

# 개발 규칙

## 1. 포매팅

### 1.1. 파일 인코딩 UTF-8 사용

### 1.2. 들여쓰기

- tab 문자 대신 space 문자 4개 사용

### 1.3. 시작하는 중괄호는 새로운 라인에서 시작하지 않고 같은 라인을 사용

```
if (value == true) {  
    System.out.println("true");  
}
```

### 1.4. 띄어쓰기

- 메소드 이름 다음에는 띄어쓰기 X

```
foo(i, j);
```

- 배열 다음에는 띄어쓰기 X

```
args[0];
```

- 이진 연산자 간에 양쪽 띄어쓰기 사용

```
a = b + c;
```

- 쉼표와 세미콜론 뒤에는 띄어쓰기 사용

```
for (int i = 0; i < 10; i++)
```

- if, while, for, switch, catch 문 뒤에는 띄어쓰기 사용

```
if (value)
```

```
while (i < 10)
```

```
for (int i = 0; i < 10; i++)
```

```
catch (Exception e)
```

## 2. 네이밍

모든 식별자는 영문 대소문자와 숫자만 사용

### 2.1. 클래스, 인터페이스명

각 단어의 첫 글자는 대문자, 나머지는 소문자 사용

```
UserService  
UserController
```

### 2.2. 패키지명

소문자만 사용

```
user  
post  
challenge
```

### 2.3. 그 외 식별자

필드 식별자, 변수, 메소드, 파라미터는 카멜케이스 사용

```
user
addUser()
deleteUser()
```

## 2.4. Controller

컨트롤러 클래스에서 메서드명을 작성할 때는 접미사를 사용

```
postList()
postAdd()
postRemove()
```

## 2.5. Service

서비스 클래스에서 메서드명을 작성할 때는 접두사를 사용

```
addPost()
removePost()
```

# 3. Structure

## 3.1. 패키지는 목적별로 묶어서 생성

- user(유저관련), post(게시글 관련), challenge(챌린지 관련)

## 3.2. Controller에서는 Service 호출과 Exception 처리만 담당

- Controller에서의 비즈니스 로직 구현은 최대한 피함

## 3.3. 메소드와 클래스는 하나의 목적만을 위해 생성

- 한개의 메소드는 한가지의 기능만 가짐

- 한개의 클래스 내부에는 같은 목적만을 가진 코드가 존재