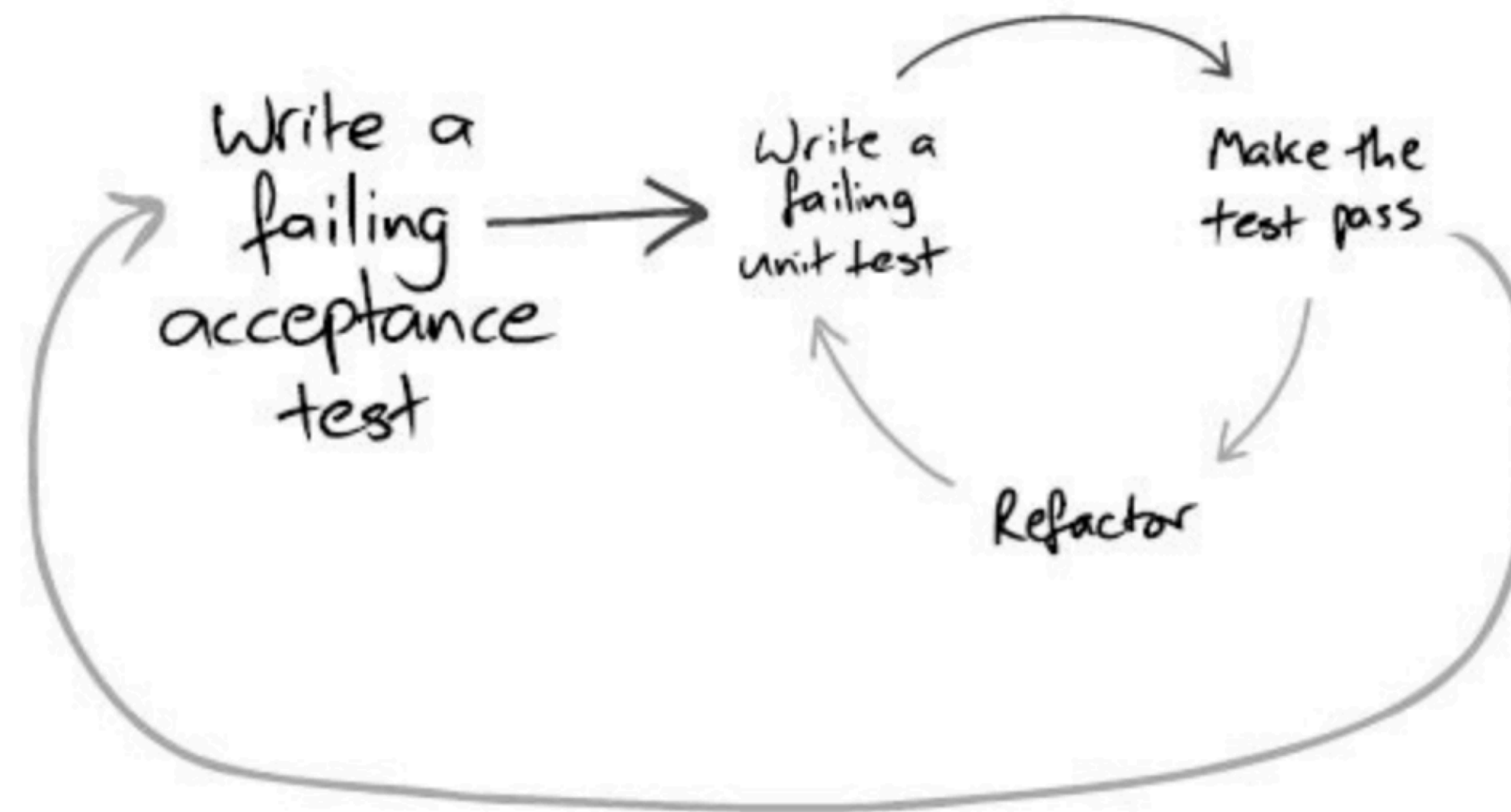


인수 테스트 주도 개발

인수 테스트 주도로 개발 하기

인수 테스트 주도로 개발은 어떻게 하나요?



인수 테스트 주도로 개발 하기

인수 조건 정의

인수 테스트 작성

기능 구현

인수 조건

- 인수 테스트가 충족해야하는 조건
- 이번 과정에서는 시나리오 형태로 표현

인수 조건 형식

- **scenario-oriented** (the Given/When/Then template)
- rule-oriented (the checklist template)
- custom formats.

인수 조건 예시

Feature: 최단 경로 구하기

Scenario: 지하철 최단 경로 조회

Given 지하철역들이 등록되어 있다.

And 지하철노선이 등록되어 있다.

And 지하철노선에 지하철역들이 등록되어 있다.

When 사용자는 출발역과 도착역의 최단 경로 조회를 요청한다.

Then 사용자는 최단 경로의 역 정보를 응답받는다.

인수 테스트

- 인수 조건을 검증하는 테스트
- 실제 요청/응답하는 환경과 유사하게 테스트 환경을 구성

인수 테스트 클래스

```
@SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.RANDOM_PORT)
public class LineAcceptanceTest {

    @DisplayName("최단 경로 구하기")
    @Test
    void findPath() {
        ...
    }
}
```

인수 테스트 예시

```
@DisplayName("지하철 노선을 조회한다.")
@Test
void getLine() {
    // given
    // 지하철_노선_등록되어_있음
    Map<String, String> params = new HashMap<>();
    params.put("name", "신분당선");
    params.put("color", "bg-red-600");
    params.put("startTime", LocalDateTime.of( hour: 05, minute: 30).format(DateTimeFormatter.ISO_TIME));
    params.put("endTime", LocalDateTime.of( hour: 23, minute: 30).format(DateTimeFormatter.ISO_TIME));
    params.put("intervalTime", "5");

    ExtractableResponse<Response> createResponse = RestAssured.given().log().all().
        contentType(MediaType.APPLICATION_JSON_VALUE).
        body(params).
        when().
        post( path: "/lines").
        then().
        log().all().
        extract();

    // when
    // 지하철_노선_조회_요청
    String uri = createResponse.header( name: "Location");

    ExtractableResponse<Response> response = RestAssured.given().log().all().
        accept(MediaType.APPLICATION_JSON_VALUE).
        when().
        get(uri).
        then().
        log().all().
        extract();

    // then
    // 지하철_노선_응답됨
    assertThat(response.statusCode()).isEqualTo(HttpStatus.OK.value());
    assertThat(response.as(LineResponse.class)).isNotNull();
}
```

인수 테스트 예시2

```
@DisplayName("지하철 노선에 등록된 지하철역을 제외한다.")
@Test
void removeLineSection1() {
    // given
    지하철_노선에_지하철역_등록_요청(신분당선, 강남역, 양재역, 2);
    지하철_노선에_지하철역_등록_요청(신분당선, 양재역, 정자역, 2);

    // when
    ExtractableResponse<Response> removeResponse
        = 지하철_노선에_지하철역_제외_요청(신분당선, 양재역);

    // then
    지하철_노선에_지하철역_제외됨(removeResponse);
    ExtractableResponse<Response> response
        = LineAcceptanceTest.지하철_노선_조회_요청(신분당선);
    지하철_노선에_지하철역_순서_정렬됨(response, Arrays.asList(강남역, 정자역, 광교역));
}
```

기능 구현

- 인수테스트 기반으로 하나씩 기능 개발을 진행

인수 테스트 주도 개발 실습

- 지하철역 삭제 기능 인수 테스트 만들기