09. 회원 가입, 수정, 탈퇴

1) 회원 가입

(1) 우편번호 검색

→ 저장된테이블의 내용을 원하는 검색어에 맞게 **불러와서 출력**

SQL 구문 : select * from zipcode where area3 like '미아 3 동%';
Select area1 from zipcode ~

필드 1 개를 저장할 때 ➔ 자료형에 맞게 반환형을 설정
필드 1 개 이상을 저장할 때
➔ 반환형의 자료형을 DTO 형(객체)로 설정한 뒤,
➔ 그 **반환형이 여러 개**가 저장되려면 List, Set 등의 타입을 사용해야한다.
> 반환형 : zipcodeBean ➔ 반환된 값'들' : Vector<ZipcodeDTO>

(2) 회원가입

→ 테이블에 **데이터를 저장**하는 방법

(3)회원가입 과정 (과정을 이해하고 따라가면 이해하기 쉽다)

Login.jsp → 회원가입(클릭) → memberReg()호출 (script.js) → **agreement.jsp**(이동) → 동의 → **Register.jsp(이동)** → 정보입력 → id 중복체크, 우편번호체크(**script.js** 에서 처리)
→ 회원가입(클릭) → 입력이 다 됐는지 체크(script.js 에서 체크한 후 **RegsterProc.jsp** 로 submit) → RegisterProc.jsp(**최종확인**페이지) → 확인완료(클릭, submit) → **MemberInsert.jsp** → MemberDAO 객체를 생성하고, 객체의 method 인 **memberInsert()**를 실행하여, 입력한 정보들을 유에 저장하고 그 성공여부를 boolean 값으로 반환받음.

Login.jsp 에서의 로그인 버튼 코드
<input type="button" value="회원가입" onclick="**memberReg()**">

: script.js 내의 memberReg() 함수를 호출한다.

Script.js 의 memberReg() 함수

```
function memberReg(){ //회원가입으로 가는 함수
        document.location="agreement.jsp";
}
```

(agreement.jsp 는 단지 약관 나열로 pass)

Register.jsp (: 회원 정보 입력 페이지)

```
<%@ page contentType="text/html;charset=utf-8" import = "member.*"%>
<jsp:useBean id="member" class="member.MemberDTO"/>
<jsp:setProperty name="member" property="*"/>
<html>
<head>
<title>회원가입</title>
</head>
<body class="register" onLoad="regForm.mem_id.focus()">
<br><br>
<table align="center" border="0" cellspacing="0" cellpadding="5">
  <tr>
    <td align="center" valign="middle" bgcolor="#FFFFCC">
      <table border="1" cellspacing="0" cellpadding="2"   align="center">
        <form name="regForm" method="post" action="RegisterProc.jsp">
          <tr> <!-- align="center" bgcolor="#996600" -->
            <td colspan="3"><b>회원 가입</b></td>
          </tr>
          <tr>
            <td width="16%">아이디</td>
            <td width="57%"> <input type="text" name="mem_id" size=
                        <input type="button" value="ID중복확인"
       onClick="idCheck(this.form.mem_id.value)"></td>
            <td width="27%">아이디를 적어 주세요.</td>
          </tr>
,,,,,,
,,,,,        (생략)
,,,,,
          <tr>
            <td>우편번호</td>
            <td> <input type="text" name="mem_zipcode" size="7">
                  <input type="button" value="우편번호찾기" onClick="zipCheck()"></td>
            <td>우편번호를 검색 하세요.</td>
          </tr>
          <tr>
            <td>주소</td>
            <td><input type="text" name="mem_address" size="70"></td
            <td>주소를 적어 주세요.</td>
          </tr>
          <tr>
            <td colspan="3" align="center">
             <input type="button" value="회원가입" onclick="inputCheck()">
                         
             <input type="reset" value="다시쓰기">
            </td>
          </tr>
        </form>
      </table>
    </td>
  </tr>
</table>
</body>
</html>
```

jsp:useBean 태그로 MemberDTO 객체 생성
jsp:setProperty 태그로 모든 객체값 setting

봐야할 부분은

script.js 의 idCheck() 함수 호출

```
function idCheck(id){
    if(id==""){
            alert("id를 입력해주세요");
            document.regForm.mem_id.focuse();
    }else{
            url="IdCheck.jsp?mem_id="+id;
            // 1. 호출할 문서명 2. 창의 제목 3.창의 옵션
window.open(url,"post","left=450, top=150, width=300, height=150");
    }
```

이 작업을 수행하기 전에, zipcode를 **db**에 먼저 넣어둬야함

```
create table zipcode(zipcode varchar(7) NOT NULL,
                     area1 varchar(10),,,,,)
insert into values(,,,,,)

function zipCheck(){
      url="ZipCheck.jsp?check=y";
      window.open(url,"post","left=450, top=150, width=300,,,,생략);
}
```

Zipcheck.jsp (: 동을 검색하여, 우편번호와 주소값을 넣어주는 페이지)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"pageEncoding="UTF-8" import="java.util.*. member.*"%>
<!DOCTYPE html>
<jsp:useBean id="memMgr" class="member.MemberDAO"/>
<%
        request.setCharacterEncoding("utf-8");
        String check = request.getParameter("check"); // self
        String area3 = request.getParameter("area3"); // self
        Vector<ZipcodeDTO> zipcodeList = memMgr.zipcodeRead(area3);
        int totalList = zipcodeList.size();
%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>우편번호 검색</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script>
        //동이름을 체크할 함수선언
        function dongCheck() {
                if (document.zipForm.area3.value == "") {
                        alert("동이름을 먼저 입력하세요!");
                        document.zipForm.area3.focus();
                        return;
                }
                document.zipForm.submit();
        }

        function sendAddress(zipcode, area1, area2, area3, area4){
                var address = area1+" "+area2+" "+area3+" "+area4;
                opener.document.regForm.mem_zipcode.value=zipcode;
                opener.document.regForm.mem_address.value=address;
                self.close();
        }
</script>
</head>
<body bgcolor="#FFFFCC">
        <b>우편번호 찾기</b>
        <form name="zipForm" method="post" action="ZipCheck.jsp">
        <table>
                <tr>
                        <td><br> 동이름 입력:<input type="text" name="area3">
                        <input type="button" value="검색" onclick="dongCheck()">
                        <input type="hidden" name="check" value="n">
                </td>
                </tr>
        </form>
        <%
                if(check.equals("n")){
                        if(zipcodeList.isEmpty()){
        %>
                <tr>
                        <td align="center"><br>검색된 레코드가 없습니다.</td>
                </tr>
```

jsp:useBean 태그로 MemberDAO객체 생성

(1) 재귀호출 부분이다.
check : (처음 넘어올땐 y)
area3 : (동 단위 주소) 처음엔 null;
입력을 하고 찾기를 하면 리프레쉬되면서 값이 들어간다.
(2) ZipcodeDTO객체들을 여러 개 갖는 Vector zipcodeList 생성
그 안에는 area3이 들어가는 zipcode를 읽어들여서 넣어준다.
(3) 마지막으로 그 Vector List의 총 레코드수를 기록해둔다.

해당 문서의 area3 값이 비어있으면, alert
입력이 되어 있다면, submit 해준다.

Javascript 에서 **opener** 는 **자기 자신을 열어준 객체, 즉 부모창**

:부모창 문서의 regForm 객체의 mem_zipcode 의 값에 자신의 zipcode를 넣어준다.
마찬가지로 mem_address의 값에 자신의 address 값을 넣는다.

```
<%          }else{ %>
            <tr>
                <td align="center"><br> *검색후 ,아래 우편번호를 클릭하면 자동으로 입력됩니다</td>
            </tr>
<%
            for(int i=0; i<totalList;i++){
                ZipcodeDTO zipBean = zipcodeList.elementAt(i);
                String tempZipcode = zipBean.getZipcode().trim();
                String tempArea1 = zipBean.getArea1().trim();
                String tempArea2 = zipBean.getArea2().trim();
                String tempArea3 = zipBean.getArea3().trim();
                String tempArea4 = zipBean.getArea4().trim();
                System.out.println(tempZipcode);
%>
            <tr>
                <td style="color:black;">
                <a href="JavaScript:sendAddress('<%=tempZipcode %>', '<%=tempArea1 %>', '<%=tempArea2 %>',
'<%=tempArea3 %>', '<%=tempArea4 %>')" style="display:block; color:black;">
                <%=tempZipcode %> <%=tempArea2 %> <%=tempArea3 %> <%=tempArea4 %></a><br>
<%
                    }//for문 여기서 종료
                }//else
            }//if
%>
                </td>
            </tr>
            <tr>
                <td align="center"><br> <a href="JavaScript:this.close()">닫기</a>
            </table>
</form>
```

for 문

(1) ZipcodeDTO 객체를 검색된 갯수만큼 생성

(2)인덱스번호는 i 값이고, trim()으로 공백을 없애고 area1 , area2 , area3, area4 를 temp 변수에 넣어준다.

(3) sendAddress() 함수를 통해 부모창으로 값들을 보낸다.

(4) 해당 temp를 출력해주는 반복구문이다.

MemberDAO.jsp 내부 zipcodeRead() method(: zipcode 를 area3 로 검색해서 ZipcodeDTO 객체 List 로 리턴)

```
public Vector<ZipcodeDTO> zipcodeRead(String area3){
        Vector<ZipcodeDTO> vecList = new Vector();
        try {
            con = pool.getConnection();
            sql = "select * from zipcode where area3 like '"+area3+"%'";
            pstmt = con.prepareStatement(sql);
            rs = pstmt.executeQuery();
            while(rs.next()) { // 찾은 레코드가 1개라도 있다면
                ZipcodeDTO tempZipcode = new ZipcodeDTO();
                tempZipcode.setZipcode(rs.getString("zipcode"));
                tempZipcode.setArea1(rs.getString("area1"));
                tempZipcode.setArea2(rs.getString("area2"));
                tempZipcode.setArea3(rs.getString("area3"));
                tempZipcode.setArea4(rs.getString("area4"));
                vecList.add(tempZipcode);
            }
        }catch(Exception e) {
            System.out.println("zipcode() error:"+e);
        }finally {
            pool.freeConnection(con,pstmt,rs);
        }
        return vecList;
```

RegisterProc.jsp (: 회원가입을 위해 입력한 정보가, 모두 맞는지 체크하는 페이지)

```
<%@ page contentType="text/html;charset=UTF-8"%>
<jsp:useBean id="member" class="member.MemberDTO"/>
<jsp:setProperty name="member" property ="*"/>
<html>
<head>
<title>회원가입 확인</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script language="JavaScript" src="script.js"></script>
</head>
<body>,,,생략,,,
<form name="regForm" method="post" action="MemberInsert.jsp">
        <tr align="center" bgcolor="#996600">
          <td colspan="3"><font color="#FFFFFF"><b>
             <jsp:getProperty name="member" property="mem_name"/>님이 작성하신 내용입니다. 확인해주세요.
        </b></font></td>
        </tr>
        <tr>
          <td width="16%">아이디</td>
          <td width="57%">
          <input type="text" name="mem_id" size="15"   value="<jsp:getProperty name="member" property="mem_id"/>" >
                    </td>
        </tr>
        ,,,생략,,,
        <tr>
          <td colspan="3" align="center">
           <input type="submit" value="확인완료">
                       
           <input type="button" value="다시쓰기" onclick="history.back()">

        ,,,생략,,,,,
</body>
</html>
```

jsp:useBean 태그로 MemberDTO 객체 생성
jsp:setProperty 태그로 모든 객체값 setting
(Register.jsp로부터 넘겨받는 값으로)

<jsp:getProperty name=" " property = " " />
태그를 이용해서 입력받는 정보들을 보여주는 페이지

MemberInsert.jsp (: MemberDAO 의 memberInsert() method 를 이용해, 입력받는 값을 DB 에 저장하는 페이지)

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" import = "member.*" %>
<%
        request.setCharacterEncoding("utf-8");//한글처리
%>
<jsp:useBean id="member" class="member.MemberDTO"/>
<jsp:setProperty name="member" property="*"/>
<%
      //회원가입시켜주는 메서드를 호출
      MemberDAO memMgr = new MemberDAO();
        boolean check = memMgr.memberInsert(member);
        System.out.println("MemberInsert.jsp의 check=>"+check);
%>
<!DOCTYPE html>
<html>
<body>
<%
          if(check){
                  out.println("<b>회원가입을 축하합니다.</b><br>");
                  out.println("<a href=Login.jsp>로그인</a>");
          }else{
                  out.println("<b>다시 입력해주세요</b><br>");
                  out.println("<a href=Register.jsp>다시 기입</a>");
          }
%>
```

jsp:useBean 태그로 MemberDTO 객체 생성
jsp:setProperty 태그로 모든 객체값 setting
(Register.jsp로부터 넘겨받는 값으로)

MemberDAO객체의 memberInsert method에
DTO 객체인 member를 매개변수로 넘겨 호출

DB 입력의 성공여부를 체크하는 boolean

MemberDAO.java 에서 memberInsert()

```
public boolean memberInsert(MemberDTO member) {
                  boolean check = false; // 성공여부를 체크할 boolean 변수
                  try {
                          con = pool.getConnection();
                          con.setAutoCommit(false); // 자동 commit 방지
                          sql="insert into member values(?,?,?,?,?,?,?,?)"; // 회원가입에 들어가는 정보를 넣는 sql 구문
                          pstmt = con.prepareStatement(sql);  //pstmt 객체에 sql구문 결합
                          pstmt.setString(1, member.getMem_id());
                          pstmt.setString(2, member.getMem_passwd());
                          pstmt.setString(3, member.getMem_name());
                          pstmt.setString(4, member.getMem_email());
                          pstmt.setString(5, member.getMem_phone());
                          pstmt.setString(6, member.getMem_zipcode());
                          pstmt.setString(7, member.getMem_address());
                          pstmt.setString(8, member.getMem_job());
                          int insert = pstmt.executeUpdate();
                          System.out.println("insert 데이터 입력 성공 :"+insert);
                          if(insert > 0) check = true;
                          con.commit();
                  }catch(Exception e) {
                          System.out.println("memberInsert() error"+e);
                  }finally {
                          pool.freeConnection(con,pstmt);
                  }
                  return check;
```

회원 정보 8가지
필드를 각 ?에 입력
순서 중요함

성공여부 판단

성공이면 true

회원정보 '수정', '삭제'도 가입과 크게 다르지 않다.

플로우를 잘 써내려간뒤, MemberDAO.java 에 method 를 잘 구현 (DB 접속, sql 문 구성, pstmt set, return)하고, 그 순서대로 jsp 파일, javascript 함수, submit 등을 구성해주면 끝

그래서 패스 ㅋㅋㅋㅋ