MemberDAO 클래스

```
package s학번
import 생략;


public class MemberDAO {

	private Connection conn;
	PreparedStatement stmt;
	ResultSet rs;

	private String url = "jdbc:mariadb://127.0.0.1:3306/mem_db";
	private String id = "root";
	private String pw = "1234";

	public MemberDAO() {

		try{
			Class.forName("org.mariadb.jdbc.Driver");
			conn = DriverManager.getConnection(url, id, pw);
		} catch(ClassNotFoundException e){
			System.err.println("JDBC 찾기 오류!!!" + e.toString());
		} catch (SQLException e) {
			System.err.println("db 연결 오류!!!" + e.toString());
		}

	}

	void closeDB(){

		try{
			if( rs!=null )		rs.close();
			if( stmt!=null )	stmt.close();
			if( conn != null )	conn.close();
		}catch(SQLException e){
			System.err.println("DB 종류 오류"+e.toString());
		}
	}

	// 회원 검색하여 검색결과를 반환하는 메소드
	public ArrayList<MemberVO> selectMember(String id){
		// 변수 선언
		ArrayList<MemberVO> list = new ArrayList<MemberVO>();
		String sql="";

		try{
			// sql문 만들고 stmt에 담기
			// id가 비어 있으면 회원 전체 검색
			// id가 있으면 해당 아이디만 검색함.

			if(id.equals("")) {
				sql = "select * from member";
				stmt = conn.prepareStatement(sql);
			}else {
				sql = "select * from member where id=?";
				stmt = conn.prepareStatement(sql);
				stmt.setString(1, id);
			}
```



```
			// 쿼리 검색수행 후 검색결과 ArrayList에 담기
			rs = stmt.executeQuery();

			while(rs.next()) {
				list.add( new MemberVO(rs.getString(1), rs.getString(2),
					rs.getString(3), rs.getString(4),rs.getString(5) ));
			}

			// db연결 종료
			closeDB();

		}catch(SQLException e){
			System.err.println("회원 검색 오류!!!" + e.toString() );
		}

		return list;

	}

	// 회원정보를 받아 테이블에 추가하기
	public boolean insertMember(MemberVO vo){
		// 변수 선언
		boolean ret = false;
		String sql="";

		try{
			// sql문 만들고 실행시키고 뒤처리
			sql = "insert into member values(?,?,?,?,?)";
			stmt = conn.prepareStatement(sql);
			stmt.setString(1, vo.getId());
			stmt.setString(2, vo.getName());
			stmt.setString(3, vo.getPass());
			stmt.setString(4, vo.getAddr());
			stmt.setString(5, vo.getJoinday());
			if( stmt.executeUpdate() > 0 ) ret = true;

			// db연결 종료
			closeDB();

		}catch(SQLException e){
			System.err.println("회원 등록 오류!!" + e.toString() );
		}

		return ret;
	}

	// 회원 삭제하기
	public boolean deleteMember(String id){
		// 변수 선언
		boolean ret = false;
		String sql="";

		try{
			sql = "delete from member where id=?";
			stmt = conn.prepareStatement(sql);
			stmt.setString(1, id);
			if( stmt.executeUpdate() > 0 ) ret = true;

			// db연결 종료
			closeDB();

		}catch(SQLException e){
			System.err.println("회원 삭제 오류!!" + e.toString() );
		}
		return ret;
	}

}
```

SwingRegister 클래스

```
package s학번
import 생략;


public class SwingRegister extends JFrame implements ActionListener{

	private String name[] = {"id", "이름", "비밀번호", "주소", "가입일자"};
	private JLabel label[] = new JLabel[5];
	private JTextField textF[] = new JTextField[5];

	private JButton btnSave = new JButton("저장");
	private JButton btnCancel = new JButton("취소");

	// 회원 등록
	SwingRegister(){

		JPanel p1 = new JPanel();
		JPanel p2 = new JPanel();

		p1.setLayout(new GridLayout(0,2));

		// 화면 그리기
		for(int i=0; i<label.length; i++) {
			label[i] = new JLabel( name[i] );
			label[i].setHorizontalAlignment(JLabel.CENTER);
			if(i==2)	textF[i] = new JPasswordField(20);
			else		textF[i] = new JTextField(20);
			p1.add(label[i]);
			p1.add(textF[i]);
		}

		p2.add(btnSave);
		p2.add(btnCancel);

		// 오늘날짜 가져오기
		LocalDate now = LocalDate.now();

		// 오늘날짜 가입일에 넣어주고 수정할 수 없도록 비활성화 하기
		textF[textF.length-1].setText(now.toString());
		textF[textF.length-1].setEnabled(false);

		btnSave.addActionListener(this);
		btnCancel.addActionListener(this);

		add(p1, BorderLayout.CENTER);
		add(p2, BorderLayout.SOUTH);

		setSize(350, 250);
		setDefaultCloseOperation(DISPOSE_ON_CLOSE);

		setTitle("회원 등록");
		setLocationRelativeTo(null);
		setVisible(true);

	}
```



```
	@Override
	public void actionPerformed(ActionEvent e) {

		// 사용자가 입력한 데이터 저장할 배열 선언 및 생성
		String data[] = new String[5];

		// 저장버튼 처리
		if(e.getSource() == btnSave ) {

			// 입력한 데이터 가져와서 배열에 담기, 데이터 앞뒤의 공백 제거하기
			for(int i=0; i<data.length; i++) {
				data[i] = textF[i].getText().trim();
			}

			// 아이디, 비밀번호, 이름 필수 체크
			// 비어있으면 메세지 창 보여주고, 커서를 해당하는 부분으로 옮기기
			// 메시지 예시> 아이디는 필수입니다
			for( int i=0; i<3; i++) {
				if(data[i].equals("")) {
					JOptionPane.showMessageDialog(this, label[i]
							+ "는 필수 입니다.");
					textF[i].requestFocus();
					return;
				}
			}

			// ---   db 테이블에 데이터 추가 작업 ------

			MemberDAO dao = new MemberDAO();

			// 입력한 회원정보로 vo변수 선언하고 생성하기
			MemberVO vo = new MemberVO(data[0], data[1], data[2], data[3], data[4]);

			// 회원 등록 성공하면 성공 메세지창 보여주고 현재 화면 닫기
			// 회원 등록 실패하면 실패 메세지창 보여주기, 현재 화면 그대로 유지
			// 성공메세지 : "회원 등록 성공", 실패메세지: "회원 등록 실패"
			if( dao.insertMember(vo) ) {
				JOptionPane.showMessageDialog(this, "회원 등록 성공");
				dispose();
			}else {
				JOptionPane.showMessageDialog(this, "회원 등록 실패");
			}
		}
		// 취소 버튼 처리
		else if(e.getSource() == btnCancel) {
			// 현재 화면 닫기, 회원 관리 화면은 종료 안됨.
			dispose();
		}

	}

}
```

SwingSearch 클래스

```
package s학번
import 생략;

public class SwingSearch extends JFrame implements ActionListener, ListSelectionListener{

	private JTextField tfId = new JTextField(10);
	private JButton btnSearch = new JButton("검색");
	private JButton btnDelete = new JButton("삭제");
	private JButton btnInsert = new JButton("회원가입");
	private JScrollPane sp = new JScrollPane();

	// 테이블과 테이블 안의 데이터 모델 선언
	private DefaultTableModel model;
	private JTable table;

	SwingSearch(){

		JPanel p1 = new JPanel();
		JPanel p2 = new JPanel();

		JLabel lbId = new JLabel("아 이 디");

		p1.add(lbId); p1.add(tfId);
		p1.add(btnSearch); p1.add(btnDelete); p1.add(btnInsert);

		// 테이블에 넣을 컬럼명 만들기 – 변수명 colNames
		String colNames[] = "아이디,이름,비밀번호,주소,가입일".split(",");

		// 테이블 모델 생성하기, 테이블 데이터 더블클릭하여 수정할 수 없도록 하기
		model = new DefaultTableModel(colNames, 0) {
			public boolean isCellEditable(int row, int column) {return false;};
		};

		// 테이블, 스크롤패널 생성하기
		table = new JTable(model);
		sp = new JScrollPane(table);

		p2.setLayout(new BorderLayout());
		p2.add(sp);

		btnSearch.addActionListener(this);
		btnDelete.addActionListener(this);
		btnInsert.addActionListener(this);

		// 테이블에 이벤트리스너 추가
		table.getSelectionModel().addListSelectionListener(this);

		add(p1, BorderLayout.NORTH);
		add(p2, BorderLayout.CENTER);

		setTitle("회원 관리");
		setSize(800,500);
		setLocationRelativeTo(null);
		setDefaultCloseOperation(DISPOSE_ON_CLOSE);
		setVisible(true);

	}
```



```
	@Override
	public void actionPerformed(ActionEvent e) {

		// 입력된 아이디값 가져오기, 앞뒤 공백 제거하기
		String strId = tfId.getText().trim();

		if(e.getSource() == btnSearch) {

			// 회원이 있으면 테이블에 회원정보 넣기
			MemberDAO dao = new MemberDAO();

			// 회원 검색하여 어레이 리스트에 저장 – 리스트명 list
			ArrayList<MemberVO> list = dao.selectMember(strId);

			// 회원이 없으면 메세지창 보여주고 스톱 메시지> XXX 회원이 없습니다.
			if(list.size() == 0) {
				JOptionPane.showMessageDialog(this, strId + "회원이 없습니다.");
				return;
			}

			// 테이블에 있는 기존 데이터 지우기
			model.setRowCount(0);

			// 검색해 온 데이터 테이블에 넣기
			for( MemberVO vo : list) {
				String arr[] = {vo.getId(), vo.getName(), vo.getPass(),
						vo.getAddr(), vo.getJoinday()};
				model.addRow(arr);
			}
		} else if( e.getSource() == btnDelete){
			// 아이디 비어 있는지 체크, 비어 있으면 메시지 창 보여주고 끝.
			// 메시지 >> 삭제할 아이디를 입력하세요.
			if(strId.equals("")) {
				JOptionPane.showMessageDialog(this,
						"삭제할 아이디를 입력하세요.");
				return;
			}

			// id에 해당하는 회원 삭제. 삭제 성공하면 회원 전체 검색하기
			// 삭제 성공하면 성공 메시지 창 보여주기. 메시지>>xxx 삭제 성공
			// 실패하면 실패메세지 창 보여주기. 메시지>> xxx 삭제 실패
			MemberDAO dao = new MemberDAO();
			if( dao.deleteMember(strId) ) {
				JOptionPane.showMessageDialog(this, strId + " 삭제 성공");
				tfId.setText("");
				btnSearch.doClick();
			}else {
				JOptionPane.showMessageDialog(this, strId + " 삭제 실패");
			}
		}else if( e.getSource() == btnInsert ) {
			// 회원 가입화면 실행시키기
			new SwingRegister();
		}

	}

	@Override
	public void valueChanged(ListSelectionEvent e) {
		// 테이블에서 선택된 행 번호를 가져옴.
		int sel = table.getSelectedRow();
		// 선택된 행이 있으면, 선택된 행의 아이디를 텍스트박스에 넣기
		if( sel >= 0  )
			tfId.setText( (String)model.getValueAt(sel, 0) );
	}

	public static void main(String[] args) {  new SwingSearch();  }
}
```