Blaine Killen

**FileReadTemp**

import java.io.\*;

import java.text.\*;

import java.util.Scanner;

/\*\*

\*

\* @author BlaineKillen

\*/

public class FileReadTemps extends javax.swing.JFrame {

/\*\*

\* Creates new form FileReadTemps

\*/

public FileReadTemps() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jButton1 = new javax.swing.JButton();

fileStatusLabel = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jLabel2 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jButton1.setText("Read Temperatures from File Feb05temps.txt");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

fileStatusLabel.setText("File Status");

jButton2.setText("Compute Average Temperature");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jLabel2.setText("Average");

org.jdesktop.layout.GroupLayout layout = new org.jdesktop.layout.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.add(15, 15, 15)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(jButton1)

.add(layout.createSequentialGroup()

.add(33, 33, 33)

.add(jButton2)))

.add(39, 39, 39)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING)

.add(fileStatusLabel)

.add(jLabel2))

.addContainerGap(39, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)

.add(layout.createSequentialGroup()

.add(30, 30, 30)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(jButton1)

.add(fileStatusLabel))

.add(38, 38, 38)

.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)

.add(jButton2)

.add(jLabel2))

.addContainerGap(174, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

// will hold at least 31 temperatures

int [] temperatures = new int [31];

// number of temperatures actually stored in array (between 0 and 31)

int numtemps = 0;

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

// count the number of temperatures read from the file

int count = 0;

// variable to hold each temperature

int temp;

try {

// create the input stream and the scanner

BufferedReader inputStream = new BufferedReader(

new FileReader("src/Resources/Feb05temps.txt")); Scanner inputScanner = new Scanner(inputStream);

// loop as long as there is still input in the file to read

while (inputScanner.hasNext())

{

// test if the next token in the file is an integer

if (inputScanner.hasNextInt())

{

// read the next integer in the file

temp = inputScanner.nextInt();

// store the temperature in the array and increment the count

temperatures[count] = temp;

count++;

// for debugging, we print each number as we read it:

System.out.println("Day " + count + ": " + temp); }

else {

// if it's not an integer, skip over it (this allows comments in the file)

inputScanner.next(); }

} // end while loop to read from file

// after file is read, save the number of items

numtemps = count;

// close the input stream

inputScanner.close();

}

catch (IOException e) {

// print the exception

System.out.println(e.toString()); }

// display a status message

fileStatusLabel.setText("File read done");

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

int total = 0;

double average;

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

{

total = total + temperatures[i];

}

average = total / temperatures.length;

DecimalFormat df = new DecimalFormat("00.000");

jLabel2.setText(String.valueOf(average));

}