

Coding Standard : Naming Convention

👤 생성자	세윤 세윤 김
🕒 생성 일시	@2024년 11월 19일 오전 11:55
⋮ 태그	

Goal : 코드의 “가독성”을 높이기 위함.

참고

(강의 자료를 참조한) 명명 규칙

1. 여러 케이스를 비교하고 그 중 카멜 케이스를 선택
 - a. 스네이크 케이스 : 언더바가 들어있는 표현 방식 → `var snake_case;`
 - b. 파스칼 케이스 : 첫 글자와 중간 글자들이 대문자인 경우 → `var PascalCase;`
 - c. 카멜 케이스 : 중간 글자는 대문자, 첫 글자가 소문자 즉, 낙타 모양과 비슷한 경우 → `var camelCase;`
 - d. 헝가리안 표기법 : 접두어에 자료형을 붙이는 경우 → `strUserName, iUserAge` 등
 - e. 케밥 표기법 : 다음단어와 '-'(대시)로 구분하는 경우 → `user-name, user-age` 등
2. 카멜 케이스(Camel Case)
 - Java 클래스, 필드, 메소드 및 변수 이름을 저장할 때 쓰는 규칙
 - 여러 단어를 띄어쓰기 없이 붙여 쓰는데, 각 단어의 첫 글자를 대문자로 작성
 - upper camel case : `ThisIsExample`
 - 클래스, 구조체, 열거형, 익스텐션, 프로토콜에 적용
 - lower camel case : `thisIsExample`
 - 변수, 상수, 함수, 속성, 메소드, 파라미터에 적용
3. 메소드 이름
 - 일반적으로 소문자 시작(동사구)
 - 함수 이름 일반적으로 값을 설명(명사구)
 - 필드(eg., title)의 값을 접근하여 리턴하는 함수 : `get`
 - 조건 묻는 boolean 함수 이름은 “is”로 시작
4. 변수 이름
 - 일반적으로 소문자 시작
 - 용도에 대한 힌트로 제공
 - 모호한 이름 사용하지 않기
 - 변수의 이름의 길이 (최소 1자, 최소 9자)
 - 대상이 사용된 위치 고려
 - 매개 변수 : 되도록 짧게

- 필드 변수 : 가능한 길고 의미 부여 가능해야 함

5. 패키지 이름

- 일반적으로 모두 소문자(명사 형태)

6. 형식

- 들여쓰기와 괄호
 - 프로그램 구조를 명확하게 하기 위해 사용
 - 문장의 일부와 선언문은 들여쓰기

7. 블록은 항상 괄호를 사용

- ```
if (flag) {
 validate();
 update();
} else {
```

- 키워드와 다음 괄호 사이에 공백을 두어 제어 구조와 메서드 호출과 구별하게
- `// Good if (username == null) { ... }`

## 8. 문장과 수식

- 문장이나 수식은 메소드나 클래스로 패키지화
- 블록 문장
  - 제어구조 중첩되었을 때 발생하는 혼란 방지
- 수식
  - 괄호를 이용하여 오퍼레이션의 순서를 명확히

## 9. 오류 & 예외 처리

- 모든
- 잘못된 데이터 어떻게 다룰 것인가
  - ex) 실제로는 없는 계좌번호
- 매개변수 오류
  - ex) `evaluate()` 라는 메소드가 매개변수로 `car, truck, bus`만을 받아들임  
`evaluate(String vehicleP)`  
`evaluate(SpecializedVehicle vehicleP)`
- 입력 오류 방지
  - 라스트 박스
  - 디폴트 값 지정

## 10. 주석

- 디버깅하는 동안 얻을 수 있는 도움
- 다른 사람들이 쉽게 프로그램 이해 가능
- 주석 다는 방법

- 불필요한 단어 생략
- 과도한 주석 금지
- 클래스 불변조건(invariant)
  - 클래스의 속성에 대한 의미와 제약 기술

#### 11. 메서드 주석

- 메서드가 하는 일을 설명하는 javadoc 스펙을 선행 조건과 함께 표기
- 관련 모든 매개 변수를 언급
  - `/** Print the sum of a and b. */`

---

## 일반적인 명명 규칙을 토대로 한 UosSense 명명 규칙

- 상수는 모두 대문자로
- 클래스와 인터페이스 이름
  - 대문자로 시작(명사 또는 명사구)
  - Upper Camel case
- 변수 이름
  - 최소 1자, 최대 20자 길이의 제한을 둘 것
- 메소드 이름
  - 최소 1자, 최대 20자 길이의 제한을 둘 것
  - lower camel case
  - 뒤에 부사구 붙이는 거 통일?
    - e.g. 위경도 → 주소 변환하는 함수
    - e.g. 주소변환with좌표 혹은 주소변환By좌표? From etc..
    - 조건 묻는 boolean 함수 이름은 "is"로 시작(PPT) 필수!!인지 아니면 유연한지
  - 검사 메소드 명명 시 check~ or validate~?
- 패키지 이름
  - 소문자?
- Json 내 파라미터 이름
  - snake case
- 블록은 일반적인 명명 규칙을 따름.
- 문장과 수식
  - 문장이나 수식은 메소드나 클래스로 패키지화
  - 블록 문장
    - 제어구조 중첩되었을 때 발생하는 혼란 방지
  - 수식
    - 괄호를 이용하여 오퍼레이션의 순서를 명확히

## 9. 오류 처리

- 오류 발생 시 주석 처리

## 10. 주석

- `/* */` 사용

## 11. 메서드 주석

- `/*`
- `* @param username`
- `* @return UserEntity`
- `*/`
- 이런 식으로 매개변수랑 리턴 값 표시

Reference :

[SE-09-coding\\_v1.1.pdf](#)

[]

<https://velog.io/@jp-share/Coding-%ED%91%9C%EA%B8%B0%EB%B2%95-%EC%B9%B4%EB%A9%9C%EC%BC%80%EC%9D%B4%EC%8A%A4-%ED%8C%8C%EC%8A%A4%EC%B9%BC%EC%BC%80%EC%9D%B4%EC%8A%A4-%EC%8A%A4%EB%84%A4%EC%9D%B4%ED%81%AC%EC%BC%80%EC%9D%B4%EC%8A%A4>