CodeTechSolution

Task -3 :-

```
package com.codetech.GradeManagementSystem;

import javax.swing.*;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.*;

import java.util.List;

class Grade

{

private String courseName;

private int score;

public Grade(String courseName, int score) {

this.courseName = courseName;

this.score = score;

}

public String getCourseName() {

return courseName;

}

public int getScore() {

return score;

}

@Override

public String toString() {

return courseName + ": " + score;
```

```
}

}

class Student {

private String studentName;

private DefaultListModel<Grade> grades;

public Student(String studentName, DefaultListModel<Grade> grades) {

this.studentName = studentName;

this.grades = grades;

}

public String getStudentName() {

return studentName;

}

public DefaultListModel<Grade> getGrades() {

return grades;

}

public void addGrade(String courseName, int score) {

Grade grade = new Grade(courseName, score);

grades.addElement(grade);

}

@Override

public String toString() {

return "Student: " + studentName + "\nGrades: " + grades;

}

}

public class GradeManagementGUI extends JFrame {

private DefaultListModel<Student> studentListModel;

private JList<Student> studentList;

public GradeManagementGUI() {
```

```
setTitle("Grade Management System");
setSize(400, 300);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLocationRelativeTo(null);
studentListModel = new DefaultListModel<>();
studentList = new JList<>(studentListModel);
JButton addStudentButton = new JButton("Add Student");
JButton editGradesButton = new JButton("Edit Grades");
JButton calculateGPAButton = new JButton("Calculate GPA");
JButton saveButton = new JButton("Save");
JButton loadButton = new JButton("Load");
addStudentButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
addStudent();
}
});
editGradesButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
editGrades();
}
});
calculateGPAButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
calculateGPA();
}
```

```
});
saveButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
saveData();
}
});
loadButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
loadData();
}
});
JPanel buttonPanel = new JPanel(new GridLayout(5, 1));
buttonPanel.add(addStudentButton);
buttonPanel.add(editGradesButton);
buttonPanel.add(calculateGPAButton);
buttonPanel.add(saveButton);
buttonPanel.add(loadButton);
JScrollPane studentScrollPane = new JScrollPane(studentList);
add(buttonPanel, BorderLayout.WEST);
add(studentScrollPane, BorderLayout.CENTER);
setVisible(true);
}
private void addStudent() {
String studentName = JOptionPane.showInputDialog("Enter student name:");
if (studentName != null && !studentName.isEmpty()) {
DefaultListModel<Grade> gradeListModel=new DefaultListModel<>();
```

```
Student student = new Student(studentName, gradeListModel);

studentListModel.addElement(student);

}

}

private void editGrades() {

Student selectedStudent = studentList.getSelectedValue();

if (selectedStudent != null) {

String courseName = JOptionPane.showInputDialog("Enter course name:");

if (courseName != null && !courseName.isEmpty()) {

String scoreString = JOptionPane.showInputDialog("Enter score:");

try {

int score = Integer.parseInt(scoreString);

selectedStudent.addGrade(courseName, score);

} catch (NumberFormatException e) {

JOptionPane.showMessageDialog(null, "Invalid score. Please enter a
number.");

}

}

}

}

private void calculateGPA() {

Student selectedStudent = studentList.getSelectedValue();

if (selectedStudent != null) {

DefaultListModel<Grade> grades = selectedStudent.getGrades();

int totalScore = 0;

for(int i=0; i<grades.size();i++){

totalScore +=grades.getElementAt(i).getScore();

}
```

```
double gpa = (double) totalScore / grades.size();

JOptionPane.showMessageDialog(null, "GPA for " +
selectedStudent.getStudentName() + ": " + gpa);

}

}

private void saveData() {

try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream("students.dat"))) {

out.writeObject(studentListModel);

JOptionPane.showMessageDialog(null, "Data saved successfully!");

} catch (IOException e) {

e.printStackTrace();

}

}

private void loadData() {

try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream("students.dat"))) {

DefaultListModel<Student> loadedModel = (DefaultListModel<Student>)
in.readObject();

studentListModel.clear();

for (int i = 0; i < loadedModel.size(); i++) {

studentListModel.addElement(loadedModel.getElementAt(i));

}

JOptionPane.showMessageDialog(null, "Data loaded successfully!");

} catch (IOException | ClassNotFoundException e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {
```

```
@Override
public void run() {
new GradeManagementGUI();
}
});
}
}
```