**package** swing;

**import** java.awt.Container;

**import** java.awt.FlowLayout;

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

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

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

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

**public** **class** JTableEx2 **extends** JFrame **implements** ActionListener {

**private** List<PersonDTO> list;

**private** DefaultTableModel model; // AbstractTableModel을 써줘도 괜찮지만 Default가 더 많이 쓰임

**private** JTable table;

**private** JButton addBtn, delBtn;

**public** JTableEx2() {

list = **new** ArrayList<PersonDTO>();

// PersonDTO만 잡아주게 되면~~

list.add(**new** PersonDTO("hong", "홍길동", "111", "010-123-1234"));

list.add(**new** PersonDTO("conan", "코난", "333", "010-777-7777")); // 한 줄당 vector가 만들어진다

// 타이틀

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

vector.addElement("아이디");

vector.addElement("이름"); // addElement와 add 둘다 가능

vector.add("비밀번호");

vector.add("핸드폰");

// model 생성

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

// vector타이틀을 가지고 생성, 테이블에 들어갈 한 줄을 미리 하나 만들어주세요 - 1

// 빈공간 만들지 마세요 - 0

// 딱 하나만 재구현하고 싶다하면 익명을 써주면 된다

// DefaultTableModel가 추상클래스가 아닌데 익명을 써도 될까? - YES

@Override

**public** **boolean** isCellEditable(**int** row, **int** column) {

// 조건 ? 참 : 거짓;

**return** column == 0 ? **false** : **true**; // 0번째 행만 수정이 안되고 나머지는 다 됨

}

};

table = **new** JTable(model); // Table에 model을 넣어라

JScrollPane scroll = **new** JScrollPane(table); // table에 scroll을 넣어라

// 버튼

addBtn = **new** JButton("추가");

delBtn = **new** JButton("삭제");

// 패널에 버튼 추가해주기

JPanel p = **new** JPanel(**new** FlowLayout(FlowLayout.***RIGHT***));

p.add(addBtn);

p.add(delBtn);

// 데이터

**for** (PersonDTO dto : list) { // list하나당 vector하나씩 만들어줘야 한다

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

v.add(dto.getId());

v.add(dto.getName());

v.add(dto.getPwd());

v.add(dto.getPhone());

model.addRow(v); // Vector를 모델에다 붙여라

}

Container c = **this**.getContentPane();

c.add(scroll);

c.add("South", p);

// Panel은 가운데로 붙는 습성이 있기 때문에 방향을 지정해주자

setBounds(900, 100, 500, 400);

setVisible(**true**);

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

// 이벤트

addBtn.addActionListener(**this**);

delBtn.addActionListener(**this**);

} // JTableEx2()

@Override

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

**if** (e.getSource() == addBtn) { // 추가

insert();

} **else** **if** (e.getSource() == delBtn) { // 삭제

delete();

}

}

**private** **void** delete() {

**boolean** value = **false**;

String name = JOptionPane.*showInputDialog*(**this**, "이름을 입력하세요");

**int** i;

**for** (i = 0; i < model.getRowCount(); i++) {

**if** (name.equals(model.getValueAt(i, 0))) {

JOptionPane.*showInputDialog*(**this**, "없는 이름입니다");

value = **true**;

**break**;

}

}

**if** (value == **false**) {

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

v.add(name);

**for** (i = 0; i < model.getRowCount(); i++) {

**if** (name.equals(model.getValueAt(i, 1))) { // 이름이 같은 행렬을 삭제

model.removeRow(i);

i = -1;

}

}

}

}

**private** **void** insert() {

**boolean** value = **false**;

String id = JOptionPane.*showInputDialog*(**this**, "아이디를 입력하세요");

**if** (!id.equals("")) {

**for** (**int** i = 0; i < model.getRowCount(); i++) {

**if** (id.equals(model.getValueAt(i, 0))) {

JOptionPane.*showMessageDialog*(**this**, "사용 중인 아이디입니다");

value = **true**;

**break**;

}

}

**if** (value == **false**) {

String name = JOptionPane.*showInputDialog*(**this**, "이름을 입력하세요");

String pwd = JOptionPane.*showInputDialog*(**this**, "비밀번호를 입력하세요");

String number = JOptionPane.*showInputDialog*(**this**, "전화번호를 입력하세요");

// Vector에 값 추가하기

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

v.add(id);

v.add(name);

v.add(pwd);

v.add(number);

JOptionPane.*showMessageDialog*(**this**, "등록 완료");

model.addRow(v);

}

}

}

**public** **static** **void** main(String[] args) {

**new** JTableEx2();

}

}

// 추상 클래스는 반드시 추상 메소드를 필수적으로 써줘야한다

// 하지만 default는 일반 클래스라 마구잡이로 쓴다

// Component는 도구의 역할 연필, 볼펜

//Button/JButton, TextField/JTextField

//독단적으로 뜨지 못하니 Container역할의 Frame/JFrame(최종 컨테이너), Dialog

// 이 기능들을 담는 것이 Container 필통

// Frame은 Frame위에 못 올라간다 따라서 따로 만들어진다

// JMenuBar와 contentPane은 각기 다른 Container이다