

# Clean Code

<https://product.kyobobook.co.kr/detail/S000001032980>

함부로 바라지 마라

- 프로퍼셔널리즘 용어에는 명예와 금지, 책임, 의미를 포함

책임감을 가져라

무엇보다도 해를 끼치지 마라

- 기능에 해를 끼치지 마라
  - QA는 아무것도 찾지 못해야 한다
  - 제대로 작동하는지 아닌지 알아야 한다
- 구조에 해를 끼치지 마라
  - 코드를 바꾸는 일 = 무자비한 리팩토링 == 보이 스카웃 규칙

직업 윤리

전산 분야 지식을 익혀라

- 디자인 패턴 : 24가지 GOF 패턴, POSA 패턴
- 설계 원칙 : SOLID 객체지향 원칙, 컴포넌트 개념
- 방법론 : XP, 스크럼, 린, 칸반, 폭포수, 구조적 분석, 구조적 설계 개념
- 도구 : UML, DFD, 흐름도, 구조차트, 페트리 넷, 상태전이 다이어그램과 테이블, 결정 테이블
- 원칙 : 테스트 주도 개발, 객체지향 설계, 구조적 프로그래밍, 지속적 통합, 짝 프로그래밍

끊임없이 배우기

## 연습

- 품새 (대개 프로그래밍으로 간단한 문제를 푸는 형식) 한두개씩 풀기
- <https://katas.softwarecraftsmanship.org>
- <http://codekata.pragprog.com>

## 회사와 고객에 동질감 갖기

## 업무 지식 익히기

## 준비된 자세

- 코드는 반드시 동작해야한다
- 코드는 고객이 제시한 문제를 반드시 풀어야한다
- 코드는 기존 시스템에 잘 녹아들어야한다
- 코드는 다른 프로그래머가 읽기 쉬어야 한다

## 속도 조절

- 언제 걸어 나가야 할지 알기
- 집까지 운전하기 (회사 문제에서 해방되기)
- 샤워

## TDD의 세가지 법칙

- 실패한 단위 테스트를 만들기 전에는 제품 코드를 만들지 않는다
- 컴파일이 안되거나 실패한 단위 테스트가 있으면 더 이상의 단위 테스트를 만들지 않는다
- 실패한 단위 테스트를 통과하는 이상의 제품 코드는 만들지 않는다

## 코딩 도장

합 맞추기

대련

경험의 폭 넓히기

테스트 자동화 피라미드

- 탐색 테스트 < 시스템 테스트 < 통합 테스트 < 컴포넌트 테스트 < 단위 테스트

반복 계획 회의

- 백로그에서 다음 반복 주기동안 처리할 항목을 고르는 일

반복 회고와 시연

PERT

- O(낙관적 추정 값), N(명목 추정 값), P(비판적 추정 값)

광대역 델파이