

# Chapter7. 오류 처리

- 오류 코드보다 예외를 사용하라
  - 오류가 발생하면 예외를 던지는 편이 낫다. 그러면 호출자 코드가 더 깔끔해진다.
- Try-Catch-Finally 문부터 작성하라
  - try 블록에서 무슨 일이 생기든지 호출자가 기대하는 상태를 정의하기 쉬워진다.
  - 먼저 강제로 예외를 일으키는 테스트 케이스를 작성한 후 테스트를 통과하게 코드를 작성하는 방법을 권장.
- 미확인(Unchecked) 예외를 사용하라
  - 때로는 확인된 예외도 유용하다. 하지만 일반적인 애플리케이션은 의존성이라는 비용이 이익보다 크다.
- 예외에 의미를 제공하라
  - 예외를 던질 때는 전후 상황을 충분히 덧붙인다.
- 호출자를 고려해 예외 클래스를 정의하라
- 정상 흐름을 정의하라
  - 특수 사례 패턴 (Special Case Pattern): 클래스를 만들거나 객체를 조작해 특수 사례를 처리하는 방식.
- null을 반환하지 마라
  - null을 반환하는 코드는 일거리를 늘릴 뿐만 아니라 호출자에게 문제를 떠넘긴다. 누구 하나라도 null 확인을 빼먹는다면 애플리케이션이 통제 불능에 빠질지도 모른다.
  - 많은 경우 특수 사례 객체가 손쉬운 해결책이다.
- null을 전달하지 마라
  - 대다수 프로그래밍 언어는 호출자가 실수로 넘기는 null을 적절히 처리하는 방법이 없다. 그렇다면 애초에 null을 넘기지 못하도록 금지하는 정책이 합리적이다.