# 1. 개발 환경

1. **개발툴**
   * Eclipse JAVA EE IDE Photon Version을 기본 개발툴로 사용한다.
   * Oracle Database11gExpress Edition을 Database 관리툴로 사용한다.
2. **버전관리계획**
   * Git 및 GitHub를 이용하여 단위 개발 및Version 관리를 한다.
   * 테스트 서버의 관리 및 작업 내용 반영은 최종 개발 책임자가 작업한다.
   * 개발 진행중인 작업 내용의 경우 개인 단위의 백업관리가 이루어지며, 개발 완료 작업 내용과 일(daily)단위의 백업은 최종 개발 책임자가 맡는다.
3. **데이터베이스 접근 계획**
   * 모든 개발자는 하나의 데이터베이스 서버를 공유하여 사용한다.
   * 데이터베이스의 접근은 WAS의 자체 Database Pooling모듈을 사용하며, DB 리소스는 독립적으로 획득할 수 없다.

# 2. 명명 규칙

1. **디렉토리 명명 규칙**
   * 디렉토리 명칭은 영문으로 사용한다.
   * <context root>하위에 메뉴의 깊이를 기준으로 디렉토리를 생성하며, 약어를 사용하지 않는다.
2. **JSP 파일 명명 규칙**
   * 해당 디렉토리 명칭을 말머리로 갖고 파일의 주요 기능을 표기한다.

|  |  |  |
| --- | --- | --- |
| XXXXX\_YYYYY.jsp  ⓐ ⓑ ⓒ | | |
| ⓐ | 디렉토리 명칭 | 메뉴명에 대한 약어 |
| ⓑ | 구분자 | ‘\_’ |
| ⓒ | 주요 기능 | 게시판 : list(목록), insert(글작성),  reply(댓글작성), update(수정),  searchlist(검색결과)  -검색 ->search  -로그인 - >login.jsp  -날짜및지역선택 -> selectdayandloc  -마이페이지->mypage.jsp  -관리자페이지->adminpage.jsp  -아이디찾기->searchid  -비밀번호찾기->searchpw  -개인정보조회->myinfo.jsp  -이용내역->myinfo\_myboard.jsp  -공지사항->notice  -차량등록->insertcar  -인증->accept  -예약->reserve  -이벤트->event |

* 구분자를 제외한 모든 섹션은 영문자로 시작하며, 대문자를 사용하지 않으며 가급적 의미를 알 수 없는 약어를 사용하지 않는다.
* 사용되는 확장자는 아래와 같다.

|  |  |
| --- | --- |
| **파일 형식** | **파일 확장자** |
| Java Servlet Page(JSP) | .jsp |
| Cascading Style Sheet(CSS) | .css |
| JavaScript | .js |
| Hyper Text Markup Language | .html |
| Web Resource | .gif, .jpg |
| Flash | .swf |
| Extensible markup language | .xml |
| JAVA | .java |

1. **패키지 명명 규칙 및 자바 파일 명명 규칙**
   * 패키지명은 3자리 이상으로 정의한다.
     + com.doncar.controller : Controller 파일 정의
     + com.doncar.common : aop 및 log 파일 정의
     + com.doncar.dto : dto 파일 정의
     + com.doncar.model : Service 및 DAO 파일 정의
   * 자바 파일은 3자리 이상의 package명 하위 단에 작성한다.
2. **자바스크립트 파일 명명 규칙**
   * 파일이 사용되는 기능명으로 명명한다
     + board : 게시판
     + list : 목록
     + rent : 렌트
     + search : 검색
     + carpool : 카풀
     + login : 로그인

예) board.js : 게시판 기능에 사용되는 자바스크립트 파일

1. **클래스 명명 규칙**

* DAO, DTO의 경우 해당 데이터베이스 테이블명칭을 그대로 갖는 클래스 이름을 사용하며, com.test.my.dto, com.test.my.model 을 상위 패키지로 갖는다.

예1) com.test.my.dto*.MemberDto.class*

예2) com.test.my.model*.MemberDao.class*

* Jar 파일의 경우 WEB-INF/lib 폴더에 위치 시킨다.

1. **객체 및 변수 명명 규칙**

* 객체의 경우 해당 객체가 페이지 내에서 유일할 경우 첫글자를 소문자로 바꾼 클래스명을 그대로 사용한다. 만약 해당 객체가 페이지 내에서 유일하지 않은 경우에는 해당 객체의 역할을 상세히 나타낼 수 있는 명칭을 말머리로 하여 선언한다.

- 예1) 클래스 MemeberDao의 객체 *memberDao*

- 예2 ) 클래스 MemberDto 의 객체 *firstMemberDto, lastMemberDto*

* 변수의 경우 단순 증감이며 for문 등의 제어문 내에서 선언되는 경우 <i,j,k> 등의 간단한 소문자를 사용한다.(‘p’는 예약된 변수로써, PreparedStatement의 index에서 사용한다.)

- 예) *for(int i=0;i<10;i++)*

* 특정 정보를 담고 있는 변수인 경우, 해당 정보의 성격을 나타낼 수 있는 명칭을 사용한다

- 예) String *rentboardlist* = “렌트게시판목록”;

- HTML내의 <input> 태그로 사용되는 필드 값을 전달받을 경우 동일한 이름의 변수에 할당한다.

1. **데이터베이스 명명 규칙**

* 테이블 명칭 : 테이블의 용도를 알수있도록 명명한다.

예) 렌트보드테이블 : RENT\_BOARD

* 필드 명칭 : 외부참조키가 아닌 경우 <TABLE NAME>+’\_’를 붙이는 형태로 사용하며, 외부참조키(FK)인 경우는 해당 FK의 원본 필드 명칭을 그대로 사용한다.

예1) RENT\_BOARD\_NO

* 시퀀스 명칭 : 말머리로 <필드명>+’\_SEQ’을 붙이는 형태로 사용한다

예 1) RENT\_BOARD\_NO\_SEQ

# 3. 개발 구조

1. **디렉토리 구조**

* 경로는 클래스 및 인터페이스 파일의 경우 java resource 폴더를, 그 외 파일들은

<context root>를 상위로 갖는다.

* Directory 이름은 의미있는 하나의 영문 단어로 구성하고 전부 소문자로 표기하도록 함. 또한 혼동을 피하기 위해 가급적 약어를 사용하지 않도록 함.
* 다음은 디렉토리 구성 규칙의 예임.
  + - JSP 파일 : <context root>+ ‘/WEB-INF/jsp’
    - JAVA Class 파일 : <java resource> + <package>
    - JAVA Source 파일 : <java resource> + <package>
    - XML 파일 : <context root>‘/WEB-INF/’

<context root> + ‘/WEB-INF/spring/appServlet/’

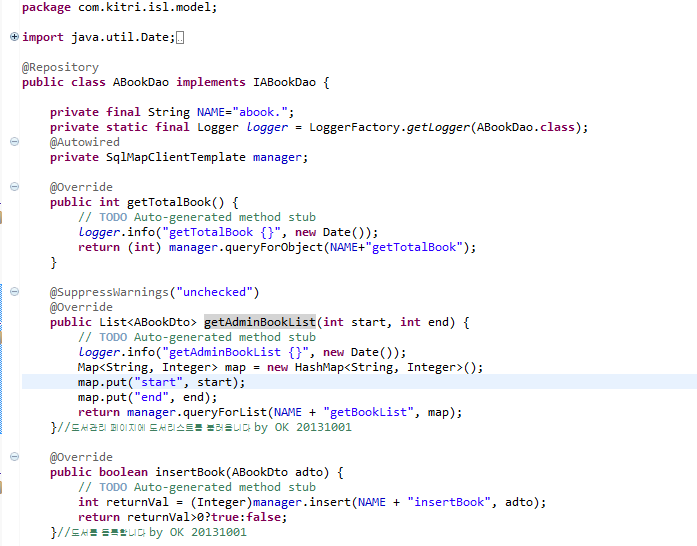
<context root> + ‘/WEB-INF/spring/sqlMap/’

<context root> + ‘/WEB-INF/views/’

* + - 이미지 파일 디렉토리 : <context root> + ’/images’
    - JavaScript 디렉토리 : <context root> + ‘/js’
    - CSS 디렉터리 : <context root> + ‘/css’

# 4. 코드 구조

1. **JAVA 코드 구조**
   * DAO 구조 예제



* + DTO 구조는 일반적인 BEAN의 형태를 갖는다

1. **JSP 및 HTML 코드 구조**

* 예제

