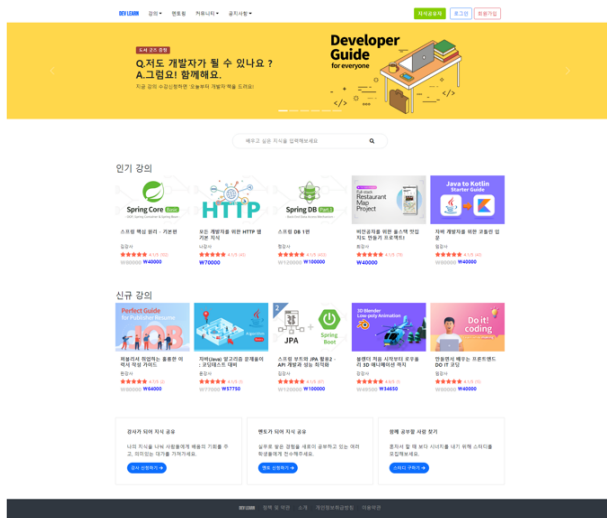


# 포트폴리오

개발 PLATFORM	프로젝트 개요
Java8, Servlet/JSP, HTML5, CSS3, JavaScript, jQuery, Spring, myBatis3, maven, bootstrap	HTML, CSS, Bootstrap, jQuery, JavaScript 를 이용한 웹 구현
Apache Tomcat 8.5.81	Maven 을 이용한 의존성 관리
Oracle	myBatis framework 를 이용하여 JDBC 연동 코드나 트랜잭션 코드 간소화
STS3 / sql Developer	AJAX 를 사용하여 웹에서 데이터 조회 시 불필요한 요청 최소화
Git	관리자 영역과 클라이언트 영역을 분리하여 효율성 제고 및 유지 보수의 편의성 제공

## 프로젝트 특징

### MAIN



이벤트 배너 : Bootstrap - Carousel

- 최근 수강 강의:  
로그인시 로그인한 유저가 최근에 수강중이던 강의가 출력 됨
- 인기 강의 / 신규 강의:  
강의 영역에 마우스 오버시  
jQuery로 강의 설명 영역이 보이게 함
- 장바구니, 찜하기 버튼 마우스 오버시 글씨 리본이 보이게 함

## 강의 - 강의상세



- bootstrap의 내비게이션 바를 사용하여 항목을 표시
- 한 화면에 표시 가능한 내용들은 scrollspy를 통해 클릭만으로 해당 위치에 즉시 이동되도록 구현

강의소개 커리큘럼 수강생 수강 전 문의 커뮤니티 새 소식

### 강의 카뎀고리

#### AJAX로 페이지 만들어 보기

강의를 수강해 주신 모든 분들께 감사드립니다.

이 강의는 AJAX를 사용해서 간단한 페이지를 만들어 보는 과정을 담고 있습니다.

강의는 지속적으로 업데이트 될 예정이며, 강의에 대한 궁금증이나 내용에 대한 질문은 문의 게시판이나 강사 이메일을 통해 해결해 드리도록 하겠습니다.

다시 한 번 수강생 분들께 감사드리며, 수강 후에는 한 계단 높은 곳에 계기를 기원합니다!

5% 할인 중인 강의

5% 28,500원 30,000원

수강신청 하기

장바구니 담기

- 지식공유자: 정영서
- 총 30개 수업(5시간)

## 강의 - 수강 전 문의

### 수강 전 문의

강의를 수강하기 전에 궁금한 사항을 질문해보세요. 강사가 직접 친절하게 답변해 드립니다!

문의 1개

김희원 2022-06-30

안녕하세요! 혹시 먼저 들으면 도움이 되는 강의가 있을까요?

답변 >

문의 작성

- 공개된 게시판이므로 전화번호 등의 개인정보는 남기지 않아주세요.
- 광고, 욕설, 비방, 도배 등 강의와 무관한 글은 예고 없이 삭제될 수 있습니다.

등록 >

### 수강 전 문의

강의를 수강하기 전에 궁금한 사항을 질문해보세요. 강사가 직접 친절하게 답변해 드립니다!

문의 1개

김희원 2022-06-30

안녕하세요! 혹시 먼저 들으면 도움이 되는 강의가 있을까요?

정영서 2022-06-30

기본적으로 필요한 것들은 강의에 소개되어 있습니다 :)

문의 작성

- 공개된 게시판이므로 전화번호 등의 개인정보는 남기지 않아주세요.
- 광고, 욕설, 비방, 도배 등 강의와 무관한 글은 예고 없이 삭제될 수 있습니다.

등록 >

- 강의 상세 페이지에서 수강 전 문의 탭을 클릭하면 문의 화면으로 이동
- 모든 회원이 자유롭게 문의를 작성하고 타 회원의 문의 내용 또한 확인 가능
- 답변 작성은 해당 강의의 강사만 가능하며, 이미 답변한 문의의 경우 답변창이 보이지 않음
- AJAX로 페이지를 이동하지 않고 문의 및 답변 출력, 작성, 삭제 가능
- 페이지 또한 페이지 이동이 아닌 더보기 버튼으로 5개씩 더 표시되도록 설정

질문과 답변  
함께 토론하며 문제를 해결해나가요

자유입력

입력과 답변

• 스타티

질문

답변

🔍

공공기관 질문을 입력하세요!

검색

최신순

내대표고집

글쓰기

해답률

spring에서 mapper에 오류가 떠졌는데 어디가 잘못된건지 봐주세요

게신가요??

아래 코드 실행하다가 갑자기 이상한 에러가 왔습니다. 아래 메시지를 봐도 무슨 에러인지 모르겠어서 어떻게 질문 바랍니다

해결이후는 어떤지 아닌지 아닌지 질문 부탁드립니다

태그

2022-06-30 08:39:53

조회수 : 17

[-] spring에서 mapper에 오류가 떠졌는데 어디가 잘못된건지 봐주세요

게신가요??

질문됨 | 2022-06-30 09:02:53 | 조회수 : 0

[-] spring에서 mapper에 오류가 떠졌는데 어디가 잘못된건지 봐주세요

게신가요??

질문됨 | 2022-06-30 09:00:48 | 조회수 : 0

[-] spring에서 mapper에 오류가 떠졌는데 어디가 잘못된건지 봐주세요

게신가요??

질문됨 | 2022-06-30 08:57:03 | 조회수 : 1

해답률

ajax 관련 질문입니다.

controller에서 html로 .jsp가 받아왔는지 출력하려는 거예요 되나요??

해답률

2022-06-17 17:00:38

조회수 : 38

[-] ajax 관련 질문입니다.

질문됨 | 2022-06-17 17:02:39 | 조회수 : 0

- 커뮤니티 - 질문과 답변 상세

**[예제 1] java 관련 질문있습니다.**

```

class Job extends Woman{
    String job;
    public void info() {
        System.out.println("여자의 직업은 "+job+"입니다.");
        System.out.println("여자의 이름은 "+name+", 나이는 "+age+"살입니다."); //주기
    }
}

```

**[예제 2]**

```

class Job extends Woman{
    String job;
    public void info() {
        super.info(); //주기
        System.out.println("여자의 직업은 "+job+"입니다.");
    }
}

```

이전글 : [re] [re] java관련 질문있습니다.

다음글 :

- 3

## 관리자 - 회원 관리

Admin

대시보드
메뉴 현황
회원 관리
콘텐츠
공지사항
나가기

회원관리

전체 회원

권한 신청

회원 신고

117명의 회원 1/2페이지

새로고침

검색조건

회원등급

활성여부

검색

#	이메일	이름	전화번호	회원 등급	활성여부
117	a@mentor.com	나멘토	010-0000-1111	멘토	비활성 계정
116	mentor1@mentor.com	멘토장영환	010-0000-0000	멘토	활성계정
115	bbb	김태스	010-0000-0000	멘토	활성계정
114	c@mentor.com	김태스	010-0000-0000	회원	비활성 계정
113	admin	관리자		관리자	활성계정
112	c@mentor.com	나멘토	010-0000-0000	멘토	비활성 계정
111	c@mentor.com	김영우	010-1111-2222	회원	활성계정
110	aaa	김태스	010-0000-1111	회원	비활성 계정
109	user@mentor.com	김태스	010-1111-1111	회원	활성계정
108	mentor@mentor.com	장태일	010-1111-1111	멘토	활성계정

1
2
3
4
5
6
7
8
9
10
>

- 회원관리 탭에서 전체 회원의 리스트 출력
- 아이디 / 이름 / 가입일 혹은 회원의 등급, 활성화 여부를 조건으로 회원 검색 가능
- 회원의 정보를 클릭하면 Modal 창으로 상세 정보 출력
- 회원의 상세 정보 내에서 등급이나 활성/ 비활성 여부를 변경할 수 있음
- 전체회원 / 권한신청 / 회원신고의 내용을 ajax 를 활용하여 데이터를 받아옴

## 관리자 - 회원 관리

회원 상세 정보

×

E-mail	a@mentor.com	이름	나멘토
비즈니스 메일	a@mentor.com	전화번호	010-0000-1111
가입일	2022-06-17 16:40:58	마지막 로그인	2021-02-19 00:00:00
정보 업데이트	2018-10-19 00:00:00	메일 수신	미동의

회원 등급

멘토

변경하기

로그인 가능

가능

상태

:: 상태 ::

자세히

메모

변경하기

Close

- 회원의 상세 정보를 모달로 확인할 수 있음
- 회원의 등급을 변경하거나 로그인 활성/ 비활성을 변경할 수 있음
- 회원의 활성 / 비활성 상태가 변동이 있었던 경우 자세히 버튼이 활성화 되어 변경 내역을 확인할 수 있음

```
1 package com.sp.dev.lectureInquiry;
2
3 public class LecInq {
4     private int inquiryNum;
5     private int lectureNum;
6     private String memberEmail;
7     private String qNickname;
8     private String aNickname;
9     private String categoryCode;
10    private String lectureSubject;
11    private String lectureContent;
12    private int lecturePrice;
13    private int dcPercent;
14    private int finalPrice;
15    private String memberNickname;
16
17    public String getCategoryCode() {
18        return categoryCode;
19    }
20    public void setCategoryCode(String categoryCode) {
21        this.categoryCode = categoryCode;
22    }
23    public String getLectureSubject() {
24        return lectureSubject;
25    }
26    public void setLectureSubject(String lectureSubject) {
27        this.lectureSubject = lectureSubject;
28    }
29    public String getLectureContent() {
30        return lectureContent;
31    }
32    public void setLectureContent(String lectureContent) {
33        this.lectureContent = lectureContent;
34    }
35    public int getLecturePrice() {
36        return lecturePrice;
37    }
38    public void setLecturePrice(int lecturePrice) {
39        this.lecturePrice = lecturePrice;
40    }
41    public int getDcPercent() {
42        return dcPercent;
43    }
44    public void setDcPercent(int dcPercent) {
45        this.dcPercent = dcPercent;
46    }
47    public String getMemberNickname() {
48        return memberNickname;
49    }
```

```

1 package com.sp.dev.lectureInquiry;
2
3 import java.net.URLDecoder;
4
5 @Controller("lectureInquiry.lectureInquiryController")
6 @RequestMapping("/lectureInquiry/*")
7 public class LecInqController {
8     @Autowired
9     private LecInqService service;
10    @Autowired
11    private LecturesService lservice;
12    @Autowired
13    private MyUtil myUtil;
14
15    // 연결하면서 강의 번호 받아오기 - 이거 없으면 404
16    @RequestMapping("inquiry")
17    public String inquiry(@RequestParam int lectureNum,
18        Model model) throws Exception {
19
20        Lectures dto = lservice.readLecture(lectureNum);
21        if(dto == null) {
22            return "redirect:/lectures/lectures";
23        }
24
25        dto.setLectureContent(dto.getLectureContent());
26
27        model.addAttribute("dto", dto);
28        model.addAttribute("lectureNum", lectureNum); // 이걸로 강의번호를 받아온다
29
30        return ".lectureInquiry.inquiry";
31    }
32
33    // AJAX - 문의 등록
34    @ResponseBody
35    @RequestMapping(value = "insert", method = RequestMethod.POST)
36    public Map<String, Object> writeSubmit(LecInq dto, HttpSession session, @RequestParam int lectureNum) throws Exception {
37        SessionInfo info = (SessionInfo) session.getAttribute("member");
38        String state = "true";
39
40        try {
41            dto.setQmember(info.getMemberEmail());
42            dto.setLectureNum(lectureNum);
43            service.insertInquiry(dto);
44        } catch (Exception e) {
45            state = "false";
46        }
47
48        Map<String, Object> model = new HashMap<>();
49        model.put("state", state);
50    }
51
52 }

```

```

LecInq.java  LecInqController.java  LecInqService.java  LecInqServiceImpl.java
67     model.put("state", state);
68
69     return model;
70 }
71
72 // 답변 등록(입력창이 필요)
73 @ResponseBody
74 @RequestMapping(value = "answer", method = RequestMethod.POST)
75 public Map<String, Object> answerSubmit(LecInq dto, HttpSession session, @RequestParam int lectureNum) throws Exception {
76     SessionInfo info = (SessionInfo) session.getAttribute("member");
77     String state = "true";
78
79     try {
80         dto.setAmember(info.getMemberEmail());
81         dto.setLectureNum(lectureNum);
82         service.insertAnswer(dto);
83     } catch (Exception e) {
84         state = "false";
85     }
86
87     Map<String, Object> model = new HashMap<>();
88     model.put("state", state);
89
90     return model;
91 }
92
93 // 리스트 (페이지에 들어가면 질문+답변 2가지가 모두, 1명의 문답만이 아니라 전체 사용자의 문답이 한꺼번에 다 나오도록)
94 // sb쪽을 보면 문의 쪽은 list/article.jsp 형태로 나뉘지만 방명록은 guest.jsp 하나만 존재한다 >> 내 경우는 list만 만들면 되는 것
95 // 따라서 방명록 쪽에는 listGuest만 있고 readGuest 같은건 없다
96 // 반면 문의는 listInquiry와 readInquiry가 모두 있어야 한다(insert, dataCount, delete는 다 있는거고 문의는 answer도 필요하긴 하다)
97 @ResponseBody
98 @RequestMapping(value = "list")
99 public Map<String, Object> list(
100     @RequestParam int lectureNum,
101     @RequestParam(defaultValue = "all") String condition,
102     @RequestParam(defaultValue = "") String keyword,
103     @RequestParam(value = "pageNo", defaultValue = "1") int current_page,
104     HttpSession session,
105     Model model
106 ) throws Exception {
107
108     keyword = URLDecoder.decode(keyword, "utf-8");
109
110     Map<String, Object> map = new HashMap<String, Object>();
111     map.put("lectureNum", lectureNum); // 이게 가장 먼저 나와야 그 아래에서 강의별 dataCount든 list든 가져올 수 있다!
112
113     int rows = 5;
114     int dataCount = service.dataCount(map);
115     int total_page = myUtil.pageCount(rows, dataCount);

```

```

116         if (current_page > total_page) {
117             current_page = total_page;
118         }
119
120         int start = (current_page - 1) * rows + 1;
121         int end = current_page * rows;
122         map.put("start", start);
123         map.put("end", end);
124
125         List<LecInq> list = service.listLecInq(map);
126         for(LecInq dto : list) {
127             dto.setQuestion(dto.getQuestion().replaceAll("\n", "<br>"));
128         }
129
130         Map<String, Object> mod = new HashMap<>();
131
132         mod.put("dataCount", dataCount);
133         mod.put("total_page", total_page);
134         mod.put("pageNo", current_page);
135         mod.put("list", list);
136
137         Lectures dto = lservice.readLecture(lectureNum);
138         model.addAttribute("dto", dto);
139         model.addAttribute("lectureNum", lectureNum);
140         return mod;
141     }
142
143     @ResponseBody
144     @RequestMapping(value = "deleteInquiry", method = RequestMethod.POST)
145     public Map<String, Object> deleteInquiry(@RequestParam int lectureNum, @RequestParam int inquiryNum, HttpSession session) throws Exception {
146         SessionInfo info = (SessionInfo) session.getAttribute("member");
147         Map<String, Object> map = new HashMap<>();
148         map.put("lectureNum", lectureNum);
149         map.put("inquiryNum", inquiryNum);
150
151         String state = "true";
152         try {
153             map.put("memberRole", info.getMemberRole());
154             map.put("memberEmail", info.getMemberEmail());
155             service.deleteInquiry(map);
156         } catch (Exception e) {
157             state = "false";
158         }
159
160         Map<String, Object> model = new HashMap<>();
161         model.put("state", state);
162         return model;
163     }
164 }

```



```
LecInq.java  LecInqController.java  LecInqService.java  LecInqServiceImpl.java
1 package com.sp.dev.lectureInquiry;
2
3 import java.util.List;
4
5
6 public interface LecInqService {
7     public void insertInquiry(LecInq dto) throws Exception;
8     public int dataCount(Map<String, Object> map);
9     public List<LecInq> listLecInq(Map<String, Object> map);
10    public void insertAnswer(LecInq dto) throws Exception;
11    public void deleteInquiry(Map<String, Object> map) throws Exception;
12    public void deleteAnswer(LecInq dto) throws Exception;
13 }
14
```

```

1 package com.sp.dev.lectureInquiry;
2
3 import java.util.List;
4
5
6
7
8
9
10
11 @Service("lectureinquiry.lecinqservice")
12 public class LecInqServiceImpl implements LecInqService {
13     @Autowired
14     private CommonDAO dao;
15
16     @Override
17     public void insertInquiry(LecInq dto) throws Exception {
18         try {
19             dao.insertData("lectureInquiry.insertInquiry", dto);
20         } catch (Exception e) {
21             e.printStackTrace();
22             throw e;
23         }
24     }
25
26
27     @Override
28     public int dataCount(Map<String, Object> map) {
29         int result = 0;
30
31         try {
32             result = dao.selectOne("lectureInquiry.dataCount", map);
33         } catch (Exception e) {
34             System.out.println(e.toString());
35         }
36
37         return result;
38     }
39
40     @Override
41     public List<LecInq> listLecInq(Map<String, Object> map) {
42         List<LecInq> list = null;
43
44         try {
45             list = dao.selectList("lectureInquiry.listLecInq", map);
46         } catch (Exception e) {
47             e.printStackTrace();
48         }
49
50         return list;
51     }
52
53     @Override
54     public void insertAnswer(LecInq dto) throws Exception {
55         try {

```

```

LecInq.java  LecInqController.java  LecInqService.java  LecInqServiceImpl.java
38     }
39
40     @Override
41     public List<LecInq> listLecInq(Map<String, Object> map) {
42         List<LecInq> list = null;
43
44         try {
45             list = dao.selectList("lectureInquiry.listLecInq", map);
46         } catch (Exception e) {
47             e.printStackTrace();
48         }
49
50         return list;
51     }
52
53     @Override
54     public void insertAnswer(LecInq dto) throws Exception {
55         try {
56             dao.updateData("lectureInquiry.insertAnswer", dto);
57         } catch (Exception e) {
58             e.printStackTrace();
59             throw e;
60         }
61     }
62
63     @Override
64     public void deleteInquiry(Map<String, Object> map) throws Exception {
65         try {
66             dao.deleteData("lectureInquiry.deleteInquiry", map);
67         } catch (Exception e) {
68             e.printStackTrace();
69             throw e;
70         }
71     }
72
73
74     @Override
75     public void deleteAnswer(LecInq dto) throws Exception {
76         try {
77             dao.updateData("lectureInquiry.deleteAnswer", dto);
78         } catch (Exception e) {
79             e.printStackTrace();
80             throw e;
81         }
82     }
83
84
85 }

```

```

63 //문의 등록
64 $(function() {
65     $(".btnSend").click(function() {
66         let question = $("#question").val().trim();
67         if(! question) {
68             $("#question").focus();
69             return false;
70         }
71
72         let url = "${pageContext.request.contextPath}/lectureInquiry/insert";
73         let query = "question=" + encodeURIComponent(question) + "&lectureNum=${lectureNum}";
74
75         const fn = function(data) {
76             $("#question").val("");
77             $(".inquiry-list-body").empty();
78             listPage(1);
79         };
80
81         ajaxFun(url, "post", query, "json", fn);
82     });
83 });
84
85 //답변 등록
86 $(function() {
87     $(".body").on("click", ".btnAns",function() {
88         let lectureNum = "${dto.lectureNum}";
89         let answer = $("#answer").val().trim();
90         if(! answer) {
91             $("#answer").focus();
92             return false;
93         }
94
95         let url = "${pageContext.request.contextPath}/lectureInquiry/answer";
96         let query = "answer=" + encodeURIComponent(answer) + "&lectureNum=${lectureNum}";
97
98         const fn = function(data) {
99             $("#answer").val("");
100             $(".inquiry-list-body").empty();
101             listPage(1);
102         };
103
104         ajaxFun(url, "post", query, "json", fn);
105     });
106 });
107
108

```

```

109 // 문의 리스트
110 function listPage(page) {
111     let url = "${pageContext.request.contextPath}/lectureInquiry/list";
112     let query = "pageNo=" + page + "&lectureNum=${lectureNum}";
113
114     const fn = function(data) {
115         printInquiry(data);
116     };
117     ajaxFun(url, "get", query, "json", fn);
118 }
119
120 function printInquiry(data) {
121     console.log(data);
122     const uid = "${sessionScope.member.memberEmail}";
123     let permission = "${sessionScope.member.memberRole == 99 ? 'true' : 'false'}";
124
125     let dataCount = data.dataCount;
126     let pageNo = data.pageNo;
127     let total_page = data.total_page;
128
129     $(".inquiry-count").attr("data-pageNo", pageNo);
130     $(".inquiry-count").attr("data-totalPage", total_page);
131
132     $("#listLecInq").show();
133     $(".inquiry-count").html("문의 " + dataCount + "개");
134
135     $(".more-box").hide();
136     if(parseInt(dataCount) === 0) {
137         $(".inquiry-list-body").empty();
138         return;
139     }
140
141     if(pageNo < total_page) {
142         $(".more-box").show();
143     }
144

```



```

LecInq.java  LecInqController.java  LecInqService.java  LecInqServiceImpl.java  inquiry.jsp
204 // 문의 삭제
205 $(function() {
206     $("body").on("click", ".deleteInquiry", function() {
207         if(! confirm("문의를 삭제하시겠습니까?")) {
208             return false;
209         }
210
211         let inquiryNum = $(this).attr("data-inquiryNum");
212         let url = "${pageContext.request.contextPath}/lectureInquiry/deleteInquiry";
213         let query = "inquiryNum=" + inquiryNum + "&lectureNum=${lectureNum}";
214         const fn = function(data) {
215             $(".inquiry-list-body").empty();
216             listPage(1);
217         };
218
219         ajaxFun(url, "post", query, "json", fn);
220
221     });
222 });
223
224 $(function() {
225     $(".more-box .more").click(function() {
226         let pageNo = $(".inquiry-count").attr("data-pageNo");
227         let total_page = $(".inquiry-count").attr("data-totalPage");
228
229         if(pageNo < total_page) {
230             pageNo++;
231             listPage(pageNo);
232         }
233     });
234 });
235

```

```

6      <!-- 문의 등록 -->
7      <insert id="insertInquiry" parameterType="com.sp.dev.LectureInquiry.LecInq">
8          insert into lectureinquiry(inquiryNum, lectureNum, qmember, question, q_regdate)
9          values (lectureinquiry_seq.NEXTVAL, #{lectureNum}, #{qmember}, #{question}, sysdate)
10     </insert>
11
12     <select id="dataCount" resultType="Integer" parameterType="map">
13         select NVL(COUNT(*), 0) from lectureInquiry
14         where lectureNum = #{lectureNum}
15     </select>
16
17     <!-- 리스트 출력 - 문의 테이블 / 강의 테이블 / 회원 테이블 join(문의자와 답변자(강사)의 닉네임을 따로 가져오느라 member가 2번 join됨. 다른 방법?) -->
18     <select id="listLecInq" parameterType="map" resultType="com.sp.dev.LectureInquiry.LecInq">
19         SELECT * FROM (
20             SELECT ROWNUM rnum, tb.* FROM (
21                 SELECT inquiryNum, qMember, question, TO_CHAR(q_regDate, 'YYYY-MM-DD') q_regDate, qm.memberNickname as qNickname,
22                     aMember, answer, TO_CHAR(a_regDate, 'YYYY-MM-DD') a_regDate,
23                     m.memberNickname as aNickname, l.memberEmail as instructorEmail,
24                     l.lecturePrice, l.dcPercent,
25                     CASE WHEN dcPercent = 0 THEN lecturePrice
26                         WHEN dcPercent > 0 THEN lecturePrice * (100-dcPercent) / 100
27                     END as finalPrice
28                 FROM lectureInquiry li
29                 JOIN lecture l ON l.lectureNum=li.lectureNum
30                 JOIN member m ON l.memberEmail=m.memberEmail
31                 JOIN member qm on qMember=qm.memberEmail
32                 WHERE l.lectureNum = #{lectureNum}
33                 ORDER BY inquiryNum DESC
34             ) tb WHERE ROWNUM <= #{end}
35             ) WHERE rnum >= #{start}
36     </select>
37
38     <!-- 문의 삭제 -->
39     <delete id="deleteInquiry" parameterType="map">
40         DELETE from lectureinquiry WHERE inquiryNum = #{inquiryNum}
41     </delete>
42
43     <!-- 답변 등록 -->
44     <update id="insertAnswer" parameterType="com.sp.dev.LectureInquiry.LecInq">
45         update lectureinquiry set amember = #{amember}, answer = #{answer}, a_regDate = sysdate
46         where inquiryNum = #{inquiryNum}
47     </update>
48
49     <!-- 답변 삭제 -->
50     <update id="deleteAnswer" parameterType="com.sp.dev.LectureInquiry.LecInq">
51         update lectureinquiry set amember = null, answer = null, a_regDate = null
52         where inquiryNum = #{inquiryNum}
53     </update>

```