Assignment 6: Student GUI with Arraylist of Objects

IB Computer Science Assignment: Building a GUI with Student Data

Assignment Title: Creating a Java GUI with Multiple Tabs and Object-Oriented Design

Objective:
In this assignment, you will build a Java application with a graphical user interface (GUI) using `JTabbedPane`. The application will allow users to input student data, display the data in a table, sort the data by various attributes, and save the data to a file. You will use object-oriented programming principles to create two classes: one for the GUI and one for the `Student` object.

Requirements

Class 1: `StudentGUI`

1. **GUI with Three Tabs**
    - **Tab 1: Data Entry Form**
        - Create text fields to enter the following information for each student:
            - Name (String)
            - GPA (double)
            - IB Courses Taken (int)
            - Height (double in inches)
        - Add a button labeled "Add Student" to save the entered student information.
        - When the button is clicked, the entered data should be stored in an `ArrayList<Student>`.
    - **Tab 2: Display Student Data in a Table**
        - Use a `JTable` to display the list of students added. The table should display columns for Name, GPA, IB Courses Taken, and Height.
        - Whenever a new student is added, update the table automatically.
    - **Tab 3: Sorting and Saving Data**
        - Create four buttons:
            - "Sort by GPA"
            - "Sort by IB Courses Taken"
            - "Sort by Height"
            - "Save to File"
        - The first three buttons should sort the students in the `ArrayList` by the respective attribute and update the table display accordingly.
        - The "Save to File" button should save the student data to a file in a readable format (e.g., CSV or plain text).

Class 2: `Student`

- **Attributes:**
  - `name` (String)
  - `gpa` (double)
  - `ibCoursesTaken` (int)
  - `height` (double)
- **Constructor:**
  - Write a constructor that accepts the name, GPA, IB courses taken, and height as parameters and initializes the respective fields.
- **Getters:**
  - Create getter methods for all the attributes (`getName()`, `getGpa()`, `getIbCoursesTaken()`, `getHeight()`).
- Setters:
  -

Instructions:

1. **Setting up the GUI:**
   - Use `JTabbedPane` to create the three tabs.
   - For Tab 1, use `JTextField` for input fields and a `JButton` for the "Add Student" button.
   - For Tab 2, use `JTable` to display the list of students.
   - For Tab 3, create four `JButton`s to handle sorting and saving functionality.
2. **Handling Events:**
   - Implement event listeners for each button. The "Add Student" button should take the data from the form in Tab 1, create a new `Student` object, and add it to the `ArrayList<Student>`.
   - The sorting buttons should sort the `ArrayList` based on the selected criteria and refresh the table view in Tab 2.
3. **Saving to File:**
   - The "Save to File" button should write all the student data from the `ArrayList` to a file (such as a CSV file).

```java
package Assignment7;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.PrintWriter;
import java.util.*;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;

import Assignment6.Book;

public class PowerSchoolGui extends JFrame implements ActionListener
{
   private JTextField nameText, GPAText, numOfIBCoursesText, heightText;
   private ArrayList<Student> studentList = new ArrayList<>();
   private String[] colNames = {"Name", "GPA", "Height", "Number of IB courses"};
   private JTable studentTable;
   private String[][] data  = new String[30][4];
   private int numOfStudents = 0;
   private Color darkGrey = new Color(160, 169, 186);


   public PowerSchoolGui()
   {
       super("PowerSchool: Snyder edition");
       setSize(750,500);
       getContentPane().setBackground(new Color(177, 230, 252));
       setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
       setLayout(null);
       this.setResizable(false);
       JTabbedPane tPane = new JTabbedPane();
           tPane.setBounds(0,0,750,450);
           JPanel dataEntryPanel = new JPanel();
               dataEntryPanel.setLayout(null);
               JLabel nameLabel = new JLabel("Enter name: ");
               nameLabel.setBounds(30, 50, 250, 30);
               nameText = new JTextField();
               nameText.setBounds(350, 50, 250, 30);
               dataEntryPanel.add(nameText);
               dataEntryPanel.add(nameLabel);
               JLabel GPALabel = new JLabel("Enter GPA: ");
               GPALabel.setBounds(30, 100, 250, 30);
```

```
        GPAText = new JTextField("decimal is ok");
        GPAText.setBounds(350, 100, 250, 30);
        dataEntryPanel.add(GPAText);
        dataEntryPanel.add(GPALabel);
        JLabel numOfIBCoursesLabel = new JLabel("Enter number of IB courses: ");
        numOfIBCoursesLabel.setBounds(30, 150, 250, 30);
        numOfIBCoursesText = new JTextField("");
        numOfIBCoursesText.setBounds(350, 150, 250, 30);
        dataEntryPanel.add(numOfIBCoursesLabel);
        dataEntryPanel.add(numOfIBCoursesText);
        JLabel heightLabel = new JLabel("Enter height: ");
        heightLabel.setBounds(30, 200, 250, 30);
        heightText = new JTextField("decmial is ok");
        heightText.setBounds(350, 200, 250, 30);
        dataEntryPanel.add(heightLabel);
        dataEntryPanel.add(heightText);
        JButton newStudentButton = new JButton("Add New Student");
        newStudentButton.addActionListener(this);
        newStudentButton.setBounds(250, 280, 200, 50);
        dataEntryPanel.add(newStudentButton);
    JPanel displayPanel = new JPanel();
        studentTable = new JTable(data, colNames);
        JScrollPane tablePane = new JScrollPane(studentTable);
        tablePane.setBounds(100,100,400,200);
        studentTable.setBackground(new Color(61, 77, 138));
        studentTable.setForeground(Color.white);
        displayPanel.add(tablePane);
    JPanel sortSavePanel = new JPanel();
        sortSavePanel.setLayout(null);
        JLabel buttonLabel = new JLabel("The buttons sort from least to greatest:");
        buttonLabel.setBounds(250, 50, 300, 50);
        sortSavePanel.add(buttonLabel);
        JButton nameButton = new JButton("Sort by Name");
        nameButton.setBounds(20, 110, 200, 50);
        nameButton.addActionListener(this);
        sortSavePanel.add(nameButton);
        JButton gpaButton = new JButton("Sort by GPA");
        gpaButton.setBounds(270, 110, 200, 50);
        gpaButton.addActionListener(this);
        sortSavePanel.add(gpaButton);
        JButton heightButton = new JButton("Sort by Height");
        heightButton.setBounds(510, 110, 200, 50);
        heightButton.addActionListener(this);
        sortSavePanel.add(heightButton);
        JButton ibButton = new JButton("Sort by Number of IB Classes");
        ibButton.setBounds(110, 160, 200, 50);
        ibButton.addActionListener(this);
```

```
            sortSavePanel.add(ibButton);
            JButton saveButton = new JButton("Save to TXT file");
            saveButton.setBounds(420, 160, 200, 50);
            saveButton.addActionListener(this);
            sortSavePanel.add(saveButton);
    tPane.add("enter data", dataEntryPanel);
    tPane.add("display", displayPanel);
    tPane.add("sort and save", sortSavePanel);

    add(tPane);
}


@Override
public void actionPerformed(ActionEvent e) {
    if (e.getActionCommand().equals("Add New Student"))
    {
        if (numOfStudents < data.length)
        {
            String name = nameText.getText();
            double GPA = Double.valueOf(GPAText.getText());
            double height = Double.valueOf(heightText.getText());
            int numOfIBCourses = Integer.valueOf(numOfIBCoursesText.getText());
            Student student = new Student(name, GPA, height, numOfIBCourses);
            studentList.add(student);
            updateTable();
            numOfStudents++;
        }
        else
        {
            nameText.setEditable(false);
            nameText.setBackground(darkGrey);
            GPAText.setEditable(false);
            GPAText.setBackground(darkGrey);
            heightText.setEditable(false);
            heightText.setBackground(darkGrey);
            numOfIBCoursesText.setEditable(false);
            numOfIBCoursesText.setBackground(darkGrey);
            nameText.setText("Class is full!");
            GPAText.setText("Class is full!");
            heightText.setText("Class is full!");
            numOfIBCoursesText.setText("Class is full!");
        }

    }
    if (e.getActionCommand().equals("Sort by Name"))
    {
```

```
            studentList.sort(Comparator.comparing(Student::getName));
            updateTable();
        }
        if (e.getActionCommand().equals("Sort by GPA"))
        {
            studentList.sort(Comparator.comparingDouble(Student::getGpa));
            updateTable();
        }
        if (e.getActionCommand().equals("Sort by Height"))
        {
            studentList.sort(Comparator.comparingDouble(Student::getHeight));
            updateTable();
        }
        if (e.getActionCommand().equals("Sort by Number of IB Classes"))
        {
            studentList.sort(Comparator.comparingInt(Student::getIbCoursesTaken));
            updateTable();
        }

        if (e.getActionCommand().equals("Save to TXT file"))
        {
           PrintWriter outputStream = null;
            try
            {
                File file = new
File("/Users/mt25190/Desktop/CS-HL2-Projects/TaheriMya/src/powerSchoolData.txt");
                outputStream = new PrintWriter(new FileOutputStream(file, false));
                outputStream.println("");
                outputStream.close();

                outputStream = new PrintWriter(new FileOutputStream(file, true));
                outputStream.println("Name, GPA, Height, Number of IB courses \n");

                for (int i = 0; i < numOfStudents; i++) {
                    outputStream.println(data[i][0] + ", " + data[i][1] + ", " + data[i][2] + ", " +
data[i][3]);
                }

                outputStream.close();
            }
            catch(FileNotFoundException err)
            {
                System.out.println("file not found");
                System.exit(0);
            }
        }
```

```
    }

    private void updateTable() {
        for (int i = 0; i < studentList.size(); i++) {
            Student student = studentList.get(i);
            data[i][0] = student.getName();
            data[i][1] = String.valueOf(student.getGpa());
            data[i][2] = String.valueOf(student.getHeight());
            data[i][3] = String.valueOf(student.getIbCoursesTaken());
        }
    }
}
```

```
package Assignment7;

public class Student
{
    private String name;
    private double GPA;
    private double height;
    private int numOfIBCourses;

    public Student (String name, double GPA, double height, int numOfIBCourses)
    {
        this.name = name;
        this.GPA = GPA;
        this.height = height;
        this.numOfIBCourses = numOfIBCourses;
    }

    public String getName()
    {
        return this.name;
    }
    public double getGpa(){
        return this.GPA;
    }
    public int getIbCoursesTaken(){
        return this.numOfIBCourses;
    }
    public double getHeight(){
        return this.height;
    }

}
```

```java
package Assignment7;

public class PowerSchoolDriver
{
   public static void main (String[] args)
   {
       PowerSchoolGui gui = new PowerSchoolGui();
       gui.setVisible(true);
   }

}
```