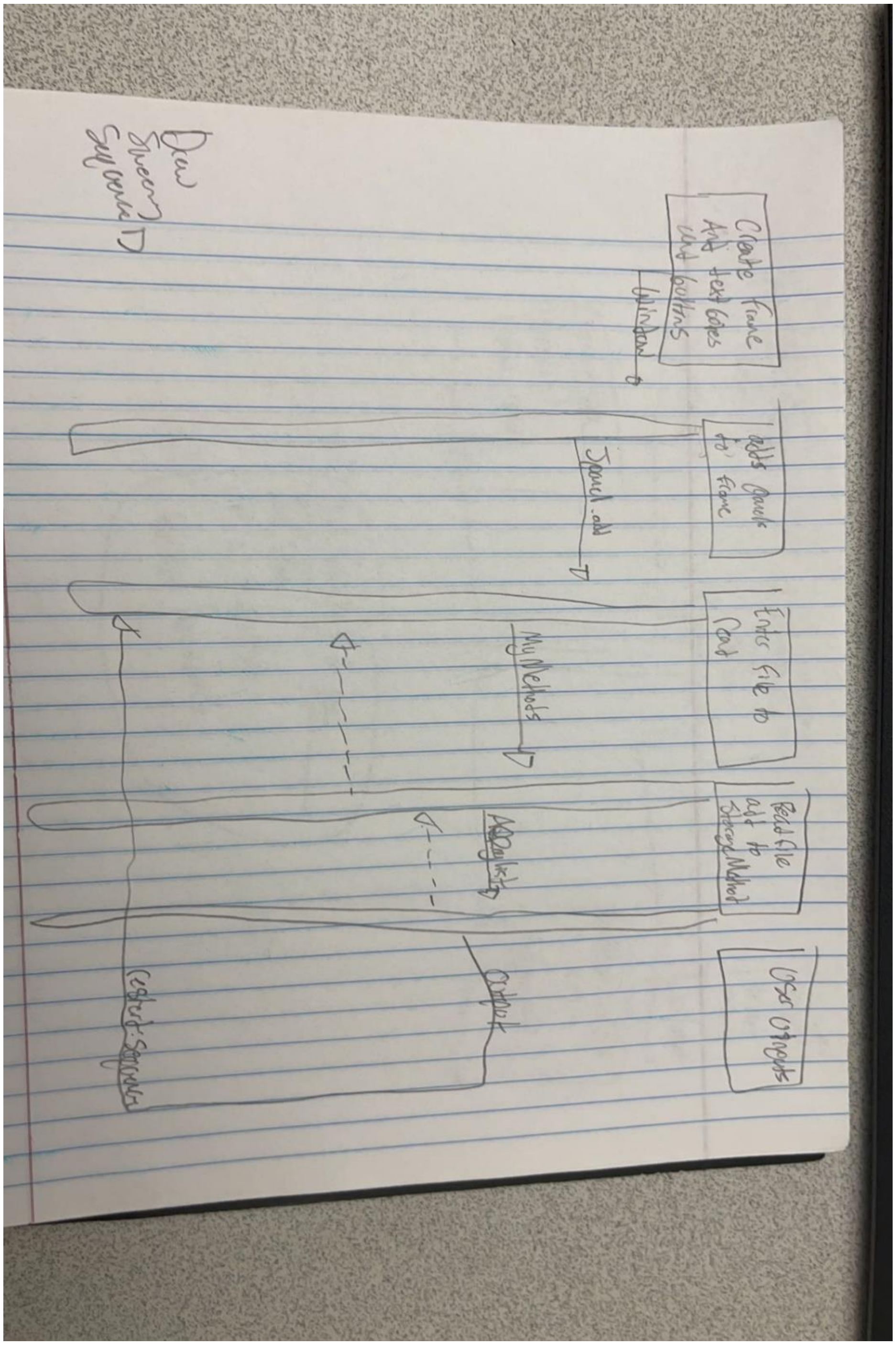
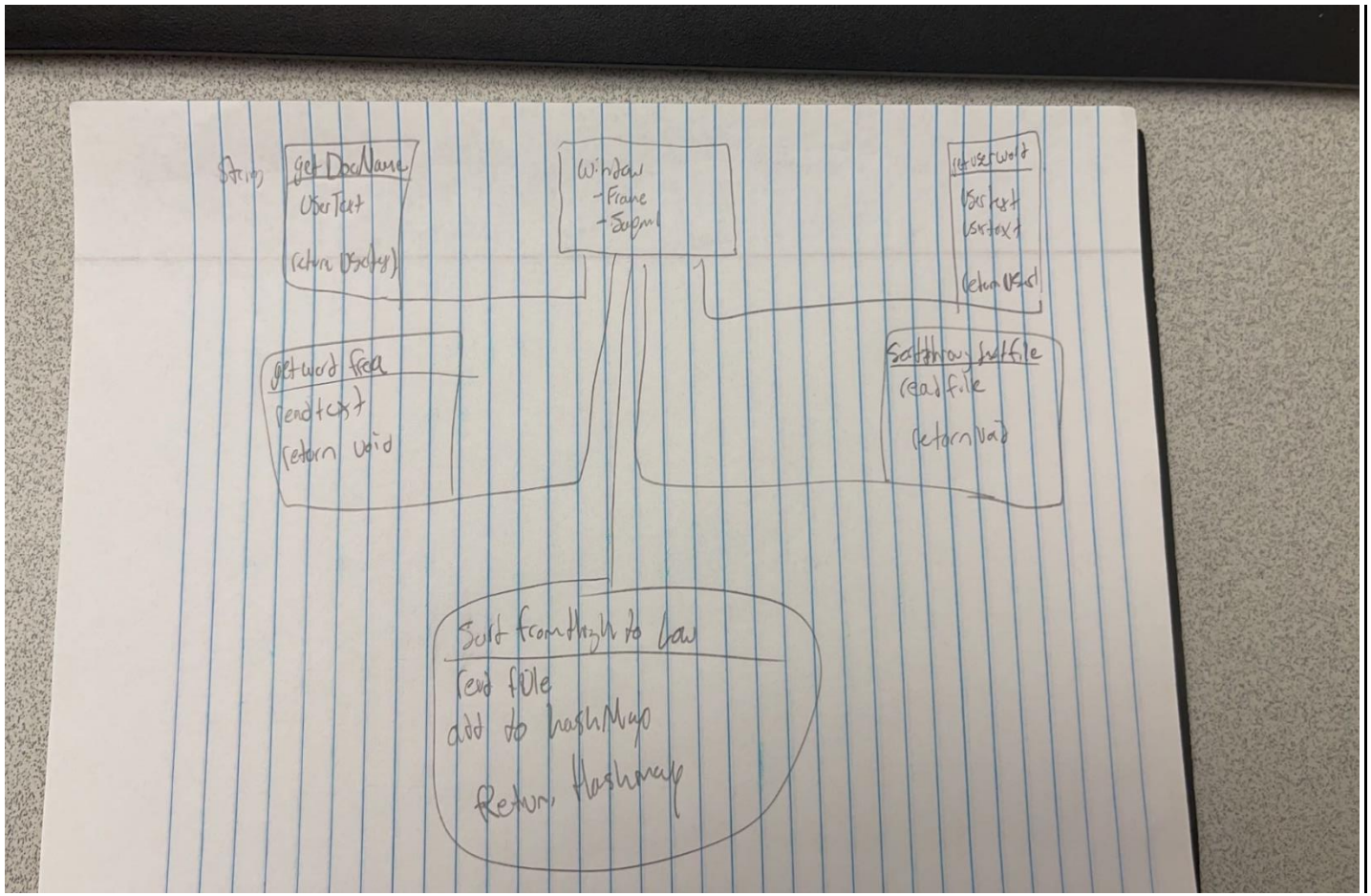


Diagram 2:


String
getDocName
UserText
return UserText
Window
Frame
Segment
getUserWord
UserText
UserText
return UserText
getWordFreq
readText
return void
SaltThrowOutFile
readFile
return Void
Sort fromHighToLow
read file
add to HashMap
Return HashMap