

Java 程序设计实验报告

学号： L170300901

姓名： 卢兑琬

专业： 计算机

班级： L1703009

哈尔滨工业大学

实验六：文件读写程序设计与异常

一、实验目的

- 1) 了解流输入输出的基本概念。
- 2) 了解异常的概念。
- 3) 掌握 File、FileReader、FileWriter 等文件读写类库的使用。
- 3) 掌握 IOException、FileNotFoundException 等异常的使用。
- 4) 掌握 Swing 常用组件及绝对布局的应用
- 5) 掌握事件驱动编程原理及应用
- 6) 掌握菜单组件的设计与使用。
- 7) 掌握对话框(Dialog)的使用。

二、实验内容

- 1) 将 CollectionBMI 类改造为 GUIFileBMI 类;
- 2) 在 GUIFileBMI 类中增加方法 saveFile(ArrayList<Student> students, String filename), 该方法可以将所有学生信息以逗号表的形式写入到当前目录下 XXX.txt 文件中(XXX 为您的学号), 每一行写入一个学生。
- 3) 在 GUIFileBMI 类中增加方法 ArrayList<Student> readFile(String filename)读文件中的数据到学生 students 中。
- 4) 在 GUIFileBMI 类中增加 4 个界面(Panel): 学生信息主界面、学生信息输入界面、学生信息维护界面、BMI 统计界面;
 - a) 学生信息主界面为默认界面, 该界面可从文件中读取并显示所有学生信息, 并按不同属性对学生排序显示。该界面还可随机生成 200 名学生, 并保存到文件中。
 - b) 学生信息输入界面每次输入单个学生, 保存后, 提示用户是否成功保存或提示用户已经存在, 请重新输入(利用 Dialog 对话框进行提示)。
 - c) 学生信息维护界面要求输入要维护的学生学号, 按学号查询并显示该学生所有信息, 可删除或修改学生信息, 并提示用户修改/删除成功或失败 (利用 Dialog 对话框进行提示)。
 - d) BMI 统计界面显示 bmi 的相关统计信息 (均值、中值、众数、方差等), 并将 bmi 值的范围由小到大划分的 10 个均分区间, 统计每个区间的学生人数, 并绘制出相应的柱状图(利用 JFreechart 实现)。
- 5) 在 GUIFileBMI 类中增加**菜单**, 可通过菜单切换显示上述 4 各不同界面。
注意, 身高、体重、及 bmi 等数值均需保留两位小数的格式进行存储和显示。

三、实验步骤

注意：将程序代码和运行结果截图粘贴在此处，注意源代码中注释行数不少于全部代码的1/3，程序源代码请压缩后上传，压缩文件按照 学号.zip 进行命名，注意源程序于报告请分别上传到不同的文件夹中！

```
package edu.hit.java.exp4;

import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Random;

import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.JToolBar;
import javax.swing.border.EmptyBorder;

import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartFrame;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
```

```

import org.jfree.data.xy.IntervalXYDataset;
import org.jfree.data.xy.XYBarDataset;
import org.jfree.data.xy.XYSeries;
import org.jfree.data.xy.XYSeriesCollection;

public class L170300901 extends JFrame {

    public static ArrayList<StudentInfo> students = new ArrayList<StudentInfo>();
    private JTextArea textAreaStudentsInfoMain = new JTextArea();
    private JPanel contentPane;
    //private static final String ProgramLanguage = "Korean";
    private static final String ProgramLanguage = "Chinese";

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    L170300901 frame = new L170300901();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    public L170300901() {
        setTitle((ProgramLanguage == "Chinese") ? "学生信息输入" : "학생 정보 입력");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 800, 620);
        setResizable(false);

        // Menu
        JMenuBar menuBar = new JMenuBar();
        JMenuItem menuItemPrintStudentsInfo = new JMenuItem((ProgramLanguage ==
"Chinese") ? "学生信息显示及录入" : "학생 정보 출력");
        JMenu menuMenu = new JMenu((ProgramLanguage == "Chinese") ? "菜单" : "메뉴");
        JMenuItem menuItemSortStudentsInfo = new JMenuItem((ProgramLanguage ==
"Chinese") ? "学生信息排序" : "학생 정보 정렬");
        JMenu menuMaintenanceStudentsInfo = new JMenu((ProgramLanguage == "Chinese") ?
"学生信息维护" : "학생 정보 유지보수");
        JMenu menuStatisticsBMI = new JMenu((ProgramLanguage == "Chinese") ? "BMI 统
计" : "BMI 통계");
    }
}

```

```

JMenuItem menuItemSearchStudentInfo = new JMenuItem((ProgramLanguage ==
"Chinese") ? "查询" : "검색");
JMenuItem menuItemModifyStudentInfo = new JMenuItem((ProgramLanguage ==
"Chinese") ? "修改" : "수정");
JMenuItem menuItemDeleteStudentInfo = new JMenuItem((ProgramLanguage ==
"Chinese") ? "删除" : "삭제");
JMenuItem menuItemShowStatistics = new JMenuItem((ProgramLanguage ==
"Chinese") ? "统计数据显示" : "통계 데이터 출력");
JMenuItem menuItemChartBMI = new JMenuItem((ProgramLanguage == "Chinese") ?
"BMI 图表" : "BMI 그래프");

```

```

setJMenuBar(menuBar);
menuBar.add(menuItemPrintStudentsInfo);
menuBar.add(menuMenu);
menuMenu.add(menuItemSortStudentsInfo);
menuMenu.add(menuMaintenanceStudentsInfo);
menuMenu.add(menuStatisticsBMI);
menuMaintenanceStudentsInfo.add(menuItemSearchStudentInfo);
menuMaintenanceStudentsInfo.add(menuItemModifyStudentInfo);
menuMaintenanceStudentsInfo.add(menuItemDeleteStudentInfo);
menuStatisticsBMI.add(menuItemShowStatistics);
menuStatisticsBMI.add(menuItemChartBMI);

menuItemSortStudentsInfo.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        showSortPanel();
    }
});
menuItemSearchStudentInfo.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        showSearchStudentPanel();
    }
});
menuItemModifyStudentInfo.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        showModifyPanel();
    }
});
menuItemDeleteStudentInfo.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Delete();
    }
});
menuItemShowStatistics.addActionListener(new ActionListener() {

```

```

        public void actionPerformed(ActionEvent e) {
            showStaticsPanel();
        }
    });
    menuItemChartBMI.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            JFreeChart chart = ChartFactory.createXYBarChart("BMI Statistics",
"Intervals", false,
                    "Number of Students", getBMIDataset(),
PlotOrientation.VERTICAL, true, false, false);
            ChartFrame frame = new ChartFrame("BMI Statistics", chart);
            frame.pack();
            frame.setVisible(true);
        }
    });

    // Toolbar
    JToolBar toolBar = new JToolBar();
    JButton btnBack = new JButton((ProgramLanguage == "Chinese") ? "返回" : "뒤로");
    btnBack.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            Back();
        }
    });
    menuBar.add(toolBar);
    toolBar.add(btnBack);

    // ContentPane
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    contentPane.setLayout(null);
    setContentPane(contentPane);

    // Input Student Information
    JScrollPane jp = new JScrollPane(textAreaStudentsInfoMain);
    JLabel labelInputStudentInfo = new JLabel((ProgramLanguage == "Chinese") ? "<输入
学生信息>" : "<학생 정보 입력>");
    JLabel labelStudentID = new JLabel((ProgramLanguage == "Chinese") ? "学号" :
"학생 ID");
    JLabel labelStudentName = new JLabel((ProgramLanguage == "Chinese") ? "姓名" :
"이름");
    JLabel labelStudentHeight = new JLabel((ProgramLanguage == "Chinese") ? "身高" :
"신장");

```

```

JLabel labelStudentWeight = new JLabel((ProgramLanguage == "Chinese") ? "体重" :
"몸무게");
JTextField textFieldStudentID = new JTextField();
JTextField textFieldStudentName = new JTextField();
JTextField textFieldStudentHeight = new JTextField();
JTextField textFieldStudentWeight = new JTextField();

jp.setBounds(180, 42, 600, 500);
labelInputStudentInfo.setBounds(15, 10, 100, 37);
labelStudentID.setBounds(15, 46, 65, 15);
labelStudentName.setBounds(15, 74, 65, 15);
labelStudentHeight.setBounds(15, 98, 65, 15);
labelStudentWeight.setBounds(15, 123, 65, 15);
textFieldStudentID.setBounds(70, 43, 100, 21);
textFieldStudentName.setBounds(70, 71, 100, 21);
textFieldStudentHeight.setBounds(70, 95, 100, 21);
textFieldStudentWeight.setBounds(70, 123, 100, 21);
textAreaStudentsInfoMain.setBounds(180, 42, 400, 500);

textFieldStudentID.setColumns(10);
textFieldStudentName.setColumns(10);
textFieldStudentHeight.setColumns(10);
textFieldStudentWeight.setColumns(10);

contentPane.add(jp);
contentPane.add(labelInputStudentInfo);
contentPane.add(labelStudentID);
contentPane.add(labelStudentName);
contentPane.add(labelStudentHeight);
contentPane.add(labelStudentWeight);
contentPane.add(textFieldStudentID);
contentPane.add(textFieldStudentName);
contentPane.add(textFieldStudentHeight);
contentPane.add(textFieldStudentWeight);

JButton btnSaveStudentsInfo = new JButton((ProgramLanguage == "Chinese") ? "保存
数据" : "정보 저장");
btnSaveStudentsInfo.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        StudentInfo student = new StudentInfo(textFieldStudentID.getText(),
textFieldStudentName.getText(),
Float.parseFloat(textFieldStudentHeight.getText()),
Float.parseFloat(textFieldStudentWeight.getText()));

```

```

        if (isExistStudentID(textFieldStudentID.getText()))
            JOptionPane.showMessageDialog(null, (ProgramLanguage ==
"Chinese") ? "学号重复! " : "중복되는 학생 ID 가 존재합니다!",
            (ProgramLanguage == "Chinese") ? "提示信息" : "프롬프트
메시지", JOptionPane.INFORMATION_MESSAGE);
        else {
            students.add(student);
            JOptionPane.showMessageDialog(null, (ProgramLanguage ==
"Chinese") ? "保存成功" : "저장 성공",
            (ProgramLanguage == "Chinese") ? "提示信息" : "프롬프트
메시지", JOptionPane.INFORMATION_MESSAGE);
            printStudentsInfo();
        }
    }
});
btnSaveStudentsInfo.setBounds(15, 155, 150, 30);
contentPane.add(btnSaveStudentsInfo);

JButton btnGenerateRandomStudentsInfo = new JButton((ProgramLanguage ==
"Chinese") ? "随机获取学生" : "학생 정보 랜덤 생성");
btnGenerateRandomStudentsInfo.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        StudentInfo student[] = new StudentInfo[200];
        student = GetGeneratedStudents(200, 175, 175, 60, 60);
        for (int i = 0; i < 200; i++)
            students.add(student[i]);
        printStudentsInfo();
    }
});
btnGenerateRandomStudentsInfo.setBounds(15, 190, 150, 30);
contentPane.add(btnGenerateRandomStudentsInfo);

JButton btnPrintStudentsInfo = new JButton((ProgramLanguage == "Chinese") ? "显示
学生信息" : "학생 정보 갱신");
btnPrintStudentsInfo.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        printStudentsInfo();
    }
});
btnPrintStudentsInfo.setBounds(180, 5, 150, 30);
contentPane.add(btnPrintStudentsInfo);
}

public void printStudentsInfo() {

```



```

        StringBuilder sb = new StringBuilder();
        for (StudentInfo student : students)
            sb.append(GetStudentInfoString(student) + "\r\n");
        textAreaStudentsInfoMain.setText(sb.toString());
    }

    public static IntervalXYDataset getBMIDataset() {
        /*
         * cntRangeBMI[] : An array that counted the number of students in each BMI range.
         * cntRangeBMI[0]: (BMI of Student)<=14
         * cntRangeBMI[1]: (BMI of Student)>14 && (BMI of Student)<=15
         * cntRangeBMI[2]: (BMI of Student)>15 && (BMI of Student)<=16
         * cntRangeBMI[3]: (BMI of Student)>16 && (BMI of Student)<=17
         * cntRangeBMI[4]: (BMI of Student)>17 && (BMI of Student)<=18
         * cntRangeBMI[5]: (BMI of Student)>18 && (BMI of Student)<=19
         * cntRangeBMI[6]: (BMI of Student)>19 && (BMI of Student)<=20
         * cntRangeBMI[7]: (BMI of Student)>20 && (BMI of Student)<=21
         * cntRangeBMI[8]: (BMI of Student)>21 && (BMI of Student)<=22
         * cntRangeBMI[9]: (BMI of Student)>22
         */
        int cntRangeBMI[] = new int[10];
        XYSeriesCollection seriesCollection = new XYSeriesCollection();
        XYSeries series = new XYSeries("BMI Statistics");

        for (StudentInfo student : students) {
            double bmi = ((student.getBMI() <= 14.0) ? 14.0 : ((student.getBMI() > 22.0) ?
23.0 : student.getBMI()));
            int index = (int) Math.ceil(bmi - 14.0);
            cntRangeBMI[index]++;
        }

        for (int i = 0; i < cntRangeBMI.length; i++)
            series.add(14 + i, cntRangeBMI[i]);
        seriesCollection.addSeries(series);

        return new XYBarDataset(seriesCollection, 0.9);
    }

    public ArrayList<StudentInfo> getStudentsInfoFromFile(String fileName) {
        BufferedReader reader = null;
        File file = new File(fileName);
        ArrayList<StudentInfo> temp = new ArrayList<StudentInfo>();

        try {

```

```

        reader = new BufferedReader(new FileReader(file));
        String tempString = null;
        while ((tempString = reader.readLine()) != null) {
            String[] token = tempString.split(",");
            StudentInfo st = new StudentInfo(token[0], token[1],
Float.parseFloat(token[2]),
            Float.parseFloat(token[3]));
            temp.add(st);
        }
        reader.close();
    } catch (IOException e) {
        e.printStackTrace();
    }

    return temp;
}

public void saveStudentsInfoToFile(ArrayList<StudentInfo> students, String fileName) {
    try {
        BufferedWriter writer = new BufferedWriter(new FileWriter(fileName, false));
        for (StudentInfo student : students) {
            writer.write(String.format("%s,%s,%.2f,%.2f,%.2f,%s\r\n", student.getID(),
student.getName(),
            student.getHeight(), student.getWeight(), student.getBMI(),
student.getHealthCondition()));
        }
        writer.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public static float getStudentsAverageBMI(List<StudentInfo> students) {
    float sum = 0;
    if (students == null || students.isEmpty())
        return 0.0f;

    for (StudentInfo student : students)
        sum += student.getBMI();
    return sum / students.size();
}

public double getStudentsMiddleBMI(List<StudentInfo> students) {
    if (students == null || students.isEmpty())

```

```

        return 0.0f;

    students.sort(new BMIComparator());
    if (students.size() % 2 == 0)
        return (students.get(students.size() / 2 - 1).getBMI() + students.get(students.size() /
2).getBMI()) / 2;
    else
        return students.get(students.size() / 2).getBMI();
}

public static float getStudentsVarianceBMI(List<StudentInfo> students) {
    if (null == students || students.isEmpty())
        return 0.0f;

    float sum = 0.0f, average = getStudentsAverageBMI(students);
    for (StudentInfo student : students)
        sum += Math.pow((student.getBMI() - average), 2.0);

    return sum / students.size();
}

public List<Float> Mode(List<StudentInfo> students) {
    Map<Float, Integer> map = new HashMap<>();
    for (StudentInfo student : students) {
        if (map.containsKey(student.getBMI()))
            map.put(student.getBMI(), map.get(student.getBMI()) + 1);
        else
            map.put(student.getBMI(), 1);
    }

    Iterator iterator = map.entrySet().iterator();
    int maxCount = Integer.MIN_VALUE;
    while (iterator.hasNext()) {
        Map.Entry entry = (Map.Entry) iterator.next();
        if (maxCount < (int) entry.getValue())
            maxCount = (int) entry.getValue();
    }
    if (maxCount == 1)
        return null;

    List<Float> list = new ArrayList<>();
    iterator = map.entrySet().iterator();
    while (iterator.hasNext()) {
        Map.Entry entry = (Map.Entry) iterator.next();

```

```

        if ((int) entry.getValue() == maxCount)
            list.add((float) entry.getKey());
    }

    return list;
}

private boolean isExistStudentID(String id) {
    for (StudentInfo student : students)
        if (id.equals(student.getID()))
            return true;
    return false;
}

private void showStaticsPanel() {
    this.setContentPane(new StaticsPanel());
    setVisible(true);
}

private void showSortPanel() {
    this.setContentPane(new SortPanel());
    setVisible(true);
}

private void showSearchStudentPanel() {
    this.setContentPane(new SearchPanel());
    setVisible(true);
}

private void showModifyPanel() {
    this.setContentPane(new ModifyPanel());
    setVisible(true);
}

public void sortStudents(Comparator<StudentInfo> c) {
    Collections.sort(students, c);
}

private class HeightComparator implements Comparator<StudentInfo> {
    @Override
    public int compare(StudentInfo st1, StudentInfo st2) {
        if (st1.getHeight() > st2.getHeight())
            return 1;
        else if (st1.getHeight() < st2.getHeight())

```

```

        return -1;
    return 0;
    }
}

private class NameComparator implements Comparator<StudentInfo> {
    @Override
    public int compare(StudentInfo st1, StudentInfo st2) {
        return st1.getName().compareTo(st2.getName());
    }
}

private class IdComparator implements Comparator<StudentInfo> {
    @Override
    public int compare(StudentInfo st1, StudentInfo st2) {
        return st1.getID().compareTo(st2.getID());
    }
}

private class WeightComparator implements Comparator<StudentInfo> {
    @Override
    public int compare(StudentInfo st1, StudentInfo st2) {
        if (st1.getWeight() > st2.getWeight())
            return 1;
        else if (st1.getWeight() < st2.getWeight())
            return -1;
        return 0;
    }
}

private class BMIComparator implements Comparator<StudentInfo> {
    @Override
    public int compare(StudentInfo st1, StudentInfo st2) {
        if (st1.getBMI() > st2.getBMI())
            return 1;
        else if (st1.getBMI() < st2.getBMI())
            return -1;
        return 0;
    }
}

class StaticsPanel extends JPanel {
    public StaticsPanel() {
        JTextArea textArea = new JTextArea();
    }
}

```

```

        textArea.setBounds(143, 10, 281, 241);
        this.add(textArea);

        List<Float> mode = Mode(students);
        StringBuilder sb = new StringBuilder();
        if (mode != null && !mode.isEmpty()) {
            sb.append(String.format("Average: %.2f\nMiddle: %.2f\n",
getStudentsAverageBMI(students),
getStudentsMiddleBMI(students)));
            for (float m : mode)
                sb.append(String.format("%.2f ", m));
            sb.append(String.format("Variance: %.2f\n",
getStudentsVarianceBMI(students)));
        } else
            sb.append(String.format("Average: %.2f\nMiddle: %.2f\nThere is no
mode\nVariance: %.2f\n",
getStudentsAverageBMI(students),
getStudentsMiddleBMI(students),
getStudentsVarianceBMI(students)));

        textArea.setText(sb.toString());
    }
}

class SortPanel extends JPanel {
    public SortPanel() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 300);
        this.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(this);
        this.setLayout(null);

        JTextArea textAreaSortedStudentsInfo = new JTextArea();
        textAreaSortedStudentsInfo.setBounds(143, 10, 400, 500);
        JScrollPane jp = new JScrollPane(textAreaSortedStudentsInfo);
        jp.setBounds(180, 42, 600, 500);
        this.add(jp);

        JLabel labelSortMethod = new JLabel((ProgramLanguage == "Chinese") ? "排序
选择" : "<정렬 방법 선택>");
        labelSortMethod.setBounds(15, 20, 100, 15);
        this.add(labelSortMethod);
    }
}

```

```

        JRadioButton radioBtnSortByID = new JRadioButton((ProgramLanguage ==
"Chinese") ? "学号" : "학생 ID");

        JRadioButton radioBtnSortByName = new JRadioButton((ProgramLanguage ==
"Chinese") ? "姓名" : "이름");

        JRadioButton radioBtnSortByHeight = new JRadioButton((ProgramLanguage ==
"Chinese") ? "身高" : "신장");

        JRadioButton radioBtnSortByWeight = new JRadioButton((ProgramLanguage ==
"Chinese") ? "体重" : "몸무게");

        JRadioButton radioBtnSortByBMI = new JRadioButton((ProgramLanguage ==
"Chinese") ? "BMI" : "BMI");

        ButtonGroup btnGroup = new ButtonGroup();

        btnGroup.add(radioBtnSortByID);
        btnGroup.add(radioBtnSortByName);
        btnGroup.add(radioBtnSortByHeight);
        btnGroup.add(radioBtnSortByWeight);
        btnGroup.add(radioBtnSortByBMI);

        radioBtnSortByID.setBounds(15, 41, 100, 23);
        this.add(radioBtnSortByID);
        radioBtnSortByID.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                sortStudents(new IdComparator());
                StringBuilder sb = new StringBuilder();
                for (StudentInfo student : students)
                    sb.append(GetStudentInfoString(student) + "\r\n");
                textAreaSortedStudentsInfo.setText(sb.toString());
            }
        });

        radioBtnSortByName.setBounds(15, 66, 100, 23);
        this.add(radioBtnSortByName);
        radioBtnSortByName.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                sortStudents(new NameComparator());
                StringBuilder sb = new StringBuilder();
                for (StudentInfo student : students)
                    sb.append(GetStudentInfoString(student) + "\r\n");
                textAreaSortedStudentsInfo.setText(sb.toString());
            }
        });

        radioBtnSortByHeight.setBounds(15, 91, 100, 23);
        this.add(radioBtnSortByHeight);

```

```

radioBtnSortByHeight.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        sortStudents(new HeightComparator());
        StringBuilder sb = new StringBuilder();
        for (StudentInfo student : students)
            sb.append(GetStudentInfoString(student) + "\r\n");
        textAreaSortedStudentsInfo.setText(sb.toString());
    }
});

radioBtnSortByWeight.setBounds(15, 117, 100, 23);
this.add(radioBtnSortByWeight);
radioBtnSortByWeight.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        sortStudents(new WeightComparator());
        StringBuilder sb = new StringBuilder();
        for (StudentInfo student : students)
            sb.append(GetStudentInfoString(student) + "\r\n");
        textAreaSortedStudentsInfo.setText(sb.toString());
    }
});

radioBtnSortByBMI.setBounds(15, 142, 121, 23);
this.add(radioBtnSortByBMI);
radioBtnSortByBMI.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        sortStudents(new BMIComparator());
        StringBuilder sb = new StringBuilder();
        for (StudentInfo student : students)
            sb.append(GetStudentInfoString(student) + "\r\n");
        textAreaSortedStudentsInfo.setText(sb.toString());
    }
});
}

}

class ModifyPanel extends JPanel {
    public ModifyPanel() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 300);
        this.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(this);
        this.setLayout(null);
    }
}

```



```

JLabel labelTargetStudentID = new JLabel((ProgramLanguage == "Chinese") ? "查询学生学号" : "수정 할 학생 ID");
labelTargetStudentID.setBounds(35, 35, 100, 15);
this.add(labelTargetStudentID);

```

```

JTextField textFieldStudentID = new JTextField();
JTextField textFieldStudentName = new JTextField();
JTextField textFieldStudentHeight = new JTextField();
JTextField textFieldStudentWeight = new JTextField();

```

```

textFieldStudentID.setBounds(170, 35, 120, 21);
this.add(textFieldStudentID);
textFieldStudentID.setColumns(10);

```

```

JLabel labelModifyStudentName = new JLabel((ProgramLanguage == "Chinese") ? "姓名" : "이름");
labelModifyStudentName.setBounds(35, 70, 54, 15);
this.add(labelModifyStudentName);

```

```

textFieldStudentName.setBounds(170, 70, 120, 21);
this.add(textFieldStudentName);
textFieldStudentName.setColumns(10);

```

```

JLabel labelModifyStudentHeight = new JLabel((ProgramLanguage == "Chinese") ? "身高" : "신장");
labelModifyStudentHeight.setBounds(35, 105, 54, 15);
this.add(labelModifyStudentHeight);

```

```

textFieldStudentHeight.setBounds(170, 105, 120, 21);
this.add(textFieldStudentHeight);
textFieldStudentHeight.setColumns(10);

```

```

JLabel labelModifyStudentWeight = new JLabel((ProgramLanguage == "Chinese") ? "体重" : "몸무게");
labelModifyStudentWeight.setBounds(35, 140, 54, 15);
this.add(labelModifyStudentWeight);

```

```

textFieldStudentWeight.setBounds(170, 140, 120, 21);
this.add(textFieldStudentWeight);
textFieldStudentWeight.setColumns(10);

```

```

JButton button = new JButton((ProgramLanguage == "Chinese") ? "修改" : "수정");
button.setBounds(35, 198, 255, 30);

```

```

        this.add(button);
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int i;
                for (i = 0; i < students.size(); i++) {
                    if (students.get(i).getID().equals(textFieldStudentID.getText())) {
                        StudentInfo student = new
StudentInfo(textFieldStudentID.getText(),
                                textFieldStudentName.getText(),
                                Float.parseFloat(textFieldStudentHeight.getText()),
                                Float.parseFloat(textFieldStudentWeight.getText()));

                        students.remove(i);
                        students.add(i, student);
                        break;
                    }
                }

                if (i < students.size()) {
                    JOptionPane.showMessageDialog(null, (ProgramLanguage ==
"Chinese") ? "修改成功! " : "수정 성공!",
                                (ProgramLanguage == "Chinese") ? "提示信息" :
"프롬프트 메시지", JOptionPane.INFORMATION_MESSAGE);
                    printStudentsInfo();
                } else
                    JOptionPane.showMessageDialog(null,
                                (ProgramLanguage == "Chinese") ? "未找到此学生! " :
"해당 학생을 찾을 수 없습니다!",
                                (ProgramLanguage == "Chinese") ? "提示信息" :
"프롬프트 메시지", JOptionPane.INFORMATION_MESSAGE);
            }
        });
    }
}

class DeletePanel extends JPanel {
    public DeletePanel() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 300);
        this.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        this.setLayout(null);
    }
}

```

```

        JLabel label = new JLabel((ProgramLanguage == "Chinese") ? "查询学生学号" :
"삭제 할 학생 ID");
        label.setBounds(283, 200, 100, 15);
        this.add(label);

        JTextField textFieldStudentID = new JTextField();
        textFieldStudentID.setBounds(418, 200, 100, 20);
        this.add(textFieldStudentID);
        textFieldStudentID.setColumns(10);

        JButton button = new JButton((ProgramLanguage == "Chinese") ? "删除" :
"삭제");
        button.setBounds(283, 250, 235, 30);
        this.add(button);
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int i;
                for (i = 0; i < students.size(); i++) {
                    if (students.get(i).getID().equals(textFieldStudentID.getText())) {
                        students.remove(i);
                        break;
                    }
                }

                if (i < students.size()) {
                    JOptionPane.showMessageDialog(null, (ProgramLanguage ==
"Chinese") ? "删除成功! " : "삭제 성공!",
                        (ProgramLanguage == "Chinese") ? "提示信息" :
"프롬프트 메시지", JOptionPane.INFORMATION_MESSAGE);
                    printStudentsInfo();
                } else
                    JOptionPane.showMessageDialog(null,
                        (ProgramLanguage == "Chinese") ? "未找到此学生! " :
"해당 학생을 찾을 수 없습니다!",
                        (ProgramLanguage == "Chinese") ? "提示信息" :
"프롬프트 메시지", JOptionPane.INFORMATION_MESSAGE);
            }
        });
    }

    class SearchPanel extends JPanel {
        public SearchPanel() {
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

```

```

        setBounds(100, 100, 450, 300);
        setBorder(new EmptyBorder(5, 5, 5, 5));
        this.setLayout(null);

        JLabel labelSearchStudentID = new JLabel((ProgramLanguage == "Chinese") ? "
查询学生学号" : "학생 ID");
        labelSearchStudentID.setBounds(35, 36, 50, 15);
        this.add(labelSearchStudentID);

        JTextField textFieldStudentID = new JTextField();
        textFieldStudentID.setBounds(100, 35, 150, 20);
        this.add(textFieldStudentID);
        textFieldStudentID.setColumns(10);

        JButton btnSearch = new JButton((ProgramLanguage == "Chinese") ? "查询" :
"검색");
        btnSearch.setBounds(280, 35, 100, 20);
        this.add(btnSearch);

        JLabel labelSearchResult = new JLabel((ProgramLanguage == "Chinese") ? "结果
" : "결과");
        labelSearchResult.setBounds(35, 70, 50, 15);
        this.add(labelSearchResult);

        JTextField textFieldStudentName = new JTextField();
        textFieldStudentName.setBounds(100, 70, 500, 20);
        this.add(textFieldStudentName);
        textFieldStudentName.setColumns(10);
        btnSearch.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int i;
                for (i = 0; i < students.size(); i++) {
                    if (students.get(i).getID().equals(textFieldStudentID.getText())) {

                        textFieldStudentName.setText(GetStudentInfoString(students.get(i)));
                        break;
                    }
                }

                if (i >= students.size())
                    textFieldStudentName.setText((ProgramLanguage == "Chinese") ? "
查无此人!" : "학생을 찾을 수 없습니다!");
            }
        });

```

```

    }
}

private void Back() {
    this.setContentPane(contentPane);
    setVisible(true);
}

private void Delete() {
    this.setContentPane(new DeletePanel());
    setVisible(true);
}

private String GetStudentInfoString(StudentInfo student) {
    return String.format("%s\t%s\t%.2f\t%.2f\t%.2f\t%s", student.getID(),
student.getName(), student.getHeight(),
        student.getWeight(), student.getBMI(), student.getHealthCondition());
}

public StudentInfo[] GetGeneratedStudents(int N, float minHeight, float maxHeight, float
minWeight,
    float maxWeight) {
    StudentInfo students[] = new StudentInfo[N];

    for (int i = 0; i < 200; i++) {
        String id = String.valueOf((int) GetRandomValueRange((float) 10000, (float)
20000));
        while (isExistStudentID(id))
            id = String.valueOf((int) GetRandomValueRange((float) 10000, (float)
20000));

        double x = new Random().nextGaussian();
        double y = new Random().nextGaussian();
        StudentInfo st = new StudentInfo(id, GetGeneratedRandomName(6),
            (float) (GetRandomValueRange(minHeight, maxHeight) + x * 0.1),
            (float) (GetRandomValueRange(minWeight, maxWeight) + y * 5));
        students[i] = st;
    }

    return students;
}

private static float GetRandomValueRange(float min, float max) {
    return (float) ((Math.random() * (max - min + 1)) + min);
}

```

```
}

private static String GetGeneratedRandomName(int length) {
    char[] randName = new char[length];

    for (int i = 0; i < length; i++)
        randName[i] = (char) Math.round((int) GetRandomValueRange((int) 'a', (int) 'z'));
    randName[0] = Character.toUpperCase(randName[0]);

    return String.valueOf(randName);
}
}
```

运行结果

学生信息输入

学生信息显示及录入

菜单返回

<输入学生信息>

显示学生信息

学号

姓名

身高

体重

保存数据

随机获取学生

学生信息输入

学生信息显示及录入

菜单返回

<输入学生信息>

显示学生信息

学号

姓名

身高

体重

保存数据

随机获取学生

12345	Taeyoon	192.00	75.00	20.35	Normal Range
-------	---------	--------	-------	-------	--------------

学生信息输入

—□×

学生信息显示及录入

菜单返回

<输入学生信息>

学号12345

姓名Taeyoon2

身高180

体重56

保存数据

随机获取学生

显示学生信息

12345	Taeyoon	192.00	75.00	20.35	Normal Range
-------	---------	--------	-------	-------	--------------

提示信息

×

i

学号重复!

확인

学生信息输入

—□×

学生信息显示及录入

菜单返回

<输入学生信息>

学号12345

姓名Taeyoon2

身高180

体重56

保存数据

随机获取学生

显示学生信息

12345	Taeyoon	192.00	75.00	20.35	Normal Range
16976	lhmyxl	175.75	59.56	19.28	Normal Range
10440	Tklpvk	175.65	64.12	20.78	Normal Range
19009	Jjufjl	175.47	56.05	18.21	Underweight
17006	Giljhs	175.45	59.88	19.45	Normal Range
14489	Mfjtaa	175.72	61.60	19.95	Normal Range
12547	Hzzdst	175.73	61.29	19.85	Normal Range
12991	Kaadgf	175.76	59.08	19.13	Normal Range
11232	Wbnovd	175.90	49.95	16.14	Underweight
10963	Oildgm	175.69	69.89	22.64	Normal Range
15995	Vowfqc	176.03	59.15	19.09	Normal Range
12911	Bfhdtj	175.39	57.69	18.75	Normal Range
12129	Jolviw	175.37	64.02	20.82	Normal Range
12627	Prkiyn	175.55	57.48	18.65	Normal Range
12007	Dugjai	175.33	61.08	19.87	Normal Range
19393	Spricn	175.63	57.30	18.58	Normal Range
12415	Tcbkei	174.88	55.53	18.16	Underweight
10733	Rshgvf	175.53	52.63	17.08	Underweight
19778	Zlhqsj	175.53	61.31	19.90	Normal Range
14532	Qcubvh	175.57	62.70	20.34	Normal Range
17404	Eycwne	176.12	57.24	18.46	Underweight
16971	Blhlqf	175.23	60.18	19.60	Normal Range
19688	Peajhy	175.70	64.94	21.04	Normal Range
17694	Evhbsy	174.95	52.59	17.18	Underweight
19696	Omdqyv	175.32	64.52	20.99	Normal Range
11653	Elusjo	175.87	63.61	20.57	Normal Range
12074	Yvwdhw	175.71	57.24	18.54	Normal Range
16227	Hvioeb	175.48	69.51	22.57	Normal Range
13167	Qevtaj	175.53	60.53	19.64	Normal Range
18298	Kiasrv	175.91	65.56	21.19	Normal Range
14365	Viihsi	175.68	60.26	19.52	Normal Range

学生信息输入						
学生信息显示及录入						
排序选择						
<input type="radio"/> 学号						
<input checked="" type="radio"/> 姓名						
<input type="radio"/> 身高						
<input type="radio"/> 体重						
<input type="radio"/> BMI						
19656	Ahsbib	175.19	59.16	19.28	Normal Range	
17832	Amvglw	175.03	50.42	16.46	Underweight	
18680	Argpci	175.83	64.46	20.85	Normal Range	
18919	Aswtgy	175.57	58.05	18.83	Normal Range	
14437	Bfdwaw	175.74	71.60	23.18	Overweight--At Risk	
12911	Bfhdjt	175.39	57.69	18.75	Normal Range	
16971	Blhlqf	175.23	60.18	19.60	Normal Range	
15202	Bmkhqz	175.27	66.26	21.57	Normal Range	
11455	Bopdes	175.88	54.82	17.72	Underweight	
19803	Boqaua	175.86	57.64	18.64	Normal Range	
16693	Bqokoe	175.38	65.97	21.45	Normal Range	
13885	Btrnhd	175.91	58.87	19.03	Normal Range	
11677	Buknfg	175.99	58.90	19.02	Normal Range	
11822	Buumbz	175.72	57.51	18.63	Normal Range	
12502	Bvmykc	175.50	58.62	19.03	Normal Range	
18227	Bwyief	176.10	60.53	19.52	Normal Range	
14786	Bzqkkr	175.65	51.27	16.62	Underweight	
15049	Cabsfh	175.11	64.70	21.10	Normal Range	
14780	Cjsrhf	175.17	61.65	20.09	Normal Range	
18596	Cjupqf	175.70	66.10	21.41	Normal Range	
12478	Clzwwz	175.15	68.71	22.40	Normal Range	
19439	Cpwlbu	175.02	60.93	19.89	Normal Range	
18715	Dcqtiw	175.03	60.44	19.73	Normal Range	
16887	Dlpjca	174.96	61.74	20.17	Normal Range	
12007	Dugjai	175.33	61.08	19.87	Normal Range	
17663	Dzfnsf	175.08	57.65	18.81	Normal Range	
19709	Ekocep	175.02	56.27	18.37	Underweight	
11653	Elusjo	175.87	63.61	20.57	Normal Range	
13432	Euyecom	176.02	62.28	20.10	Normal Range	
17694	Evhbsy	174.95	52.59	17.18	Underweight	
10527	Ewhqoz	175.84	54.20	17.53	Underweight	

学生信息输入						
学生信息显示及录入						
排序选择						
<input type="radio"/> 学号						
<input type="radio"/> 姓名						
<input checked="" type="radio"/> 身高						
<input type="radio"/> 体重						
<input type="radio"/> BMI						
19803	Boqaua	175.86	57.64	18.64	Normal Range	
18703	Vpznmd	175.86	68.71	22.21	Normal Range	
11653	Elusjo	175.87	63.61	20.57	Normal Range	
11992	Fviwpg	175.88	68.95	22.29	Normal Range	
11455	Bopdes	175.88	54.82	17.72	Underweight	
14089	Ngncuq	175.88	60.15	19.45	Normal Range	
16068	Teworf	175.89	67.06	21.68	Normal Range	
12466	Ugxdca	175.89	59.04	19.08	Normal Range	
11232	Wbnovd	175.90	49.95	16.14	Underweight	
18298	Kiasrv	175.91	65.56	21.19	Normal Range	
13885	Btrnhd	175.91	58.87	19.03	Normal Range	
18471	Hecynf	175.91	66.78	21.58	Normal Range	
18649	Nqqghz	175.93	65.20	21.06	Normal Range	
18606	Zpjzqz	175.93	49.56	16.01	Underweight	
12360	Hgkrde	175.95	53.59	17.31	Underweight	
15060	Scqoug	175.96	61.86	19.98	Normal Range	
11677	Buknfg	175.99	58.90	19.02	Normal Range	
17308	Gtlqkr	176.00	63.55	20.52	Normal Range	
12305	Fdsvrz	176.00	67.82	21.89	Normal Range	
18583	Vjjrci	176.00	56.39	18.20	Underweight	
16443	Lucary	176.00	66.97	21.62	Normal Range	
14229	Hwcrzv	176.01	54.62	17.63	Underweight	
13010	Pzyabd	176.01	64.01	20.66	Normal Range	
13432	Euyecom	176.02	62.28	20.10	Normal Range	
15995	Vowfqc	176.03	59.15	19.09	Normal Range	
11390	Rbrnvp	176.05	57.80	18.65	Normal Range	
18735	Tzrntv	176.07	58.25	18.79	Normal Range	
18227	Bwyief	176.10	60.53	19.52	Normal Range	
17404	Eycwne	176.12	57.24	18.46	Underweight	
12345	Taeyoon	192.00	75.00	20.35	Normal Range	

学生信息输入

学生信息显示及录入

菜单返回

排序选择

☐ 学号

☐ 姓名

☐ 身高

☒ 体重

☐ BMI

18298	Kiasrv	175.91	65.56	21.19	Normal Range
11984	lwtvui	175.16	65.59	21.38	Normal Range
16427	Jbnuul	175.08	65.64	21.41	Normal Range
16721	Rpdchf	175.47	65.65	21.32	Normal Range
16693	Bqokoe	175.38	65.97	21.45	Normal Range
15618	Tkvqva	175.60	66.03	21.41	Normal Range
18596	Cjupqf	175.70	66.10	21.41	Normal Range
15202	Bmkhqz	175.27	66.26	21.57	Normal Range
13282	Nqpzxd	175.41	66.48	21.61	Normal Range
16697	Qtabkl	175.33	66.49	21.63	Normal Range
11860	Hsavrj	175.25	66.62	21.69	Normal Range
10228	Jenmyh	175.19	66.67	21.72	Normal Range
18471	Hecynf	175.91	66.78	21.58	Normal Range
16443	Lucary	176.00	66.97	21.62	Normal Range
16068	Teworf	175.89	67.06	21.68	Normal Range
16846	Zxjtyr	175.08	67.07	21.88	Normal Range
15058	ldholb	175.56	67.78	21.99	Normal Range
12305	Fdsvrz	176.00	67.82	21.89	Normal Range
18703	Vpznmd	175.86	68.71	22.21	Normal Range
12478	Clzwwz	175.15	68.71	22.40	Normal Range
11992	Fviwpg	175.88	68.95	22.29	Normal Range
15452	Uyullr	175.10	69.37	22.63	Normal Range
15161	Tecuzn	175.82	69.42	22.46	Normal Range
16227	Hvioeb	175.48	69.51	22.57	Normal Range
16798	Fisbuc	175.28	69.65	22.67	Normal Range
18944	lifrwn	175.67	69.68	22.58	Normal Range
10963	Oiidgm	175.69	69.89	22.64	Normal Range
11895	Vctuoo	175.26	70.58	22.98	Normal Range
14437	Bfdwaw	175.74	71.60	23.18	Overweight-At Risk
12345	Taeyoon	192.00	75.00	20.35	Normal Range

学生信息输入

—□×

学生信息显示及录入

菜单返回

排序选择

☐ 学号

☐ 姓名

☐ 身高

☐ 体重

☒ BMI

10804	Hpuhue	175.19	65.10	21.21	Normal Range
19040	Kwkgez	175.45	65.41	21.25	Normal Range
16721	Rpdchf	175.47	65.65	21.32	Normal Range
11984	lwtvui	175.16	65.59	21.38	Normal Range
15618	Tkvqva	175.60	66.03	21.41	Normal Range
18596	Cjupqf	175.70	66.10	21.41	Normal Range
16427	Jbnuul	175.08	65.64	21.41	Normal Range
16693	Bqokoe	175.38	65.97	21.45	Normal Range
15202	Bmkhqz	175.27	66.26	21.57	Normal Range
18471	Hecynf	175.91	66.78	21.58	Normal Range
13282	Nqpzxd	175.41	66.48	21.61	Normal Range
16443	Lucary	176.00	66.97	21.62	Normal Range
16697	Qtabkl	175.33	66.49	21.63	Normal Range
16068	Teworf	175.89	67.06	21.68	Normal Range
11860	Hsavvj	175.25	66.62	21.69	Normal Range
10228	Jenmyh	175.19	66.67	21.72	Normal Range
16846	Zxjtyn	175.08	67.07	21.88	Normal Range
12305	Fdsvrz	176.00	67.82	21.89	Normal Range
15058	Idholb	175.56	67.78	21.99	Normal Range
18703	Vpznmd	175.86	68.71	22.21	Normal Range
11992	Fviwpg	175.88	68.95	22.29	Normal Range
12478	Cizwzw	175.15	68.71	22.40	Normal Range
15161	Tecuzn	175.82	69.42	22.46	Normal Range
16227	Hvioeb	175.48	69.51	22.57	Normal Range
18944	lifnwk	175.67	69.68	22.58	Normal Range
15452	Uyullr	175.10	69.37	22.63	Normal Range
10963	Olidgm	175.69	69.89	22.64	Normal Range
16798	Fisbuo	175.28	69.65	22.67	Normal Range
11895	Vctuoo	175.26	70.58	22.98	Normal Range
14437	Bfdwaw	175.74	71.60	23.18	Overweight-At Risk

学生信息输入

学生信息显示及录入

菜单

返回

查询学生学号

查询

结果

学生信息输入

学生信息显示及录入

菜单

返回

查询学生学号

12345

查询

结果

12345	Taeyoon	192.00	75.00	20.35	Normal Range
-------	---------	--------	-------	-------	--------------

学生信息输入

学生信息显示及录入

菜单返回

查询学生学号

12346

查询

结果

查无此人!

学生信息输入

学生信息显示及录入

菜单返回

查询学生学号

姓名

身高

体重

修改

学生信息输入

学生信息显示及录入

菜单

返回

查询学生学号

12345

姓名

Taeyon

身高

175

体重

50

修改

提示信息

×

i

修改成功!

확인

学生信息输入

学生信息显示及录入

菜单

返回

查询学生学号

12346

姓名

Taeyon

身高

175

体重

50

修改

提示信息

×

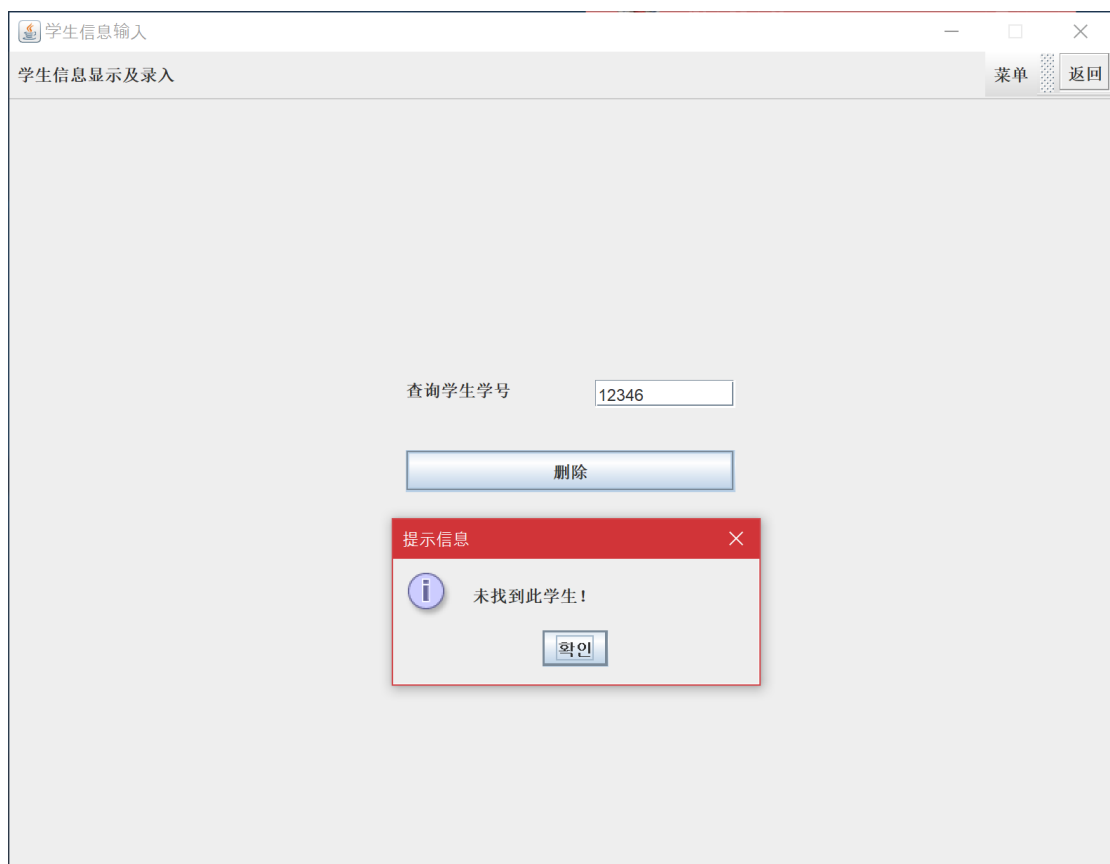
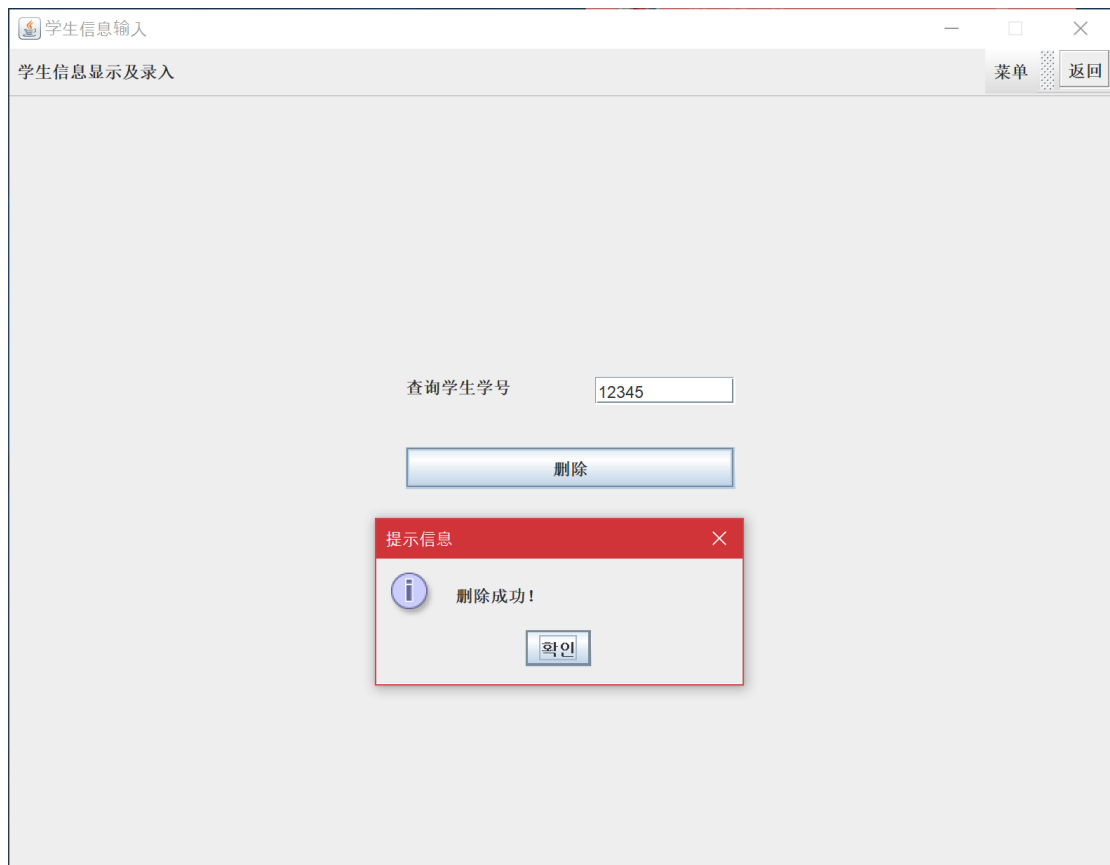
i

未找到此学生!

확인

学生信息输入						
学生信息显示及录入						菜单 返回
排序选择						
<input type="radio"/> 学号						
<input checked="" type="radio"/> 姓名						
<input type="radio"/> 身高						
<input type="radio"/> 体重						
<input type="radio"/> BMI						
10912	Rwrnga	175.44	56.32	18.30	Underweight	
18256	Rxiuxf	175.38	62.09	20.19	Normal Range	
15060	Scqoug	175.96	61.86	19.98	Normal Range	
18540	Sgwagb	175.13	59.45	19.39	Normal Range	
10077	Sntdpw	175.28	56.27	18.31	Underweight	
19393	Spricn	175.63	57.30	18.58	Normal Range	
18883	Sqkjqp	175.33	59.30	19.29	Normal Range	
16236	Swkjmc	175.54	63.53	20.62	Normal Range	
18535	Swyqrr	175.65	65.01	21.07	Normal Range	
12345	Taeyon	175.00	50.00	16.33	Underweight	
12415	Tcbkei	174.88	55.53	18.16	Underweight	
13719	Tctabf	175.05	60.90	19.87	Normal Range	
11437	Tdgzut	175.57	58.81	19.08	Normal Range	
15161	Tecuzn	175.82	69.42	22.46	Normal Range	
16068	Teworf	175.89	67.06	21.68	Normal Range	
11227	Tjmqkp	175.75	63.62	20.60	Normal Range	
10440	Tklpvk	175.65	64.12	20.78	Normal Range	
15618	Tkvqva	175.60	66.03	21.41	Normal Range	
18735	Tzrntv	176.07	58.25	18.79	Normal Range	
16199	Ubwwym	175.17	58.89	19.19	Normal Range	
12287	Udrbnr	175.07	61.89	20.19	Normal Range	
12466	Ugxdca	175.89	59.04	19.08	Normal Range	
11638	Uhorrl	175.18	59.73	19.46	Normal Range	
16999	Ussgju	175.25	45.81	14.92	Underweight	
15452	Uyullr	175.10	69.37	22.63	Normal Range	
16014	Vafvzb	174.91	61.98	20.26	Normal Range	
11895	Vctuoo	175.26	70.58	22.98	Normal Range	
13489	Vgfprs	175.48	60.45	19.63	Normal Range	
13631	Vhtxag	175.32	56.96	18.53	Normal Range	
14365	Viihsi	175.68	60.26	19.52	Normal Range	
18583	Vjjrci	176.00	56.39	18.20	Underweight	
10066	Vjstta	175.99	69.98	20.90	Normal Range	

学生信息输入	
学生信息显示及录入	菜单 返回
<div> <div>查询学生学号</div> <div></div> </div> <div>删除</div>	



学生信息输入						
学生信息显示及录入						
排序选择						
<input checked="" type="radio"/> 学号	11908	Zxtipl	175.21	60.53	19.72	Normal Range
<input type="radio"/> 姓名	11922	Jeezpv	175.55	63.82	20.71	Normal Range
<input type="radio"/> 身高	11945	Rekwzu	175.57	55.23	17.92	Underweight
<input type="radio"/> 体重	11980	Weicus	175.26	52.23	17.00	Underweight
<input type="radio"/> BMI	11984	Iwtvui	175.16	65.59	21.38	Normal Range
	11992	Fviwpg	175.88	68.95	22.29	Normal Range
	12007	Dugjai	175.33	61.08	19.87	Normal Range
	12074	Yvwdhw	175.71	57.24	18.54	Normal Range
	12085	Rsinpu	175.47	51.45	16.71	Underweight
	12129	Jolviw	175.37	64.02	20.82	Normal Range
	12283	Vrjxjq	175.43	60.02	19.50	Normal Range
	12287	Udrbnr	175.07	61.89	20.19	Normal Range
	12305	Fdsvrz	176.00	67.82	21.89	Normal Range
	12360	Hgkrde	175.95	53.59	17.31	Underweight
	12415	Tcbkei	174.88	55.53	18.16	Underweight
	12431	Yhosts	175.45	49.54	16.09	Underweight
	12466	Ugxdca	175.89	59.04	19.08	Normal Range
	12478	Clzwwz	175.15	68.71	22.40	Normal Range
	12502	Bvmykc	175.50	58.62	19.03	Normal Range
	12547	Hzjdst	175.73	61.29	19.85	Normal Range
	12595	Jfwyyq	175.61	59.01	19.13	Normal Range
	12627	Prkiyn	175.55	57.48	18.65	Normal Range
	12673	Lsbmih	175.47	61.88	20.10	Normal Range
	12911	Bfhdjt	175.39	57.69	18.75	Normal Range
	12990	Rbwxtet	175.29	60.90	19.82	Normal Range
	12991	Kaadgf	175.76	59.08	19.13	Normal Range
	13010	Pzyabd	176.01	64.01	20.66	Normal Range
	13075	Xunaoj	175.21	58.13	18.93	Normal Range
	13108	Zbnkijw	175.80	61.04	19.75	Normal Range
	13167	Qevtaj	175.53	60.53	19.64	Normal Range
	13282	Nqpzxd	175.41	66.48	21.61	Normal Range

学生信息输入						
学生信息显示及录入						
<div> <div>Average:19.61</div> <div>Middle:19.64</div> <div>There is no mode</div> <div>Variance:2.48</div> </div>						

