

# 코드 작성 규칙

---



## 디자인, 개발시

---

- ▶ 디자인은 기획서의 화면UI에 나와있는 부분만 디자인한다.
  - 추가적으로 디자인을 해야한다면 반드시 기획서의 화면UI에도 추가하는 디자인을 넣어야한다.
  - 기획서의 화면UI와 조금이라도 다른 디자인을 한다면 반드시 기획서의 화면UI에도 원래 디자인에서 변경된 디자인으로 바뀌어야한다.
- ▶ 개발은 기획서의 전체시스템흐름, 상세시스템흐름 그리고 기능 설명만을 보고 개발한다.
  - 기획서와 조금이라도 다른 개발을 하게된다면 반드시 기획서에 변경된 개발이 반영되어야한다.

# 개발시

---

## ➤ 기획서

- 기획서의 기능 설명은 상세시스템 흐름도 또는 화면UI를 통해 파악할 수 있는 기능만을 설명한다.

예시)

- 피키캐스트의 Head Search Text(변수화파일의 첫번째 슬라이드 참고)를 클릭해보면 검색어를 입력할 수도있고 추천검색어가 나오기도한다. 그러나 프로젝트 기획서.hwp파일의 1.2.2 메인UI의 검색 기능 설명에는 추천검색어 기능에 대한 설명이 빠졌다. 그러므로 추천검색어 기능을 추가하고 싶다면 반드시 프로젝트 기획서.hwp파일에 추천검색어 기능에 대한 설명을 하고 기능을 추가하여야한다.

기능 설명에 추천검색어 기능을 설명할때는 “ Head Search Text를 클릭시 Head Search Text 밑에 추천검색어 목록이 나타나며 추천검색어는 임의로 선정한다. “

이정도로 화면을 보고 파악할 수 있는 정도의 기능만을 설명한다.

사실 내부적으로는 DB에 접속해서 추천검색어 목록을가져와서 컨트롤러를 통해서 뷰에 출력하는 흐름이겠지만 이것을 UI만보고 파악할 순 없다.

# 자바언어 코드작성 규칙

---

## ➤ 메소드

1. 오버라이드 시 반드시 `@override` 애너테이션을 단다.



# 자바언어 코드작성 규칙

- ▶ 블록을 포함하고 있는 모든 문장
  - 클래스, 조건문, 제어문, 메서드, 블록문 등... 블록 ( { ... } )을 포함하고 있는 모든 문장은 반드시 블록의 닫는괄호 ( } )뒤에 왼쪽의 그림, 아래의 예시와 같은 규칙을 따른다.
- ▶ 예시 )
  1. public class Calculator{  
...  
} // end class Calculator
  2. if(!name.equal(lcs)){  
...  
} // end if(!name.equal(lcs))
  3. for(int i =0; i <10; i++){  
...  
} // end for(int i =0; i <10; i++)
  4. public void factorial(int num){  
...  
} //end factorial(int num)

```
42  
43     while (true) {  
44         System.out.println("명령어를 입력하세요 : ");  
45         String command = reader.readLine();  
46  
47         if (command.equalsIgnoreCase("exit")) {  
48             System.out.println("종료합니다.");  
49             break;  
50         } // end if(command.equalsIgnoreCase("exit"))  
51  
52         if (command.startsWith("new")) {  
53             processNewCommand(command.split(""));  
54             continue;  
55         } else if (command.startsWith("change")) {  
56             processChangeCommand(command.split(""));  
57             continue;  
58         } // end if (command.startsWith("new"))  
59  
60         printHelp();  
61     } // end while(true)  
62  
63 } // end main
```

```
1 package controller;  
2  
3 import javax.servlet.http.HttpSession;  
4  
14 @Controller  
15 @RequestMapping("/edit/changePassword")  
16 public class ChangePwdController {  
17  
18     private ChangePasswordService changePasswordService;  
19  
20     public void setChangePasswordService(  
21  
22     @RequestMapping(method = RequestMethod.GET)  
23     public String form(  
24         @ModelAttribute("command") ChangePwdCommand pwdCmd) {  
25         return "edit/changePwdForm";  
26     } //end public String form(@ModelAttribute("command") ChangePwdCommand pwdCmd)  
27  
28     @RequestMapping(method = RequestMethod.POST)  
29     public String submit(  
30         @ModelAttribute("command") ChangePwdCommand pwdCmd,  
31         Errors errors,  
32         HttpSession session) {  
33         new ChangePwdCommandValidator().validate(pwdCmd, errors);  
34         if (errors.hasErrors()) {  
35             return "edit/changePwdForm";  
36         }  
37         AuthInfo authInfo = (AuthInfo) session.getAttribute("authInfo");  
38         try {  
39             changePasswordService.changePassword(  
40                 authInfo.getEmail(),  
41                 pwdCmd.getCurrentPassword(),  
42                 pwdCmd.getNewPassword(),  
43                 pwdCmd.getConfirmPassword());  
44         } catch (Exception e) {  
45             errors.rejectValue("password", "password.error");  
46         }  
47         return "edit/changePwdForm";  
48     }  
49 }  
50
```

# 자바언어 코드작성 규칙

---

## ▶ 인터페이스

- 인터페이스명은 모두 소문자로 작성한다.
- 인터페이스명은 xxxable을 사용한다.
  1. `public interface selectable { ... }`
- 인터페이스는 interface 패키지 내에 위치시킨다.

# GIT, GITHUB 사용 규칙

---

## ➤ 이슈

- 디자인, 개발 모두 어떠한 일을 할 경우 반드시 이슈를 생성한다.
- 이슈번호는 브랜치이름이된다.

## ➤ 브랜치 생성

- 디자인은 Design/# 으로 브랜치를 생성한다.
- 개발은 Feature/# 으로 브랜치를 생성한다.
- #은 이슈의 번호를 따른다.

## ➤ 커밋

- 하나의 커밋은 하나의 클래스에 대한 내용만 담고 있다.

예시)

1. 회원가입처리 클래스, 회원탈퇴처리 클래스의 내용을 수정하면 두 개의 커밋이 생성되어야 한다.
2. 회원가입처리 클래스에 있는 두개 이상의 필드 또는 메소드 수정시 하나의 커밋이 생성되어야 한다.