

cleanCode3 summary

20150560 Dongwon Kim

Unit Test(1)

- 3 laws of TDD(Test Driven Development)
 - 1. You may not write production code until you have written a failing unit test.
 - 2. You may not write more of a unit test than is sufficient to fail, and not compiling is failing.
 - 3. You may not write more production code than is sufficient to pass the currently failing test.
 - -> 즉 코드와 테스트 코드 작성은 동시에 이루어져야 한다.
 - -> 테스트 코드를 많이 작성하게 되고, 테스트 코드 관리 문제가 발생함.

Unit test(2)

- 지저분한 테스트는 없느니만 못하다.
- 테스트 코드는 필수적이다. Clean unit test가 필요함.
- Test가 코드의 유연성/유지가능성/재사용성을 증대시킨다.
- 적절한 test suite가 있어야만이 다음에 code를 변경할 수 있음.

Unit test(3)

- Test code is a also code – Readability is a king
- Build-Operate-Check pattern
 - test data를 build하고, 이를 가지고 operation을 수행하고, 결과값을 비교하는 것을 각각 method extraction 한다.
- Clear 한 domain specific test lang. 즉, 좋은 test API를 유지하기 위해서 refactor 하라.
- Test code에 대해서는 다른 기준이 필요하다.
 - Production code 만큼 efficient할 필요 없다.
 - Mental mapping 등의 문제에 대해서도 보다 flexible하게.
- Test 하나는 하나의 concept 만을 검사하도록 한다.

Unit test(3)

- Five rules for clean tests – FIRST
 - Fast : test는 자주 실행되어야 하므로 빨라야 한다.
 - Independent: test 간의 의존관계가 있어서는 안된다. – test 하나의 fail이 연쇄적인 결과를 야기할 수 있음.
 - Repeatable : 여러 환경에서 다시 반복해서 결과를 내놓을 수 있어야 한다.
 - Self-validating : test 는 Boolean output 만을 내놓아야 한다. Pass하거나 fail하거나. Log를 읽어가며 분석해야 하거나 등의 쓸모없는 동작은 있어서는 안된다.
 - Timely : test code는 production code 작성 전에 쓰여져야 한다.