**package** io;

**import** java.awt.Container;

**import** java.awt.GridLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

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

**public** **class** ScoreForm **extends** JFrame **implements** ActionListener {

**private** JLabel hakL, nameL, korL, engL, mathL;

**private** JTextField hakT, nameT, korT, engT, mathT;

**private** JButton inputBtn, printBtn, searchBtn, rankBtn, saveBtn, loadBtn;

**private** DefaultTableModel model;

**private** JTable table;

**private** Score score; // inteface 잡기

**public** ScoreForm() {

// JLabel 생성

hakL = **new** JLabel("학번");

nameL = **new** JLabel("이름");

korL = **new** JLabel("국어");

engL = **new** JLabel("영어");

mathL = **new** JLabel("수학");

// JTextField 생성

hakT = **new** JTextField("", 22); // 크기(22)를 안주면 쪼그라든다

// TextField는 크기가 없기 때문에 데이터가 들어갈 공간이 있어야 하니까 크기를 잡아준다

nameT = **new** JTextField("", 22);

korT = **new** JTextField("", 22);

engT = **new** JTextField("", 22);

mathT = **new** JTextField("", 22);

// JButton 생성

inputBtn = **new** JButton("입력");

printBtn = **new** JButton("출력");

searchBtn = **new** JButton("학번검색");

rankBtn = **new** JButton("순위");

saveBtn = **new** JButton("파일저장");

loadBtn = **new** JButton("파일읽기");

// JTable의 타이틀

Vector<String> vector = **new** Vector<String>();

vector.add("학번");

vector.add("이름");

vector.add("국어");

vector.add("영어");

vector.add("수학");

vector.add("총점");

vector.add("평균");

model = **new** DefaultTableModel(vector, 0);

table = **new** JTable(model);

// 화면에 뿌려주는 건 Table. 그 안에서 데이터를 넣다 뺏다 하는 역할은 model이 한다

JScrollPane scroll = **new** JScrollPane(table);

score = **new** ScoreImpl();

// Panel 생성

// West - 왼쪽

JPanel hakP = **new** JPanel(); // 학점 패널 위에

hakP.add(hakL); // 학점 라벨과

hakP.add(hakT); // 학점 텍스트필드를 넣어주고

JPanel nameP = **new** JPanel();

nameP.add(nameL);

nameP.add(nameT);

JPanel korP = **new** JPanel();

korP.add(korL);

korP.add(korT);

JPanel engP = **new** JPanel();

engP.add(engL);

engP.add(engT);

JPanel mathP = **new** JPanel();

mathP.add(mathL);

mathP.add(mathT);

JPanel p = **new** JPanel(**new** GridLayout(5, 1)); // 전체 패널 위에 균등하게 패널을 또 넣어줌(5행, 1열)

p.add(hakP); // 학점 패널

p.add(nameP); // 이름 패널

p.add(korP); // 국어 패널

p.add(engP); // 영어 패널

p.add(mathP); // 수학 패널

JPanel p2 = **new** JPanel(**new** GridLayout(1, 6, 5, 5)); // 또 다른 패널(p2)에 입력, 출력, 학번검색, 순위, 파일저장, 파일읽기 버튼을 넣어줌

p2.add(inputBtn);

p2.add(printBtn);

p2.add(searchBtn);

p2.add(rankBtn);

p2.add(saveBtn);

p2.add(loadBtn);

Container c = getContentPane(); // 패널을 넣어줄 Container를 생성

c.add("West", p); // 서쪽에 p를

c.add("South", p2); // 남쪽에 p2를

c.add("Center", scroll); // scroll은 기본 센터에 배치

setBounds(900, 100, 700, 300);

setVisible(**true**);

setDefaultCloseOperation(***EXIT\_ON\_CLOSE***);

// 이벤트

event(); // event 함수 호출!

// 원래는 이벤트 안에 6개의 버튼들의 이벤트들이 다 들어올테지만

// 지금보다 훨씬 더 많은 이벤트들을 다룰 수 있으니 event함수를 적어놓고

// event메소드를 만든 걸 호출 하는 식으로 만들어 준 것이다~~~

} // ScoreForm() - 생성자 끝!

// 메소드는 따로 잡을 수 있움

**public** **void** event() {

inputBtn.addActionListener(**this**);

printBtn.addActionListener(**this**);

searchBtn.addActionListener(**this**);

rankBtn.addActionListener(**this**);

saveBtn.addActionListener(**this**);

loadBtn.addActionListener(**this**);

}

@Override

**public** **void** actionPerformed(ActionEvent e) {

// 버튼들이 다 넘어오니까~

**if** (e.getSource() == inputBtn) { // 넘어오는 버튼이 inputBtn이냐~

input();

// 많은 데이터를 model에다 넣어야 하고 그 model을 Impl에다 뿌리려면 데이터를 보관해야 한다

// 데이터를 보관하려면 list가 필요하다. (list 대신 DB를 쓰기도)

// score.print(model);

// ScoreImpl에서는 JTable이 있는지 없는지 모르니

// model을 보내줘야한다 그래야 Table에 관련된 일을 할 수 있다

// 데이터를 넣고 빼는 역할은 model이 하고 있으니까

// score는 interface라 Score interface에 만들어진다

} **else** **if** (e.getSource() == printBtn) { // printBtn이냐~

score.print(model);

} **else** **if** (e.getSource() == searchBtn) { // searchBtn이냐~

score.search(model);

} **else** **if** (e.getSource() == rankBtn) { // rankBtn이냐~

score.tot\_desc();

} **else** **if** (e.getSource() == saveBtn) { // saveBtn이냐~

score.save();

} **else** **if** (e.getSource() == loadBtn) { // loadBtn이냐~

score.load();

score.print(model); // 파일 안에 있는 걸 읽어오기

}

} // actionPerformed(ActionEvent e)

**private** **void** input() {

// JTextField에 입력된 데이터 얻기

String hak = hakT.getText(); // hak TextField에 hak 데이터 가져오기

// 그렇담 데이터가 들어왔는 지 어떻게 확인할 수 있을까?

// = if(hak.equals("")) {

**if** (hak.length() == 0) { // 글자수가 0? 아무 문자도 들어오지 않았다는 것

JOptionPane.*showMessageDialog*(**this**, "학번을 입력하세요");

**return**;

}

String name = nameT.getText();

**int** kor = Integer.*parseInt*(korT.getText()); // TextField에 들어오는 건 모두 문자열이다. 문자열 숫자로 바꿔버리자

**int** eng = Integer.*parseInt*(engT.getText()); //

**int** math = Integer.*parseInt*(mathT.getText());

// 계산

**int** tot = kor + eng + math;

**double** avg = (**double**) tot / 3;

ScoreDTO dto = **new** ScoreDTO(); // DTO를 만들어서 ScoreDTO안에 데이터들을 한곳으로 모아야 데이터를 들고 다니기 편하다

dto.setHak(hak);

dto.setName(name);

dto.setKor(kor);

dto.setEng(eng);

dto.setMath(math);

dto.setTot(tot);

dto.setAvg(avg);

// 첫번째 방법 - ScoreDTO를 list에 담아서 JTable에 뿌리고...파일에 보관

score.intput(dto);

// 두번째 방법- ScoreDTO를 가지고 ScoreDAO.java에서 DB에 insert한다

// 초기화 - 입력을 하고 저장하고 나면 다시 입력할 수 있도록 입력값을 초기화시켜줘야 한다

// 저장하고나서 글을 일일이 지워줄 수 없으니

hakT.setText("");

nameT.setText("");

korT.setText("");

engT.setText("");

mathT.setText("");

}

}

// 입력 버튼을 누름과 동시에 ActionPerformed쪽으로 넘어와서 actionperformed에서 inputBtn입니까?

// 하고 확인한 뒤 input메소드로 보낸다

// 자바에선 null을 하나의 상수로 인식한다

// 예를들어

// String a = null; String b = "";

// a + "apple" b = "apple"

// "nullapple" "apple"

// -> 결합이나 연산에는 null을 집어넣으면 안된다. null값 자체가 들어가버리기 때문

// 자바가 제공하는 메소드는 throws가 안된다