

# ILE3-027 테스트

소프트웨어의 품질을 높이기 위한 애플리케이션 테스트에 대하여 학습

## 소프트웨어 테스트

구현된 어플리케이션 or 시스템에 대하여 결함을 찾아내는 활동

- 오류 발견 관점  
소프트웨어의 잠재된 오류 발견
- 오류 예방 관점  
코드 리뷰, 크로스 체크 등을 이용하여 정적으로 오류 발견
- 품질 향상 관점  
요구사항을 만족시키기 위해 반복적인 테스트를 수행

## 기본 원칙

1. 테스트는 결함이 존재함을 밝히는 활동이다
2. 완벽한 테스트는 불가능하다.
3. 테스트는 개발 초기에 시작해야 한다.
4. 테스트 코드의 지속적인 개선

## 테스트의 유형

- 정적 테스트  
프로그램을 실행시키지 않고 소스 코드를 이용하여 논리적으로 검증
- 동적 테스트  
프로그램을 실행하여 수행하는 테스트로 블랙박스 테스트, 화이트박스 테스트가 존재
  - 화이트박스 테스트  
내부의 로직을 모두 확인하면서 테스트
  - 블랙박스 테스트  
입/출력 값을 이용하여 테스트

## 테스트 방법

1. 안전 테스트  
소스코드 내의 보안적인 결함 테스트
2. 구조 테스트  
시스템의 내부 논리 경로, 소스코드의 복잡도를 평가하는 테스트
3. 회복 테스트  
고의로 오류를 유도하고, 이 후에도 정상적으로 작동하는지 테스트
4. 스트레스 테스트  
과도한 데이터를 입력하여 시스템 과부하 시에도 정상적으로 작동하는지 테스트
5. 성능 테스트  
응답 시간, 출력량 등의 반응 속도의 테스트
6. 병행 테스트  
다른 시스템에서도 입력이 동일한 경우, 같은 출력이 나오는지 테스트

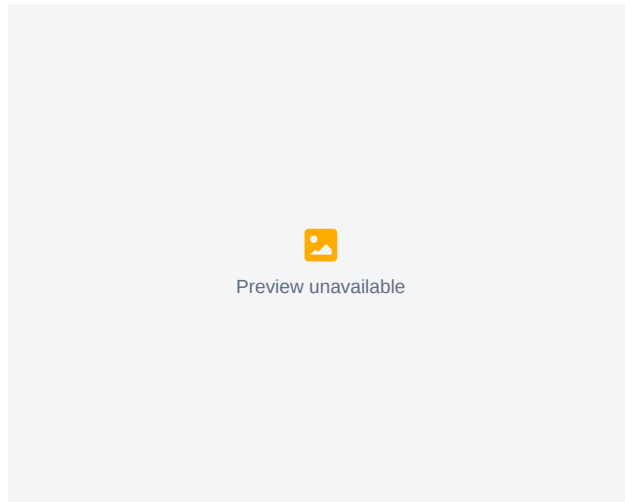
## 테스트의 구성 [↗](#)

### 통합 테스트 [↗](#)

애플리케이션 통합 테스트는 소프트웨어 각 모듈 간의 인터페이스 관련 오류 및 결함을 찾아내는 테스트

#### Top Down 방식 [↗](#)

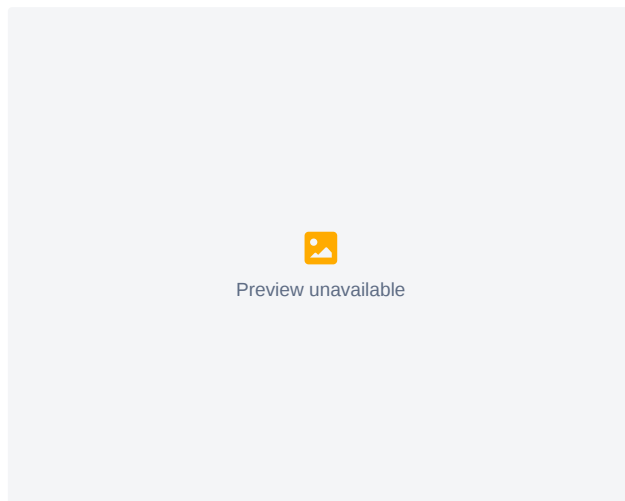
메인 프로그램으로 부터 하위로 Flow를 따라 테스트를 수행



1. 메인 제어 모듈은 작성된 프로그램을 사용하고, 아직 작성되지 않은 하위 제어 모듈 및 모든 하위 컴포넌트를 대신하여 더미 모듈인 스텝(Stub)을 개발한다.
2. 깊이-우선 방식 또는 너비-우선 방식에 따라, 하위 모듈인 스텝이 한 번에 하나씩 실제 모듈로 대체된다.
3. 각 모듈 또는 컴포넌트를 통합하면서 테스트가 수행된다.
4. 테스트가 완료되면 스텝이 실제 모듈 또는 컴포넌트로 작성된다.

#### Bottom Top 방식 [↗](#)

최하위부터 상위로 Flow를 따라 테스트를 수행



1. 최하위 레벨의 모듈 또는 컴포넌트들이 하위 모듈의 기능을 수행하는 클러스터(Cluster)로 결합된다.
2. 상위의 모듈에서 데이터의 입력과 출력을 확인하기 위한 더미 모듈인 드라이버(Driver)를 작성한다.
3. 각 통합된 클러스터 단위를 테스트한다.

4. 테스트가 완료되면 각 클러스터들은 프로그램의 위쪽으로 결합되며, 드라이버는 실제 모듈 또는 컴포넌트로 대체된다.