31. 리팩토링.

- TDD에서 중요한 것은 프로그램의 의미를 상실시키지 않는것

차이점 일치시키기.

- 비슷해보이는 두 코드조각. 합치려면?
  - 단계적으로 유사하도록 수정.
- 작은 단계. 명확한 feedback를 사용해서 테스트하려는 것이. "분할된 믿음이" 기반한 refactoring
  - 발생 빈도를 줄일 수 있다.
- 분기 / 반복문 / 클래스 등에서 이런 패턴 응용.

변화 격리하기.

- 객체 / 메서드의 일부만 변경하려면?
  - 바꿀부분을 격리한다.
- 객체추출 / method object / 메서드 추출 등이 있다.

레이어 이주시키기.

- 표현 양식을 변경시키려면?
  - 일시적 Data 중복을 고려한다.
    - → 일시적으로 중복시켰다. → 기존코드 삭제.

메서드 추출하기.

- 길고 복잡한 method를 읽기쉽게 만들려면?
  - 긴메서드의 일부를 별도로 분리하고 호출하도록.

메서드 인라인

- 너무 짧거나 제어흐름이 복잡하면 호출하는 메서드 본문을 가져온다 → 메서드 호출한 곳으로..

Interface 추출하기.

- operation에 대한 2번째 구현을 추가하려면.
  → 공통 Interface를 추가한다.

메서드 옮기기

- method를 원래 있던 곳으로 옮기려면?
  - 옮겨갈 class에 method를 추가하고 해당 method를 호출시킨다.

method object.

- 여러개의 매개변수와. 지역변수를 갖는 복잡한 method를 어떻게 분해할까?
  - → 객체로 만들자.
- 다수의 지역변수. 임시변수 등이 얽혀있는 경우 method 추출도 힘들기 때문.

매개변수 추가.

- method에 매개 변수를 추가하려면?
  - interface에 추가 → 매개변수 추가.
  - compiler가 가르쳐 준다.

method 매개변수를 생성자 매개변수로 바꾸려면?

- 하나 이상의 method 매개변수를 생성자로 옮기려면?
  - this를 활용 (인스턴스 변수 / 속성).