**AddressBook – 부분 수정됨 >> 전체 조회, 삭제, 상세조회**

**처음 사용자가 신호를 송출하는 클래스 (메인클래스)**

<import >

import java.time.LocalTime;

import java.time.format.DateTimeFormatter;

<106번라인>

DateTimeFormatter formatter = DateTimeFormatter.*ofPattern*("HH:mm:ss");

String lt = LocalTime.*now*().format(formatter);

labelTimer = new JLabel("현재 시간 "+lt);

~~<수정 아이콘-263>~~

~~updateActionPerformed();~~

<조회 actionperformed-331>

private void detailActionPerformed() {

System.***out***.println("상세조회 구현");

int index[] = table.getSelectedRows();

AddressVO vo = new AddressVO();

if(index.length == 0) {

JOptionPane.*showMessageDialog*(this, "조회할 데이터를 선택하세요...", "Error", JOptionPane.***ERROR\_MESSAGE***);

return;

} else if(index.length > 1) {

JOptionPane.*showMessageDialog*(this, "데이터를 한건만 선택하세요...", "Error", JOptionPane.***ERROR\_MESSAGE***);

return;

} else {

try {

// 테이블에서 선택된 컬럼의 id를 읽어옵니다.

// Integer id= (Integer)myTableModel.getValueAt(index[0], 0);

Integer id = Integer.*parseInt*

(myTableModel.getValueAt(index[0], 0).toString());

vo.setCommand("select");

vo.setId(id);

AddressCtrl ctrl = new AddressCtrl();

//ctrl.send(vo);

mDialog.set("상세조회", false, ctrl.send(vo), *abook*);

mDialog.setVisible(true);

} catch (Exception e) {

JOptionPane.*showMessageDialog*(this, "데이터를 가져오는 중 발생했습니다."+e.toString(),"Error", JOptionPane.***ERROR\_MESSAGE***);

}

}

}

<삭제 ActionPerformed – 407>

private void deleteActionPerformed() {

int index[] = table.getSelectedRows();

if(index.length == 0) {

JOptionPane.*showMessageDialog*(this, "삭제할 데이터를 선택하세요...", "Error", JOptionPane.***ERROR\_MESSAGE***);

return;

}else {

try {

Integer id = Integer.*parseInt*(myTableModel.getValueAt(index[0], 0).toString());

System.***out***.println("사용자가 선택한 id : "+id);

AddressVO vo = new AddressVO();

vo.setCommand("delete");

vo.setId(id);

AddressCtrl ctrl = new AddressCtrl();

ctrl.send(vo);

refreshData();

} catch (Exception e) {

JOptionPane.*showMessageDialog*(this, "데이터를 가져오는 중 발생했습니다."+e.toString(),"Error", JOptionPane.***ERROR\_MESSAGE***);

}

}

System.***out***.println("삭제하기");

}

<전체조회 ActionPerformed – 449>

if(vos.length >= 0) {

/\*vos = new AddressVO[2];

AddressVO rvo = new AddressVO("이순신","서울시 마포구 공덕동","010-555-6677","1"

,"고교동창","1990-05-28","Back-End개발자","2022-03-15",1);

vos[0] = rvo;

rvo = new AddressVO("강감찬","서울시 영등포구 당산동","010-777-6677","1"

,"대학동창","1992-01-28","Back-End개발자","2022-01-25",2);

vos[1] = rvo;\*/

for (int i = 0; i < vos.length; i++) {

AddressVO aVO = vos[i];

Vector<Object> oneRow = new Vector<>();

oneRow.addElement(aVO.getId());

oneRow.addElement(aVO.getName());

oneRow.addElement(aVO.getAddress());

oneRow.addElement(aVO.getTelephone());

myTableModel.addRow(oneRow);

}

while(myTableModel.getRowCount()> vos.length) {

myTableModel.removeRow(0);

}

}

AddressCtrl – clean >> 중계기

AddressVO – clean >> 사용자가 송출한 신호를 받아 중개기로 전달하는 클래스

DBConnectionMgr – clean

- 실질적인 처리를 하는 클래스 4개 -

ModifyAddrEty – clean

**DeleteAddrEty – 다 바뀜 >> 삭제처리(채워져 있는 칸에만 반응해야함)**

**~~RegisterAddrEty – 다 바뀜 >> 입력처리(비워져 있는 칸에만 반응해야함)~~**

**~~RetrieveAddrEty – 다 바뀜 >> 조회처리(채워져 있는 칸에만 반응해야함)~~**

**ModifyDialog – 부분 수정됨 >> 입력, 수정 (단독 클래스)**

<확인버튼 선택 – 206>

if(avo !=null){

JOptionPane.*showMessageDialog*(this, "수정하기에서 확인 입니다.","INFO", JOptionPane.***INFORMATION\_MESSAGE***);

try{

AddressVO vo = new AddressVO();

vo.setCommand("update");

vo.setName(getName());

vo.setAddress(getAddress());

vo.setTelephone(getTel());

vo.setGender(getGender());

vo.setRelationship(getRelationShip());

vo.setBirthday(getBirthDay());

vo.setComments(getComment());

AddressCtrl ctrl = new AddressCtrl();

ctrl.send(vo);

if(vo.getResult()==1) {

*abook*.refreshData();

}

}catch(Exception e){

JOptionPane.*showMessageDialog*(this, "수정중 에러가 발생했습니다." + e,

"Error", JOptionPane.***ERROR\_MESSAGE***);

}

/\*

\* 아이디가 0보다 큰값이 아니면 입력모드 이다.

\*/

}else{

try {

JOptionPane.*showMessageDialog*(this, "입력하기에서 확인 입니다.","INFO", JOptionPane.***INFORMATION\_MESSAGE***);

AddressVO vo = new AddressVO();

vo.setCommand("insert");

vo.setName(getName());

vo.setAddress(getAddress());

vo.setTelephone(getTel());

vo.setGender(getGender());

vo.setRelationship(getRelationShip());

vo.setBirthday(getBirthDay());

vo.setComments(getComment());

// 이거 왜 위로 가면 안되지...?

AddressCtrl ctrl = new AddressCtrl();

ctrl.send(vo);

if(vo.getResult()==1) {

*abook*.refreshData();

}

} catch (Exception e) {

JOptionPane.*showMessageDialog*(this, "입력중 에러가 발생했습니다." + e,

"Error", JOptionPane.***ERROR\_MESSAGE***);

}

}