

프로그래밍 역량 강화 전문기관, 민코딩

문자열 유사도 검사



Method Level TDD

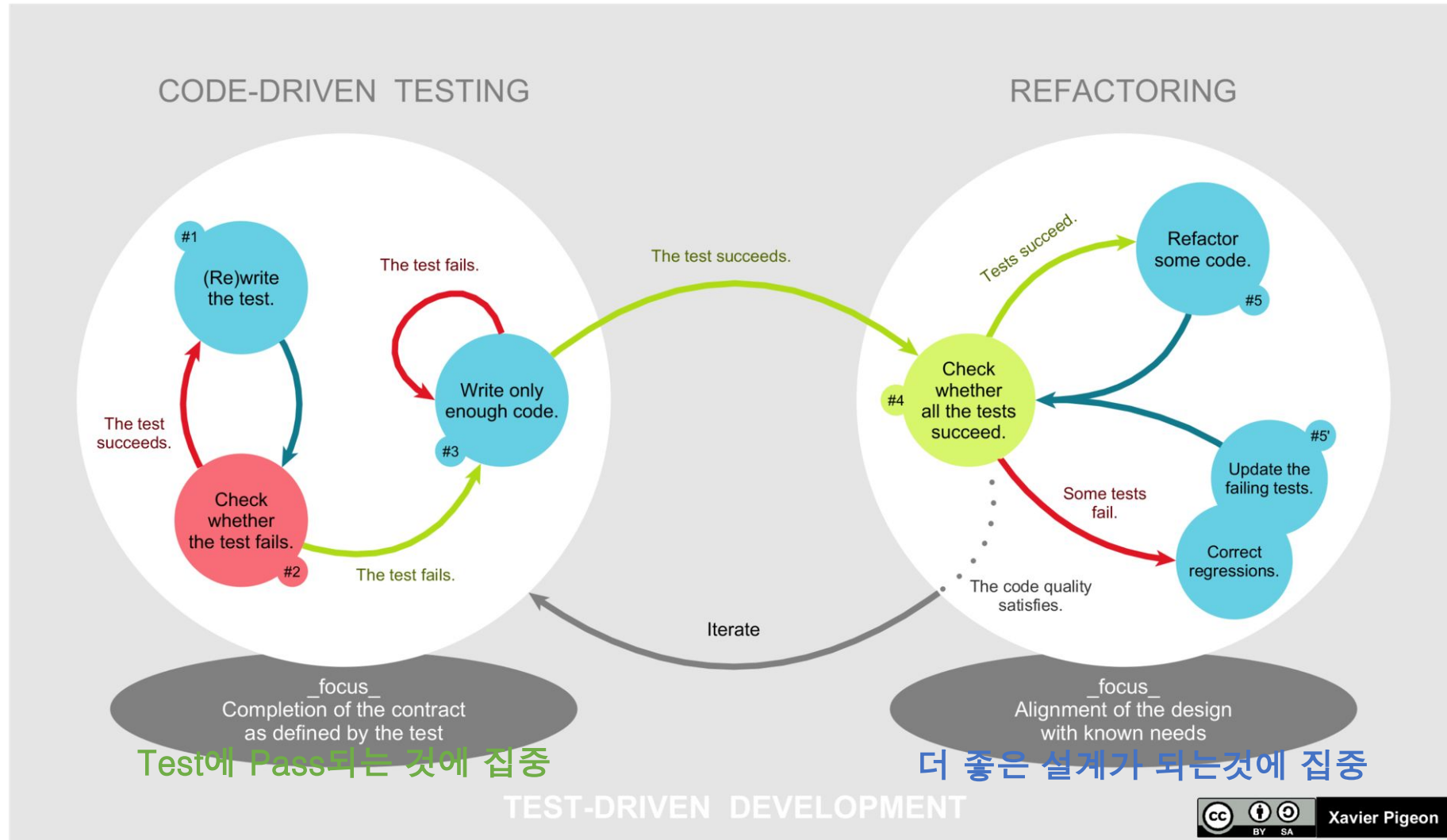
- ✓ 어떤 함수가 개발에 필요할지에 대해 고민하고, 이를 먼저 구현한다.
- ✓ “12-43-16” 의 Split & Sum 예시
 - split() -> TDD로 개발
 - toInts() □ TDD로 개발
 - sum() □ TDD로 개발
 - 위 내용 통합된 splitAndSum() □ TDD로 개발

[참고] Non TDD 대비, TDD의 이점

✓ Non TDD 대비, TDD 프로젝트의 품질 측정 비교

- 코드 품질 지표
 - <https://www.samsungsds.com/kr/insights/test-driven-development.html>
- TDD 이점에 대한 통계와 연구
 - https://www.hanbit.co.kr/media/channel/view.html?cms_code=CMS6706987091

[참고] TDD Cycle



Unit Test + 구현 + 리팩토링 훈련

TDD Project : 문자열 유사도 검사

두 문자열 유사도 검사기 (0 ~ 100점)

요구사항

- 두 String이 주어지면, 유사도를 검사한다.
- 0점 ~ 100점 으로 유사도 검사 결과를 출력한다.

예시) ABCDE, ABCDE = 100점

두 문자열 유사도 검사기 (0 ~ 100점)

유사도 검사법

1. 글자 수 검사 (최대 60점)
2. 등장 알파벳 검사 (최대 40점)

글자수 검사와 등장 알파벳 검사에서 각각 60, 40 점을 받으면,
최대 유사도 점수인 100 점을 획득할 수 있다.

1번 검사 : 글자 수

✓ 글자 수 검사 (max 60점)

- 같은 글자수일때, 최대 점수 60점 획득
- 문자열 길이가 2배 이상 차이가 난다면, 0점

✓ 부분 점수 측정 방법

- $Gap = A - B$
 - A = 긴 문자열 길이
 - B = 짧은 문자열의 길이

▪ 부분점수 공식

$$\left(1 - \frac{Gap}{B}\right) \times 60$$

ASD, DSA = 60점
A, BB = 0점
AAABB, BAA = 부분점수
AA, AAE = 부분점수

2번 검사 : 등장 알파벳 검사

✓ 등장 알파벳 검사 (max 40점)

- 두 문자열의 같은 종류의 알파벳이 사용되었다면 최대 점수 40점 획득
- 두 문자열의 사용된 알파벳이 모두 다르다면 0점
- 대문자만 취급.

✓ 부분 점수 측정 방법

- TotalCnt = 등장하는 알파벳 총 개수
- SameCnt = 두 문자열 중, 같은 알파벳의 개수

▪ 부분점수 공식

$$\left(\frac{SameCnt}{TotalCnt} \right) \times 40$$

ASD, DSA = 40점
A, BB = 0점
AAABB, BA = 40점
AA, AAE = 부분점수

진행방법

Repository 준비

- 각자 Repository 를 준비한다. (Repo명 : SimilarityChecker-번호)
- README 파일
- 팀원 Collaborator 등록
- Branch Protection Rule – Approve 1인 필수

TDD로 개발 후 코드리뷰 요청

- Branch 1 : feature/length
- Branch 2 : feature/alpha
- PR 요청 후 코드리뷰
 - 팀원들은 미세한 개선사항을 제안 하면서, Request change 1회 필수
 - 정말 미세한 개선사항으로 Request change를 해도 좋음 (기분 상하지 말자)