* 10장 클래스
  + 클래스체계
    - 클래스를 정의하는 표준 자바 관례에 따르면 가장 먼저 변수 목록이 나옴
    - 정적 공개 상수가 있다면 맨 처음에 나옴
    - 다음으로 정적 비공개 변수가 나오며 비공개 인스턴스 변수가 나옴
    - 공개 변수가 필요한 경우는 거의 없다.
    - 변수다음엔 함수가 나오는데 비공개 함수는 자신을 호출하는 공개 함수 직후에 넣는다
  + 캡슐화
    - 변수와 유틸리티 함수는 가능한 공개하지 않는 편이 낫지만 반드시 숨겨야 한다는 법칙은 없다
    - 떄로는 변수나 유틸리티 함수를 protected로 선언해 테스트 코드에 접근을 허용하기도 한다.
  + 클래스는 작아야 한다.
    - 클래스를 만들 때 첫번쨰 규칙은 크기이다. 클래스는 작아야 한다.
    - 두번째 규칙도 크기이다. 더 작아야 한다.
    - 앞서 함수 장에서 했던 이야기를 되풀이할 의도는 없다 단지 클래스를 설계할 때도 함수와 마찬가지로 작게 가 기본 규칙이라는 의미이다.
    - 단일 책임 원칙
      * 클래스나 모듈을 변경할 이유가 하나. 단 하나 뿐이어야 한다는 원칙
      * SRP는 책임 이라는 개념을 정의하며 적절한 클래스 크기를 제시한다.
      * 클래스는 책임, 즉 변경할 이유가 하나여야 하는 의미이다.
      * 책임, 즉 변경할 이유를 파악하려 애쓰다보면 코드를 추상화하기도 쉬워진다.
      * 더 좋은 추상화가 더 쉽게 떠오른다.
      * SRP는 객체지향 설계에서 더욱 중요한 개념이다. 또한 이해하고 지키지 수월한 개념이기도 하다. 하지만 이상하게 SRP는 클래스 설계자가 가장 무시하는 규칙 중 하나라는 것
      * 큰 클래스 몇 개가 아니라 작은 클래스 여럿으로 이뤄진 시스템이 더 바람직하다.
    - 응집도
      * 클래스는 인스턴스 변수 수가 작아야 한다. 각 클래스 메서드는 클래스 인스턴스 변수를 하나 이상 사용해야 한다. 일반적으로 메서드가 변수를 더 많이 사용할수록 메서드와 클래스의 응집도가 더 높아진다.
      * 응집도가 높은 클래스는 가능하지도 바람지하지도 않지만 대부분은 응집도가 높은 클래스를 선호한다.
      * 함수를 작게 매개변수 목록을 짧게 라는 전략을 따르다 보면 떄떄로 몇몇 메서드만이 사용하는 인스턴스 변수가 아주 많아진다. 이는 십중팔구 새로운 클래스로 쪼개야 한다는 신호이다.
    - 응집도를 유지 하면 작은 클래스 여럿이 나온다.
      * 큰함수를 작은함수 여럿으로 나누기만 해도 클래스 수가 많아지는데 새로운 변수를 클래스 인스턴스 변수로 승격한다면 새 함수는 인수가 필요없이 함수 쪼개기 쉬워진다.
  + 변경하기 쉬운 클래스
    - 대다수 시스템은 지속적인 변경이 가해진다. 그리고 변경할떄마다 시스템이 의도대로 동작하지 않을 위험이 따른다. 깨끗한 시스템을 클래스를 체계적으로 정리해 변경에 수반하는 위험을 낮춘다.
    - 저자의 경험에 의하면 클래스 일부에서만 사용되는 비공개 메서드는 코드를 개선할 잠재적인 여지를 시사한다.
    - sql클래스를 논리적으로 완셩으로 여긴다면 책임을 분리하려 시도할 필요가 없다. 가까운 장래 update문이 필요하지 않다면 sql클래스를 내버려두는 편이 좋다.