TDD

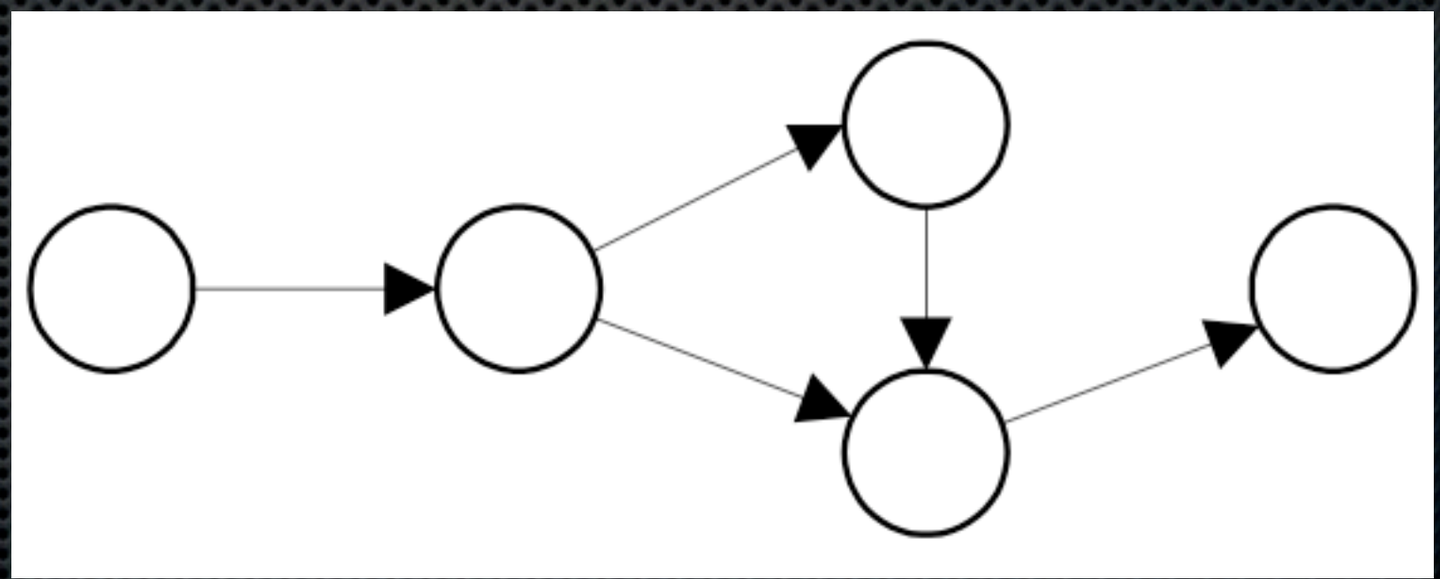
Daum Corp.
백명석

발표목차

- TDD 소개
  - Why TDD
  - TDD Life Cycle
- TDD Examples
  - Movie / Tyrant Client / Bowling Game
- What is Mock Object
  - Common Structure
  - Mock Test Example
- Q&A

TDD

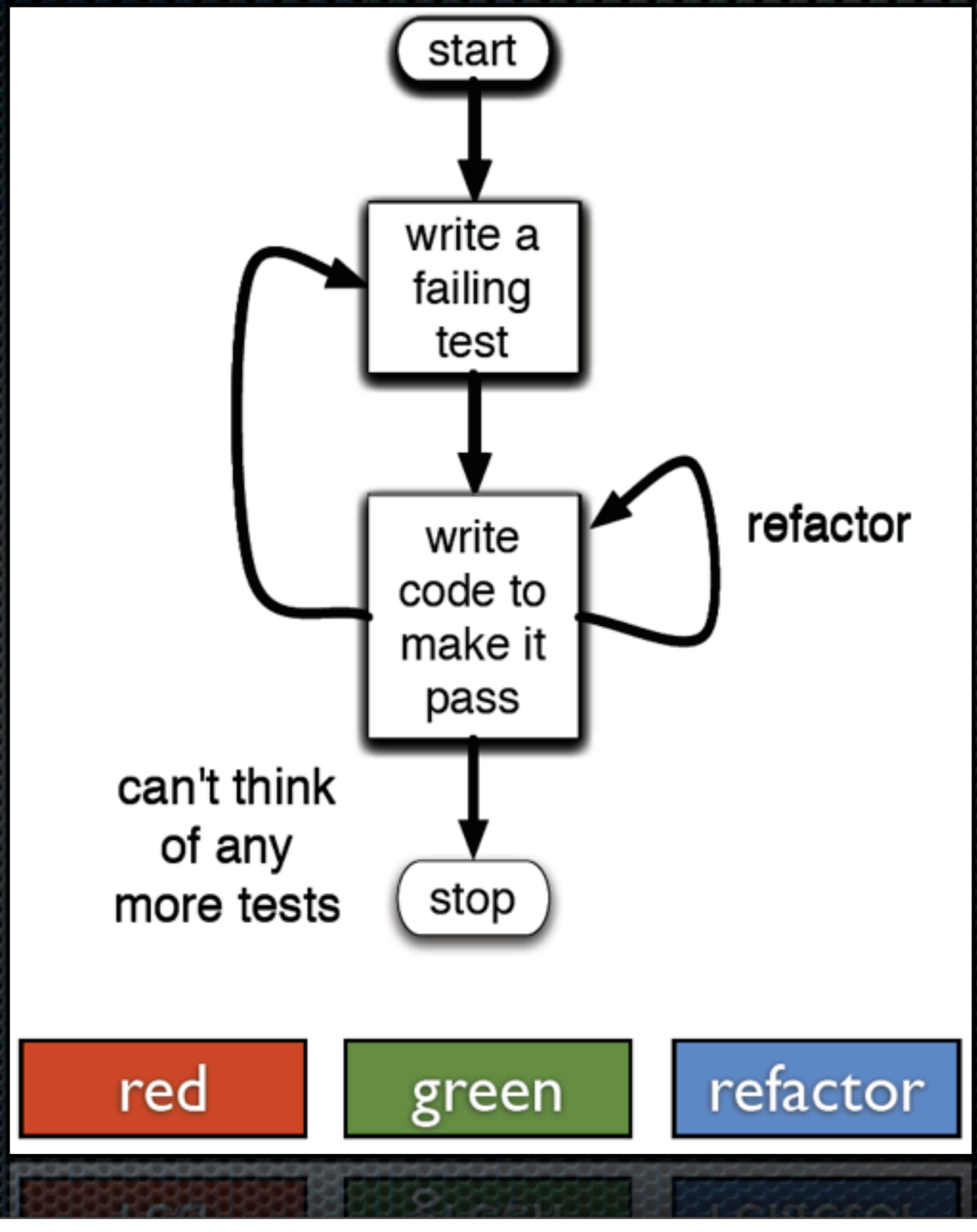
- Test Driven Development
  - Test Driven Design
  - Test First Development(not TAD)
- Find Interfaces/Collaborators Iteratively
- A Running OOP :A Web of Collaborating Objects



Why TDD

- Testing is only a side effect of TDD
  - … the import part is how it changes your code for the better
- Benefits of TDD
  - Debugging Champion vs Good Designer
  - First Consumer, Manual, Design Document
  - Decoupling(force mocking dependent objects)

TDD Life Cycle



TDD Example1

- 영화 등급을 입력 받아 평균 등급을 제공하는 기능
- “Test-Driven Development by Example”, Kent Beck
- movie.mov

TDD Example2

- Tyrant Client
  - http://vimeo.com/10789674
- tyrant.mov

TDD Example2

- Test First
  - Test가 없을 수 없다.
- Always Executable/Testable
- Refactoring이 자유롭다.
- Legacy에 코드를 추가할 때도 적용 가능
  - Sprout / Wrap

TDD Example 3

- Bowling Game
  - Red / Green / Refactor(Blue)
  - Red: next most interesting case but still really simple
  - Green: make it pass
  - Blue: refactor

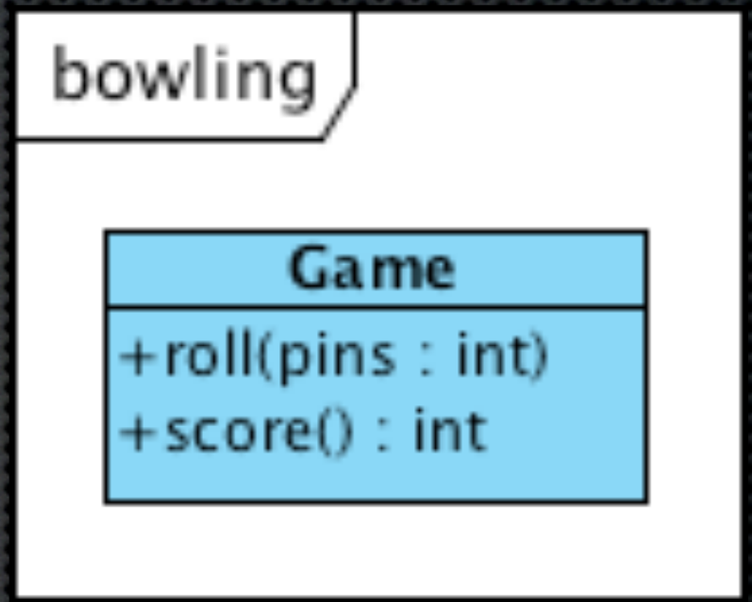
TDD Example 3

- Rule
  - Game has 10 Frames
  - Each Frame has 2 Rolls
  - Spare: 10 + next first roll에서 쓰러 뜨린 핀수.
  - Strike: 10 + next two rolls에서 쓰러 뜨린 핀수.
  - 10th 프레임은 특별. spare 처리하면 3번 던질 수 있음.

1	4	4	5	6	/	5	/	X	0	1	7	/	6	/	X	2	6
5		14		29		49		60	61		77		97		117	133	

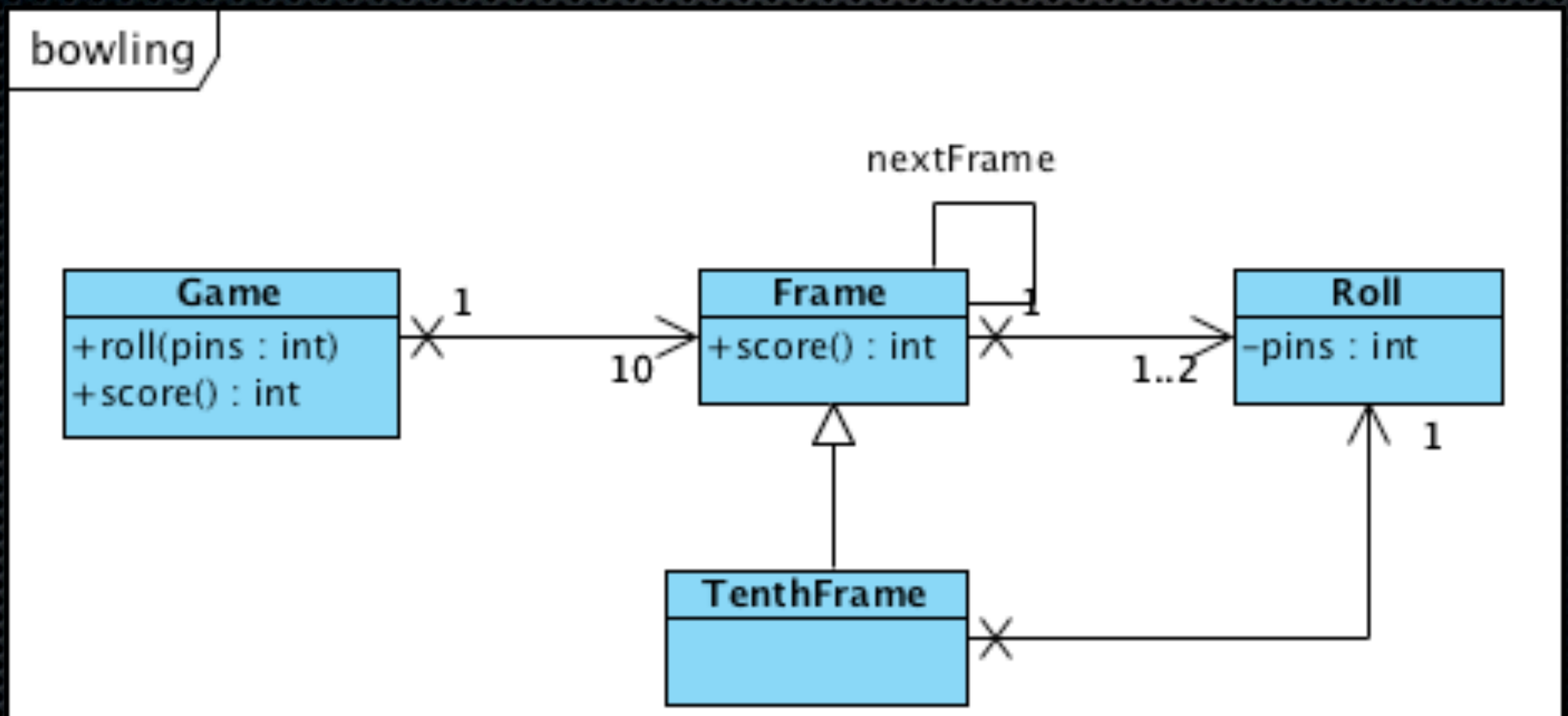
TDD Example 3

- 목적
  - Game 클래스 생성
- Game 클래스
  - roll과 score라는 2개의 메소드
  - roll
    - ball을 roll할 때마다 호출
    - 인자 pins는 쓰러진 핀수
  - score
    - 게임이 끝난 후에만 호출
    - 게임의 점수를 반환



TDD Example 3

- 간단한 설계



TDD Example 3

- 시연
  - http://www.cleancoders.com/codecast/clean-code-episode-6-part-2/show
- bowling-game.mov

TDD Example 3

- Red-Green-Blue
- 사전설계: 무시하지 않지만 따르지도 않는다.
  - 감을 잡기 위함. Scratch Refactoring
- Nothing으로 시작(Always runnable)
- 생각나는 Production Code를 작성하도록 유도하는 테스트를 추가
- 가장 쉽고, 간단하면서 흥미로운 테스트를 추가
- 각 Phase에 있을 때는 그 Phase에서 해야 할 일만
  - 뒷통수

What is Mock Object

- Extension to TDD
- Test Doubles
- Support OOD by guiding the discovery of types
- Identifying types based on the object's roles
- Verifying Interactions Not State

Common Structure

- Create the test fixture including any mock objects
- Define expectations on the mock objects
- Invoke the method to be tested
- Verify expectations and assert any postconditions

Mock Test Example

- WriteArticleController#writeArticle
  - parse user's request
  - delegate to domain object
    - save
    - display
  - decide next page
- mock-test.mov

Benefits of Mock

- Clarifying the interactions between classes
- Iterative “Interface Discovery”
- Modularity & Responsibility Assignment
- Reducing dependencies
- Independent Development
  - No Disturbance
  - No Wait
  - DI

Q&A

Q&A. TDD 아는데...

- TDD 알긴 하는데.
- 익혀야 한다.
- 學而時習之 不亦悅乎.
- 10만번 연습.
- 연습은 연습장에서.

–시합에서 하는 것은 연습이 아니다.

Q&A. TAD ???

- TAD(Test After Development)
  - 프로젝트 완료 후 작성하는 문서
  - 되는 것만 해 본다.
  - 개발하기 위해 협의한 문서가 가치
  - 완료한 후 인수인계를 위해 작성한 문서는 무의미
    - 형식적
    - 도움이 안됨
  - Scenario Test 개념으로는 의미있음

Q&A. Legacy Project

- Sprout/Wrap Method/Class
- Extract Till Drop - fitness
- Extract Method Object
- Characterization Test - using replacing stdout

Q&A. Grand Refactoring Project

- Grand Refactoring Project or Make a New System
  - always fails
  - Pareto's Rule
  - Needs Based Refactoring
- 장애/버그
  - 재현하는 테스트 추가
  - Make It Pass

Q&A. TDD & BDUF

- BDUF(Big Design Up Front)
- 허공에 설계 vs 동작하는 코드로 설계
- 오늘 결정할 수 없는 일: unknown unknowns
  - 7% always, 13% often, 45% Never
- 2주 단위의 Iteration에서 2주 후에 보여줄 Executable(Unit Test, Scenario Test …)

Q&A. Mock Library

- 가급적 안 쓰는 것이 가장 좋은
- 신규 코드 작성
  - subclassing & override
  - or Mockito
- 레거시 테스트 작성
  - mockito, powermock
  - jMockit
    - http://martinfowler.com/articles/modernMockingTools.html
    - https://github.com/Hypoport/welc_examples_java_jmockit.git

jMockit

CurrencyConversion.java – [jmockit-test] – jmockit-test – [~/git/DaumDevCulture/samples/jmockit-test]

CurrencyConversionTest.java

```
                CurrencyConversion.currencySymbols();
                result = Arrays.asList(new String[] { "FOO" });
            }
        };
        CurrencyConversion.convertFromTo("FOO", "BAR");
    }

    @Test
    public void convertsCorreclty() throws Exception {
        final ByteArrayInputStream bais = new ByteArrayInputStream(
                "<div id=\"converter_results\"><ul><li><b>5 x = 42 Y</b>".ge

        new NonStrictExpectations() {
            DefaultHttpClient httpclient;
            HttpResponse response;
            HttpEntity entity;
            {
                httpclient.execute((HttpUriRequest) any);
                result = response;
                response.getEntity();
                result = entity;
                entity.getContent();
                result = bais;
            }
        };

        new NonStrictExpectations(CurrencyConversion.class) {
            {
                CurrencyConversion.currencySymbols();
                result = Arrays.asList(new String[] { "X", "Y" });
            }
        };

        BigDecimal result = CurrencyConversion.convertFromTo("X", "Y");
        assertEquals(new BigDecimal(42), result);
    }
}
```

CurrencyConversion.java

```
        String url = "http://www.jhall.demon.co.uk/currency/by_currency.html'
        try {
            HttpClient httpclient = new DefaultHttpClient();
            HttpGet httpget = new HttpGet(url);
            HttpResponse response = httpclient.execute(httpget);
            HttpEntity entity = response.getEntity();
            if (entity != null) {
                InputStream instream = entity.getContent();
                InputStreamReader irs = new InputStreamReader(instream);
                BufferedReader br = new BufferedReader(irs);
                String l;
                boolean foundTable = false;
                while ((l = br.readLine()) != null) {
                    if (foundTable) {
                        if (l.matches("\\s+<td valign=top>[A-Z]{3}</td>")) {
                            result.add(l.replaceAll(".*top>", "").replaceAll
                        }
                    }
                    if (l.startsWith("<h3>Currency Data"))
                        foundTable = true;
                    else
                        continue;
                }
            }

        } catch (Exception e) {
            throw new RuntimeException(e);
        }

        allCurrenciesCache = result;
        lastCacheRead = System.currentTimeMillis();

        return result;
    }

    public static BigDecimal convertFromTo(String fromCurrency, String toCur
        List<String> valid = currencySymbols();
        if (!valid.contains(fromCurrency))
            throw new IllegalArgumentException(String.format("Invalid from c
        if (!valid.contains(toCurrency))
            throw new IllegalArgumentException(String.format("Invalid to cur
        String url = String.format("http://www.gocurrency.com/v2/dorate.php?
```

Q&A. Naming

- 이름은 매우 중요
- 구현되지 않은 것에 대해서 이름이 잘 안 떠오를 수 있음.
- foo, bar 등으로 시작하고, 의미가 정확해 질 때 이름도 정확하게 변경
- Suffering-Oriented Programming
  - http://nathanmarz.com/blog/suffering-oriented-programming.html
  - 1. make it possible
  - 2. make it beautiful
  - 3. make it fast

Q&A. Private Method는 ???

- protected로 scope을 변경하여
- reflection을 이용해서
- public으로 변경해서
- 테스트 하지 말라

Q&A - GUI는 어떻게 테스트하나 ?

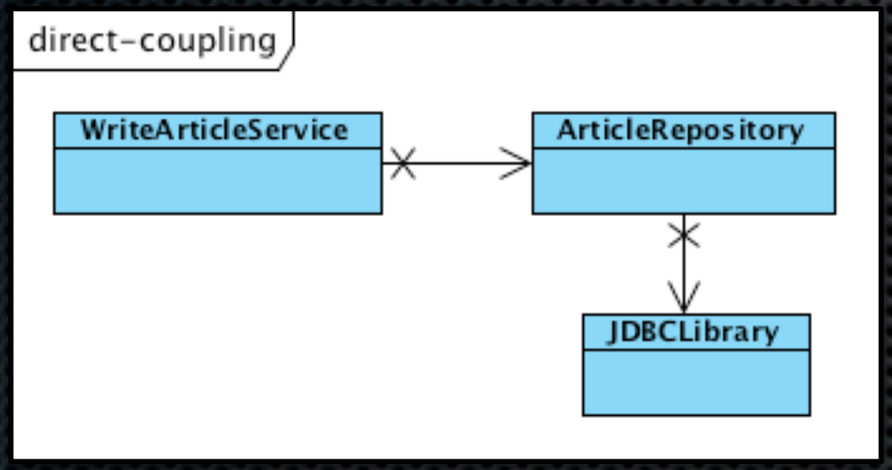
- 대개는 안 한다.
- the content of the screen can be tested, event if the final format can not.
  - Gray로 표현할 지 말지를 결정하는 로직 - O
  - Gray로 표현하는 것 자체 - X
- GUI를 2개의 컴포넌트로 분리한다.
  - Presenter(O)/View(X)
  - Presenter: integrate the business objects and then constructs the data structure
  - View: renders that data structure.

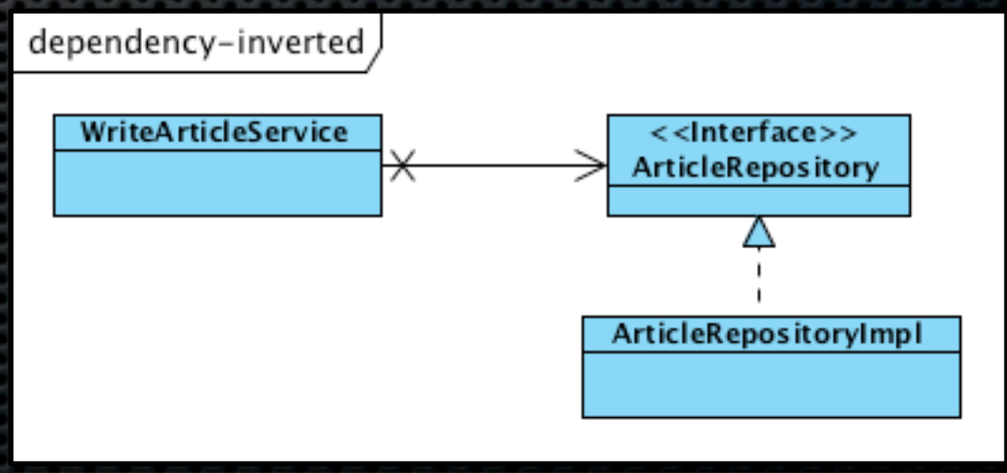
Q&A -DB는 어떻게 테스트하나 ?

- 대개의 경우는 테스트 안한다(블랙박스).
- 테스트 하고 싶은 것
  - 스키마
  - 쿼리가 원하는 플랜대로 동작하는가
  - 수작업으로 한번만
- Layering 기법을 이용
  - Application과 DB를 분리
  - DB를 mock이나 stub으로 교체할 수 있다.
  - Application을 실제 DB 없이 테스트

Q&A - Layering 기법(DIP)

- High Level Module should not depend on Low Level Detail
- Separate Complexity





End