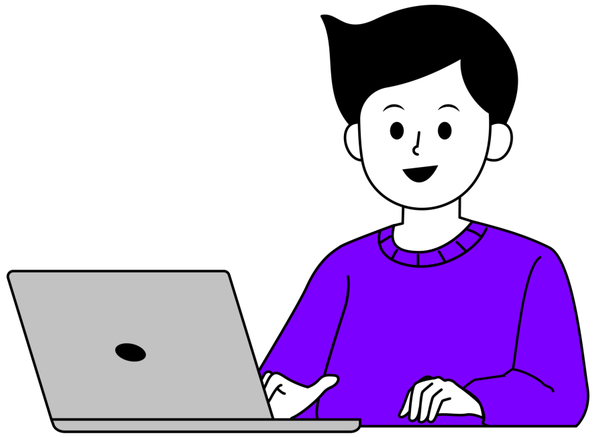
3주차 발표_민서연

TDD

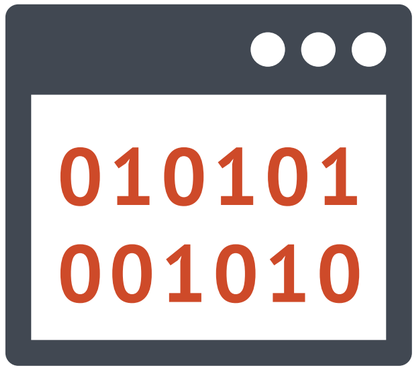
01 정의

TDD (Test-driven delvelopment)

짧은 개발 사이클을 반복하는 소프트웨어 개발 프로세스



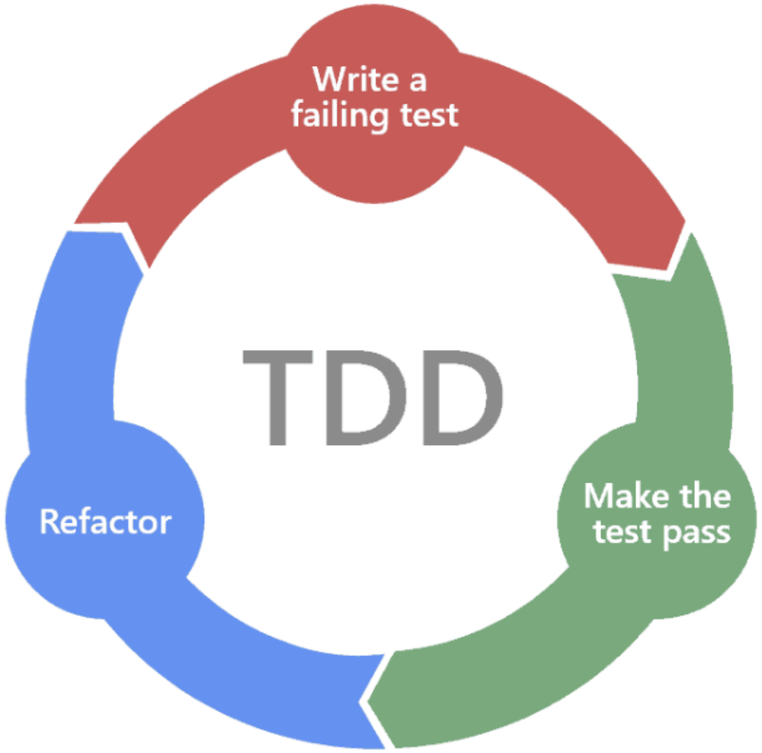
요구사항을 검증하는 자동화된 테스트 케이스를 작성



테스트 코드를 작성하기 위한 최소한의 코드

표준에 맞도록 리펙토링

01 정의



Red

- 하려는 행위를 변수, 객체, 메소드가 있다는 전제 하에 코드 작성

Green

- 테스트를 통과 시키기 위한 코드 작성

Refactor

- 통과된 코드를 리팩토링

02 TDD 장단점

장점

높은 코드 안정성

재설계 시간의 단축

디버깅 시간 단축

단점

생산성 저하

구조에 얽매임

03 실제 예시 - 간단한 계산기 프로그램

1. 요구 사항: 두 숫자를 더하고 빼는 계산기 프로그램 개발

2. 첫 번째 요구 테스트 작성

```
def test_addition():
    assert add(1, 2) == 3  # 실패하는 테스트
```

3. 테스트 통과를 위한 최소한의 코드 작성

```
def add(a, b):
    return a + b
```

03 실제 예시 - 간단한 계산기 프로그램

4. 두번째 테스트 작성

```
def test_subtraction():
    assert subtract(4, 2) == 2  # 실패하는 테스트
```

5. 테스트 통과를 위한 최소한의 코드 작성

```
def subtract(a, b):
    return a - b
```

6. 추가 테스트와 리팩토링

- 여러가지 케이스에 대한 테스트를 추가하고 필요하다면 코드 리팩토링함 (ex 음수를 처리하는 지)