

코드정의서

- 개발자 커뮤니티

극락조(3 조)

목 차

1. 개요.....	
1.1. 목적.....	
2. 공통 법칙.....	
2.1. 작업 디렉토리.....	
2.2 소스 파일	
3. 명명법.....	
3.1. 공통 규칙	
3.2. 폴더명.....	
3.3. 패키지명	
3.4. 파일명.....	
3.5. 메소드명	
3.6. 필드명.....	
4. 주석문.....	
4.1. 공통 주석문	
4.2. 변수 주석문	
5. 기타 스타일	
5.1. 들여쓰기	

5.2. 선언.....

5.2.1. 패키지 / 임포트.....

5.2.2. 클래스 / 인터페이스.....

5.3. 공백.....

6. 내부 구조.....

6.1. DAO.....

6.2. Controller.....

6.3. Service.....

1. 개요

1.1. 목적

소프트웨어를 개발하는 모든 과정에 들어가는 비용 중 80%가 유지보수에 쓰여진다. 소프트웨어의 직접 개발한 개발자가 그 소프트웨어의 유지보수를 담당하는 경우는 거의 볼 수 없는 경우다. 규칙을 정하고 코드를 작성하게 되면 가독성이 높아져 다른 개발자가 소스 코드를 처음 보더라도 더 빠르고 확실하게 이해할 수 있게 된다. 따라서 규칙을 정하고 지침은 개발자 간 반드시 지켜야 할 약속임을 항상 인지해야 한다.

2. 공통 법칙

2.1. 작업 디렉토리

프로젝트, 패키지, 모듈, 클래스 형태로 형성한다.

2.2 소스 파일

2.2.1. 각 자바 소스 파일은 하나의 Public Class 혹은 Public Interface 를 포함한다.

2.2.2. 하나의 Method 혹은 Class 에서 사용될 객체의 선언 및 생성은 상단에 위치시킨다.

3. 명명법

3.1. 공통 규칙

두 개 이상의 단어로 이루어져 있는 고유명사의 경우 첫번째 단어를 제외, 시작 문자를 대문자로 한다.

3.2 폴더명

3.2.1. webapp

3.2.2. webapp/views: 모든 view 처리 파일의 집합

3.2.3. webapp/views/common: view 파일 중 사용자, 관리자를 제외한 공통 파일의 집합

3.2.4. webapp/views/include: 공통적으로 include 하여 사용할 파일의 집합

3.2.5. webapp/views/admin: 관리자 페이지 파일의 집합

3.2.6. webapp/views/web: 사용자 페이지 파일의 집합

3.2.7. webapp/views/board: 게시판 파일의 집합

3.3 패키지명

dao, dto, controller 패키지의 경우 부모 테이블 당 하나의 파일을 생성한다.

service 패키지의 경우 위 부모 테이블을 기준으로 하위 폴더를 구성한다.

Ex. 게시판, 사용자 테이블이 존재할시 kr.or.kosa.service 패키지 하위로 명칭을 정의한다.

3.4. 파일명

3.4.1. Back 단의 Java 파일들은 첫번째 글자를 대문자, 이후 각 단어의 시작 문자 또한 대문자로 한다.

3.4.2. Front 단의 파일들은 첫번째 글자를 소문자, 이후 각 단어의 시작 문자를 대문자로 한다.

3.5. 함수명

함수의 이름은 주로 동사이며 두 개 이상의 단어로 구성될 경우 첫번째 단어를 소문자인 동사로 하고, 이후 각 단어의 시작 문자는 대문자로 한다.

Ex. viewAllList(), deleteUser() 등

3.6. 필드명

3.4.1. 변수, 모든 인스턴스 등의 첫번째 글자는 소문자로 하고, 이후 각 단어의 시작 문자는 대문자로 한다.

3.4.2. 클래스 변수의 이름은 타입의 이름과 동일하게 지정한다.

3.4.3. select, count 등은 입력 값, 제어문 혹은 반복문의 임시 변수명으로 사용할 수 있다.

4. 주석문

4.1. 공통 주석문

모든 주석문은 해당 코드 바로 윗줄에 위치하도록 한다.

브라우저에서 보내고 받는 데이터의 경우 반드시 주석문을 추가한다.

Ex. document.joinForm.submit(); <c:set> 등

4.2 변수 주석문

변수 주석문은 예외로 해당 코드 옆에 위치하도록 하고, 라인 단위 `//` 주석으로 한다.

5. 기타 스타일

5.1. 들여쓰기

5.1.1. 들여쓰기는 이클립스 프로그램 내 자동 정렬 기능(ctrl+shift+F)을 사용한다.

5.1.2. '{' 기호는 메소드 이름과 같은 줄에 위치하고, '}' 기호는 다른 줄에 위치하며 해당 줄에는 주석을 제외한 어떤 코드도 위치할 수 없다.

Ex.

```
public void setTime() {  
    System.out.println("");  
}
```

5.2. 선언

5.2.1. 패키지 / 임포트

필요에 따라 Package 문이 나타날 수 있다. Package 문 이후 한 줄을 띄우고 다음으로 import 문이 위치하도록 한다. Import 의 경우 한 줄에 하나의 import 만 명시한다.

5.2.2. 클래스 / 인터페이스

Import 문 이후 class 혹은 interface 선언문을 명시한다. class 혹은 interface 선언문의 구성은 다음과 같은 순서로 나타낸다.

5.2.2.1. class/interface 주석

5.2.2.2. class/interface 선언문

5.2.2.3. class/interface 에 속하는 멤버 필드 (private, protected, public 순으로 선언)

5.2.2.4. class 객체 생성자

5.2.2.5. class method 주석

5.2.2.6. class method 객체에서 사용될 메소드

5.3. 공백

조건문, 제어문의 키워드 (if, while, return, switch, for 등)와의 관계

5.3.1. 시작 시 '(', '{'의 괄호 앞에 한 칸의 빈칸을 포함하여 작성한다.

5.3.2. 종료 시 ')', '}'의 괄호 뒤에는 주석을 제외한 어떠한 코드도 작성하지 않는다.

6. 내부 구조

6.1. DAO

6.1.1. DAO 의 최상위 구조

아래를 기준으로 사용한다.

- DataSource ds = **null**;
- **private** Connection conn;
- **private** PreparedStatement pstmt;
- **private** ResultSet rs;
- **private** String sql;
- **private** int resultRow;
- **private boolean** idCheck; //아이디 체크여부
- **private** User Dto;
- **private** List<User> dtoList;
- **private** String pwdCheck;

6.1.2. try ~ catch ~ finally

catch문에 반드시 `System.out.println(e.getMessage());` 코드를 포함한다.

finally문에서는 try ~ catch문에서 사용된 DB 함수들을 반환한다.

6.2. Controller

6.2.1. 변수

doProcess 메소드의 최상위 변수는 아래 내용을 고정으로 사용한다.

```
String requestURI = request.getRequestURI();
String contextPath = request.getContextPath();
String url_Command = requestURI.substring(contextPath.length());
Action action=null;
ActionForward forward=null;
```

조건문의 urlcommand 구분은 웹 서비스의 구동 순서 및 관련된 서비스를 고려하여 작성한다.

Ex. 메인 페이지 호출 → 로그인/로그아웃 → 로그인 체크 → 세션 확인 → 회원 가입 → 회원 가입 처리

6.2.2. 조건문 Back 단 처리

Ex.

```
action = new BoardListService();
```

```
forward = action.execute(request, response);
```

6.2.3. 조건문 view 페이지 접근

Ex.

```
forward = new ActionForward();  
forward.setRedirect(false);  
forward.setPath("/WEB-INF/views/board/board_write.jsp");
```

6.3. Service

최상위에 ActionForward forward = null; 코드를 작성한다.

Forward 아래에 parameter 값을 정의하며, Back 단 처리는 try ~ catch 문을 사용한다.