

cleanCode 4 – CH.10 summary

20150560 Dongwon Kim

Class(1)

- Standard convention – class는 variable list로 시작하고, public이 그중 에서도 먼저 온다.
- Class should be small! – responsibility 가 작아야 한다. (class가 얼마나 많은 일들을 수행해야 하는지를 의미.)
- Single Responsibility Principle – class should have ONLY ONE reason to change.
- SRP 를 잘 실천한다면? 프로그램은 많은 수의 작은 class 로 구성되게 될 것. – 코드가 작동한다고 멈추지 말 것!

Class(2)

- Class 는 cohesive하게 만들자.
 - Each variable이 each method 에서 사용되도록.
 - 즉, 반드시 필요한 variable 만을 가지는 단위로 class 를 구성하자는 것.
- Large function 을 나누자!
 - 근데 이러면 less cohesive 하는 방향으로 변하지 않나요?
 - 그 말인즉슨, class 를 더 작은 class로 나누어야 한다는 것!
 - 다시 한번, Smaller class is always better!
- PrimeGenerator 예제
 - 각 responsibility 에 따라서 나눔.
 - Class 는 반드시 object 로만 사용되어야 하는 것이 아니다. Static 을 이 용해서 사용할 수 있다. – useful 한 scope를 제공 가능.

Class(3)

- SQL examples
 - Sql string 을 생성하는 class.
 - Class 가 다양한 functionality 를 포함하기 때문에 수정되어야 할 경우가 매우 많다! -> SRP principle!
 - 각 functionality 를 base class를 상속 받는 class 들로 나눈다.
 - 이 경우 functionality 추가 시에 기존 코드를 수정할 필요가 없다. -> 새로운 class 를 생성하면 됨. (Isolating from Change, reduce potential risk)

Class(4)

- Dependency Invariant Programming(DIP)
 - Class 의존성은 오로지 abstraction에 한해서만 구성되어야 한다.
 - 주식 예제에서 볼 수 있듯이, 가격을 '어떻게' 얻어 오는지는 class scope 에서는 전혀 중요하지 않은 정보.
 - 따라서 이는 다른 class로 abstraction 하는 것이 적합하다.
 - 이런 경우 abstract class/interface 를 사용하여 이들을 상속하는 class 가 '어떻게' 를 representation 하도록 한다.
 - 이 경우에는 package 간 의존성도 약해지므로, test code를 만들기도 적합하다.