package edu.mills.cs64.extraProgramming;

import java.awt.\*;

import java.text.DecimalFormat;

import javax.swing.\*;

import javax.swing.plaf.FontUIResource;

/\*\* A JFrame used for calculating grades.

\*

\* @author Nell Dosker

\*

\*/

public class CalculateGUI extends JFrame{

private final static Font DISPLAY\_FONT = new FontUIResource("Courier New", Font.PLAIN, 14);

private final static Font BUTTON\_FONT = new FontUIResource("Arial", Font.PLAIN, 12);

private final static Font SAVED\_FONT = new FontUIResource("Courier New", Font.BOLD, 14);

private static int min = 400;

private final static Dimension MIN\_SIZE = new Dimension(min,min);

private static double[] grades;

private static double[] credits;

private static double gpa;

private static JButton getCourses;

private static JButton getCategories;

private static JButton getGrades;

private static JButton getGPA;

private static JButton resetButton;

private static JButton clearButton;

private static JScrollPane displaySP;

private static JFrame frame;

private static JTextArea saved;

private static JScrollPane savedSP;

private static String categories;

private static String info;

/\*\*A display of information generated by the calculator.

\*

\*/

public static JTextArea display;

/\*\*A list of courses.

\*

\*/

public static Course[] courseList;

/\*\*Creates a new instance of CalculateGUI.

\*

\* A new calculator GUI is created, with one button and one text field visible.

\*

\*/

public CalculateGUI(){

frame = new JFrame("Grade Calculator");

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

frame.setMinimumSize(MIN\_SIZE);

frame.setAlwaysOnTop(true);

getCourses = new JButton("Get Courses");

getCourses.setFont(BUTTON\_FONT);

getCategories = new JButton("Get Categories");

getCategories.setFont(BUTTON\_FONT);

getGrades = new JButton("Get Grades");

getGrades.setFont(BUTTON\_FONT);

getGPA = new JButton("Get GPA");

getGPA.setFont(BUTTON\_FONT);

resetButton = new JButton("Reset Calculator");

resetButton.setFont(BUTTON\_FONT);

clearButton = new JButton("Clear Saved Results");

clearButton.setFont(BUTTON\_FONT);

display = new JTextArea("Calculator results appear here.");

display.setFont(DISPLAY\_FONT);

displaySP = new JScrollPane(display,

ScrollPaneConstants.VERTICAL\_SCROLLBAR\_AS\_NEEDED,

ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_NEVER);

displaySP.setPreferredSize(new Dimension(min, (min/3)));

saved = new JTextArea("Previously Calculated: ");

saved.setFont(SAVED\_FONT);

savedSP = new JScrollPane(saved,

ScrollPaneConstants.VERTICAL\_SCROLLBAR\_AS\_NEEDED,

ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_NEVER);

savedSP.setPreferredSize(new Dimension(min/3, min/3));

getCourses.addActionListener((ae) -> getCoursesAction());

getCategories.addActionListener((ae) -> getCategoriesAction());

getGrades.addActionListener((ae) -> getGradesAction());

getGPA.addActionListener((ae) -> getGPAAction());

resetButton.addActionListener((ae) -> resetButtonAction());

clearButton.addActionListener((ae) -> clearButtonAction());

GroupLayout layout = new GroupLayout(frame.getContentPane());

frame.getContentPane().setLayout(layout);

layout.setAutoCreateGaps(true);

layout.setAutoCreateContainerGaps(true);

layout.setHorizontalGroup(

layout.createParallelGroup(GroupLayout.Alignment.CENTER)

.addComponent(getCourses)

.addComponent(getCategories)

.addComponent(getGrades)

.addComponent(getGPA)

.addComponent(resetButton)

.addComponent(displaySP)

.addComponent(clearButton)

.addComponent(savedSP));

layout.setVerticalGroup(

layout.createSequentialGroup()

.addComponent(getCourses)

.addComponent(getCategories)

.addComponent(getGrades)

.addComponent(getGPA)

.addComponent(resetButton)

.addComponent(displaySP)

.addComponent(clearButton)

.addComponent(savedSP));

getCategories.setVisible(false);

getGrades.setVisible(false);

getGPA.setVisible(false);

resetButton.setVisible(false);

clearButton.setVisible(false);

frame.pack();

setVisibility(true);

}

private void getCoursesAction(){

int courses = 0;

while(true){

try{

courses = Integer.parseInt(inputDialog("How many classes do you have?"));

}

catch(NumberFormatException e){

display.selectAll();

display.replaceSelection("The course list could not be initialized.");

break;

}

resetButton.setVisible(true);

display.selectAll();

if(courses != 0){

if(courses == 1){

display.replaceSelection("You have 1 class.\n");

}

else{

display.replaceSelection("You have "

+ courses + " classes.\n");

}

createCourses(courses);

getCourses.setVisible(false);

getCategories.setVisible(true);

frame.pack();

setVisibility(true);

break;

}

else{

display.replaceSelection("The course list could not be initialized.");

break;

}

}

}

private void getCategoriesAction(){

for(Course course : courseList){

categories = course.getCategories();

display.append(categories);

}

getCategories.setVisible(false);

getGrades.setVisible(true);

}

private void getGradesAction(){

grades = new double[courseList.length];

int i = 0;

for(Course course : courseList){

String name = course.getName();

String answer = inputDialog("Is "

+ name + " weighted?");

String weights = course.getWeights(answer);

display.append(weights);

try{

double total = course.total();

display.append("\n" + name +

" Final Grade: " + total);

grades[i] = total;

}

catch(NumberFormatException e){

display.append(Course.NUMBER\_ERROR);

return;

}

i++;

}

getGrades.setVisible(false);

getGPA.setVisible(true);

}

private void getGPAAction(){

info = "Transcript: ";

for(Course course : courseList){

info += "\n" + course.getName()

+ "\nCredit: "

+ course.getCredit()

+ "\nFinal Grade: "

+ course.total + "\n";

}

display.selectAll();

display.replaceSelection(info);

gpaCalculate(courseList);

info += "\nGPA: " + gpa;

}

private void resetButtonAction() {

getCourses.setVisible(true);

getGPA.setVisible(false);

display.selectAll();

display.replaceSelection("Calculator results appear here.");

if(info != ""){

saved.append("\n" + info + "\n---------------------\n");

clearButton.setVisible(true);

}

resetButton.setVisible(false);

}

private void clearButtonAction(){

saved.selectAll();

saved.replaceSelection("Previously Calculated: ");

clearButton.setVisible(false);

}

private static String inputDialog(String message){

String userInput = "";

new JOptionPane();

userInput = JOptionPane.showInputDialog(null, message,

"Grade Calculator",

JOptionPane.QUESTION\_MESSAGE);

return userInput;

}

/\*\*Changes the visibility of the frame.

\* The frame becomes invisible or visible.

\* @param visibility If true, visible. If false, invisible.

\*/

public static void setVisibility(boolean visibility){

frame.setVisible(visibility);

}

/\*\*Creates an array of courses.

\*

\* @param courses the number of courses in the array

\*/

public static void createCourses(int courses){

courseList = new Course[courses];

credits = new double[courseList.length];

if(courseList.length != 0){

createCourses(courseList,0);

return;

}

}

private static void createCourses(Course[] courseList, int i){

String course = "";

if(i == 0){

course = inputDialog("Enter first class name");

}

else{

course = inputDialog("Enter next class name");

}

try{

double credit = Double.parseDouble(inputDialog("Enter credit for "

+ course));

credits[i] = credit;

courseList[i] = new Course(course,credit);

display.append("\n" + course + "\nCredit: "

+ credit + "\n");

i++;

}

catch (NumberFormatException e){

display.append(Course.NUMBER\_ERROR);

}

if(i != courseList.length){

createCourses(courseList,i);

}

}

private static void gpaCalculate(Course[] courses){

double credit = 0.0;

double totalGradePoints = 0.0;

try{

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

credit += credits[i];

totalGradePoints += courses[i].getGradePoints(grades[i]);

}

gpa = totalGradePoints/credit;

String pattern = "#.##";

DecimalFormat decimalFormat = new DecimalFormat(pattern);

gpa = Double.parseDouble(decimalFormat.format(gpa));

}

catch(NumberFormatException e){

display.append(Course.NUMBER\_ERROR);

}

display.append("\nSemester GPA: " + gpa);

}

/\*\*Creates an instance of CalculateGUI.

\*

\* @param args

\*/

public static void main(String[] args){

new CalculateGUI();

}

}