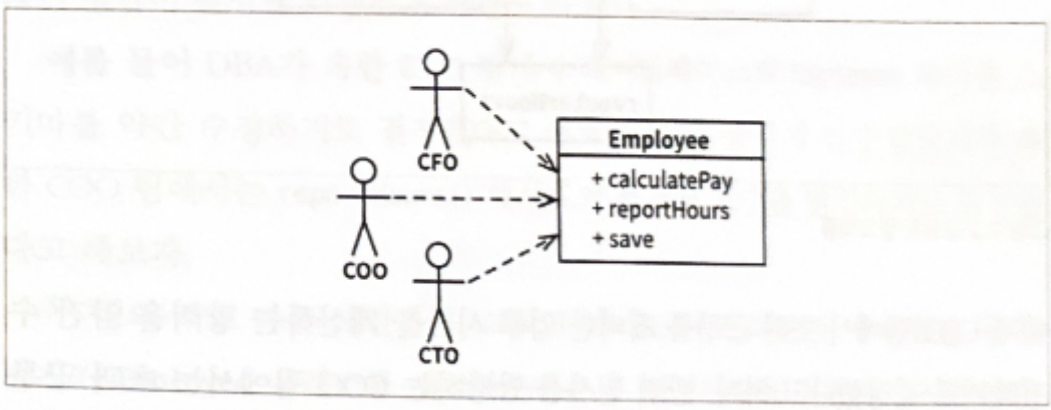
07장. SRP: 단일 책임 원칙

- 단일 모듈은 변경의 이유가 오직 하나뿐이어야 한다.
  - 다시 말하면 “하나의 모듈은 오직 하나의, 오직 하나의 사용자 또는 이해관계자에 대해서만 책임져야 한다.”
  - 하나의 모듈은 하나의, 오직 하나의 액터에 대해서만 책임져야 한다. (액터는 집단 또는 조직 또는 개인)
- 여기서 모듈은 소스파일이다. 응집된 이라는 단어가 SRP를 암시한다. 단일 액터를 책임지는 코드를 함께 묶어주는 힘이 바로 응집성이다.

징후 1: 우발적 중복

- 급여 애플리케이션의 Employee 클래스



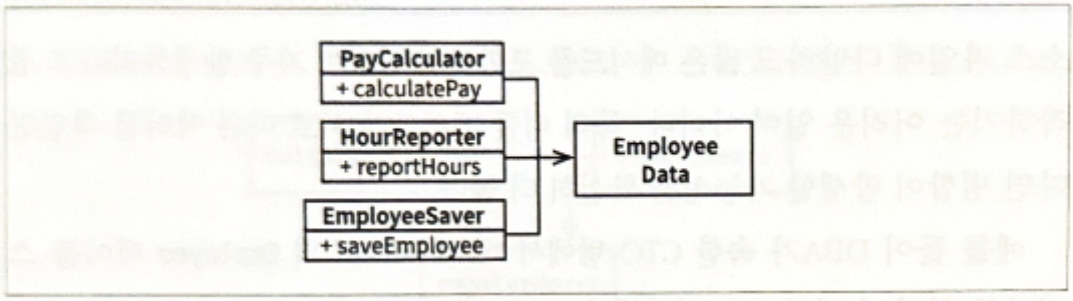
- SRP를 위반하는 이유는 세가지 메서드가 매우 다른 세명의 액터를 책임지기 때문이다.
  - calculatorPay 메서드는 CFO 보고를 위해 사용한다.
  - reportHours 메서드는 COO 보고를 위해 사용한다.
  - save 메서드는 CTO 보고를 위해 사용한다.
- CFO 팀에서 결정한 조치가 COO팀이 의존하는 어떤 것에 영향을 줄 수 있다.

징후 2: 병합

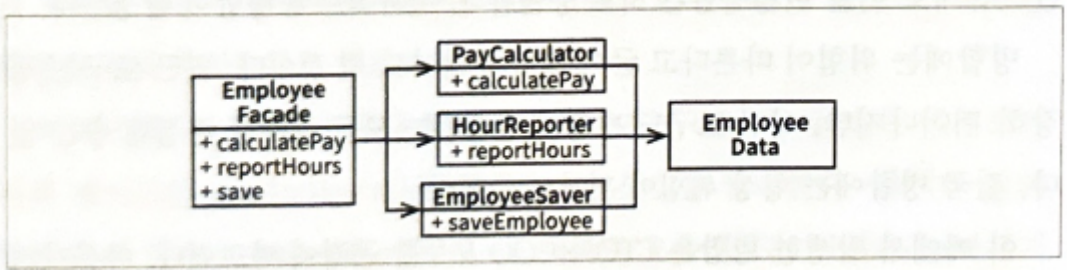
- 여러 액터가 사용되고있는 클래스에 같이 사용중인 알고리즘을 수정한다면 병합이 일어날 수 밖에 없다.
- 병합이 나쁜 것은 아니지만 위험이 뒤따르게 된다.

해결책

- 그 모두가 메서드를 각기 다른 클래스로 이동시키는 것
- 다음과 같이 모든 메소드들이 제각기 구현되어 EmployeeData 클래스에서만 사용되도록 한다.



- 하지만 개발자가 세 가지 클래스를 인스턴스화하고 추적해야하는 것이 단점이다. 이를 해결하기 위해 퍼사드(Facade)패턴이 있다.



결론

단일 책임 원칙은 메서드와 클래스 수준의 원칙이다. 하지만 이보다 상위의 두 수준에서도 다른 형태로 다시 등장한다.