

코드 컨벤션

명명 규칙

- 모든 이름은 반드시 용도를 추론할 수 있도록 의미를 담아 작성한다. 반복문에서 반복자 변수 이름을 i, j, k로 정하는 경우나 잠깐 쓸 숫자 변수 이름으로 n을 쓰는 등의 관습적인 명명은 허용한다.

- 변수 이름은 camelCase로, 명사로 선언한다.

```
String email = "abc@example.com";
```

- ID처럼 일반적으로 대문자로 쓰는 약어도 camelCase로 작성한다.

```
String id, String userId
```

- 메소드 이름은 camelCase로, 동사로 선언한다.

```
getEmail()
```

```
extractDomain(String email)
```

```
isValid(String input);
```

- 패키지 이름은 반드시 소문자로 작성한다.

```
com.shop.petpal.domain
```

- 클래스, 인터페이스 이름은 PascalCase로, 명사로 작성한다.

```
public class UserInfo {}
```

```
public class Pager {}
```

- 컬렉션 이름은 안에 들어간 데이터가 뭔지 추론할 수 있도록 의미를 담아서 작성한다.

복수형으로 작성하거나 컬렉션 이름을 포함하도록 작성한다.

```
List<String> emails;
```

```
List<User> userList;
```

- 상수의 이름은 대문자와 밑줄(언더스코어/언더바)를 이용해 작성한다.

```
public static final ADMIN_ID = 100;
```

```
public enum Membership { BASIC, VIP, VVIP }
```

- 데이터를 꺼내오는 메소드의 이름의 경우, Mapper 클래스에서는 `findXX()`, Service 클래스에서는 `getXX()`로 작성한다.

참고) 시간복잡도가 낮은 경우는 get, 시간복잡도가 상대적으로 높은 경우는 find를 사용하는 것이 일반적인 네이밍 컨벤션이다. JPA의 getByld, findById처럼 null을 반환할 수 없으면 get, null을 반환할 수 있으면 find를 사용하는 컨벤션도 있다.

이외 규칙

- 조건문에는 불가피한 경우를 제외하고는 부정 조건을 넣지 않는다. 불가피한 경우란 부정 조건을 사용하지 않을 경우 역으로 코드의 가독성이 떨어지는 경우를 의미한다. -> 느낌표 쓴 조건이라도 바로 이해될 정도로 간단하면 그냥 써도 된다는 뜻

- 들여쓰기 철저히 지키기

중괄호나 태그를 열고 줄바꿈한 경우에 들여쓰기

닫는 괄호 및 태그는 반드시 여는 괄호 및 태그와 위치를 맞추어 같은 열에 작성

그냥 **Ctrl+Shift+F** 누르면 자동 포매팅 되긴 하는데, HTML 태그는 줄바꿈을 좀 이상하게 해주는 듯

- 이항 연산자는 좌우 띄어쓰기

`a = b + c;` (O)

`a=b+c` (X)

- 중괄호 열기 전에 띄어쓰기
- if, while, for, switch, catch 문 뒤에는 띄어쓰기
- 메소드 이름, 배열 이름 다음에 오는 괄호는 붙여 쓰기

`void fun() {}` (O)

`void fun ()` (X)

`void fun(){} (X)`

- 쉼표나 세미콜론 뒤에도 띄어쓰기

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

`int[] arr = {1, 2, 3, 4};`

- 제어문이 한 줄이라도 중괄호 꼭 쓰기
- 변수의 범위는 가능한 좁게 선언한다. 특정한 메소드 안에서만 쓰일 변수는 클래스의 필드가 아니라 그 메소드의 지역변수로 선언하고, 특정한 제어문 안에서만 쓰일 변수는 그 제어문 블록 내부에서 선언해서 사용한다.
- 긴 메소드**의 경우, 변수를 실제로 사용되는 위치와 가까운 곳에 선언한다. 메소드 맨 밑에서 쓰는 변수를 메소드 맨 위에 선언하지 않는다. 변수가 최초로 사용되는 지점에 가까운 곳에 선언하라는 의미이다.
- 변하지 않을 값을 선언할 때는 `final` 키워드를 적극 활용한다. 실수로 재할당 했을 때 컴파일 에러나 IDE 경고를 볼 수 있도록 하기 위함

`@Autowired private final MemberService memberService`

- SQL 쿼리의 경우 키워드는 대문자, 테이블명이나 컬럼명 등은 소문자로 작성한다. 가독성이 좋게끔 줄바꿈을 많이 한다.

- 주석은 아래의 접두어를 활용해서 작성한다.
 - **TODO** : 아직 작성하지 않았는데 추후에 구현해야 하는 로직이 있는 경우
`// TODO: 로그인 로직 구현`
 - **FIXME** : 코드에 문제가 있어서 수정해야 할 내용이 있는 경우
 - **NOTE** : 기타 알아두어야 할 사항이 있는 경우
`// NOTE: 넘어올 파라미터 - 글번호, 페이지번호, ...`
`// NOTE: 세션에 저장할 데이터 - 회원번호, 아이디, 닉네임, 회원등급, 회원권한, ...`
 - 그 외 로직이 어려워서 이해를 돕기 위한 설명 목적의 주석은 접두어 없이 그냥 작성
- 단번에 이해하기 어려운 코드는 반드시 주석을 작성한다.
- 한 줄에 너무 긴 코드를 쓰지 않는다.

STS 설정 - 'print margin' 검색 - Show print margin 체크하고 컬럼 값 100으로 지정

에디터에 보이는 회색 줄 안 넘어가게끔 가독성 좋게 잘 끊어서 줄바꿈하기