**GUI Subsystem**

**Group 2:**

John Abueg, William Rios Crespo, Joshua Kerley, Michael Lancaster

CMSC 495 63802

**Author Note:**

Document Version GS002

Date: 2018-10-12

**Version Control**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document** | **Date** | **Action** | **Name** | **Email** |
| GS001 | 2018-09-30 | Created | Joshua Kerley | jkillakerlz@gmail.com |
| GS002 | 2018-10-04 | Modified | William Rios Crespo | william.rioscrespo19@gmail.com |
| GS003 | 2018-10-07 | Modified | Michael Lancaster | lancastermc@gmail.com |
| GS004 | 2018-10-12 | Modified | Michael Lancaster | lancastermc@gmail.com |

import java.awt.\*;

import javax.swing.\*;

import javax.swing.border.TitledBorder;

import javax.swing.filechooser.FileNameExtensionFilter;

import java.awt.event.\*;

import java.io.File;

public class GUI {

private JFrame frmGuiCalculator;

private JScrollPane scrollPane;

private JTextArea textArea;

private JLabel assignmentNameLabel = new JLabel("Assignment Name:");

private JLabel weightLabel = new JLabel("Weight:");

private JLabel gradeLabel = new JLabel("Grade:");

private JLabel valueLabel;

private JTextField assignmentNameText = new JTextField();

private JTextField weightText = new JTextField();

private JTextField gradeText = new JTextField();

private JTextField valueTextField;

private JPanel userInputPanel;

private JPanel buttonPanel;

private JPanel backgroundPanel;

private JPanel inputPanel;

private JPanel whatIfPanel;

private JPanel optionPanel;

private JButton btnInput;

private JButton btnOutput;

private JButton btnEnter;

private JButton btnCalculate;

private JButton btnEnter2;

private JButton btnClear;

private Input input = null;

private double[] whatIfNumbers = null;

private int listCounter = 0;

private boolean calculate = false;

// Launch the application.

public static void main(String[] args) {

new GUI().frmGuiCalculator.setVisible(true);

}

// Create the application.

public GUI() {

initialize();

}

// Initialize the contents of the frame.

private void initialize() {

frmGuiCalculator = new JFrame();

frmGuiCalculator.setResizable(false);

frmGuiCalculator.setTitle("GPA Calculator");

frmGuiCalculator.setBounds(100, 100, 529, 318);

frmGuiCalculator.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frmGuiCalculator.getContentPane().setLayout(new BorderLayout());

frmGuiCalculator.setLocationRelativeTo(null);

buttonPanel = new JPanel();

frmGuiCalculator.getContentPane().add(buttonPanel, BorderLayout.SOUTH);

buttonPanel.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));

btnInput = new JButton("Load File");

btnInput.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent event) {

inputFile();

}

});

buttonPanel.add(btnInput);

btnOutput = new JButton("Save File");

btnOutput.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

outputFile();

}

});

buttonPanel.add(btnOutput);

backgroundPanel = new JPanel();

frmGuiCalculator.getContentPane().add(backgroundPanel, BorderLayout.CENTER);

backgroundPanel.setLayout(new GridLayout(0, 2, 0, 0));

textArea = new JTextArea();

textArea.setEditable(false);

scrollPane = new JScrollPane(textArea);

backgroundPanel.add(scrollPane);

inputPanel = new JPanel();

backgroundPanel.add(inputPanel);

inputPanel.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));

userInputPanel = new JPanel();

inputPanel.add(userInputPanel);

userInputPanel.setBorder(new TitledBorder("User Input"));

GridLayout gl\_userInputPanel = new GridLayout(3, 2);

gl\_userInputPanel.setVgap(10);

gl\_userInputPanel.setHgap(10);

userInputPanel.setLayout(gl\_userInputPanel);

userInputPanel.add(assignmentNameLabel);

userInputPanel.add(assignmentNameText);

userInputPanel.add(weightLabel);

userInputPanel.add(weightText);

userInputPanel.add(gradeLabel);

userInputPanel.add(gradeText);

optionPanel = new JPanel();

inputPanel.add(optionPanel);

btnEnter = new JButton("Enter");

btnEnter.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent event) {

if (input == null) {

input = new Input();

}

input.userInput(assignmentNameText.getText(), weightText.getText(), gradeText.getText());

displayResults(input);

}

});

optionPanel.add(btnEnter);

btnCalculate = new JButton("Calculate");

btnCalculate.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent event) {

if (input != null) {

calculate = true;

displayResults(input);

}

}

});

optionPanel.add(btnCalculate);

btnClear = new JButton("Clear");

btnClear.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent event) {

textArea.setText("");

input = null;

}

});

optionPanel.add(btnClear);

whatIfPanel = new JPanel();

inputPanel.add(whatIfPanel);

whatIfPanel.setBorder(new TitledBorder("What-If"));

GridLayout gl\_whatIfPanel = new GridLayout(1, 3);

gl\_whatIfPanel.setHgap(10);

gl\_whatIfPanel.setVgap(10);

whatIfPanel.setLayout(gl\_whatIfPanel);

valueLabel = new JLabel("Value:");

whatIfPanel.add(valueLabel);

valueTextField = new JTextField();

valueTextField.setColumns(6);

whatIfPanel.add(valueTextField);

btnEnter2 = new JButton("Enter");

whatIfPanel.add(btnEnter2);

btnEnter2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent event) {

if (input != null) {

try {

whatIfNumbers = Calculator.whatIf(input.getGradeList(), input.getWeightList(),

Integer.parseInt(valueTextField.getText()));

calculate = true;

displayResults(input);

} catch (Exception exception) {

JOptionPane.showMessageDialog(new JFrame(),

"Please enter an integer value for the what-if scenario.", "What-If - Error",

JOptionPane.INFORMATION\_MESSAGE);

}

}

}

});

}

private void inputFile() {

JFileChooser chooser = new JFileChooser();

chooser.setDialogTitle("Please Load file");

chooser.addChoosableFileFilter(new FileNameExtensionFilter("Dat Files", "dat"));

int result = chooser.showOpenDialog(null);

if (result == JFileChooser.APPROVE\_OPTION) {

try {

Input input = new Input(chooser.getSelectedFile().getCanonicalPath());

input.readFile();

calculate = true;

displayResults(input);

input = null;

} catch (Exception exception) {

System.out.println("Input Exception.");

}

}

}

private void outputFile() {

JFileChooser chooser = new JFileChooser();

chooser.setDialogTitle("Write File");

chooser.addChoosableFileFilter(new FileNameExtensionFilter("Dat Files", "dat"));

int result = chooser.showSaveDialog(null);

if (result == JFileChooser.APPROVE\_OPTION) {

Output output = new Output();

File file = chooser.getSelectedFile();

String filePath = file.getPath();

if(!filePath.toLowerCase().endsWith(".dat")){

file = new File(filePath + ".dat");

}

output.inputFile(file);

output.write(textArea.getText());

}

}

private void displayResults(Input input) {

if (!(input.getAssignmentNameList().isEmpty() || input.getGradeList().isEmpty()

|| input.getWeightList().isEmpty())) {

for (int i = listCounter; i < input.getAssignmentNameList().size(); i++) {

textArea.append("Assignment Name: " + input.getAssignmentNameList().get(i)

+ System.getProperty("line.separator"));

textArea.append("Weight: " + input.getWeightList().get(i) + System.getProperty("line.separator"));

textArea.append("Grade: " + input.getGradeList().get(i) + System.getProperty("line.separator")

+ System.getProperty("line.separator"));

}

if (calculate == true) {

textArea.append(

"Maximum: " + Calculator.maximum(input.getGradeList()) + System.getProperty("line.separator"));

textArea.append(

"Minimum: " + Calculator.minimum(input.getGradeList()) + System.getProperty("line.separator"));

textArea.append("Mean: " + Calculator.mean(input.getGradeList(), input.getWeightList())

+ System.getProperty("line.separator"));

textArea.append(

"Median: " + Calculator.median(input.getGradeList(), input.getWeightList()) + System.getProperty("line.separator"));

textArea.append("Standard Deviation: " + Calculator.stdDev(input.getGradeList(), input.getWeightList())

+ System.getProperty("line.separator"));

textArea.append("Final Grade: " + Calculator.letterGrade(input.getGradeList(), input.getWeightList())

+ System.getProperty("line.separator") + System.getProperty("line.separator"));

}

if (whatIfNumbers != null) {

textArea.append("Grade (1 - 100) needed to reach what-if scenario: " + whatIfNumbers[0]

+ System.getProperty("line.separator"));

textArea.append("Weight needed to reach what-if scenario: " + whatIfNumbers[1]

+ System.getProperty("line.separator") + System.getProperty("line.separator"));

whatIfNumbers = null;

}

}

listCounter = (calculate == true) ? 0 : listCounter + 1;

calculate = false;

input = null;

}

}