



현장 리뷰어 양성 과정

Refactoring

Hands-on

CONTENTS

1. 개발환경설정
2. Test Case 소개
3. Legacy Code 소개
4. Test case Coverage 100%
5. IDE를 활용한 리팩토링
6. 추상화를 활용한 리팩토링
7. 패턴을 활용한 리팩토링
8. 회고

현장 리뷰어 양성과정

Refactoring

Hands-on



★
★ 김동식(삼성SDS)
★ SW 엔지니어링팀

- 전문영역 : 개발플랫폼, 개발생산성
- . SW 개발방법론 리딩
- . DEVOPS 개발환경 구축 지원
- . Agile Coach
- . Clean Code & Refactoring

dongsik2026.kim@Samsung.com

오늘 할 일

GildedRose Refactoring

- Test Case Coverage 100% 만들어 보기
- IDE를 활용한 Refactoring
- 추상화를 활용한 Refactoring
- 패턴을 활용한 Refactoring



1. 개발환경설정

실습환경

소스 다운로드 : <https://github.com/cleancodeclass/gilded-rose>
(master 브랜치)

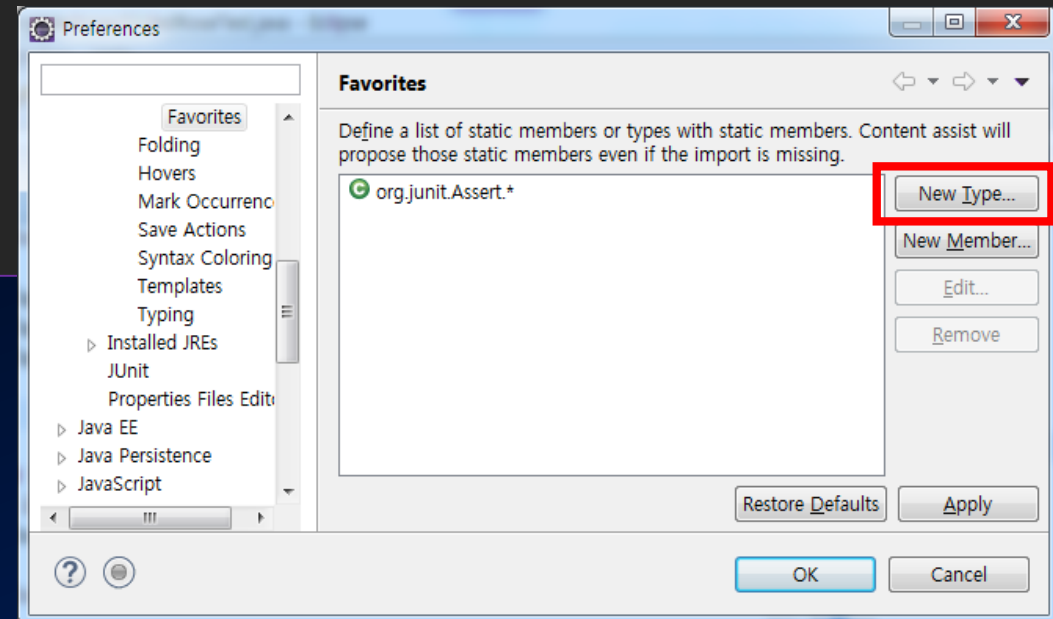
Static import 추가

Windows -> Preferences 메뉴

Java -> Editor -> Content Assist -> Favorites 항목

"New Type..." 버튼 : " 클릭하여 아래내용 추가

org.junit.Assert



실습환경

테스트 코드

```
GildedRoseTest.java
34 // act (when)
35 gildedRose.updateQuality();
36
37 // assert (then)
38 assertEquals(NORMAL_ITEM, items[0].name);
39 assertEquals(-1, items[0].sellIn);
40 assertEquals(0, items[0].quality);
41 }
42
43 @Test
44 public void Normal아이템_판매기간0_품질5_테스트() {
45
46 // arrange (given)
47 Item[] items = new Item[] { new GildedRose(0, 5) };
48 GildedRose gildedRose = new GildedRose(items);
49
50 // act (when)
```

원본코드

```
GildedRose.java
7 this.items = items;
8 }
9
10 public void updateQuality() {
11     for (int i = 0; i < items.length; i++) {
12         if (!items[i].name.equals("Aged Brie")
13             && !items[i].name.equals("Sulfur Dioxide")
14             && items[i].quality > 0) {
15             if (items[i].name.equals("Backstage Passes")) {
16                 items[i].quality = Math.min(50, items[i].quality + 1);
17             } else {
18                 items[i].quality = Math.max(1, items[i].quality - 1);
19             }
20             if (items[i].quality < 50) {
21                 items[i].quality = items[i].quality + 1;
22             }
23             if (items[i].name.equals("Sulfur Dioxide")) {
24                 items[i].quality = Math.max(1, items[i].quality - 1);
25             }
26         }
27     }
28 }
```

테스트 결과

```
Finished after 0.033 seconds
Runs: 14/14 Errors: 0 Failures: 0
com.gildedrose.GildedRoseTest [Runner: JUnit 4] (0.007 s)
BackstagePass아이템 판매기간0 품질0 테스트 (0.000 s)
BackstagePass아이템 판매기간0 품질10 테스트 (0.000 s)
```

테스트 커버리지

Element	Coverage	covered
GildedRoseMaster	61.1 %	

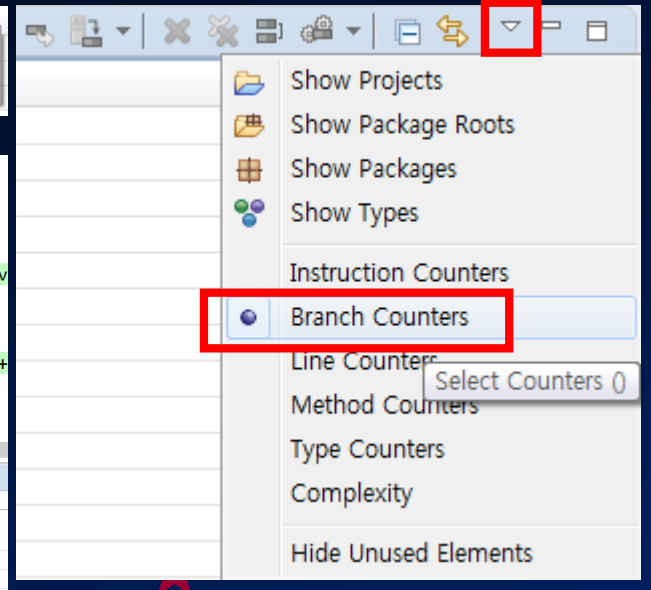
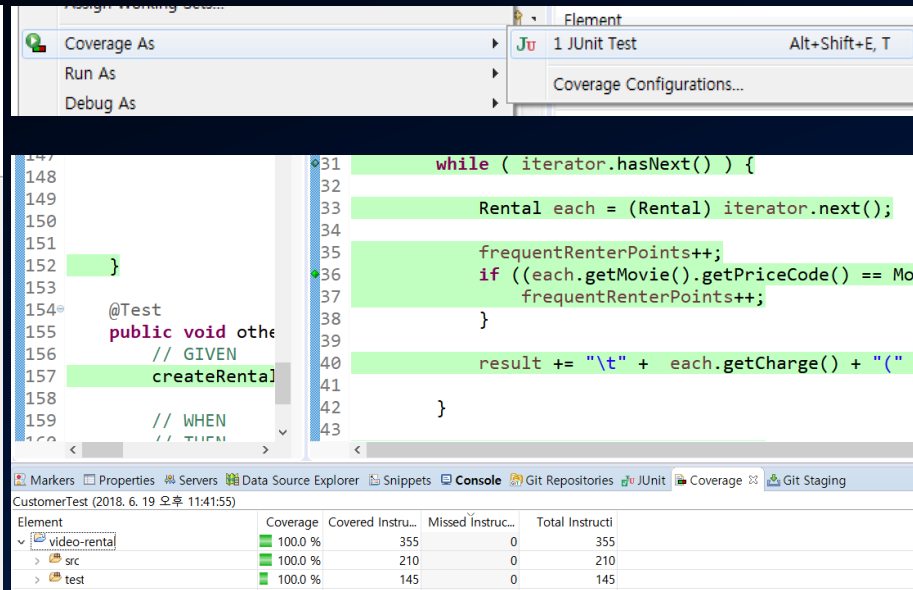
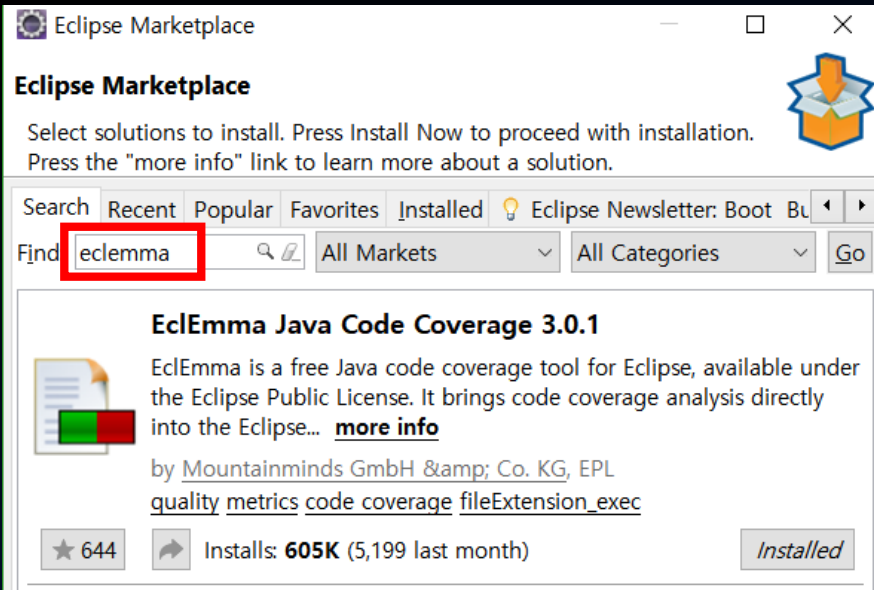
실습환경

테스트케이스 커버리지 : Eclemma

이클립스 - Help

- Eclipse Marketplace – eclemma – 설치 – 재기동 – [Branch Counters]로 설정

① - Alt + Shift + E + T 또는 TestCase 선택 후 [우클릭] – [Coverage As] – [JUnit Test] 실행



실습환경

```
GildedRose.java
/  this.items = items;
8  }
9
10 public void updateQuality() {
11     for (int i = 0; i < items.length; i++) {
12         if (!items[i].name.equals("Aged Brie")
13             && !items[i].name.equals("Backstage passes to a TAFK..."))
14             if (items[i].quality > 0) {
15                 if (!items[i].name.equals("Sulfuras, Hand of Ragnaros"))
16                     items[i].quality = items[i].quality - 1;
17             }
18         }
19     } else {
20         if (items[i].quality < 50) {
21             items[i].quality = items[i].quality + 1;
22
23             if (items[i].name.equals("Backstage passes to a TAFK..."))
24                 if (items[i].sellIn < 11) {
25                     if (items[i].quality < 50) {
26                         items[i].quality = items[i].quality + 1;
27                     }
28                 }
29     }
```

테스트 코드에 의해
분기문의 조건이 **모두 실행된** Line

테스트 코드에 의해
분기문의 조건이 **일부 실행된** Line

테스트 코드에 의해
분기문이 **전혀 실행되지 않은** Line



2. Test Case 소개

Test Case 소개

Junit 테스트 메서드 구조

```
import static org.junit.Assert.*;
import org.junit.Test;
```

```
public class PersonTest {
```

```
    @Test
```

```
    public void testGetDisplayName() {
```

```
        // arrange
```

```
        Person person = new Person("Gogh", "Vincent");
```

```
        // act
```

```
        String displayName = person.getDisplayName();
```

```
        // assert
```

```
        assertEquals("Vincent Gogh", displayName);
```

```
    }
```

```
}
```

1. '@Test' method annotation

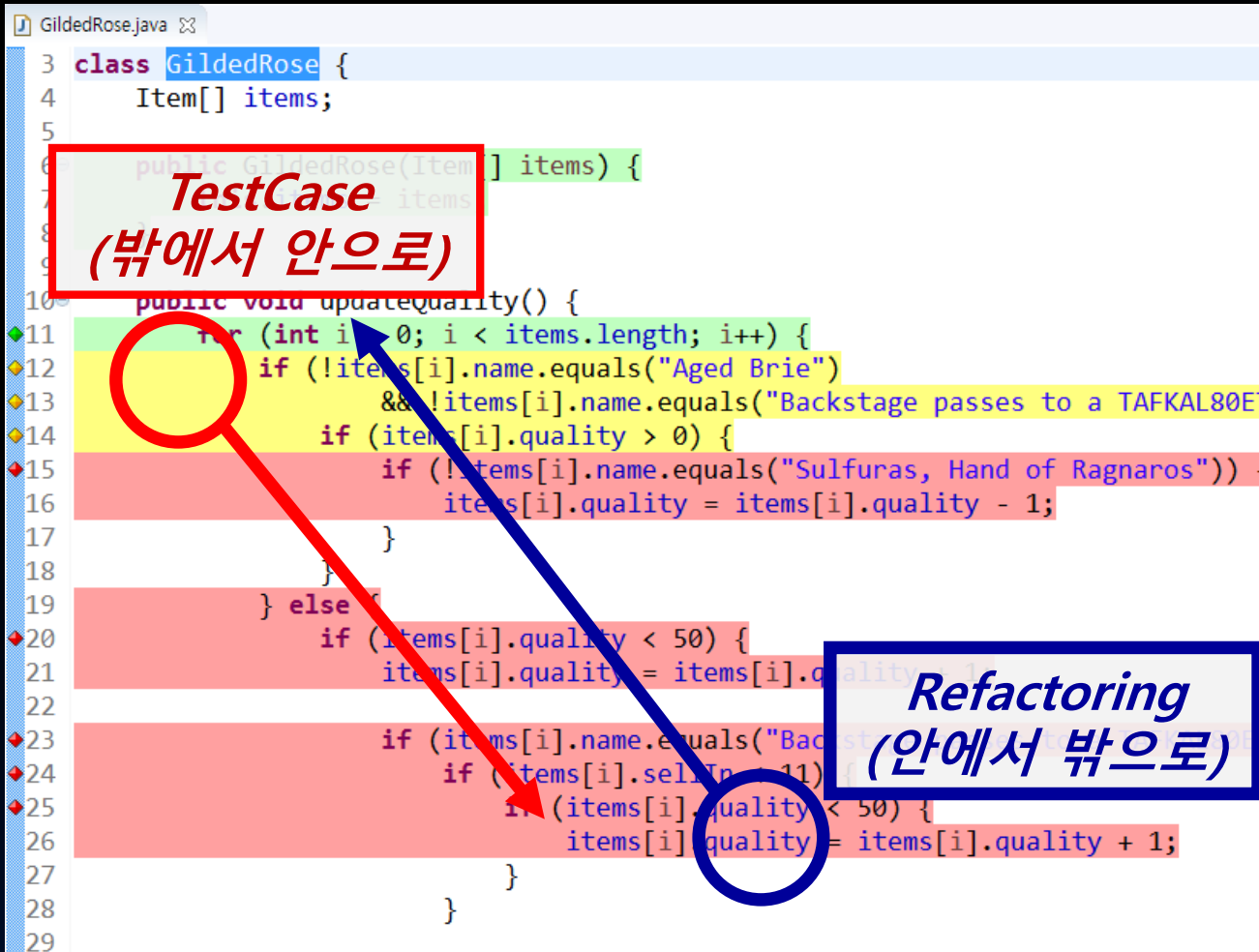
2. Arrange : 조건설정

3. Act : 테스트 할 메소드 실행

4. Assert : 결과 검증

Test Case 소개

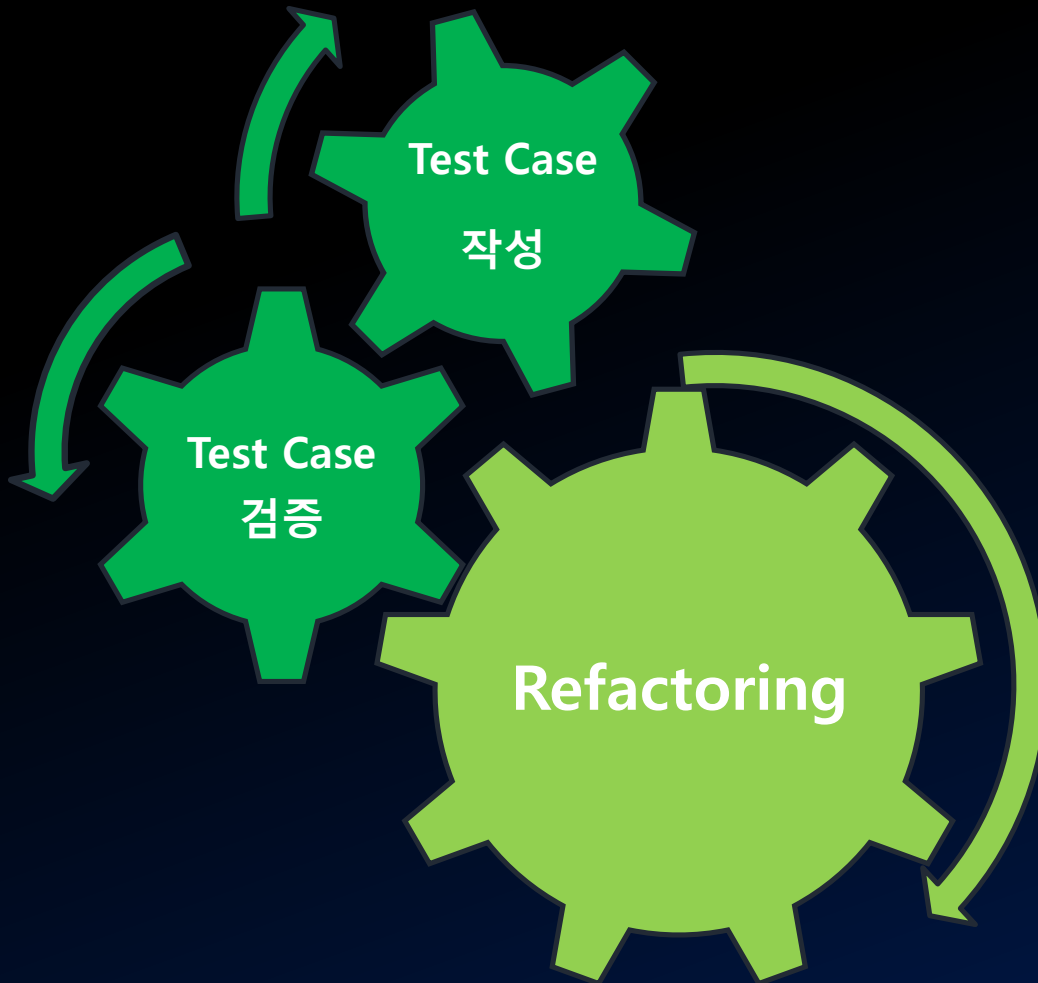
Test Case 작성 순서와 Refactoring 순서



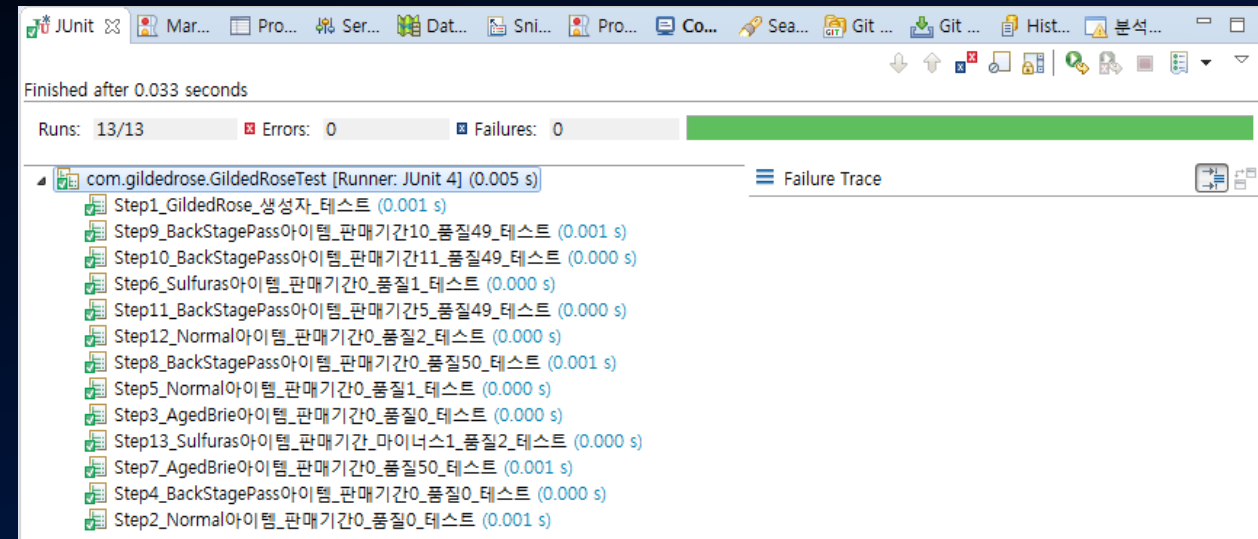
1. Test Case 작성은 밖에서 부터 안으로
(안에서 부터 작성하려면,
그 전까지의 로직을 모두 파악해야 한다.)
2. Refactoring은 안에서부터 밖으로
(기존 업무에 끼치는 영향을
최소화 하면서 검증)

Test Case 소개

Refactoring 후 Test Case로 검증



Refactoring을 하고나면
반드시 Test Case를 실행하여
SW의 기존 동작을 만족시키는지 확인





3. Legacy Code 소개

Gilded Rose

Gilded Rose 는 Allison 이 운영하는 작은 여관입니다.
이 여관은 유명한 도시의 요지에 자리잡고 있습니다.
우리는 이 여관에서 finest goods를 사고 팝니다.

- 상품의 유통기한(sellIn)이 다가올수록 상품들의 **품질(quality)**은 지속적으로 떨어집니다.
- 모든 아이템에는 **유통기한(sellIn)**이 있습니다.
 - 유통기한은 아이템을 팔아야 하는 날까지 남아 있는 일 수로 표시합니다.
- 모든 아이템에는 **품질값(quality)**이 있습니다.
 - 품질값은 아이템이 얼마나 가치가 있는지를 나타냅니다.
- 시스템은 매일 자정에 모든 아이템의 값들을 갱신합니다.

Gilded Rose



Gilded Rose

Gilded Rose 라는 여관에서
판매하는 4개의 상품들에 대한
유통기한과 품질을 관리하는
업무로직을 가진 어플리케이션

Gilded Rose

- 유통기한이 지나면, 품질은 두 배씩 빨리 떨어집니다.
- 아이템의 품질은 음수가 될 수 없습니다.
- 아이템의 품질은 50보다 클 수 없습니다.



- **"Aged Brie"**는 시간이 지날수록 품질이 증가합니다.

- **"Sulfuras"**는 전설의 아이템입니다.
절대 팔지도 않고 품질이 떨어지지도 않습니다.



- **"Backstage passes"** 는 유통기한이 다가올수록 품질이 증가합니다.
유통기한이 10일 이하일 때 품질은 2씩 증가하고,
유통기한이 5일 이하일 때는 3씩 증가합니다.
콘서트가 끝나고 유통기한이 지나면 품질은 0이 됩니다.



Gilded Rose



<Aged Brie>



<Backstage Pass>



<Sulfuras>

NORMAL

<Normal>

코드를 살펴보아요

Item

GildedRose

GildedRoseTest

GildedRoseTest

GildedRoseTest.java

```
1 package com.gildedrose;
2
3 import static org.junit.Assert.*;
4
5 import org.junit.Test;
6
7 public class GildedRoseTest {
8
9     @Test
10     public void Step1_GildedRose_생성자_테스트() {
11
12         // arrange
13         Item[] items= new Item[]{};
14
15         // act
16         GildedRose gildedRose= new GildedRose(items);
17
18         // assert
19         assertNotNull(gildedRose);
20     }
21 }
```

1. '@Test' method annotation

2. Arrange : 조건설정

3. Act : 테스트 할 메소드 실행

4. Assert : 결과 검증

Gilded Rose 메인코드

```
10 public void updateQuality() {  
11     for (int i = 0; i < items.length; i++) {  
12         if (!items[i].name.equals("Aged Brie")) {  
13             if (items[i].name.equals("Backstage passes to a TAFKAL80ETC concert")) {  
14                 if (items[i].quality > 0) {  
15                     if (items[i].name.equals("Sulfuras, Hand of Ragnaros")) {  
16                         items[i].quality = items[i].quality - 1;  
17                     }  
18                 }  
19             } else {  
20                 if (items[i].quality < 50) {  
21                     items[i].quality = items[i].quality + 1;  
22                 }  
23                 if (items[i].name.equals("Backstage passes to a TAFKAL80ETC concert")) {  
24                     if (items[i].sellIn < 11) {  
25                         if (items[i].quality < 50) {  
26                             items[i].quality = items[i].quality + 1;  
27                         }  
28                     }  
29                 }  
30                 if (items[i].sellIn < 6) {  
31                     if (items[i].quality < 50) {  
32                         items[i].quality = items[i].quality + 1;  
33                     }  
34                 }  
35             }  
36         }  
37     }  
38 }  
39 if (!items[i].name.equals("Sulfuras, Hand of Ragnaros")) {
```

복잡한 분기문

부정문

반복사용되는 문자열

Magic Number

역시 어렵다.

코드를 한번 살펴 보시면
if else 문이
아주 뒤죽박죽 섞여 있거든요?

국내 최고의 **복잡도**를 가진
코드라 할 수 있겠어요!

생각해 보아요



<Aged Brie>



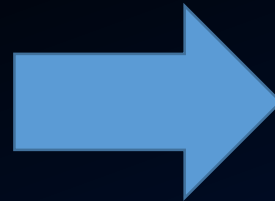
<Backstage Pass>



<Sulfuras>



<Normal>



새로운 아이템이
추가될 경우에
어떤 어려움이
예상될까요?

생각해 보아요

업무별로 분기만 잘 되어 있어도...

분기문이 좀 적었으면...

기존 코드의 동작을 보장하는 TestCase가 있었으면...

리팩토링 전략

GildedRose Refactoring

- 기존로직 작동을 보장하기 위한 T.C. Coverage 100%
- IDE를 활용한 Refactoring
- 추상화를 활용한 Refactoring
- 패턴을 활용한 Refactoring



4. Test case Coverage 100%

Test Case의 효과



1. 수정시의 생산성 향상
2. 버그 잡기가 빨라진다.
3. 버그 잡기가 쉬워진다.
4. 시스템 구조가 좋아진다.
5. 리팩토링의 조건이 된다.
6. 회귀테스트를 제공한다.
7. 하위호환성 보증의 방법을 제공한다.
8. 전체 시스템의 이해 없이 부분의 수정이 가능하다.
9. 샘플로 활용된다.
10. 코드리뷰시에 부담감이 준다.
11. CI가 제대로 활용된다.
12. 설계와 구현을 분리할 수 있다.

**Test Case의 효과중에서
오늘 실습과 관련있는 것중 하나는?**

실습시작~

Pair 프로그래밍으로 진행 (조종사 - 아바타)



조종사



아바타

Test coverage 100% (1/4)



Alt + Shift + E + 잠깐 기다렸다가 + T

단축키로 테스트 성공확인 및

Test Coverage 테스트!

```
8 import org.junit.Test;
9
10 public class GildedRoseTest {
11
12     @Test
13     public void GildedRose_생성자_테스트() {
14         // arrange
15         Item[] items= new Item[]{};
16
17         // act
18         GildedRose gildedRose= new GildedRose(items);
19
20         // assert
21         assertNotNull(gildedRose);
22     }
```



XX 버튼을 눌러서 컬러링을 제거

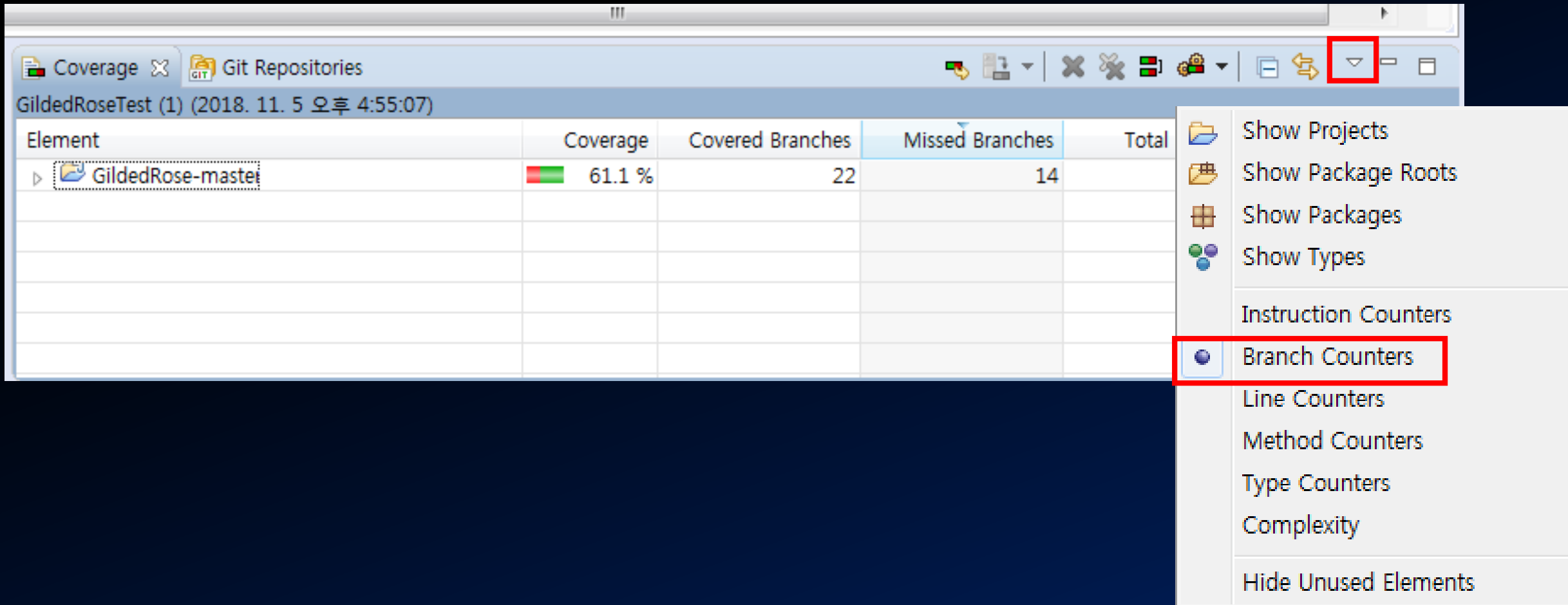
```
items[i].quality < 50) {
    items[i].quality = items[i].quality + 1;

    if (items[i].name.equals("Backstage passes to a TAFKAL80ETC concert")) {
        if (items[i].sellIn < 11) {
            if (items[i].quality < 50) {
                items[i].quality = items[i].quality + 1;
            }
        }
    }
}
```

Element	Coverage	Covered Branches	Missed Branches	Total Branches
GildedRose-master	61.1 %	22	14	36

Test coverage 100% (2/4)

Branch Counter 로 설정되어 있는지 확인



The screenshot shows the IntelliJ IDEA Coverage tool window. The top bar includes tabs for 'Coverage' and 'Git Repositories'. The main area displays a table for 'GildedRoseTest (1) (2018. 11. 5 오후 4:55:07)'. The table has columns for 'Element', 'Coverage', 'Covered Branches', 'Missed Branches', and 'Total'. The 'GildedRose-master' element is listed with a coverage of 61.1%, 22 covered branches, and 14 missed branches. A context menu is open on the right, showing options like 'Show Projects', 'Show Package Roots', 'Show Packages', 'Show Types', 'Instruction Counters', 'Branch Counters' (highlighted with a red box), 'Line Counters', 'Method Counters', 'Type Counters', 'Complexity', and 'Hide Unused Elements'.

Element	Coverage	Covered Branches	Missed Branches	Total
▶ GildedRose-master	61.1 %	22	14	

Test coverage 100% (3/4)

테스트 케이스 추가 (같이 해보기 1)

```
22 @Test
23 public void Step2_Normal아이템_판매기간0_품질0_테스트() {
24
25     // arrange
26     Item[] items= new Item[]{new Item("Normal Item", 0, 0)};
27     GildedRose gildedRose= new GildedRose(items);
28
29     // act
30     gildedRose.updateQuality();
31
32     // assert
33     assertEquals("Normal Item", items[0].name);
34     assertEquals(-1, items[0].sellIn);
35     assertEquals(0, items[0].quality);
36 }
37
```

기대값과 실제값을

assertEquals 메서드를

활용하여 검증

기대값

실제값

Test coverage 100% (3/4)

테스트 케이스 추가 (같이 해보기 2)

```
38 @Test
39 public void Step3_AgedBrie아이템_판매기간0_품질0_테스트() {
40
41     // arrange
42     Item[] items= new Item[]{new Item("Aged Brie", 0, 0)};
43     GildedRose gildedRose= new GildedRose(items);
44
45     // act
46     gildedRose.updateQuality();
47
48     // assert
49     assertEquals("Aged Brie", items[0].name);
50     assertEquals(-1, items[0].sellIn);
51     assertEquals(2, items[0].quality);
52
53 }
54
```

Test coverage 100% (4/4)

Coverage를 100% 만족하도록 Test Case 작성 (10 min.)





5. IDE를 활용한 리팩토링

Constant 처리 (1/23)

⑥ *Alt + Shift + T + a* 단축키 활용

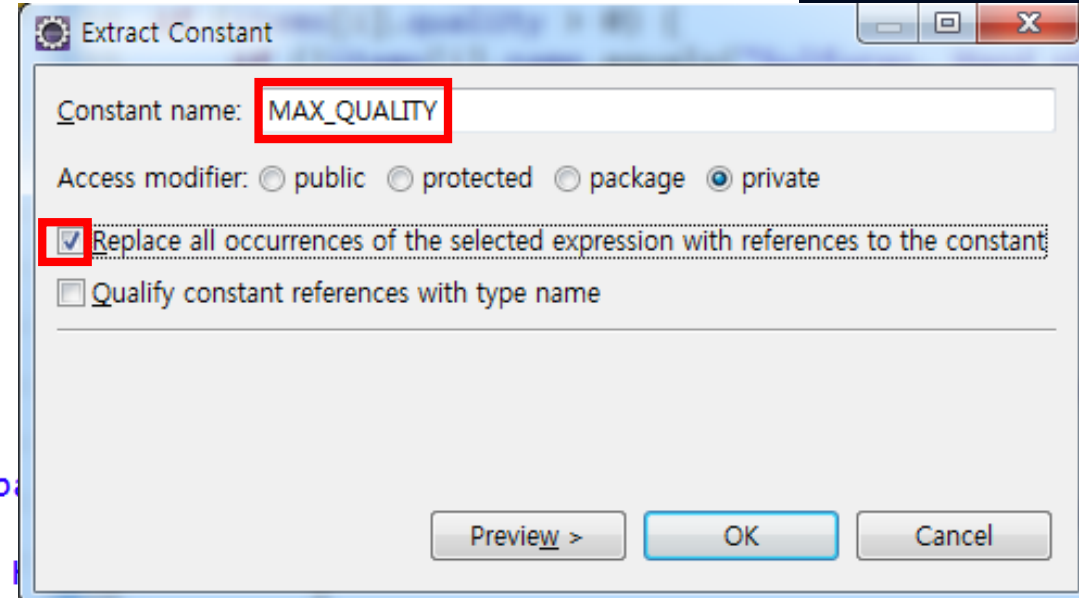


Extract to Constant 클릭하여

"Replace all occurrences" 체크

GildedRose.java

```
1 package com.gildedrose;
2
3 class GildedRose {
4     private static final int MAX_QUALITY = 50;
5     Item[] items;
6
7     public GildedRose(Item[] items) {
8         this.items = items;
9     }
10
11     public void updateQuality() {
12         for (int i = 0; i < items.length; i++) {
13             if (!items[i].name.equals("Aged Brie")
14                 && !items[i].name.equals("Backstage pass"))
15                 if (items[i].quality > 0) {
16                     if (!items[i].name.equals("Sulfuras, Hand of Magnificence"))
17                         items[i].quality = items[i].quality - 1;
18                 }
19             }
20         } else {
21             if (items[i].quality < MAX_QUALITY)
22                 items[i].quality = items[i].quality + 1;
```

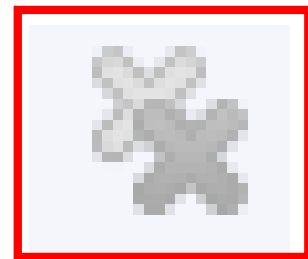


실행내용 컬러링 제거 (2/23)

```
10 public void updateQuality() {
11     for (int i = 0; i < items.length; i++) {
12         if (!items[i].name.equals("Aged Brie")
13             && !items[i].name.equals("Backstage passes to a TAFKAL80ETC concert")) {
14             if (items[i].quality > 0) {
15                 if (!items[i].name.equals("Sulfuras, Hand of Ragnaros")) {
16                     items[i].quality = items[i].quality - 1;
17                 }
18             }
19         } else {
20             if (items[i].quality < 50) {
21                 items[i].quality = items[i].quality + 1;
22             }
23             if (items[i].name.equals("Backstage passes to a TAFKAL80ETC concert")) {
24                 if (items[i].sellIn < 11) {
25                     if (items[i].quality < 50) {
26                         items[i].quality = items[i].quality + 1;
27                     }
28                 }
29             }
30         }
31     }
32 }
```



Xx 버튼을 클릭하여 *eclemma*
실행내용 컬러링 제거




IDE interface showing the bottom toolbar and a table.

Toolbar icons: Console, JUnit, Coverage, Git Repositories, and a red square button with a white 'X' icon.

Element	Coverage	Covered Branches	Missed Branches	Total Branches

부정문 제거해 보기 (3/23)

Aged Brie.... if (!~) 조건 -> invert if statement



```
GildedRose.java
1 package com.gildedrose;
2
3 class GildedRose {
4     private static final int MAX_QUALITY = 50;
5     Item[] items;
6
7     public GildedRose(Item[] items) {
8         this.items = items;
9     }
10
11     public void updateQuality() {
12         for (int i = 0; i < items.length; i++) {
13             if (!items[i].name.equals("Aged Brie"))
14
15             ...
16             for (int i = 0; i < items.length; i++) {
17                 if (items[i].name.equals("Aged Brie") ||
18                     items[i].name.equals("Backstage passes to a TAFKAL80ETC
19                     concert")) {
20                     if (items[i].quality < MAX_QUALITY {
21                         ...
22                     }
23                 }
24             }
25         }
26     }
27 }
```

부정문으로 되어 있는

분기문 괄호 앞에서

Ctrl + 1 단축키 활용하여

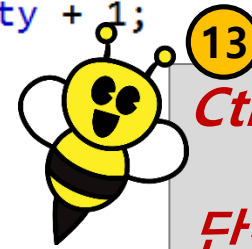
Invert 'If' statement 클릭

```
if (items[i].name.equals("Aged Brie")
    || items[i].name.equals("Backstage passes to a TAFKAL80ETC concert")) {
```

들여쓰기 정리 (4/23)

들여쓰기 정리

```
17         if (items[i].name.equals("Backstage passes to a TAFKAL80ETC concert")) {
18             if (items[i].sellIn < 11) {
19                 if (items[i].quality < MAX_QUALITY) {
20                     items[i].quality = items[i].quality + 1;
21                 }
22             }
23
24             if (items[i].sellIn < 6) {
25                 if (items[i].quality < MAX_QUALITY) {
26                     items[i].quality = items[i].quality + 1;
27                 }
28             }
29         }
30     }
31     } else {
32         if (items[i].quality > 0) {
33             if (!items[i].name.equals("Sulfuras, Hand of Ragnaros")) {
34                 items[i].quality = items[i].quality - 1;
35             }
36         }
37     }
```



Ctrl + Shift + F

**단축키를 활용하여
들여쓰기 정리**

부정문 제거해 보기 (5/23)

invert if statement를 위한 else 추가

```
32         } else {  
33             if (items[i].quality > 0) {  
34                 if (!items[i].name.equals("Sulfuras, Hand of Ragnaros")) {  
35                     items[i].quality = items[i].quality - 1;  
36                     } else {  
37                         }  
38             }  
39     }  
40 }
```



*Invert if statemet 기능을
활용하려면,*

Else 분기문이 있어야 해요



Ctrl + 1

단축키를 활용하여

add else statement

부정문 제거해 보기 (6/23)



부정문으로 되어 있는

분기문 괄호 앞에서

10 Ctrl + 1 단축키 활용하여

Invert 'If' statement 클릭

if (!~) 조건 -> invert if statement

```
32         } else {  
33             if (items[i].quality > 0) {  
34                 if (items[i].name.equals("Sulfuras, Hand of Ragnaros")) {  
35  
36                     } else {  
37                         items[i].quality = items[i].quality - 1;  
38                     }  
39                 }  
40             }
```

부정문 제거해 보기 (7/23)



*Ctrl + F 로 !를 검색했을때
아무것도 검색이 되면 안됩니다*

부정문으로 되어 있는 if 문을 긍정문으로 바꿔보세요 (10 min.)



if 문에 else 문이 없으면

else를 추가해 주셔야

리팩토링 기능을 이용하실 수 있습니다.



부정문으로 되어 있는

분기문 괄호 앞에서

10

Ctrl + 1 단축키 활용하여

Invert 'If' statement 클릭

Constant 처리 (8/23)

"Aged Brie", "Backstage passes...", "Sulfuras,..."

```
GildedRose.java
1 package com.gildedrose;
2
3 class GildedRose {
4     private static final int MAX_QUALITY = 50;
5     Item[] items;
6
7     public GildedRose(Item[] items) {
8         this.items = items;
9     }
10
11     public void updateQuality() {
12         for (int i = 0; i < items.length; i++) {
13             if (items[i].name.equals("Aged Brie")
14                 || items[i].name.equals("Backstage passes")) {
15                 if (items[i].quality < MAX_QUALITY) {
16                     items[i].quality++;
17                 }
18                 if (items[i].name.equals("Sulfuras")) {
19                     items[i].quality = 0;
20                 }
21             }
22         }
23     }
24 }
```



6

Ctrl + 1

또는

Alt + Shift + T + A
단축키를 활용하여
Extract to Constant 실행

- Invert equals
- Extract to local variable (replace all occurrences)
- Extract to local variable
- Extract to constant**
- Put '||' expression in parentheses

Constant 처리 (9/23)

“Backstage passes...”, “Sulfuras...” 를 모두 Constant로 만들어 보세요 (5 min.)



Ctrl + 1

또는

Alt + Shift + T + A

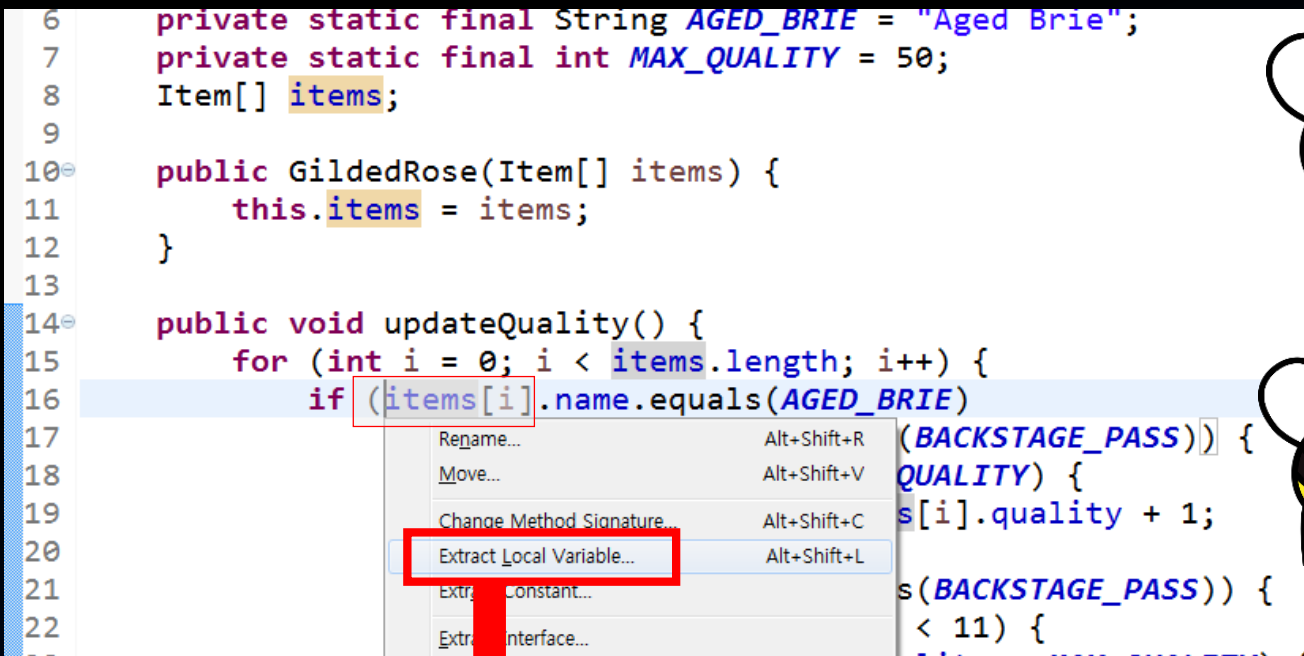
단축키를 활용하여

Extract to Constant 실행

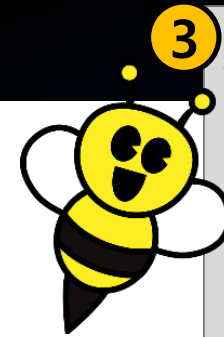
로컬 변수로 처리 (10/23)

items[i] -> Extract Local Variable

```
6 private static final String AGED_BRIE = "Aged Brie";
7 private static final int MAX_QUALITY = 50;
8 Item[] items;
9
10 public GildedRose(Item[] items) {
11     this.items = items;
12 }
13
14 public void updateQuality() {
15     for (int i = 0; i < items.length; i++) {
16         if (items[i].name.equals(AGED_BRIE)
17             || items[i].name.equals(BACKSTAGE_PASS)) {
18             if (items[i].quality < MAX_QUALITY) {
19                 items[i].quality + 1;
20             }
21         }
22     }
23 }
```

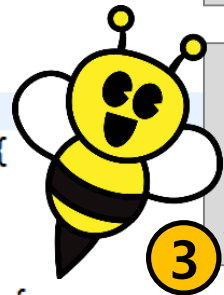


```
public void updateQuality() {
    for (int i = 0; i < items.length; i++) {
        Item item = items[i];
        if (item.name.equals(AGED_BRIE)
            || item.name.equals(BACKSTAGE_PASS)) {
            if (item.quality < MAX_QUALITY) {
                item.quality = item.quality + 1;
            }
        }
    }
}
```



③ Alt + Shift + L 단축키 활용

Extract to local variable 클릭하여
"Replace all occurrences..." 체크




Items[i] 를 선택하고

Alt + Shift + L 단축키를 활용!

Split Condition (11/23)

if (~||~) -> Split Condition

```
14 public void updateQuality() {  
15     for (int i = 0; i < items.length; i++) {  
16         Item item = items[i];  
17         if (item.name.equals(AGED_BRIE)  
18             || item.name.equals(BACKSTAGE_PASS)) {  
19             if (  
20  
21  
22  
23  
24  
25  
26  
27
```



- Exchange left and right operands for infix expression
- Split || condition**
- Extract to local variable (replace all occurrences)
- Extract to local variable
- Put '||' expression in parentheses

10

Ctrl + 1 단축키를 활용하여

Split || Condition 실행

Quiz

QUIZ

```
17     if (item.name.equals(AGED_BRIE)) {
18         if (item.quality < MAX_QUALITY) {
19             item.quality = item.quality + 1;
20         }
21     }
22     if (item.name.equals(BACKSTAGE_PASS)) {
23         if (item.sellIn < 11) {
24             if (item.quality < MAX_QUALITY) {
25                 item.quality = item.quality + 1;
26             }
27         }
28     }
29     if (item.sellIn < 6) {
30         if (item.quality < MAX_QUALITY) {
31             item.quality = item.quality + 1;
32         }
33     }
34 }
35 } else if (item.name.equals(BACKSTAGE_PASS)) {
```



분기문 정리 중에 Coverage가 100%
미만으로 줄어는 이유는?

실행되지 않는 조건 삭제 (12/23)

Spilt에 의해 동일한 block이 2개의 분기로 분리되면서, 실행되지 않는 block이 생김
실행되지 않는 block을 삭제

```
17     if (item.name.equals(AGED_BRIE)) {
18         if (item.quality < MAX_QUALITY) {
19             item.quality = item.quality + 1;
20
21     if (item.name.equals(BACKSTAGE_PASS)) {
22         if (item.sellIn < 11) {
23             if (item.quality < MAX_QUALITY) {
24                 item.quality = item.quality + 1;
25             }
26         }
27
28         if (item.sellIn < 6) {
29             if (item.quality < MAX_QUALITY) {
30                 item.quality = item.quality + 1;
31             }
32         }
33     }
34 }
35 } else if (item.name.equals(BACKSTAGE_PASS)) {
```

동일한 조건에 대하여 조건 삭제 (13/23)

상위 조건(BACKSTAGE_PASSES)와 내부 조건(BACKSTAGE_PASSES)이 같기 때문에 내부 조건 삭제

```
14 public void updateQuality() {  
15     for (int i = 0; i < items.length; i++) {  
16         Item item = items[i];  
17         if (item.name.equals(AGED_BRIE)) {  
18             if (item.quality < MAX_QUALITY) {  
19                 item.quality = item.quality + 1;  
20             }  
21         } else if (item.name.equals(BACKSTAGE_PASS)) {  
22             if (item.quality < MAX_QUALITY) {  
23                 item.quality = item.quality + 1;  
24             }  
25         } if (item.name.equals(BACKSTAGE_PASS)) {  
26             if (item.sellIn < 11) {  
27                 if (item.quality < MAX_QUALITY) {  
28                     item.quality = item.quality + 1;  
29                 }  
30             }  
31         }
```

들여쓰기 정리 (14/23)

```
14 public void updateQuality() {
15     for (int i = 0; i < items.length; i++) {
16         Item item = items[i];
17         if (item.name.equals(AGED_BRIE)) {
18             if (item.quality < MAX_QUALITY) {
19                 item.quality = item.quality + 1;
20             }
21         } else if (item.name.equals(BACKSTAGE_PASS)) {
22             if (item.quality < MAX_QUALITY) {
23                 item.quality = item.quality + 1;
24             }
25             if (item.sellIn < 11) {
26                 if (item.quality < MAX_QUALITY) {
27                     item.quality = item.quality + 1;
28                 }
29             }
30             if (item.sellIn < 6) {
31                 if (item.quality < MAX_QUALITY) {
32                     item.quality = item.quality + 1;
33                 }
34             }
35         }
36     }
37 } else {
38     if (item.quality > 0) {
39         if (item.name.equals(SULFURAS)) {
```



12

Ctrl + Shift + F

단축키를 활용하여

들여쓰기 정리

분기문 정리하기 (15/23)



BIGPICTURE

이름 비교 조건을 상위 단계로 이동

```
36         } else {  
37             if (item.quality > 0) {  
38                 if (item.name.equals(SULFURAS)) {  
39                       
40                 } else {  
41                     item.quality = item.quality - 1;  
42                 }  
43             }  
44         }
```

```
36         } else {  
37             if (item.name.equals(SULFURAS)) {  
38                 if (item.quality > 0) {  
39                       
40                 } else {  
41                     if (item.quality > 0) {  
42                         item.quality = item.quality - 1;  
43                     }  
44                 }  
45             }
```

```
if(AGED_BRIE) {  
  
} else if (BACKSTAGE_PASS)  
  
} else if (SULFURAS) {  
  
} else {  
  
}
```

위와같이 업무기준으로 분기문을 정렬한뒤

-> 메서드 추출

-> 클래스 추출

-> 추상화

-> FactoryPattern 적용

분기문 정리하기 (16/23)

이름 비교 조건을 상위 단계로 이동

```
53     if (item.sellIn < 0) {  
54         if (item.name.equals(AGED_BRIE)) {  
55             if (item.quality < 0) {  
56                 item.quality = item.quality + 1;  
57             }  
58         } else {  
59             if (item.name.equals(BACKSTAGE_PASS)) {
```

```
53     if (item.name.equals(AGED_BRIE)) {  
54         if (item.sellIn < 0) {  
55             if (item.quality < 50) {  
56                 item.quality = item.quality + 1;  
57             }  
58         }  
59     } else if (item.name.equals(BACKSTAGE_PASS)) {  
60         if (item.sellIn < 0) {  
61             item.quality = item.quality - item.quality;  
62         }  
63     } else if (item.name.equals(SULFURAS)) {  
64         if (item.sellIn < 0) {  
65             if (item.quality > 0) {
```

```
Else {  
    if {  
        ....  
    }  
}
```

형태의 분기문을

Else if

형태의 분기문으로 변경하기 위해

10

Ctrl + 1 단축키를 활용하여

Change 'else' block to statement 기능을
활용해 보세요.

분기문 정리하기 (17/23)

BACKSTAGE_PASS, SULFURAS

이름 비교 조건을 상위 단계로 이동 (10 min.) •



```
else {  
  if {  
    ....  
  }  
}
```

형태의 분기문을

Else if

형태의 분기문으로 변경하기 위해

10 Ctrl + 1 단축키를 활용하여

Change 'else' block to statement 기능을
활용해 보세요.

분기문 모으기 (18/23)

같은조건의 2개 if에 대하여 하나로 합침

```
GildedRose.java
52     } else {
53         item.sellIn = item.sellIn - 1;
54     }
55
56     if (item.name.equals(AGED_BRIE)) {
57         if (item.sellIn < 0) {
58             if (item.quality < MAX_QUALITY) {
59                 item.quality = item.quality + 1;
60             }
61         }
62     }
```

```
14 public void updateQuality() {
15     for (int i = 0; i < items.length; i++) {
16         Item item = items[i];
17         if (item.name.equals(AGED_BRIE)) {
18             if (item.quality < MAX_QUALITY) {
19                 item.quality = item.quality + 1;
20             }
21         }
22         if (item.sellIn < 1) {
23             if (item.quality < MAX_QUALITY) {
24                 item.quality = item.quality + 1;
25             }
26         }
27     } else if (item.name.equals(BACKSTAGE_PASS)) {
```

하위 조건(AGED_BRIE)의 내용을 상위
조건(AGED_BRIE)과 묶음

나머지 조건도 합침

(합치면서
sellIn 조건 변경
(0 → 1))



분기문 모으기 (19/23)

같은조건의 2개 if에 대하여 하나로 합침 (10 min.)



하위 조건(AGED_BRIE)의 내용을 상위
조건(AGED_BRIE)과 묶음

나머지 조건도 합침
(합치면서 sellIn 조건 변경 (0 → 1))

(합치면서
sellIn 조건 변경
(0 → 1))



분기문 모으기 (20/23)

하위 if 그룹 삭제

```
75  
76         if (item.name.equals(AGED_BRIE)) {  
77  
78         } else if (item.name.equals(BACKSTAGE_PASS)) {  
79  
80         } else if (item.name.equals(SULFURAS)) {  
81  
82         } else {  
83  
84         }  
85     }  
86 }
```

분기문 모으기 (21/23)

불필요 코드 삭제 (조건 안에 내용 없는 것) -> Extract Method 수행을 위하여 삭제하지 않고 놔 두도록 함

```
GildedRose.java
44
45         item.quality = item.quality - item.quality;
46     }
47     } else if (item.name.equals(SULFURAS)) {
48         if (item.quality > 0) {
49
50         }
51     } else {
```

분기문이 예쁘게 정리 되었습니다.

고급 리팩토링으로 가기 위한 좋은 구조입니다.

```
public void updateQuality() {
    for (int i = 0; i < items.length; i++) {
        Item item = items[i];
        if (item.name.equals(AGED_BRIE)) {
            if (item.quality < MAX_QUALITY) {
                item.quality = item.quality + 1;
            }
            if (item.sellIn < 1) {
                if (item.quality < MAX_QUALITY) {
                    item.quality = item.quality + 1;
                }
            }
        } else if (item.name.equals(BACKSTAGE_PASS)) {
            if (item.quality < MAX_QUALITY) {
                item.quality = item.quality + 1;
            }
            if (item.sellIn < 11) {
                if (item.quality < MAX_QUALITY) {
                    item.quality = item.quality + 1;
                }
            }
        }
    }
}
```

```
    } else if (item.name.equals(SULFURAS)) {
    } else {
        if (item.quality > 0) {
            item.quality = item.quality - 1;
        }
    }
}
```


각 조건별로 Extract Method (22/23)

메서드 추출

GildedRose.java

```
12     }
13
14     public void updateQuality() {
15         for (int i = 0; i < items.length; i++) {
16             Item item = items[i];
17             if (item.name.equals(AGED BRIE)) {
18                 updateQualityForAgedBrie(item);
19             } else if (item.name.equals(BACKSTAGE PASS)) {
20                 updateQualityForBackstagePass(item);
21             } else if (item.name.equals(SULFURAS)) {
22                 //메서드 생성을 위한 임시코드 작성후
23                 updateQualityForSulfuras(item);
24             } else {
25                 updateQualityForNormal(item);
26             }
27         }
28     }
29 }
```



Alt + Shift + M

단축키를 활용하여 메서드 추출



Sulfuras 아이템의 경우

임시코드를 작성하고

Alt + Shift + C 단축키 활용하여

Item 파라미터 추가

각 조건별로 Extract Method (23/23)

메서드 추출 (5 min.)



Alt + Shift + M

단축키를 활용하여 메서드 추출

IDE를 활용한 리팩토링 완료





6. 추상화를 활용한 리팩토링

재설계 Refactoring, AgedBrieItem 생성 (1/21)

생성자로 item 객체 넘김



AgedBrieItem agedBrieItem = new AgedBrieItem(item);

타이핑 후, 이클립스 assist 기능 활용

```
14 public void updateQuality() {
15     for (int i = 0; i < items.length; i++) {
16         Item item = items[i];
17         if (item.name.equals(AGED BRIE)) {
18             AgedBrieItem agedBrieItem = new AgedBrieItem(item);
19             updateQualityForAgedBrie(agedBrieItem);
20         }
21     }
22 }
23
24
25
```

AgedBrieItem cannot be resolved to a type

6 quick fixes available:

- Create class 'AgedBrieItem'
- Create interface 'AgedBrieItem'
- Create enum 'AgedBrieItem'
- Add type parameter 'AgedBrieItem' to 'GildedRose'
- Add type parameter 'AgedBrieItem' to 'updateQuality()'
- Fix project setup...

The constructor AgedBrieItem(Item) is undefined

2 quick fixes available:

- Remove argument to match 'AgedBrieItem()'
- Create constructor 'AgedBrieItem(Item)'

Press 'F2' for focus

재설계 Refactoring, updateQuality* 메소드 AgedBriemtem 파라미터 추가 (2/21)

Move Method를 위한 준비



Alt + Shift + C **2**

단축키를 활용하여

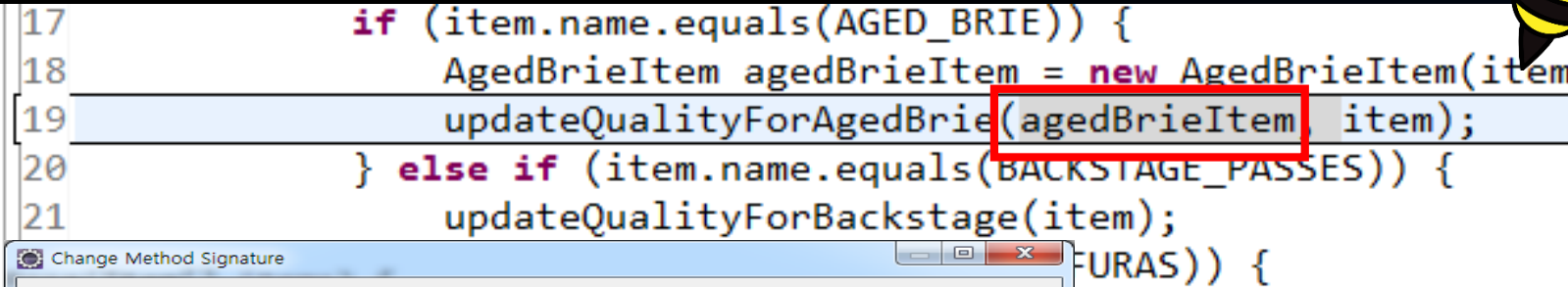
Change Method Signature 기능으로 파라미터 추가

추가하는 파라미터는

updateQualityForAgedBrie

메서드를 AgedBrieltem

클래스로 이동시키기 위한 연결 정보



The screenshot shows the 'Change Method Signature' dialog box. The 'Method name' field contains 'updateQualityForAgedBrie'. The 'Access modifier' is set to 'private' and the 'Return type' is 'void'. The 'Parameters' tab is active, showing a table with two parameters: 'AgedBrieItem' (Name: agedBrieItem, Default value: agedBrieItem) and 'Item' (Name: item, Default value: -). The 'Add' button is highlighted with a red box. The 'Method signature preview' at the bottom shows the code: `private void updateQualityForAgedBrie(AgedBrieItem agedBrieItem, Item item)`.

재설계 Refactoring, AgedBrieItem 생성자 처리 (3/21)

```
GildedRose.java  AgedBrieItem.java  ⌕
1 package com.gildedrose;
2
3 public class AgedBrieItem {
4
5     private Item item;
6
7     public AgedBrieItem(Item item) {
8         this.item= item;
9     }
10
11 }
```


재설계 Refactoring, updateQuality* -> 메서드 이동 (4/21)

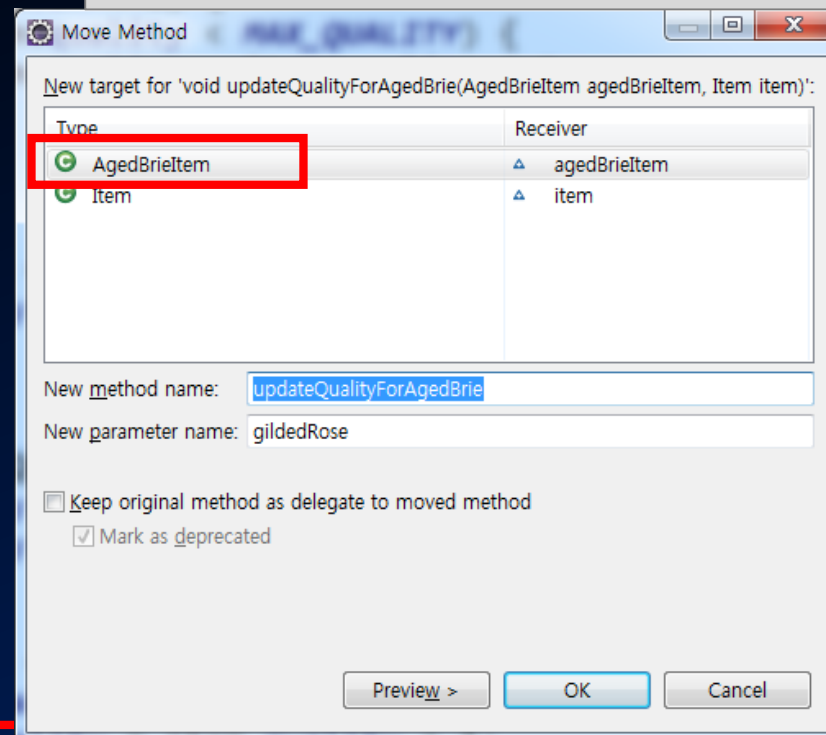
```
70 private void updateQualityForAgedBrie(AgedBrieItem agedBrieItem, Item item) {  
71     if (item.quality < MAX_QUALITY) {  
72         item.quality = item.quality + 1;  
73     }
```

Alt + Shift + V **8**

단축키를 활용하여

Move Method 기능실행

```
3 public class AgedBrieItem {  
4  
5     private Item item;  
6  
7     public AgedBrieItem(Item item) {  
8         this.item = item;  
9     }  
10  
11 void updateQualityForAgedBrie(GildedRose gildedRose, Item item) {  
12     if (item.quality < GildedRose.MAX_QUALITY) {  
13         item.quality = item.quality + 1;  
14     }  
15  
16     if (item.sellIn < 1) {  
17         if (item.quality < GildedRose.MAX_QUALITY) {  
18             item.quality = item.quality + 1;  
19         }  
20     }  
21 }  
22 }
```



재설계 Refactoring, updateQuality* -> 불필요 파라미터 삭제 (5/21)

```
3 public class AgedBrieItem {  
4  
5     private Item item;  
6  
7     public AgedBrieItem(Item item) {  
8         this.item = item;  
9     }  
10  
11     void updateQualityForAgedBrie(Item item, GildedRose gildedRose) {  
12         if (item.quality < GildedRose.MAX_QUALITY) {  
13             item.quality = item.quality + 1;  
14         }  
15  
16         if (item.sellIn < 1) {  
17             if (item.quality < GildedRose.MAX_QUALITY) {  
18                 item.quality = item.quality + 1;  
19             }  
20         }  
    }
```



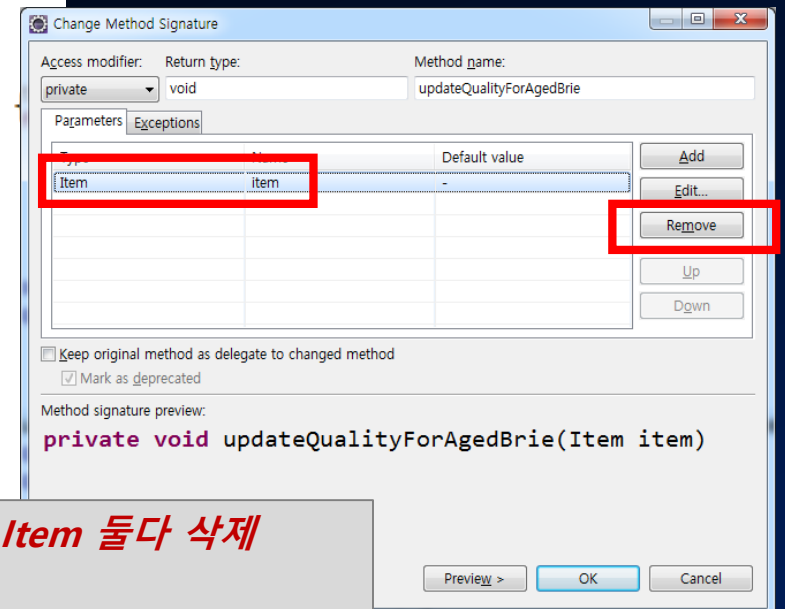
Alt + Shift + C

2

단축키를 활용하여

Change Method Signature 기능 실행하여

파라미터 삭제



GildedRose, Item 둘다 삭제

재설계 Refactoring, 객체생성 (6/21)

BackstagePassItem, SulfurasItem, NormalItem 클래스를 만들어 보세요. (15 min.)



updateQualityForSulfuras 메서드가

파라미터 없다면,

Item 파라미터

8

추가 후 Move Method (Alt + Shift + V)



Alt + Shift + C 2

단축키를 활용하여

Change Method Signature 기능실행하여

파라미터 추가 및 삭제

재설계 Refactoring, 메서드명 통일 (7/21)

AgedBriemItem, BackstagePassItem, SulfurasItem, NormalItem 클래스 모두 수정

```
3 public class AgedBriemItem {  
4  
5     private Item item;  
6  
7     public AgedBriemItem(Item item) {  
8         this.item = item;  
9     }  
10  
11     void updateQuality() {  
12         if (item.quality < GildedRose.MAX_QUALITY) {  
13             item.quality = item.quality + 1;  
14         }  
15  
16         if (item.sellIn < 1) {  
17             if (item.quality < GildedRose.MAX_QUALITY) {  
18                 item.quality = item.quality + 1;  
19             }  
20         }  
21     }  
22 }
```



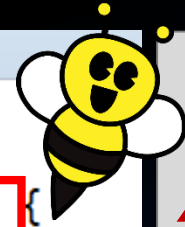
Alt + Shift + R ⑤

단축키를 활용하여

메서드명 Rename

재설계 Refactoring, extract Super class (8/21)

```
GildedRose.java  AgedBrieItem.java  GildedRoseItem.java
1 package com.gildedrose;
2
3 public class AgedBrieItem extends GildedRoseItem {
4
5     private Item item;
6
7     public AgedBrieItem(Item item) {
8         this.item = item;
9     }
10
11     void updateQuality() {
12         if (item.quality < GildedRose.MAX_QUALITY)
13             item.quality = item.quality + 1;
14     }
15 }
```

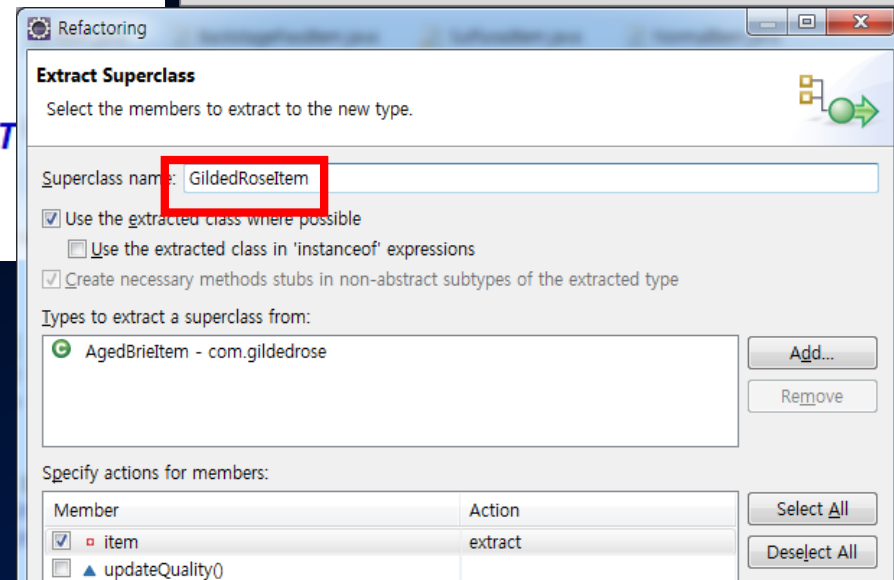


`private Item item;` 선택후

Alt + Shift + T + T ⑦

단축키를 활용하여

Extract Super Class 활용



- Item를 포함하여 상위 클래스 생성

재설계 Refactoring, 생성자 수정 (9/21)

AgedBrieItem, SulfurasItem 같이 실습

```
3 public class AgedBrieItem extends GildedRoseItem {  
4  
5 public AgedBrieItem(Item item) {  
6     super(item);  
7 }  
8
```

```
3 public class GildedRoseItem {  
4  
5     protected Item item;  
6  
7 public GildedRoseItem(Item item) {  
8     this.item= item;  
9 }  
10 }
```


재설계 Refactoring, 생성자 수정 (10/21)

BackstagePassItem, NormalItem 클래스가
GildedRoseItem 을 확장하도록 만들어 보세요. (15 min.)



재설계 Refactoring, abstract 처리 (11/21)

@Override 추가 후, Quick Fix 활용,

AgedBrieItem, SulfurasItem 같이 실습

```
3 public class AgedBrieItem extends GildedRoseItem {
4
5     public AgedBrieItem(Item item) {
6         super(item);
7     }
8
9     @Override
10    void updateQuality() {
11        if (item.quality < GildedRose.MAX_QUALITY) {
12            item.quality = item.quality + 1;
13        }
14
15        if (item.sellIn < 1) {
16            if (item.quality < GildedRose.MAX_QUALITY) {
17                item.quality = item.quality + 1;
18            }
19        }
20    }
21 }
```

```
3 public abstract class GildedRoseItem {
4
5     protected Item item;
6
7     public GildedRoseItem(Item item) {
8         this.item = item;
9     }
10
11    abstract public void updateQuality();
12 }
13
```



@Override 추가 후,
Quick Fix 활용

```
9 @Override
10 void updateQuality() {
11     if (item.quality < GildedRose.MAX_QUALITY) {
12         item.quality = item.quality + 1;
13     }
14
15     if (item.sellIn < 1) {
16         if (item.quality < GildedRose.MAX_QUALITY) {
17             item.quality = item.quality + 1;
18         }
19     }
20 }
```

The method updateQuality() of type AgedBrieItem must override a superclass method
2 quick fixes available:
Create 'updateQuality()' in super type 'GildedRoseItem'
Remove '@Override' annotation
Press 'F2' for focus



GildedRoseItem 클래스를 abstract 처리,
updateQuality 메서드를 abstract 처리

재설계 Refactoring, asbstract 처리 (12/21)

BackstagePassItem, NormalItem 클래스가
GildedRoseItem 클래스의 updateQuality 메서드를 Override 하도록 수정. (5 min.)



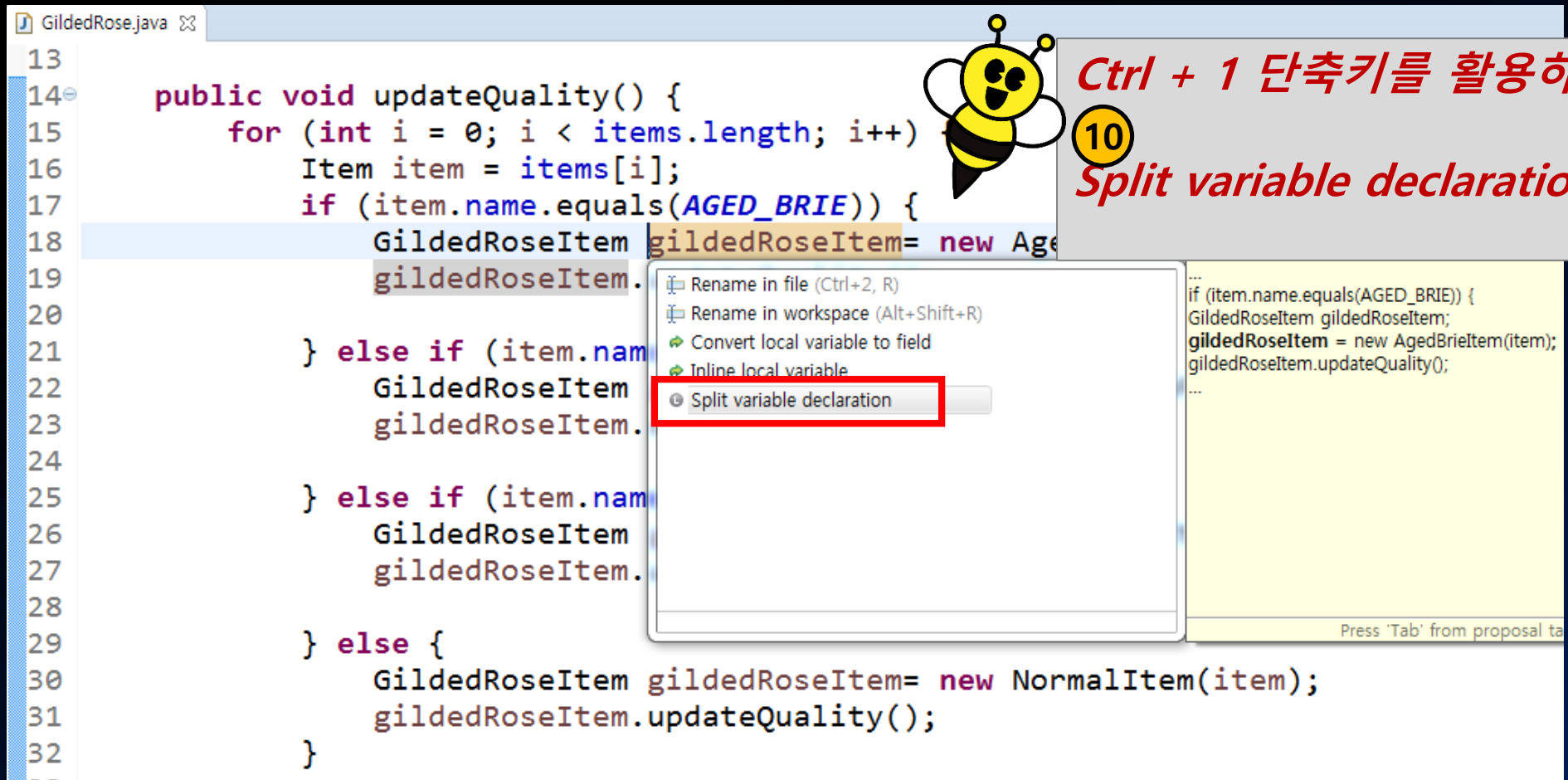
재설계 Refactoring, 기본 if 부분 Superclass로 변경 (13/21)

```
GildedRose.java
13
14 public void updateQuality() {
15     for (int i = 0; i < items.length; i++) {
16         Item item = items[i];
17         if (item.name.equals(AGED_BRIE)) {
18             GildedRoseItem agedBrieItem= new AgedBrieItem(item);
19             agedBrieItem.updateQuality();
20
21         } else if (item.name.equals(BACKSTAGE_PASS)) {
22             GildedRoseItem backstagePassItem= new BackstagePassItem(item);
23             backstagePassItem.updateQuality();
24
25         } else if (item.name.equals(SULFURAS)) {
26             GildedRoseItem sulfurasItem= new SulfurasItem(item);
27             sulfurasItem.updateQuality();
28
29         } else {
30             GildedRoseItem normalItem= new NormalItem(item);
31             normalItem.updateQuality();
32     }
```

재설계 Refactoring, 변수명 통일 (14/21)

```
GildedRose.java  ✖
13
14 public void updateQuality() {
15     for (int i = 0; i < items.length; i++) {
16         Item item = items[i];
17         if (item.name.equals(AGED_BRIE)) {
18             GildedRoseItem gildedRoseItem = new AgedBrieItem(item);
19             gildedRoseItem.updateQuality();
20
21         } else if (item.name.equals(BACKSTAGE_PASS)) {
22             GildedRoseItem gildedRoseItem = new BackstagePassItem(item);
23             gildedRoseItem.updateQuality();
24
25         } else if (item.name.equals(SULFURAS)) {
26             GildedRoseItem gildedRoseItem = new SulfurasItem(item);
27             gildedRoseItem.updateQuality();
28
29         } else {
30             GildedRoseItem gildedRoseItem = new NormalItem(item);
31             gildedRoseItem.updateQuality();
32         }
33     }
```

재설계 Refactoring, Split variable declaration (15/21)



```
13
14 public void updateQuality() {
15     for (int i = 0; i < items.length; i++) {
16         Item item = items[i];
17         if (item.name.equals(AGED_BRIE)) {
18             GildedRoseItem gildedRoseItem = new AgedBrieItem(item);
19             gildedRoseItem.updateQuality();
20         } else if (item.name.equals(BRIE)) {
21             GildedRoseItem gildedRoseItem = new BrieItem(item);
22             gildedRoseItem.updateQuality();
23         } else if (item.name.equals(NORMAL_ITEM)) {
24             GildedRoseItem gildedRoseItem = new NormalItem(item);
25             gildedRoseItem.updateQuality();
26         }
27     }
28 }
29
30
31
32
```

Ctrl + 1 단축키를 활용하여
10 Split variable declaration 실행

if (item.name.equals(AGED_BRIE)) {
GildedRoseItem gildedRoseItem;
gildedRoseItem = new AgedBrieItem(item);
gildedRoseItem.updateQuality();
}

Press 'Tab' from proposal ta

재설계 Refactoring, 중복 코드 제거 (선언 상단, update 하단) (16/21)

```
GildedRose.java
13
14 public void updateQuality() {
15     for (int i = 0; i < items.length; i++) {
16         Item item = items[i];
17         GildedRoseItem gildedRoseItem;
18         if (item.name.equals(AGED_BRIE)) {
19             gildedRoseItem = new AgedBrieItem(item);
20             gildedRoseItem.updateQuality();
21
22         } else if (item.name.equals(BACKSTAGE_PASS)) {
23             gildedRoseItem = new BackstagePassItem(item);
24             gildedRoseItem.updateQuality();
25
26         } else if (item.name.equals(SULFURAS)) {
27             gildedRoseItem = new SulfurasItem(item);
28             gildedRoseItem.updateQuality();
29
30         } else {
31             gildedRoseItem = new NormalItem(item);
32             gildedRoseItem.updateQuality();
33     }
```


재설계 Refactoring, if 및 update 부분 Extract Method (17/21)

```
8 Item[] items;
9
10 public GildedRose(Item[] items) {
11     this.items = items;
12 }
13
14 public void updateQuality() {
15     for (int i = 0; i < items.length; i++) {
16         Item item = items[i];
17         updateQuality(item);
18
19         if (item.name.equals(SULFURAS)) {
20
21         } else {
22             item.sellIn = item.sellIn - 1;
23         }
24     }
25 }
26
27 private void updateQuality(Item item) {
28     GildedRoseItem gildedRoseItem;
29
30     if (item.name.equals(AGED_BRIE)) {
31         gildedRoseItem = new AgedBrieItem(item);
32         gildedRoseItem.updateQuality();
33
34     } else if (item.name.equals(BACKSTAGE_PASS)) {
35         gildedRoseItem = new BackstagePassItem(item);
```

```
8 Item[] items;
9
10 public GildedRose(Item[] items) {
11     this.items = items;
12 }
13
14 public void updateQuality() {
15     for (int i = 0; i < items.length; i++) {
16         Item item = items[i];
17         GildedRoseItem gildedRoseItem;
18
19         if (item.name.equals(AGED_BRIE)) {
20             gildedRoseItem = new AgedBrieItem(item);
21             gildedRoseItem.updateQuality();
22
23         } else if (item.name.equals(BACKSTAGE_PASS)) {
24             gildedRoseItem = new BackstagePassItem(item);
25             gildedRoseItem.updateQuality();
26
27         } else if (item.name.equals(SULFURAS)) {
28             gildedRoseItem = new SulfurasItem(item);
29             gildedRoseItem.updateQuality();
30
31         } else {
32             gildedRoseItem = new NormalItem(item);
33             gildedRoseItem.updateQuality();
34         }
35     }
```



Alt + Shift + M 단축키를 활용하여

Extract Method 실행

함수명 : updateQuality

재설계 Refactoring, 공통로직 추출, gildedRoseItem.updateQuality(); (18/21)

```
27 private void updateQuality(Item item) {  
28     GildedRoseItem gildedRoseItem;  
29  
30     if (item.name.equals(AGED_BRIE)) {  
31         gildedRoseItem = new AgedBrie(item);  
32     } else if (item.name.equals(BACKSLASH_ITEM)) {  
33         gildedRoseItem = new BackslashItem(item);  
34     } else if (item.name.equals(SULFURAS_ITEM)) {  
35         gildedRoseItem = new SulfurasItem(item);  
36     } else {  
37         gildedRoseItem = new NormalItem(item);  
38     }  
39  
40     gildedRoseItem.updateQuality();  
41 }  
42  
43  
44 }
```



Ctrl + D 로 선택된 라인 삭제

범위 선택후

Alt + ↑, ↓ 방향으로 선택된 구문 이동

재설계 Refactoring, if 부분 Extract Method (create) (19/21)

```
27 private void updateQuality(Item item) {  
28     GildedRoseItem gildedRoseItem;  
29     gildedRoseItem = createGildedRoseItem(item);  
30     gildedRoseItem.updateQuality();  
31 }  
32  
33 private GildedRoseItem createGildedRoseItem(Item item) {  
34     GildedRoseItem gildedRoseItem;  
35     if (item.name.equals(AGED_BRIE)) {  
36         gildedRoseItem = new AgedBrieItem(item);  
37  
38     } else if (item.name.equals(BACKSTAGE_AGE)) {  
39         gildedRoseItem = new BackstagePassItem(item);  
40  
41     } else if (item.name.equals(SULFURAS)) {  
42         gildedRoseItem = new SulfurasItem(item);  
43  
44     } else {  
45         gildedRoseItem = new NormalItem(item);  
46     }  
47     return gildedRoseItem;  
48 }
```

4

Alt + Shift + M

단축키를 활용하여 메서드 추출

함수명 : createGildedRoseItem

재설계 Refactoring, sellIn 처리 부분 Extract Method (20/21)

```
14 public void updateQuality() {  
15     for (int i = 0; i < items.length; i++) {  
16         Item item = items[i];  
17         updateQuality(item);  
18         updateSellIn(item);  
19     }  
20 }  
21  
22 private void updateSellIn(Item item) {  
23     if (item.name.equals(SULFURAS)) {  
24  
25     } else {  
26         item.sellIn = item.sellIn - 1;  
27     }  
28 }
```



Alt + Shift + M

단축키를 활용하여 메서드 추출

함수명 : updateSellIn

재설계 Refactoring, Join Variable declaration (21/21)

```
private void updateQuality(Item item) {  
    GildedRoseItem gildedRoseItem;  
    gildedRoseItem = createGildedRoseItem(item);  
}
```

Rename in file (Ctrl+2, R)
Rename in workspace (Alt+Shift+R)
Convert local variable to field
Join variable declaration
Extract to local variable (replace all occurrences)
Extract to local variable

Link all references for a local name (does not change references in other files)



Ctrl + 1

단축키를 활용하여

Join variable declaration 실행

```
30 private void updateQuality(Item item) {  
31     GildedRoseItem gildedRoseItem = createGildedRoseItem(item);  
32     gildedRoseItem.updateQuality();  
33 }  
34
```

추상화를 활용한 리팩토링 완료





7. 패턴을 활용한 리팩토링

재설계 Refactoring, Factory 생성 (1/10)

```
30 private void updateQuality(Item item) {  
31     GildedRoseFactory gildedRoseFactory= new GildedRoseFactory();  
32     GildedRoseItem gildedRoseItem = createGildedRoseItem(item);  
33     gildedRoseItem.updateQuality();  
34 }
```



GildedRoseFactory 정의 후

Eclipse Create Class 기능으로

GildeRoseFactory 클래스 생성

재설계 Refactoring, Signature 변경 및 Move 메소드 (2/10)

create* 메소드를 Move → GildedRoseFactory 클래스로

```
30 private void updateQuality(Item item) {  
31     GildedRoseFactory gildedRoseFactory= new GildedRoseFactory();  
32     GildedRoseItem gildedRoseItem = createGildedRoseItem(gildedRoseFactory, item);  
33     gildedRoseItem.updateQuality();  
34 }  
35  
36 private GildedRoseItem createGildedRoseItem(GildedRoseFactory gildedRoseFactory, Item  
37     GildedRoseItem gildedRoseItem;  
38     if (item.name.equals(AGED_BRIE)) {  
39         gildedRoseItem = new AgedBrieItem(  
40
```



2

Alt + Shift + C

단축키를 활용하여

Change Method Signature 기능실행하여

파라미터 추가

재설계 Refactoring, Signature 변경 (Factory 파라미터 삭제) (3/10)

```
30 private void updateQuality(Item item) {  
31     GildedRoseFactory gildedRoseFactory= new GildedRoseFactory();  
32     GildedRoseItem gildedRoseItem = createGildedRoseItem(gildedRoseFactory, item);  
33     gildedRoseItem.updateQuality();  
34 }
```

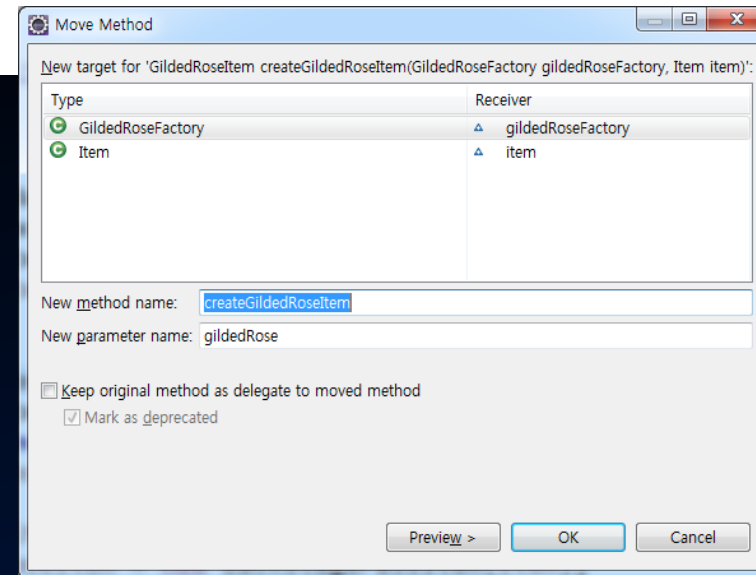


Alt + Shift + C ②

단축키를 활용하여

Move Method 기능실행

GildedRoseItem 삭제



```
30 private void updateQuality(Item item) {  
31     GildedRoseFactory gildedRoseFactory= new GildedRoseFactory();  
32     GildedRoseItem gildedRoseItem = gildedRoseFactory.createGildedRoseItem(this, item);  
33     gildedRoseItem.updateQuality();  
34 }
```

재설계 Refactoring, Factory Singleton 적용 (4/10)

```
3 public class GildedRoseFactory {
4
5     private static GildedRoseFactory instance;
6
7     private GildedRoseFactory() {
8
9     }
10
11     public static synchronized GildedRoseFactory getInstance() {
12         if(instance == null) {
13             instance= new GildedRoseFactory();
14         }
15
16         return instance;
17     }
18
19     GildedRoseItem createGildedRoseItem(Item item) {
20         GildedRoseItem gildedRoseItem;
```

재설계 Refactoring, Inline Local Variable (5/10)

GildedRose 클래스 updateQuality(Item item) 메서드를 정리

```
30 private void updateQuality(Item item) {  
31     GildedRoseItem gildedRoseItem = GildedRoseFactory.getInstance().createGildedRoseItem(item);  
32     gildedRoseItem.updateQuality();  
33 }  
34 }
```

재설계 Refactoring, updateSellIn 메소드 Signature 변경 (super class : GildedRoseItem) (6/10)

```
14 public void updateQuality() {  
15     for (int i = 0; i < items.length; i++) {  
16         Item item = items[i];  
17         updateQuality(item);  
18         updateSellIn(item);  
19     }  
20 }  
21  
22 private void updateSellIn(Item item) {  
23     if (item.name.equals(SULFURAS)) {
```

```
13  
14 public void updateQuality() {  
15     for (int i = 0; i < items.length; i++) {  
16         Item item = items[i];  
17         updateQuality(item);  
18  
19         GildedRoseItem gildedRoseItem= GildedRoseFactory.getInstance().createGildedRoseItem(item);  
20  
21         updateSellIn(gildedRoseItem, item);  
22     }  
23 }  
24  
25 private void updateSellIn(GildedRoseItem gildedRoseItem, Item item) {  
26     if (item.name.equals(SULFURAS)) {
```



Alt + Shift + C ②

단축키를 활용하여

Change Method Signature 기능실행하여

파라미터 추가

- 추가된 파라미터에는 Factory로 부터 생성된 객체를 넘김

재설계 Refactoring, updateSellIn Move (7/10)

```
25 private void updateSellIn(GildedRoseItem gildedRoseItem, Item item) {  
26     if (item.name.equals(SULFURAS)) {  
27  
28     } else {  
29         item.sellIn = item.sellIn - 1;  
30     }  
31 }
```

```
3 public abstract class GildedRoseItem {  
4  
5     protected Item item;  
6  
7     public GildedRoseItem(Item item) {  
8         this.item = item;  
9     }  
10  
11     abstract public void updateQuality();  
12  
13     void updateSellIn(GildedRose gildedRose, Item item) {  
14         if (item.name.equals(GildedRose.SULFURAS)) {  
15  
16         } else {  
17             item.sellIn = item.sellIn - 1;  
18         }  
19     }  
20 }
```



Alt + Shift + V ⑧

단축키를 활용하여

GildedRose 클래스에서, GildedRoseItem 클래스로

Move Method

재설계 Refactoring, 파라미터 삭제 (Change Signature) (8/10)

Factory 클래스에서 Item 생성처리 및 구현클래스 생성을 완료 했으므로,
updateSellIn의 파라미터 제거 (2개 모두 삭제)

```
14 public void updateQuality() {  
15     for (int i = 0; i < items.length; i++) {  
16         Item item = items[i];  
17         updateQuality(item);  
18  
19         GildedRoseItem gildedRoseItem= GildedRoseFact  
20  
21         gildedRoseItem.updateSellIn();  
22     }  
23 }
```

```
14 public void updateQuality() {  
15     for (int i = 0; i < items.length; i++) {  
16         Item item = items[i];  
17         updateQuality(item);  
18  
19         GildedRoseItem gildedRoseItem= GildedRose  
20  
21         gildedRoseItem.updateSellIn(this, item);  
22     }  
23 }
```



Alt + Shift + C ②

단축키를 활용하여

Change Method Signature 기능실행하여

파라미터 제거

재설계 Refactoring, updateQuality 호출 수정 (9/10)

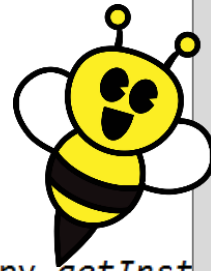
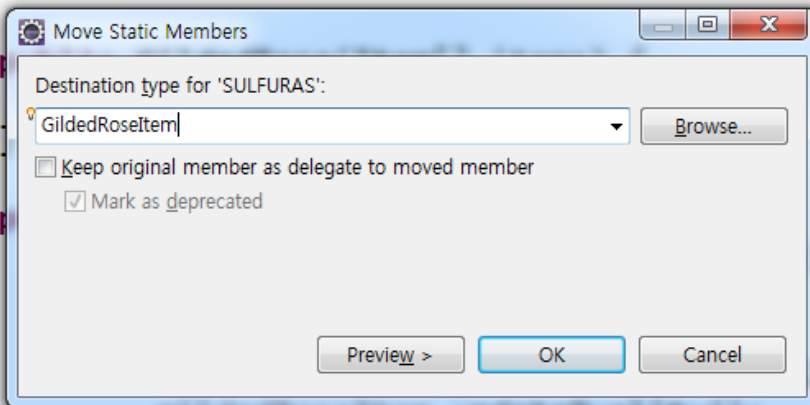
- `updateQuality(item) -> gildedRoseItem.updateQuality()`
- 하단 `updateQuality(Item)` 메소드는 삭제

```
14 public void updateQuality() {  
15     for (int i = 0; i < items.length; i++) {  
16         Item item = items[i];  
17  
18         GildedRoseItem gildedRoseItem= GildedRoseFact  
19         updateQuality(item);  
20         gildedRoseItem.updateSellIn()  
21     }  
22 }
```

```
14 public void updateQuality() {  
15     for (int i = 0; i < items.length; i++)  
16         Item item = items[i];  
17  
18     GildedRoseItem gildedRoseItem= GildedRoseFactory.getInstance().createGildedRoseItem(item);  
19     gildedRoseItem.updateQuality();  
20     gildedRoseItem.updateSellIn();  
21 }  
22  
23 }
```

재설계 Refactoring, 상수 이동 (GildedRose -> GildedRoseItem) (10/10)

```
3 class GildedRose {  
4     static final String SULFURAS = "Sulfuras, Hand of Ragnaros";  
5     static final String BACKSTAGE_PASS = "Backstage passes to a TAFKAL80ETC concert";  
6     static final String AGED_BRIE = "Aged Brie";  
7     static final int MAX_QUALITY = 50;  
8     Item[] items;  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19
```



Alt + Shift + V ⑧

단축키를 활용하여

GildedRose 클래스에서,

GildedRoseItem 클래스로 상수 이동

```
3 public abstract class GildedRoseItem {  
4  
5     protected Item item;  
6     static final int MAX_QUALITY = 50;  
7     static final String AGED_BRIE = "Aged Brie";  
8     static final String BACKSTAGE_PASS = "Backstage passes to a TAFKAL80ETC concert";  
9     static final String SULFURAS = "Sulfuras, Hand of Ragnaros";  
10
```

패턴을 활용한 리팩토링 완료



8. 회고



회고

(Plus, Minus, Insight)



마치며

