



Clean Code [5장 형식 맞추기]

~ 적절한 행 길이를
유지하라

Main Reader : 위승빈



형식 맞추기

1. 프로그래머라면
2. 형식을 맞추는 목적
3. 적절한 행 길이를 유지하라

프로그래머라면

프로그래머라면 형식을 깔끔하게 맞춰 코드를 짜야 한다.

코드 형식을 맞추기 위한 규칙을 정하고 따르자.

팀으로 일한다면 팀의 규칙을 정하고 모두 그 규칙을 따르도록 한다.

형식을 맞추는 목적

코드 형식은 중요하다.

코드 형식은 의사소통의 일환이다.

의사소통은 전문 개발자의 일차적인 의무이다.

코드의 가독성은 앞으로 바뀔 코드의 품질에 많은 영향을 미친다.

구현 스타일과 가독성 수준은 유지보수 용이성과 확장성에 계속 영향을 미친다.

원래 코드는 사라지더라도 개발자의 스타일과 규율은 사라지지 않는다.

적절한 행 길이를 유지하라

적은 코드의 파일들로도 커다란 시스템을 구축할 수 있다.

일반적으로 큰 파일보다 작은 파일이 이해하기 쉽다.

반드시 지킬 규칙은 아니지만 바람직한 규칙으로 삼아보자.

적절한 행 길이를 유지하라

신문 기사처럼 작성하라

- 이름은 간단하면서도 설명이 가능하게
- 소스 파일의 첫 부분은 고차원 개념과 알고리즘을 설명
- 아래로 내려갈수록 의도를 세세하게 묘사
- 마지막에는 가장 저차원 함수와 세부 내역

개념은 빈 행으로 분리하라

- 일련의 행 묶음은 완결된 생각 하나를 표현
- 생각 사이에는 빈 행을 넣어 분리
- 빈 행은 새로운 개념을 시작한다는 시각적 단서
- 보통 빈 행 바로 다음 줄에 눈길이 멈춤

적절한 행 길이를 유지하라

세로 순서

- 일반적으로 함수 호출 종성석은 아래 방향으로 유지 (호출되는 함수를 호출하는 함수보다 나중에 배치)
 - 소스 코드 모듈이 고차원에서 저차원으로 자연스럽게 내려감
- 가장 중요한 개념을 먼저 표현하고, 세세한 사항은 가장 마지막에 표현
- 소스 파일에서 몇 개만 읽어도 개념을 파악하기 쉬워지도록

수직 거리

- 서로 밀접한 개념은 세로로 가까이 두자
- 타당한 근거가 없다면 서로 밀접한 개념은 한 파일에 속하도록
 - protected 변수를 피하도록 : 개념을 찾기 위해 다른 파일을 봐야 하므로
- 밀접한 두 개념은 세로 거리로 연관성을 표현
 - 연관성 : 한 개념을 이해하는 데 다른 개념이 중요한 정도

적절한 행 길이를 유지하라 - 수직 거리

변수 선언

- 변수는 사용하는 위치에 최대한 가까이 선언
- 루프를 제어하는 변수는 루프 문 내부에 선언

인스턴스 변수

- 인스턴스 변수는 클래스 맨 처음에 선언
- 변수 간에 세로로 거리를 두지 않도록
 - 클래스 메서드가 인스턴스 변수를 사용하기 때문

적절한 행 길이를 유지하라 - 수직 거리

종속 함수

- 한 함수가 다른 함수를 호출한다면 두 함수는 세로로 가까이 배치
- 호출하는 함수보다 먼저 배치

개념적 유사성

- 개념적 친화도가 높을수록 코드를 가까이 배치
- 한 함수가 다른 함수를 호출하는 직접적인 종속성은 친화도가 높은 예
- 비슷한 동작을 수행하는 함수도 친화도가 높은 예
- 명명법이 똑같고 기본기능이 유사해도 친화도가 높은 예

형식을 깔끔하게