

학습 테스트와 단위 테스트

테스트 연습 1단계 - 학습 테스트

학습 테스트란?

기능 구현을 위한 테스트라기 보다 새로운 API, 라이브러리, 프레임워크가 어떻게 동작하는지를 검증하기 위한 테스트

학습 테스트 장점

[학습 테스트로 배우는 스프링](#)에서 인용

- 다양한 조건에 따른 기능을 손쉽게 확인해 볼 수 있다.
- 학습 테스트 코드를 개발 중에 참고할 수 있다.
- 프레임워크나 제품을 업그레이드할 때 호환성 검증을 보여준다.
- 테스트 작성에 대한 좋은 훈련이 된다.
- 새로운 기술을 공부하는 과정이 즐거워진다.

학습 테스트 예시 1

```
import org.junit.Test;

import static org.assertj.core.api.Assertions.assertThat;

public class StringTest {
    @Test
    void isEmpty() {
        assertThat("").isEmpty().isTrue();
        assertThat(" ".isEmpty()).isFalse();
        assertThat("a".isEmpty()).isFalse();
    }

    @Test
    void lastIndexOf() {
        assertThat("abcde".lastIndexOf("b")).isEqualTo(1);
    }
}
```

학습 테스트 예시 2

```
import org.junit.Test;

import java.util.ArrayList;

import static org.assertj.core.api.Assertions.assertThat;

public class CollectionTest {
    @Test
    public void arrayList() {
        ArrayList<String> values = new ArrayList<>();
        values.add("first");
        values.add("second");
        assertThat(values.add("third")).isTrue();
        assertThat(values.size()).isEqualTo(3);
        assertThat(values.get(0)).isEqualTo("first");
        assertThat(values.contains("first")).isTrue();
        assertThat(values.remove(0)).isEqualTo("first");
        assertThat(values.size()).isEqualTo(2);
    }
}
```



참고 자료

- [학습테스트\(Learning Test\)를 이용해서 공부하기](#)
- [학습테스트](#)

테스트 연습 2단계 - 단위 테스트

단위 테스트는 프로덕션 코드를 구현한 후 JUnit과 같은 도구를 활용해 작은 단위를 테스트하는 것을 의미한다.

단위 테스트와 TDD는 다르다.

내가 구현하는 메소드(함수) 중

Input과 Output이 명확한 클래스 메소드(보통 Util 성격의 메소드)

에 대한 단위 테스트 연습

알고리즘을 학습한다면 알고리즘 구현에 대한 검증을 단위 테스트로 한다.

알고리즘은 Input, Output이 명확하기 때문에 연습하기 좋다.



미션 완료



나의 팀



슬라이드