

Testing #2

Team 2 : "제품 관리 알림기능이 있는" 편의점 정산 프로그램

Team 5 : 자동분류 자료관리 시스템

T2

200511318 김희재

200511326 박현진

200711436 서영주

200913987 이승효

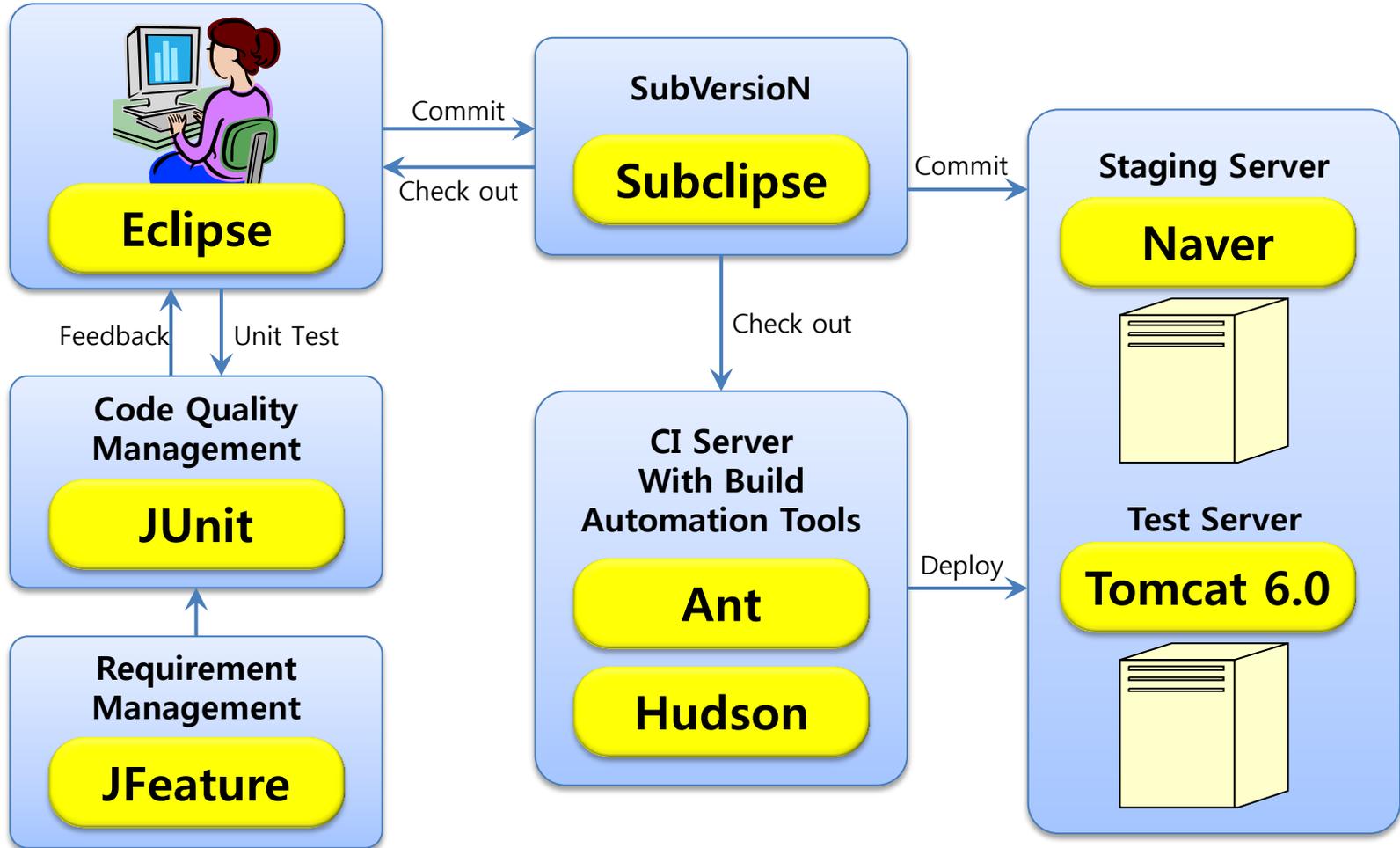
Contents.

1. Our CTIP Environment

2. Testing [Team 2 / Team 5]

- ◆ **System Testing**
- ◆ **Inspection**
- ◆ **Unit Testing**
- ◆ **Coverage**
- ◆ **Build**

CTIP Environment



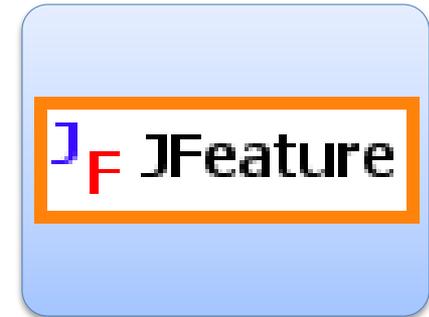
Tools of CTIP



Develop Tool



Unit Testing Tool



RE Tool



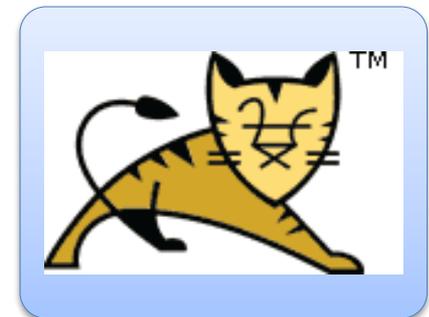
CM Tool



Build Automation



CI Server



WAS
(Web Application Server)

Testing Process



'12. 5. 24, 21:30이 되어서야 정상적 테스트가 가능해짐.

Testing [Team 2]

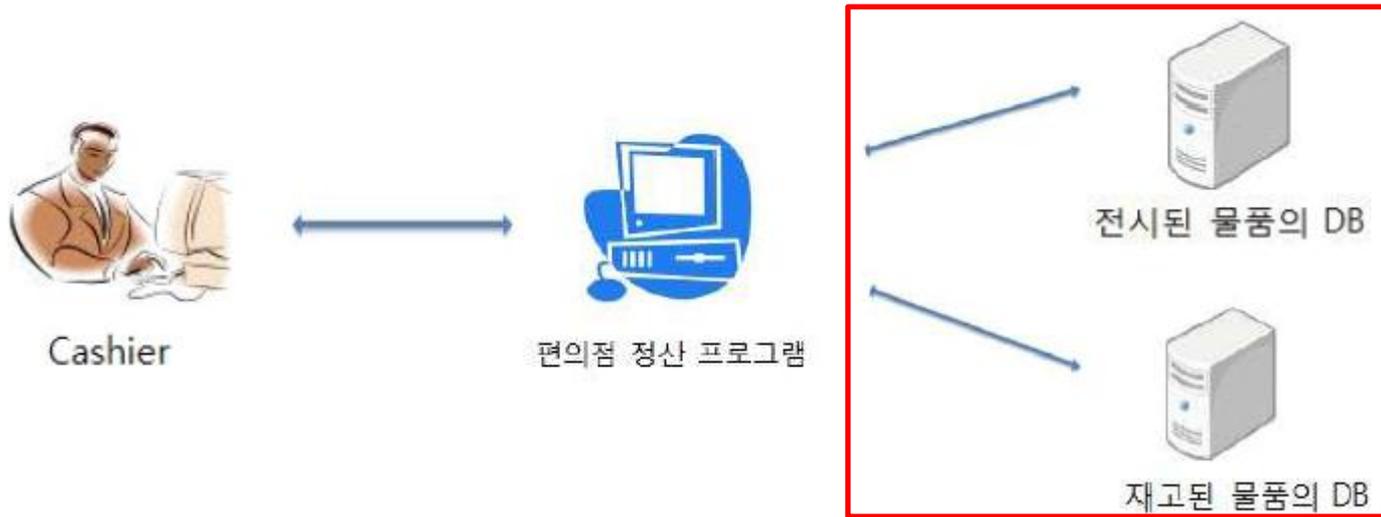
"제품 관리 알림기능이 있는" 편의점 정산 프로그램

Testing [Team 2]

"제품 관리 알림기능이 있는" 편의점 정산 프로그램



물품 정보 입력



명세에 있으나 존재하지 않던 DB가 파일 시스템으로 구현됨.
하지만 '**물품 정보**'만 저장하고 '**결제 정보**'는 저장하지 않는다.



① 물품관리

◆ 물품정보 입력

The screenshot displays a desktop environment with three windows. The background window is titled '편의점 관리 시스템' (Convenience Store Management System). The foreground window is titled '새로운 물품 추가' (Add New Item) and contains a form with the following fields:

상품명 :	토르의함마
물품가격 :	1000
재고할물품수량 :	-1
전시할물품수량 :	-1

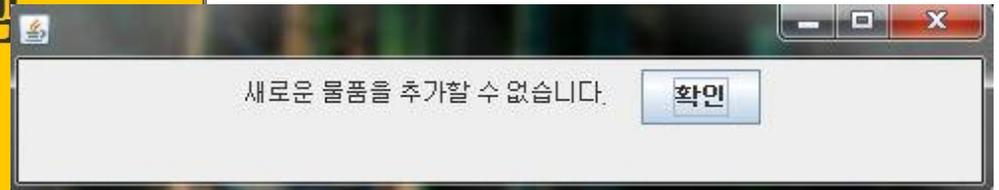
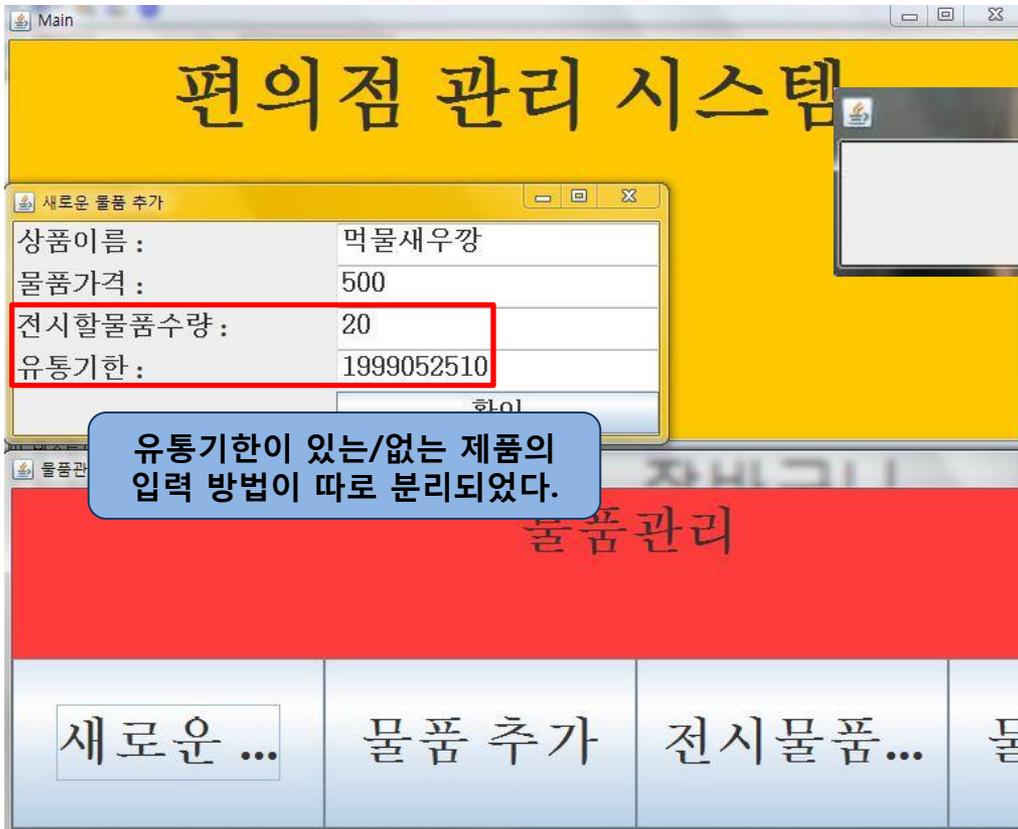
Below the form is a '확인' (Confirm) button. To the right, a message box displays the text '새로운 물품을 추가할 수 없습니다.' (Cannot add new item.) with an '확인' (Confirm) button.

At the bottom, a window titled '물품관리' (Inventory Management) shows a red header and four buttons: '새로...' (New...), '물품 ...' (Item ...), '전시...' (Display...), and '물품 ...' (Item ...).

음수의 물품수량 입력 시
물품을 추가할 수 없게 바뀜.

① 물품관리

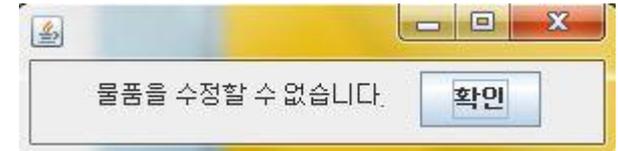
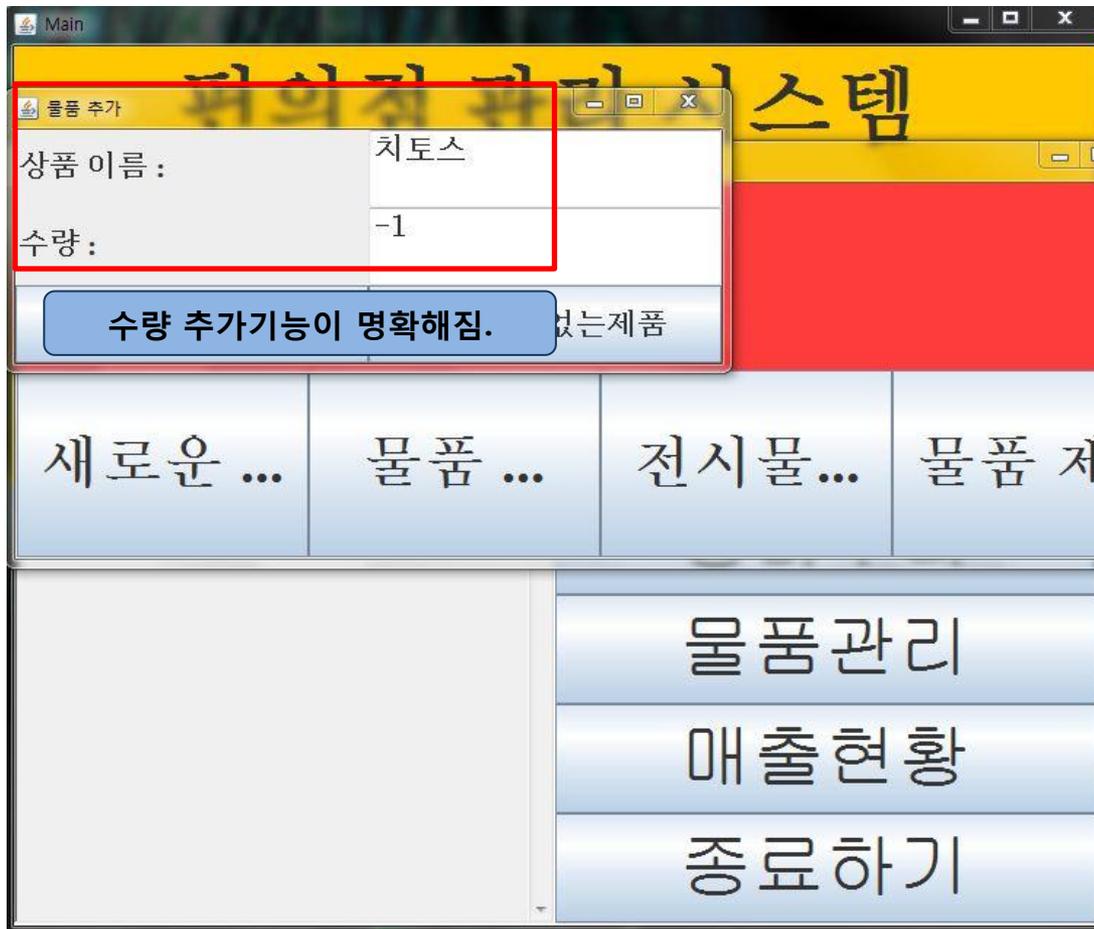
◆ 유통기한 입력



유통기한이 지난 물품의 입력이 불가능하게 바뀜.

① 물품관리

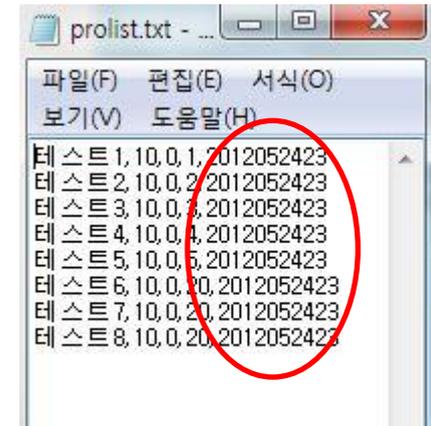
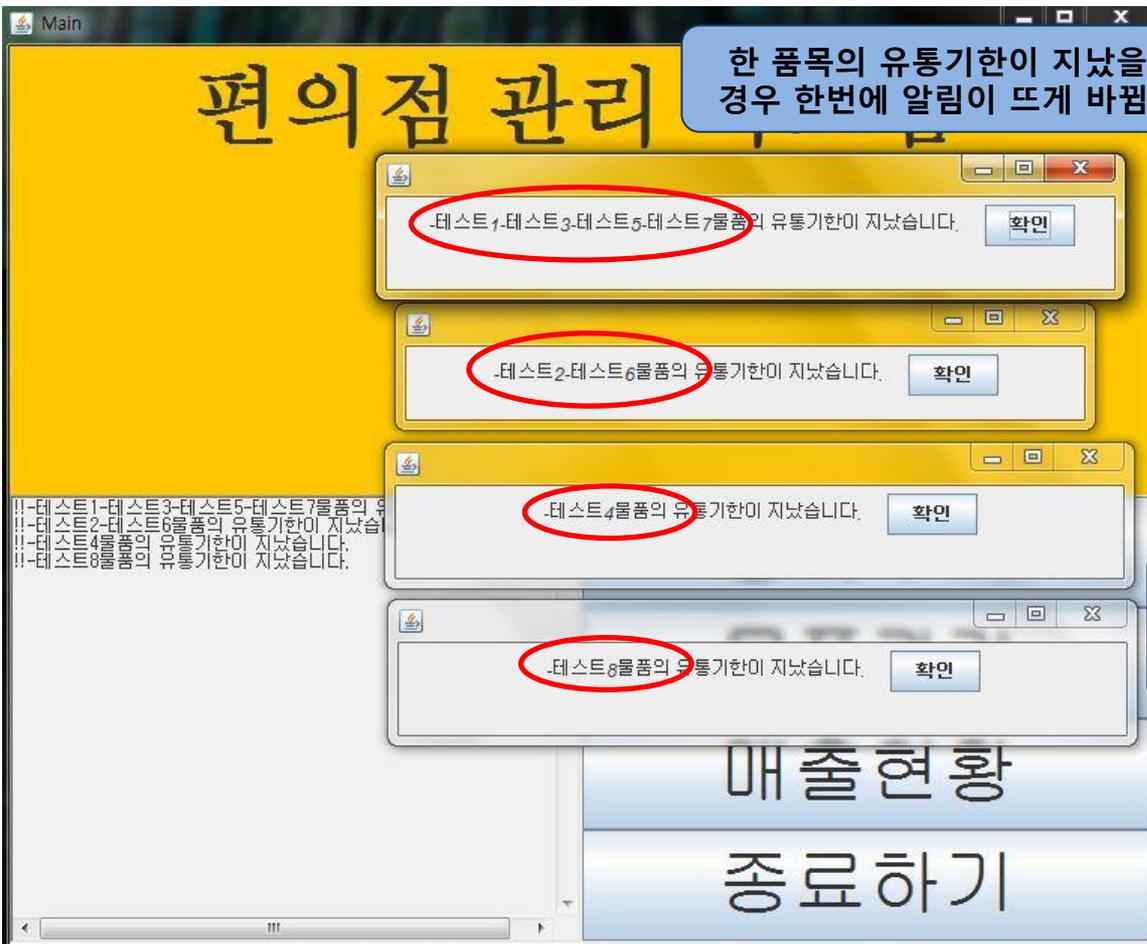
◆ 수량 수정 - 유통기한 없는 제품



수량 감소가 불가능하게 바뀜.

① 물품관리

◆ 유통기한 알림 기능



클래스 다름알림창이 뜨는 규칙을 알 수 없음.
파일로 품목을 입력 받을 때에는 알림기능이
문제가 없지만 프로그램 상에서 입력 받을 때에는
알림기능이 동작하지 않는다.

② 장바구니

◆ 물품 결제

약간 개선되었으나 아직 메시지 출력부분에 문제가 있음

The screenshot shows a graphical user interface for a shopping cart system. A yellow banner at the top reads "약간 개선되었으나 아직 메시지 출력부분에 문제가 있음". Below it, a window titled "시스템" (System) displays a payment confirmation screen. A red box highlights the error message: "아이스크림샌드위치의 전시수량이 부족합니다." (The display quantity of ice cream sandwich is insufficient). The screen also shows the items "아이스크림" (Ice cream) and "샌드위치" (Sandwich) with quantities of 10 each, and a total payment amount of 30,000. A "확인" (Confirm) button is visible. At the bottom, a menu lists options: 장바구니 (Shopping Cart), 물품관리 (Product Management), 매출현황 (Sales Status), and 종료하기 (End). A terminal window at the bottom left shows the following output:

```
아이스크림 상품 10개를 장바구니에 추가했습니다.  
샌드위치 상품 10개를 장바구니에 추가했습니다.  
!!아이스크림샌드위치 상품의 전시수량이 부족합니다.  
====장바구니에 결제된 사항====  
이름 :아이스크림      수량 :10  
이름 :샌드위치       수량 :10  
결제 금액 :30000
```

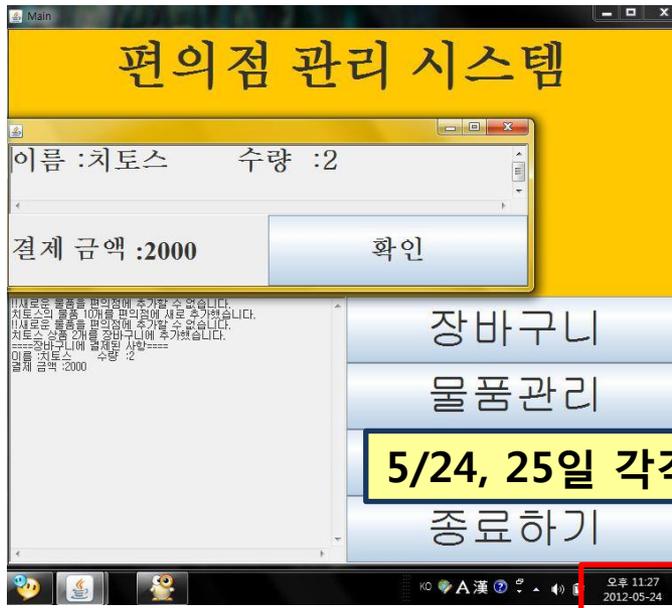
③ 매출 현황

◆ 하루 매출 현황

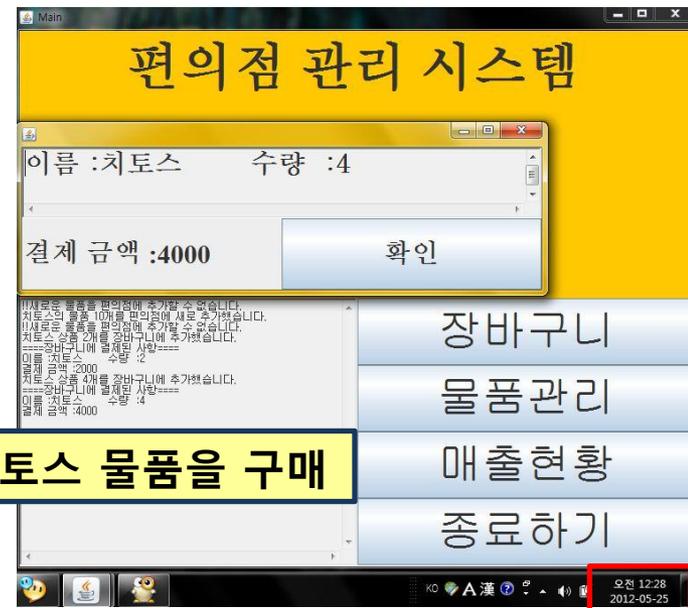
The screenshot shows a system window titled 'Main' with a yellow background. At the top, it says '편의점 카카 시스템'. A yellow callout box contains the text: '하루 매출 현황 열람 시 품목의 이름만 순서대로 쌓이기만 할 뿐 각 품목의 가격과 수량은 확인할 수 없다.' A red circle highlights a list of items: '이: :간초', '이: :메로나', '이: :메로나', '이: :붕어빵', '이: :간초'. A red oval highlights the text '오늘 하루 총매출 :6400'. A blue callout box at the bottom right says: '장바구니 구매가 2번 이상 일어나도 매출 현황 조회가 정상적으로 이루어지게 바뀐.' A red box highlights a terminal-style log showing purchase history: '장바구니에 결재된 사항', '이: :간초 수량 :2', '이: :메로나 수량 :3', '결제 금액 :2100', '메로나 상품 1개를 장바구니에 추가했습니다.', '장바구니에 결재된 사항', '이: :메로나 수량 :1', '결제 금액 :500', '붕어빵 상품 5개를 장바구니에 추가했습니다.', '간초 상품 1개를 장바구니에 추가했습니다.', '장바구니에 결재된 사항', '이: :붕어빵 수량 :5', '이: :간초 수량 :1', '결제 금액 :3800', '오늘 하루 총 매출 현황', '이: :간초', '이: :메로나', '이: :메로나', '이: :붕어빵', '이: :간초', '오늘 하루 총매출 :6400'. At the bottom right, there are buttons for '장바구니', '매출 현황', and '종료하기'.

③ 매출현황

◆ 기간별 매출 현황



오후 11:27
2012-05-24



오전 12:28
2012-05-25

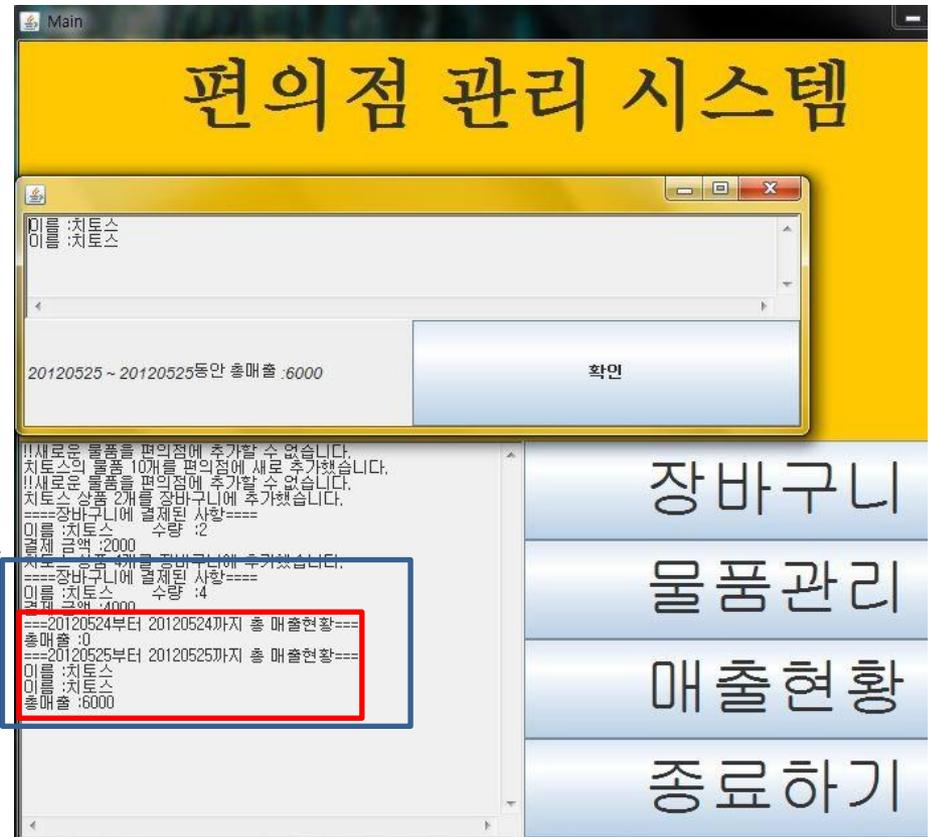
5/24, 25일 각각 치토스 물품을 구매

③ 매출현황

◆ 기간별 매출 현황

```
치토스 상품 4개를 장바구니에 추가했습니다.  
====장바구니에 결제된 사항====  
이름 :치토스      수량 :4  
결제 금액 :4000  
====20120524부터 20120524까지 총 매출현황====  
총매출 :0  
====20120525부터 20120525까지 총 매출현황====  
이름 :치토스  
이름 :치토스  
총매출 :6000
```

기간별 매출 현황을 제대로 출력하지 못하고
이제까지의 모든 구매가 프로그램의 실행 시점에
이루어진 것처럼 출력된다.



Testing [Team 2]

"제품 관리 알림기능이 있는" 편의점 정산 프로그램



Inspection – Document & Code

#1. 선택된 물품의 '장바구니' 추가

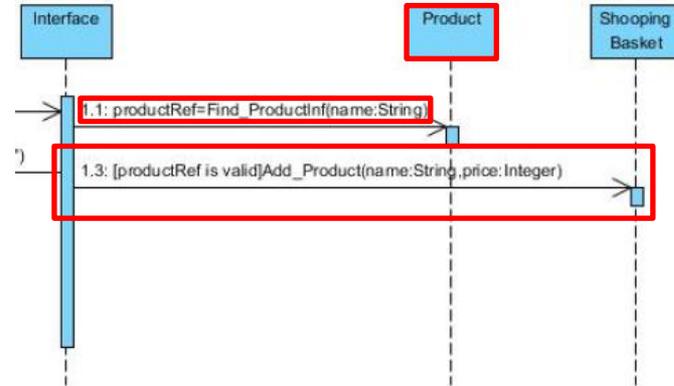
```

Product_Mgr.java
clean(AFind_ProductInf(String Name, int number) {
    return false; // 추가하려는 물품의 개수가 0이거나 음수이면 false
}
if(number >= 50)
.....
return false; // 추가하려는 개수가 전시된 물품보다 많다
}
System.out.println(p.getDis_num() + " 추가하려는 개수");
shoppingbasket.Add_Product(Name, p.getPrice(), number);
return true;
}

public void Add_Product(String name, int price, int number) {
    int i;

```

Class명 다름
함수명 다름
인자 다름
호출위치 다름
인자 다름



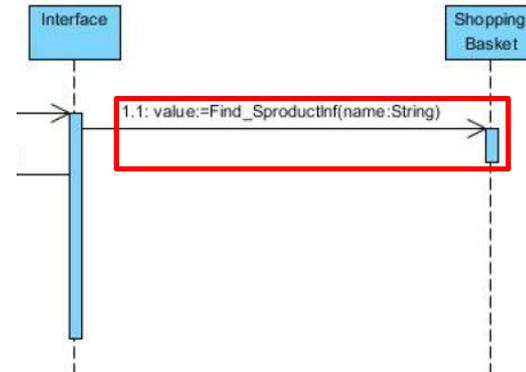
#2. '장바구니'안의 물품 삭제

```

public boolean DFind_ProductInf(String name, int number) {
    int i;
    for(i=0; i<50; i++)
    {
        if(sale_product[i][0]!=null)
        {
            if(sale_product[i][0].equals(name))
            {
                if(Integer.parseInt(number) >= 0)
                {
                    Rmv_Product(name, number);
                }
            }
        }
    }
    return true;
}

```

함수명 다름
인자 다름
UML에 없는 함수 호출

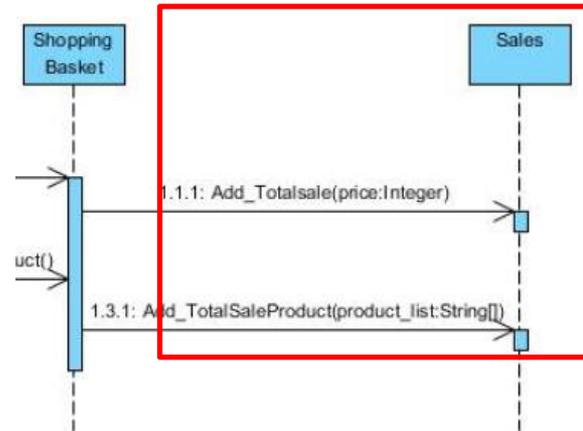


Inspection – Document & Code

#3. '장바구니'에 추가된 물품 결제

```

Sales_Mgr.java
200 Add_Totalsale(int price_total)
클래스 다름
126 public void Request_Pay_prod(){
127     Font f1=new Font("Serif",Font.BOLD,30);
128     int pricetotal;
129     String sp;
130     String str;
131     pricetotal=shoppingbasket.Get_SalePrice();
132     sp=shoppingbasket.Get_SaleProduct();
133     str=product.Reduce_CheckNum(sp);
134     if(str.equals("")){
    
```



#8. 물품 추가 후 재고수량을 수정

```

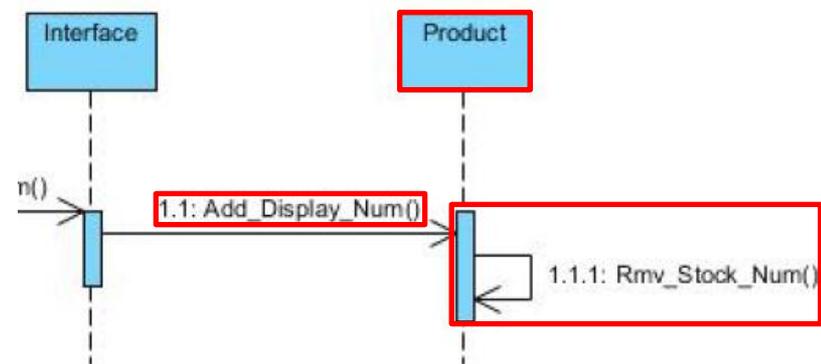
Interface.java
174 public void Request_chg_stock_num(String name,int num){
175     String str;
176     String []c;
177     str=product.Control_Product_Num(name,num);
178     c=str.split("-");
    
```

UML에 없는 함수 호출 한번만 있음.

```

Product_Mgr.java
254 public String Control_Product_Num(String Name,int num) {
    
```

클래스가 다름.
함수 자체가 다름.
인자가 다르고 UML에 나온 다음 함수도 찾을 수 없음.



Inspection – Document & Code

#18. 물품의 정보를 수정

클래스 다름

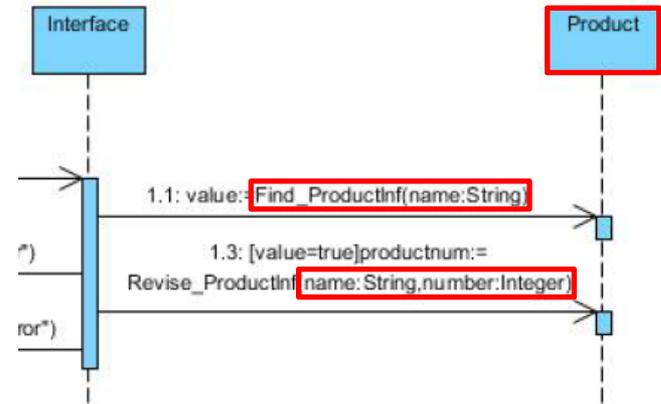
```

Product_Mgr.java
public boolean MFind_ProductInf(String Name,int num,String expDate)
    if(num<=0||num>=200)
    {
        return false;//수정하려는 물품의 개수가 음수이면 안되니까.너무 :
    }
    if(expDate==null)
    .....
    Product p=(Product)v.elementAt(i);
    if(p.getName().equals(Name)
    {
        this.Revise_ProductInf(Name,num,expDate)
        return true;
    }
  
```

함수명 다름

인자 다름

클래스 다름



#19. 하루에 판매된 물품 목록과 총 매출액 출력

클래스 다름

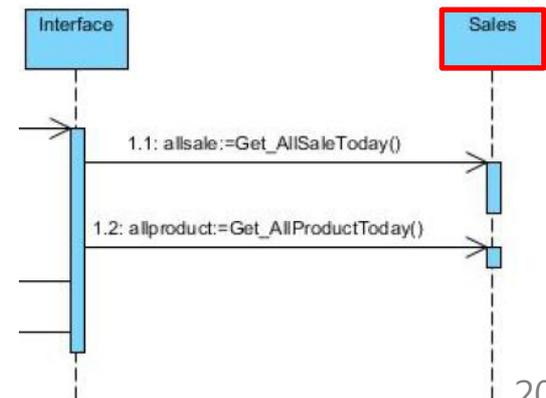
```

Sales_Mgr.java
public int Get_AllSaleToday() {
  .....
}
  
```

클래스 다름

```

Sales_Mgr.java
public String[] Get_AllProductToday() {
  // System.out.println(sale_list[i].getT
  .....
}
  
```

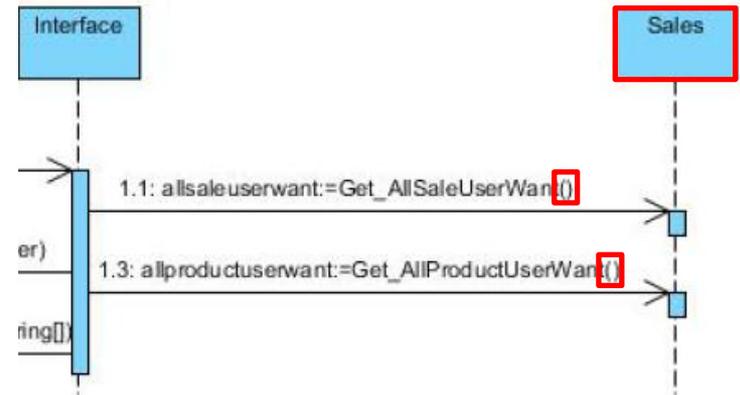


Inspection – Document & Code

#20. 원하는 기간의 총 매출액 출력

```
Sales_Mgr.java ☒
148 public int Get_AllSaleUserWant(String beginDate, String endDate) {
149
150     int price = 0;

Sales_Mgr.java ☒
164 public String[] Get_AllProductUserWant(String beginDate, String endDate) {
165
166     int i = 0;
```



Testing [Team 2]

"제품 관리 알림기능이 있는" 편의점 정산 프로그램



Testing 기법

Black Box Test

코드레벨 이해 없이 어플리케이션을 테스트하는 방법

White Box Test

소스코드를 기반으로 테스트하는 방법

Gray Box Test

어플리케이션을 기반으로 테스트하지만 소스코드의 정확성 및 오류사항을 함께 테스트하는 방법.

Testing 기법

	White Box Test	Black Box Test
개요	내부구조 및 논리 위주 테스트 기법	기능 및 I/O 위주 테스트 기법
기법	커버리지(수행, 분기, 조건, 루프)	동등분할, 경계값, 원인/결과그래프, 오류예측
단계	디버깅 및 단위테스트 단계에서 수행	통합테스트 및 시스템, 인수 테스트 단계
수행자	개발자 / 테스터	개발자 / 테스터 / 사용자
공통점	테스트 통한 소프트웨어 결함의 사전 제거 및 품질향상	

Gray Box Test

테스트 기간이 오래 걸린다고 생각할 수도 있지만, 사전에 코드오류를 미리 확인하여, 오류로 인한 추가 코딩 작업이 많이 줄어들어 테스트 시간이 감소된다

In [Team2] Testing

- ① 명세서, 보고문서와 실제 프로그램의 차이가 큼
- ② Pairwise Testing 적용이 어려움
- ③ Void Method의 경우 내부의 값에 변화를 줌
- ④ 각각의 Use-case에서 선행되는 Method의 Success여부가 다음의 Method의 Test 결과를 좌우



Gray Box Test

프로젝트의 소스코드를 면밀히 분석



예상되는 Fail과 Success case를 도출



Test Case를 작성 하여 Unit Test수행

Unit Testing – JUnit 4

#16. 새로운 물품 품목 추가

Finished after 0.146 seconds

Runs: 27/27 Errors: 1 Failures: 12

Product_Mgr_testAdd_NewProduct [Runner: JUnit 4]

- testAdd_NewProduct_SameProductFromFile (0.000 s)
- testAdd_NewProduct_PriceMinusFromFile (0.000 s)
- testAdd_NewProduct_NameNull (0.000 s)
- testAdd_NewProduct_PriceOver (0.001 s)
- testAdd_NewProduct_Exp (0.000 s)
- testAdd_NewProduct_ExpExceed (0.000 s)
- testAdd_NewProduct_NameNullFromFile (0.000 s)
- testAdd_NewProduct_StkOverFromFile (0.000 s)
- testAdd_NewProduct_StkMinusFromFile (0.000 s)
- testAdd_NewProduct_DisOverFromFile (0.000 s)
- testAdd_NewProduct_DisMinusFromFile (0.000 s)
- testAdd_NewProduct_PriceOverFromFile (0.000 s)

이전 버전과 다르게
파일 입출력으로 물품을
받았는데, 오히려 Fail이
늘었음

```
private SimpleDateFormat today;  
private String sTime;  
  
@Before  
public void instantinit() throws Exception {  
    product_mgr = new Product_Mgr();  
    today = new SimpleDateFormat("yyyyMMddHH", Locale.KOREA);  
    day = new Date();  
    sTime = today.format(day);  
}  
  
@Test  
public void testAdd_NewProduct_SameProductFromFile() { /*=== 추가 이점이 Null*/  
    Product p, t;  
    for(int i = 0; i < product_mgr.v.size(); i++) {  
        for(int j = 0; j < product_mgr.v.size(); j++) {  
            p = product_mgr.v.elementAt(i);  
            t = product_mgr.v.elementAt(j);  
            assertFalse(!=j && p.getName().equals(t.getName()));  
        }  
    }  
}  
  
@Test  
public void testAdd_NewProduct_NameNullFromFile() { /*=== 추가 이점이 Null*/  
    Product p;  
    for(int i = 0; i < product_mgr.v.size(); i++) {  
        p = product_mgr.v.elementAt(i);  
        assertFalse(p.getName().equals(""));  
    }  
}  
  
@Test  
public void testAdd_NewProduct_StkOverFromFile() { /*=== 추가 이점이 Null*/  
    Product p;  
    for(int i = 0; i < product_mgr.v.size(); i++) {  
        p = product_mgr.v.elementAt(i);  
        assertFalse(p.getStk_num() >= 200);  
    }  
}
```

Testing [Team 2]

"제품 관리 알림기능이 있는" 편의점 정산 프로그램



Requirement Coverage

Requirement Coverage Report

Requirement Coverage Summary

Summary (28)	6 (21.43%)	21 (75%)	1 (3.57%)
--------------	------------	----------	-----------

Requirement Coverage Details

Sr#	Category	Coverage
1.	15.유통기한이 있는 물품의 '우측' 사용자에게 알림 (1)	1 (100%)
2.	20.원하는 기간의 총 매출액 출력 (2)	2 (100%)
3.	16.새로운 물품 품목 추가 (1)	1 (100%)
4.	7.사용자에게 부족한 물품 알림 (1)	1 (100%)
5.	9.재고 물품 부족함 확인 (1)	1 (100%)
6.	19.하루에 판매된 물품 목록과 총 매출액 출력 (2)	1 (50%) 1 (50%)
7.	5.판매된 물품 수량 조정 (1)	1 (100%)
8.	13.유통기한 지난 물품 폐기 후 재고 수량을 (1)	1 (100%)
9.	18.물품의 정보를 수정 (2)	2 (100%)
10.	12.사용자에게 유통기한 지난 물품 알림 (1)	1 (100%)
11.	8.물품 추가 후 재고수량을 수정 (2)	2 (100%)
12.	3.'장바구니'에 추가된 물품 결제 (4)	2 (50%) 2 (50%)

All Requirements

- testFind_Product (80%)
- testAdd_Product (50%)
- testFind_Sproduc... (100%)
- testGet_SalePrice (100%)
- testAdd_Totalsale (66.67%)
- testGet_SaleProd... (100%)
- testAdd_TotalSal... (33.33%)
- testAdd_TotalSal... (33.33%)
- testReduce_Check... (100%)
- testReduce_Check... (100%)
- testReduce_Check... (100%)
- testAdd_Display_... (87.5%)
- testRmv_Stock_Num (0%)
- testControl_Prod... (87.5%)
- New (0%)
- testCheck_OverExp (0%)
- testCheck_OverExp (0%)
- testRevise_ExpPr... (60%)
- testReduce_Check... (100%)
- testReduce_Check... (100%)

지금까지 작성한 Test Case에 대해 Coverage를 산출

Requirement Coverage

지금까지 작성한 Test Case에 대해 Coverage를 산출

Requirement Coverage Report

Requirement Coverage Summary

testFind_Product (5) 4 (80%) 1 (20%)

Requirement Coverage Details

Sr#	Test Methods	Coverage
1.	Product_Mgr_testAFind_ProductInfn.testAFind_ProductInfn	100%
2.	Product_Mgr_testAFind_ProductInfn.testAFind_ProductInfn_NumMinus	100%
3.	Product_Mgr_testAFind_ProductInfn.testAFind_ProductInfn_NumOver	100%
4.	Product_Mgr_testAFind_ProductInfn.testAFind_ProductInfn_ProductFromFile	0%
5.	Product_Mgr_testAFind_ProductInfn.testAFind_ProductInfn_WrongName	100%

Requirement Coverage Report

Requirement Coverage Summary

3.장바구니에 추가된 물품 결제 (4) 2 (50%) 2 (50%)

Number of Requirements	4
Unique Test Methods	8
Requirements:Test Methods Ratio	1:2
Missing Test Methods	None

Requirement Coverage Details

Sr#	Coverage Item	Coverage
1.	Get_SalePrice()_Test (1)	1 (100%)
2.	Add_Totalsale()_Test (1)	0%
3.	Get_SaleProduct()_Test (1)	1 (100%)
4.	Add_TotalSaleProduct()_Test (1)	0%

Testing [Team 2]

"제품 관리 알림기능이 있는" 편의점 정산 프로그램



Requirement Coverage

Ant에서 Build.xml을 만들어 Hudson에서 Build

Hudson

Hudson > Team2 2ndTest > #9 > Test Results > (root) > Product_Mgr_testAdd_NewProduct

프로젝트로 돌아가기
상태
변경사항
Build Now
콘솔 출력
Configure
History
Tag this build
Test Result
Previous Build

Test Result : Product_Mgr_testAdd_NewProduct

14 failures (±0)

28 tests (±0)
Took 0.16 sec.
소개 내용 읽기

All Tests

Test name	Duration	Status
testAdd_NewProduct_DisMinusFromFile	0 ms	Failed
testAdd_NewProduct_DisMinusIntMIN	0 ms	Passed
testAdd_NewProduct_DisMinusONE	0 ms	Passed
testAdd_NewProduct_DisOverFromFile	0 ms	Failed
testAdd_NewProduct_DisPlusIntMAX	0 ms	Passed
testAdd_NewProduct_DisPlusMAX	0 ms	Failed
testAdd_NewProduct_DisPlusOne	0 ms	Passed
testAdd_NewProduct_Exp	0 ms	Passed
testAdd_NewProduct_ExpExceed	0 ms	Passed
testAdd_NewProduct_ExpWrongDate	0 ms	Failed
testAdd_NewProduct_ExpWrongFromFile	30 ms	Failed
testAdd_NewProduct_ExpWrongString	10 ms	Failed
testAdd_NewProduct_NameNotNull	0 ms	Passed

Hudson

Hudson > Team2 2ndTest > #9 > Test Results > (root) > Product_Mgr_testRevise_ProductInf

프로젝트로 돌아가기
상태
변경사항
Build Now
콘솔 출력
Configure
History
Tag this build
Test Result
Previous Build

Test Result : Product_Mgr_testRevise_ProductInf

4 failures (±0)

10 tests (±0)
Took 20 ms.
소개 내용 읽기

All Tests

Test name	Duration	Status
testRevise_ProductInf_Exp	0 ms	Passed
testRevise_ProductInf_ExpExceed	0 ms	Passed
testRevise_ProductInf_ExpFromFile	0 ms	Failed
testRevise_ProductInf_NameNull	0 ms	Passed
testRevise_ProductInf_NonExp	0 ms	Passed
testRevise_ProductInf_NonExpFromFile	0 ms	Failed
testRevise_ProductInf_NumMinus	10 ms	Passed
testRevise_ProductInf_NumOver	0 ms	Passed
testRevise_ProductInf_OverDis	0 ms	Failed
testRevise_ProductInf_OverStk	10 ms	Failed

Hudson Build

Hudson

search

자동 갱신형 보기

Hudson > Team2 2ndTest

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

Build History [\(trend\)](#)

●	#9	2012. 5. 25 오전 2:38:47	
●	#8	2012. 5. 24 오후 11:19:06	
●	#7	2012. 5. 24 오후 4:42:56	
●	#6	2012. 5. 24 오후 4:41:31	
●	#5	2012. 5. 23 오후 3:03:07	
●	#4	2012. 5. 23 오후 3:02:48	
●	#3	2012. 5. 23 오후 3:00:36	
●	#2	2012. 5. 23 오후 2:52:16	
●	#1	2012. 5. 23 오후 2:48:58	

Project Team2 2ndTest

[Workspace](#)

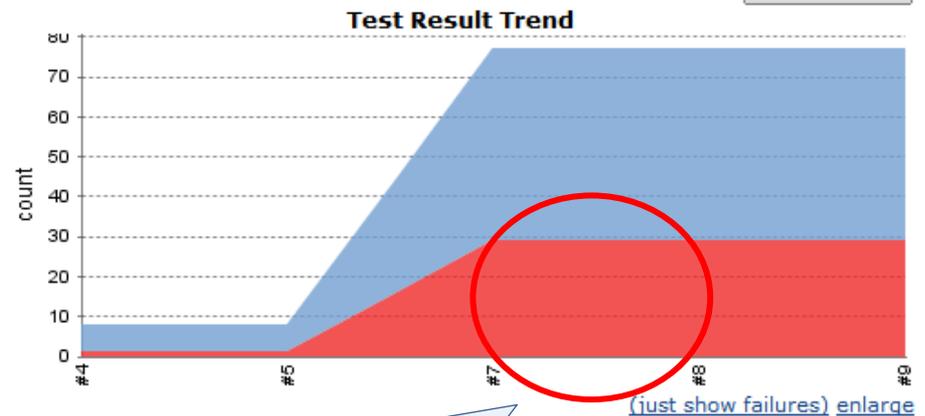
[Recent Changes](#)

[Latest Test Result \(29 failures / ±0\)](#)

[Latest Console output](#)

소개 내용 입력

Disable Project



Permalinks

- [Last build \(#9\), 2 hr 12 min ago](#)
- [Last successful build \(#9\), 2 hr 12 min ago](#)
- [Last failed build \(#6\), 12 hr ago](#)
- [Last unstable build \(#9\), 2 hr 12 min ago](#)
- [Last unsuccessful build \(#9\), 2 hr 12 min ago](#)

지난 테스트 요구사항에 피드백을 잘 해줘서 fail율이 36% 정도로 생각보다 작게 나왔습니다. :)

결과보고서

테스트 보고서 - Team2

1. 프로그램 및 인터페이스

- ① 프로그램 종료 시 기존 물품판매정보Sales 유지 불가능
- ② 파일로부터 입력받은 물품에 대한 정보가 프로그램 실행 시 화면에 표시되지 않음
- ③ 프로그램 실행시간 기준으로 프로그램이 동작하여 현재시간을 반영하지 못함
- ④ 따라서 유통기한 지난 품목 검색, 장바구니의 결제 시간저장, 기간별 매출 검색기능에 정상적으로 작동하지 않음.
- ⑤ 유통기한이 지난 물품의 검색 기능이 Verification을 만족하지 못함.
- ⑥ '종료버튼'외에 Alt+F4 X버튼으로 종료시 여전히 비정상적으로 종료됨.

2. Use-Case별 Failed Test Case

Use-Case #1 선택된 물품의 '장바구니' 추가

Test Method	Failed Test Case	원인분석
Product_Mgr.AFind_ProductInf	파일에서 입력받은 유통기한에 있는 물품으로 테스트 시	<ul style="list-style-type: none"> • Product_Mgr.생성시 파일에서 부분에서 문제발생 • Product_Mgr.Add_Product의
ShoppingBasket.Add_Product	장바구니의 물품의 종류수가 50종을 초과할 경우	<ul style="list-style-type: none"> • 장바구니당 최대 구매가능 • 제한물품 수를 늘리고 제한을

Use-Case #2 '장바구니'안의 물품 삭제

Test Method	Failed Test Case	원인분석
ShoppingBasket.DFind_ProductInf	테스트 성공	

Use-Case #3 '장바구니'에 추가된 물품 결제

Test Method	Failed Test Case	원인분석
ShoppingBasket.Get_SalePrice	테스트 성공	<ul style="list-style-type: none"> • 50종 이하의 물품 구매 시 테스트 성공
ShoppingBasket.Get_SaleProduct	테스트 성공	<ul style="list-style-type: none"> • 50종 미만의 물품 구매 시 테스트 성공 • 50종 물품 구매시엔 제한사항과 별개로 여러 발생 • Sales_Mgr.Add_TotalSaleProduct의 Test Case 참조
Sales_Mgr.Add_Totalsale	서로 다른 날짜의 구매현황 (sales)가 생성될 경우 50종의 물품 구매시	<ul style="list-style-type: none"> • 매소드 내의 새로운 날짜의 구매 기록 생성시 구매 총액에 값이 0인 변수 totalPrice로 Sales를 생성 • Sales_Mgr.Add_TotalSaleProduct의 Test Case 참조
Sales_Mgr.Add_TotalSaleProduct	50종의 물품 구매시	<ul style="list-style-type: none"> • product_list[listNum][0]=null로 개수 확인시 50번째 검사후 listNum이 51로 증가하면서 product_list[51][0]을 참조함으로써 여러 발생

Use-Case #4 결제 후 총 매출액 업데이트

Test Method	Failed Test Case	원인분석
Sales_Mgr.Add_Totalsale	서로 다른 날짜의 구매현황 (sales)가 생성될 경우 50종의 물품 구매시	<ul style="list-style-type: none"> • 매소드 내의 새로운 날짜의 구매 기록 생성시 구매 총액에 값이 0인 변수 totalPrice로 Sales를 생성 • Sales_Mgr.Add_TotalSaleProduct의 Test Case 참조

Use-Case #5,6,7 판매된 물품 수량 조정, 전시물품 부족함 확인, 사용자에게 부족한 물품 알림

Test Method	Failed Test Case	원인분석
Product_Mgr.Reduce_CheckNum	테스트 성공	<ul style="list-style-type: none"> • 전시 물품 부족시 출력부분 개선을 위해 tName 데 name넣는 부분 수정 필요

Testing [Team 5]

“ 자동분류 자료관리 시스템 ”

Testing [Team 5]

“ 자동분류 자료관리 시스템 ”



#1. 파일 다운로드 & 업로드

- Usecase #4 파일 업로드 하기(업로드 모드로 로그인 했을 시만 사용 가능)
- Usecase #5 파일 다운로드 하기

#2. 확장자 추가/삭제

- Usecase #6 확장자 추가 하기(업로드 모드에서만 가능)
- Usecase #7 확장자 삭제 하기(업로드 모드에서만 가능)
- Usecase #8 카테고리별 확장자 보기

#3. 기타 버그

- Usecase #9 홈화면 버튼
- Usecase #11 파일 삭제하기(업로드모드시에만 가능)
 - (번외_1) 같은 파일을 다시 다운로드
 - (번외_2) 프로그램 종료시

#1. 파일 다운로드 & 업로드

Usecase #4 파일 업로드 하기(업로드 모드로 로그인 했을 시만 사용 가능)

Usecase #5 파일 다운로드 하기

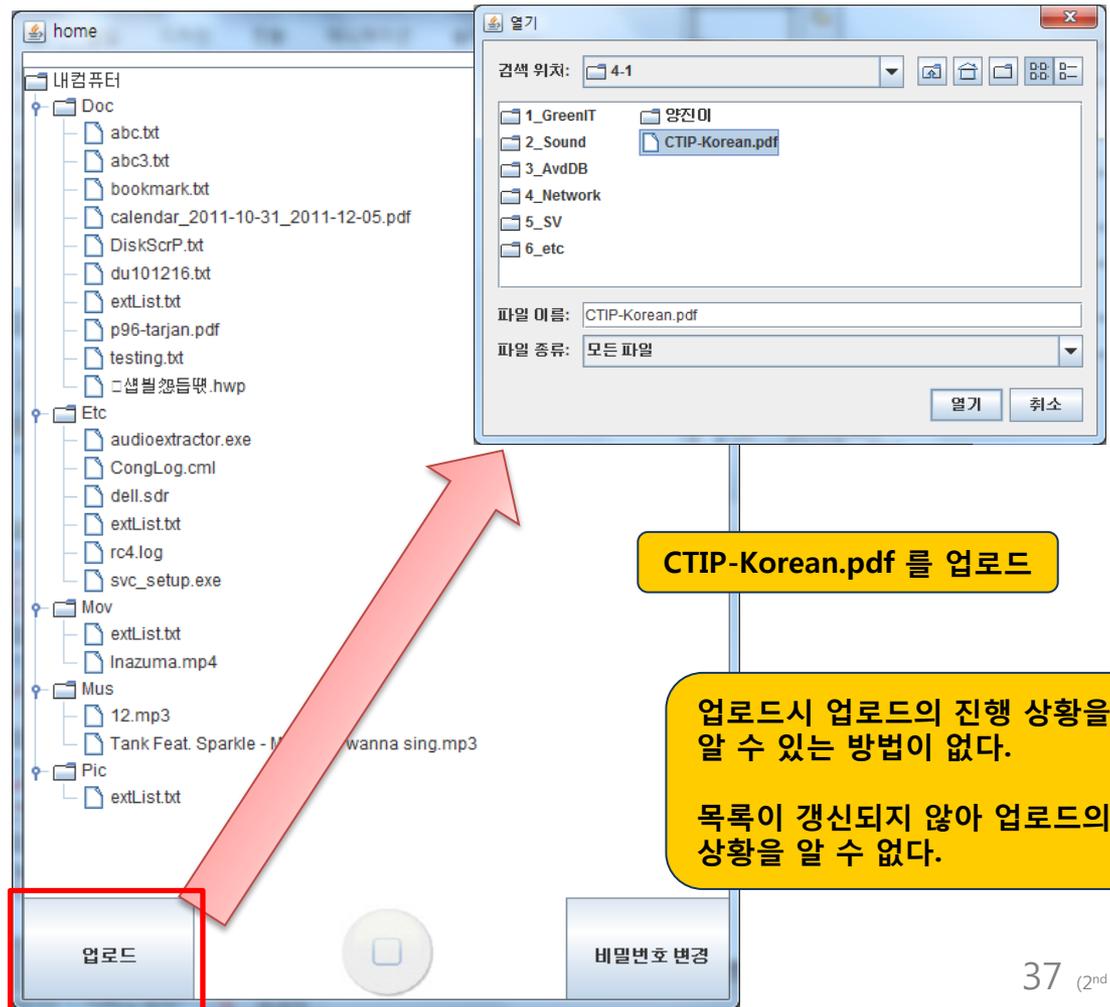
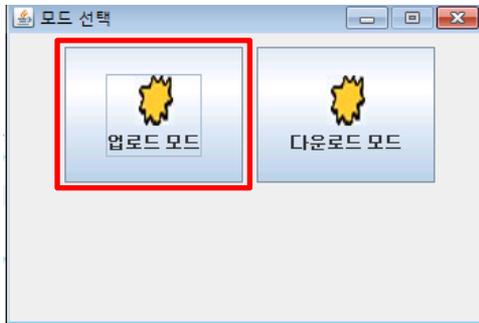
→ 파일을 업로드 해서 다운로드 받은 파일이 업로드한 파일과 일치 여부를 테스트

- ① 파일을 업로드 하여 다시 다운로드
- ② 서버에 있는 파일을 다운로드

#1. 파일 다운로드 & 업로드

- 파일을 업로드 하여 다시 다운로드

① 로컬저장소 파일 업로드



CTIP-Korean.pdf 를 업로드

업로드시 업로드의 진행 상황을 알 수 있는 방법이 없다.
목록이 갱신되지 않아 업로드의 상황을 알 수 없다.

#1. 파일 다운로드 & 업로드

- 파일을 업로드 하여 다시 다운로드

② 파일 다운로드

The image illustrates the process of uploading and downloading a file. It consists of several overlapping windows and a table:

- File Explorer:** Shows a directory structure with a file named "CTIP-Korean.pdf" highlighted in a red box. A red arrow points from this file to the save dialog.
- Mode Selection Dialog:** A small dialog box with two buttons: "업로드 모드" (Upload Mode) and "다운로드 모드" (Download Mode). The "업로드 모드" button is highlighted with a red box.
- Save Dialog:** A "저장" (Save) dialog box with "Desktop" as the save location. The filename is "CTIP-Korean.pdf" and the file type is "모든 파일" (All files). A red arrow points from the save dialog to the file list table.
- File List Table:** A table showing a list of files. The "CTIP-Korean" file is highlighted in a red box. A yellow callout box points to the "0KB" size, indicating a successful upload.

업로드 모드에서 보이지 않던 CTIP-Korean.pdf

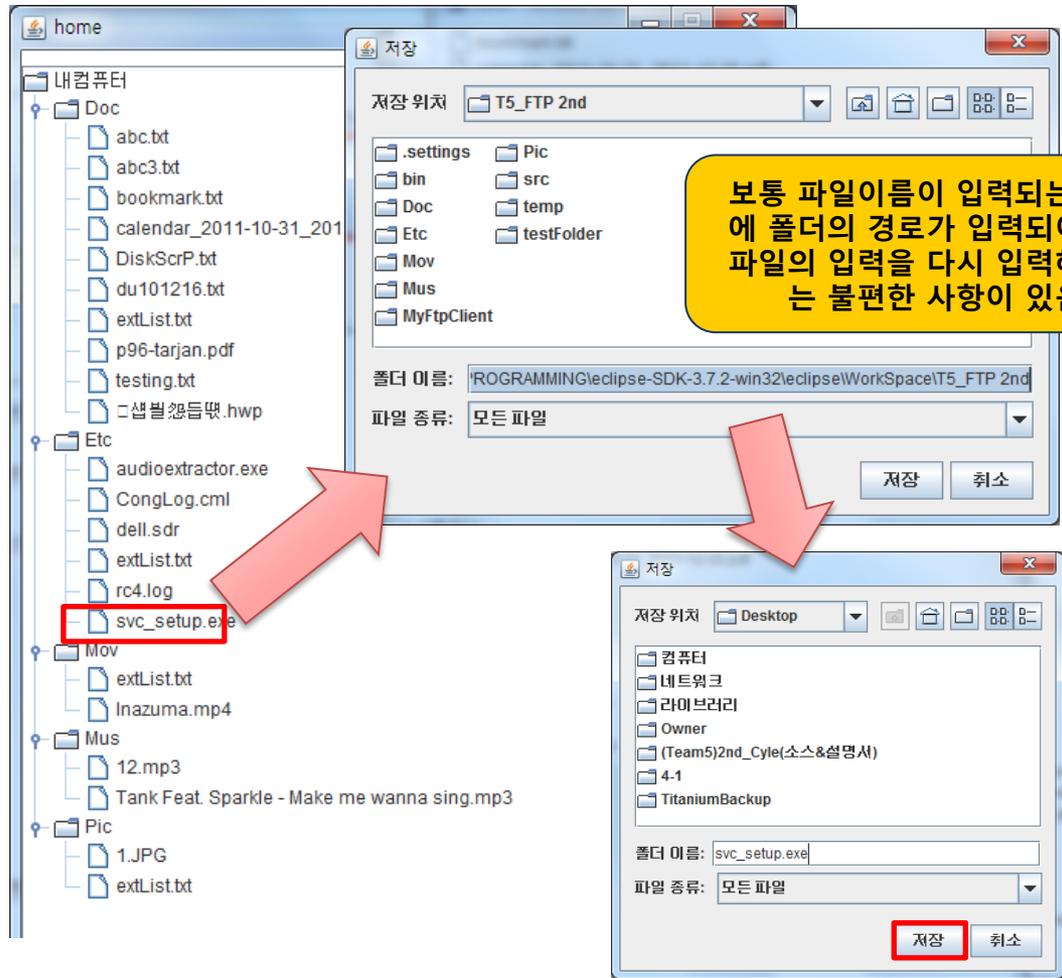
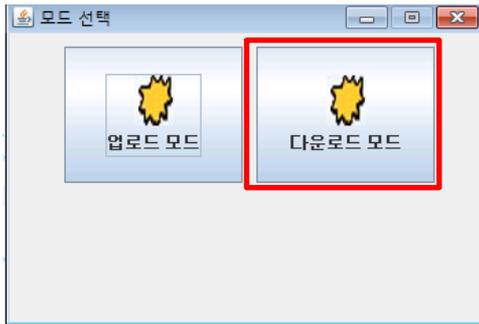
파일이 0KB인 것으로 보아 업로드, 다운로드 모두 안됨을 확인

이름	크기	항목 유형	수정한 날짜
☆ 즐겨찾기			
다운로드			
바탕 화면			
최근 위치			
4-1			
라이브러리			
Subversion			
문서			
비디오			
사진			
dddd	555KB	PDF 파일	2012-05-25 오전...
CTIP-Korean	0KB	PDF 파일	2012-05-25 오전...

CTIP-Korean 수정한 날짜: 2012-05-25 오전 12:44 만든 날짜: 2012-05-25 오전 12:44
PDF 파일 크기: 0바이트

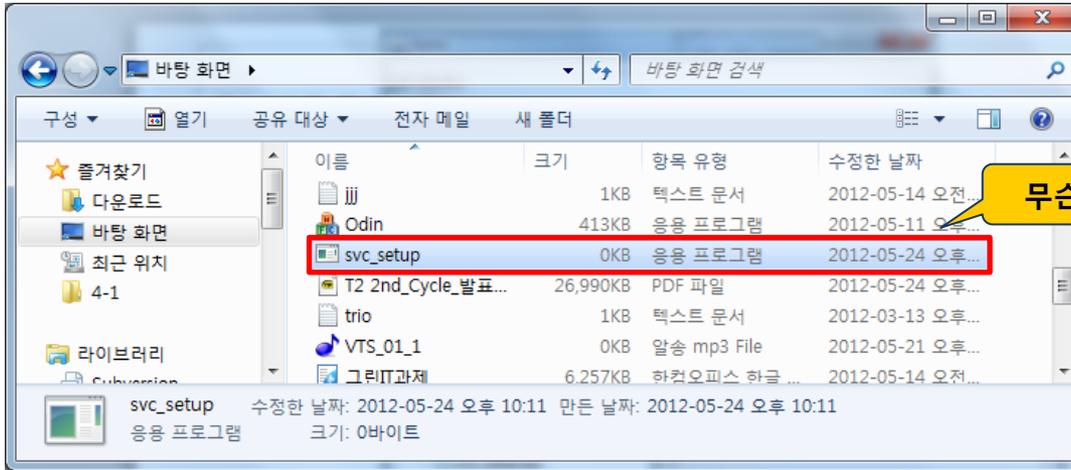
#1. 파일 다운로드 & 업로드

- 서버에 있는 파일을 다운로드

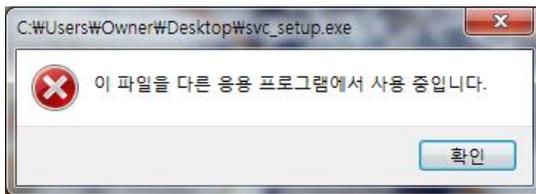


#1. 파일 다운로드 & 업로드

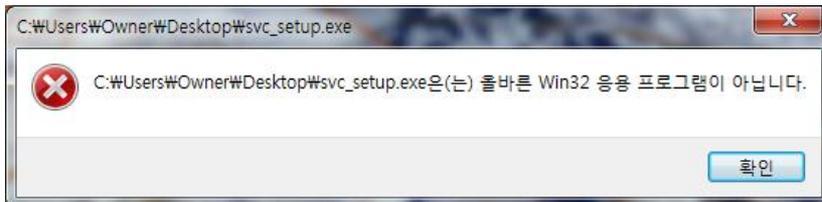
- 서버에 있는 파일을 다운로드



무슨 파일인지 알 수 없으나 0KB....



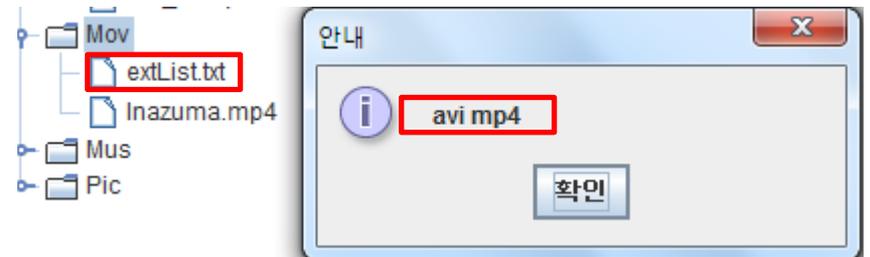
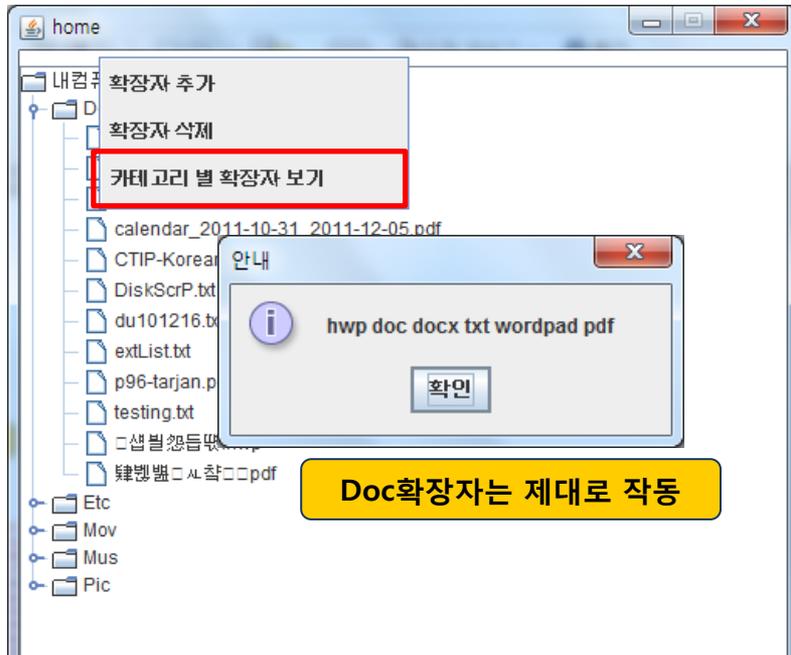
다운로드 완료 시에도 FTP 프로그램이 파일을 계속 access하는 상태여서 파일에 대해 아무런 동작을 할 수 없음



프로그램을 종료한 후, 실행하면, 에러가 뜬. → 다운로드가 제대로 작동 되지 않음

#2. 확장자 추가/삭제

Usecase #8 카테고리별 확장자 보기



avi, mp4가 저장되어있는데 txt가 존재

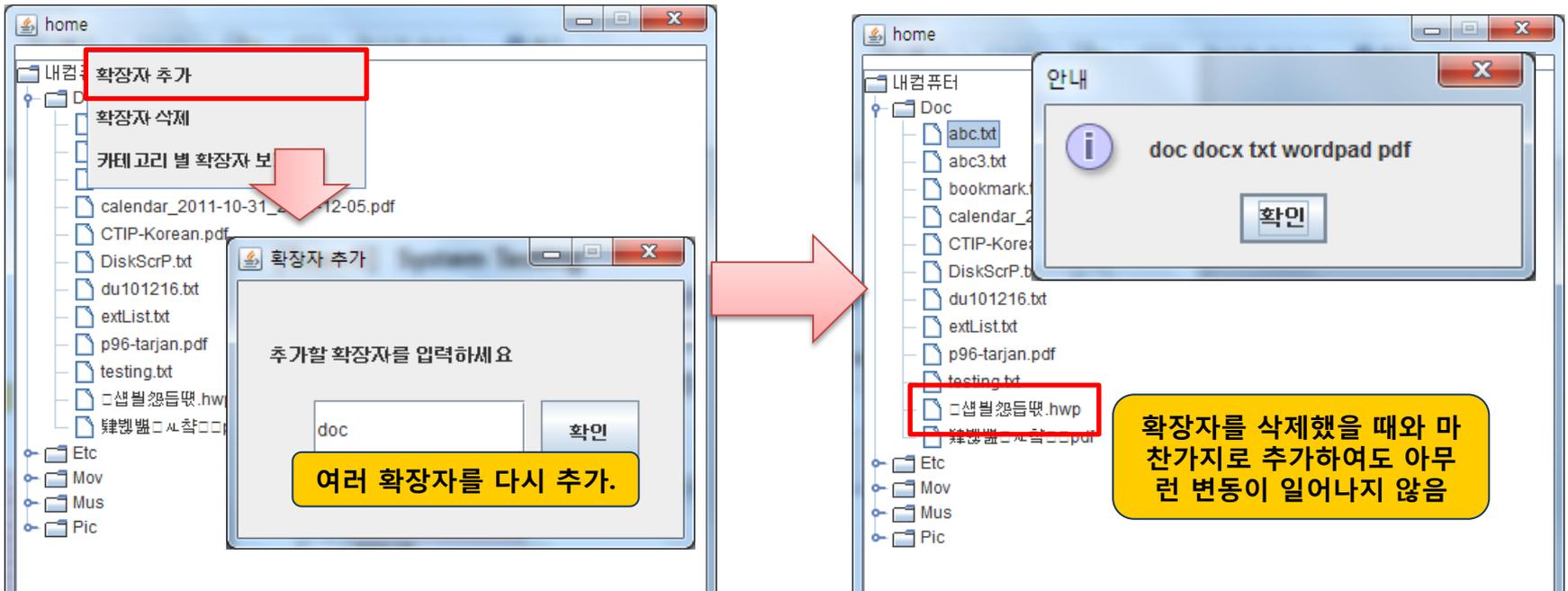
#2. 확장자 추가/삭제

Usecase #7 확장자 삭제하기

The image shows a sequence of two screenshots from a Windows file explorer. In the first screenshot, a '확장자 삭제' (Delete Extension) dialog box is open, with 'hwp' entered in the input field. A red box highlights the '확장자 삭제' option in the menu. A yellow callout box at the bottom states: 'Wordpad를 제외한 모든 확장자를 삭제' (Delete all extensions except Wordpad). In the second screenshot, the 'wordpad' file in the 'Doc' folder is highlighted with a red box. A yellow callout box at the bottom states: '대부분의 확장자를 삭제했지만, doc 폴더에 모든 파일이 그대로 있음!' (Most extensions were deleted, but all files in the doc folder remain!).

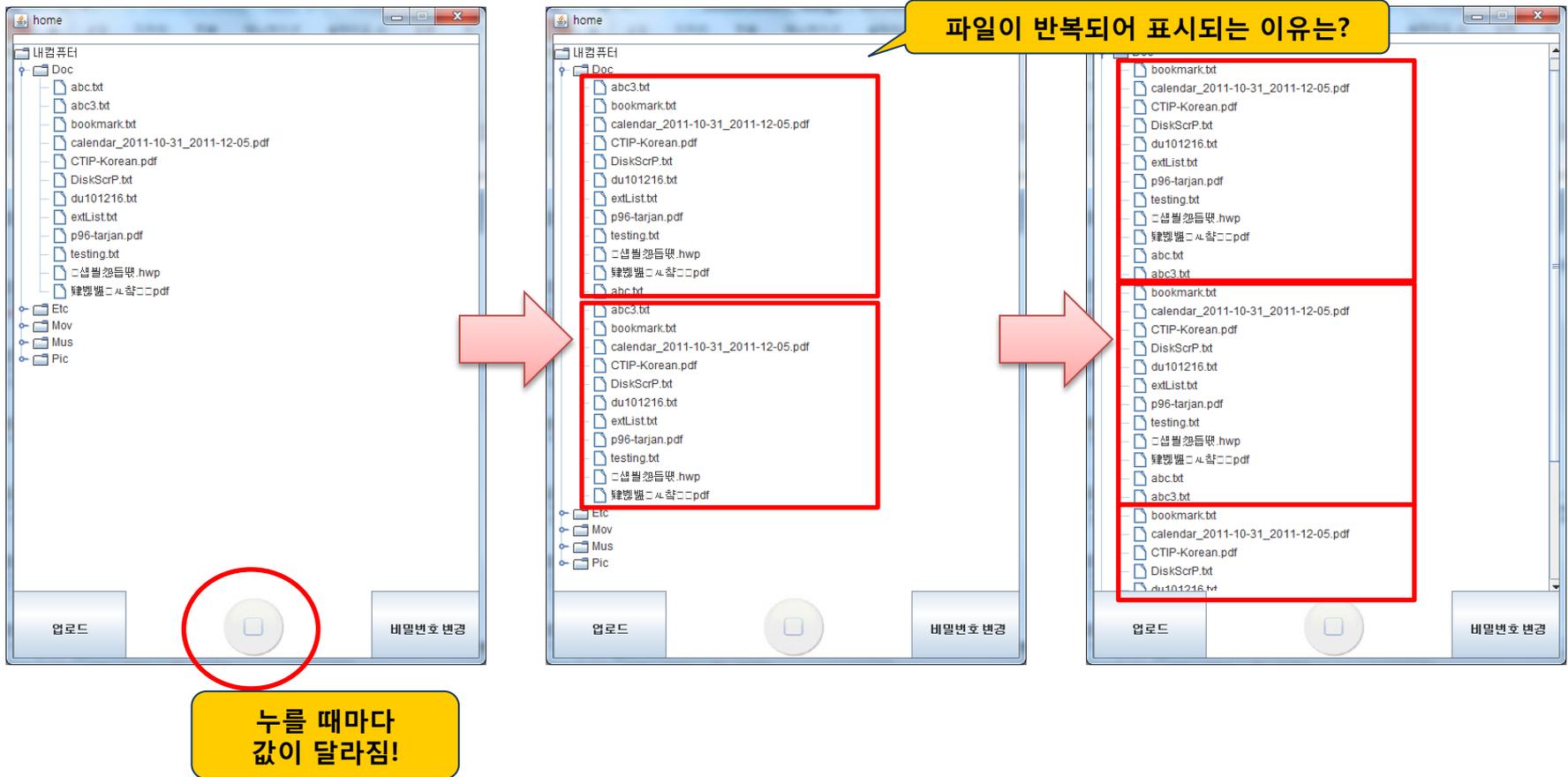
#2. 확장자 추가/삭제

Usecase #6 확장자 추가하기



#3. 기타 버그

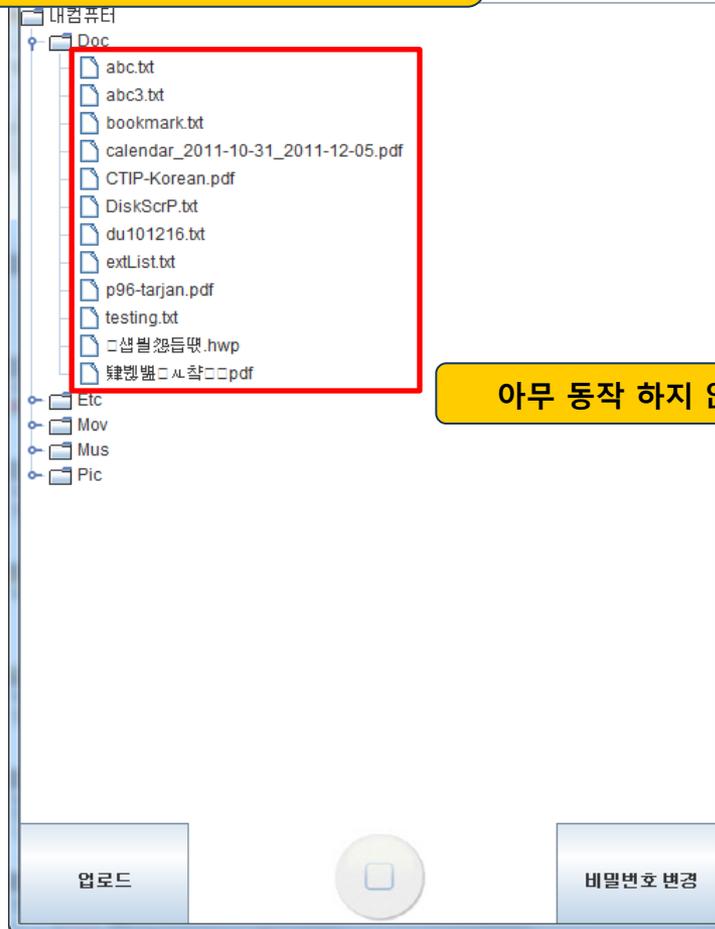
Usecase #9 홈화면 버튼



#3. 기타 버그

Usecase #11 파일 삭제하기 (업로드모드시에만 가능)

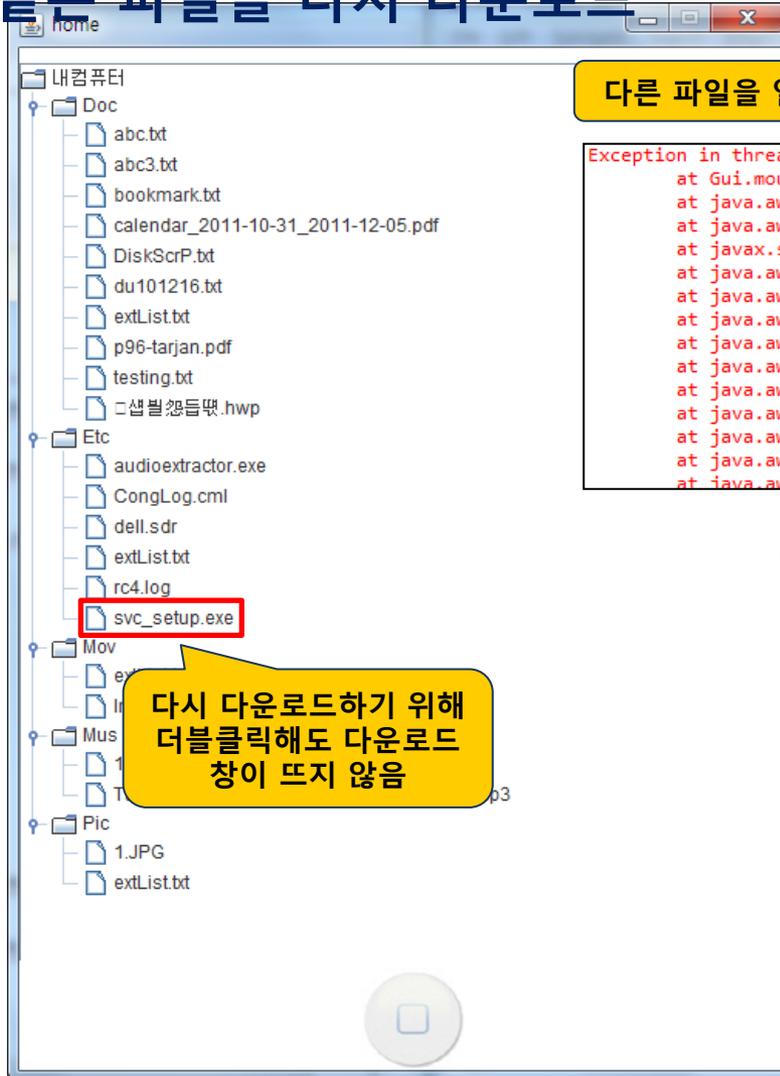
삭제할 파일을 오른쪽 더블클릭하여
파일을 삭제한다?



아무 동작 하지 않음!

#3. 기타 버그

- 같은 파일을 다시 다운로드

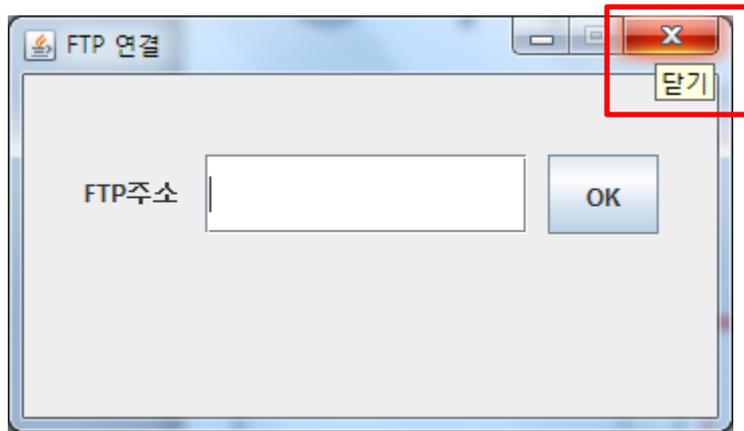


다른 파일을 열고, 취소를 했을때, 콘솔창에 뜨는 에러창

```
Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException
    at Gui.mouseClicked(Gui.java:1026)
    at java.awt.AWTEventMulticaster.mouseClicked(Unknown Source)
    at java.awt.Component.processMouseEvent(Unknown Source)
    at javax.swing.JComponent.processMouseEvent(Unknown Source)
    at java.awt.Component.processEvent(Unknown Source)
    at java.awt.Container.processEvent(Unknown Source)
    at java.awt.Component.dispatchEventImpl(Unknown Source)
    at java.awt.Container.dispatchEventImpl(Unknown Source)
    at java.awt.Component.dispatchEvent(Unknown Source)
    at java.awt.LightweightDispatcher.retargetMouseEvent(Unknown Source)
    at java.awt.LightweightDispatcher.processMouseEvent(Unknown Source)
    at java.awt.LightweightDispatcher.dispatchEvent(Unknown Source)
    at java.awt.Container.dispatchEventImpl(Unknown Source)
    at java.awt.Window.dispatchEventImpl(Unknown Source)
```

#3. 기타 버그

- 프로그램 종료시



닫기가 되지 않는다!

```
Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException
    at Gui.windowClosing(Gui.java:1214)
    at java.awt.Window.processWindowEvent(Unknown Source)
    at javax.swing.JFrame.processWindowEvent(Unknown Source)
    at java.awt.Window.dispatchEvent(Unknown Source)
    at java.awt.Component.dispatchEventImpl(Unknown Source)
    at java.awt.Container.dispatchEventImpl(Unknown Source)
    at java.awt.Window.dispatchEventImpl(Unknown Source)
    at java.awt.Component.dispatchEvent(Unknown Source)
    at java.awt.EventQueue.dispatchEventImpl(Unknown Source)
    at java.awt.EventQueue.access$000(Unknown Source)
    at java.awt.EventQueue$1.run(Unknown Source)
    at java.awt.EventQueue$1.run(Unknown Source)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.security.AccessControlContext$1.doIntersectionPrivilege(Unknown Source)
    at java.security.AccessControlContext$1.doIntersectionPrivilege(Unknown Source)
    at java.awt.EventQueue$2.run(Unknown Source)
    at java.awt.EventQueue$2.run(Unknown Source)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.security.AccessControlContext$1.doIntersectionPrivilege(Unknown Source)
    at java.awt.EventQueue.dispatchEvent(Unknown Source)
    at java.awt.EventDispatchThread.pumpOneEventForFilters(Unknown Source)
    at java.awt.EventDispatchThread.pumpEventsForFilter(Unknown Source)
    at java.awt.EventDispatchThread.pumpEventsForHierarchy(Unknown Source)
    at java.awt.EventDispatchThread.pumpEvents(Unknown Source)
    at java.awt.EventDispatchThread.pumpEvents(Unknown Source)
    at java.awt.EventDispatchThread.run(Unknown Source)
```

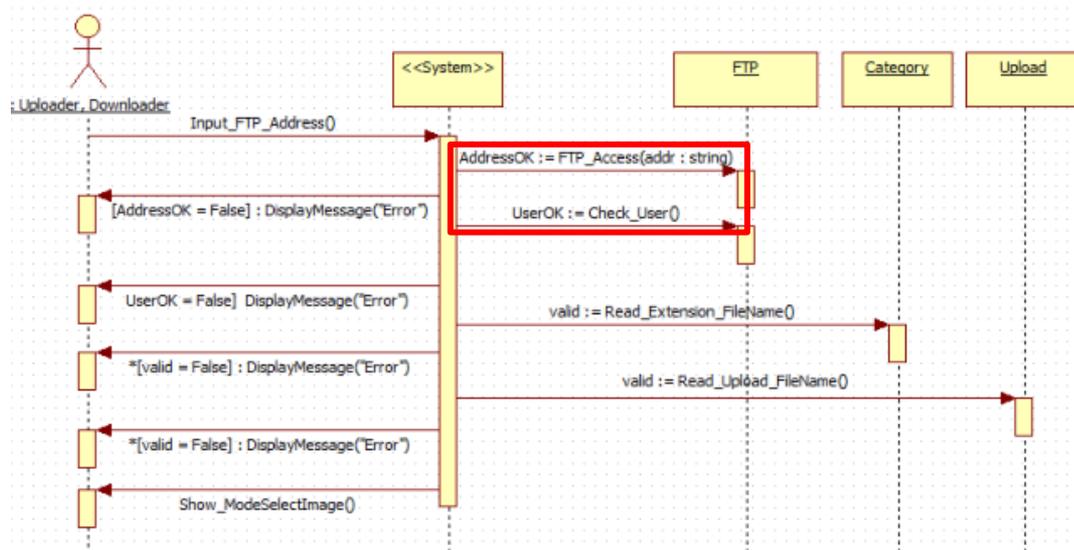
Testing [Team 5]

“ 자동분류 자료관리 시스템 ”



Inspection – Document & Code

#6. FTP 연결

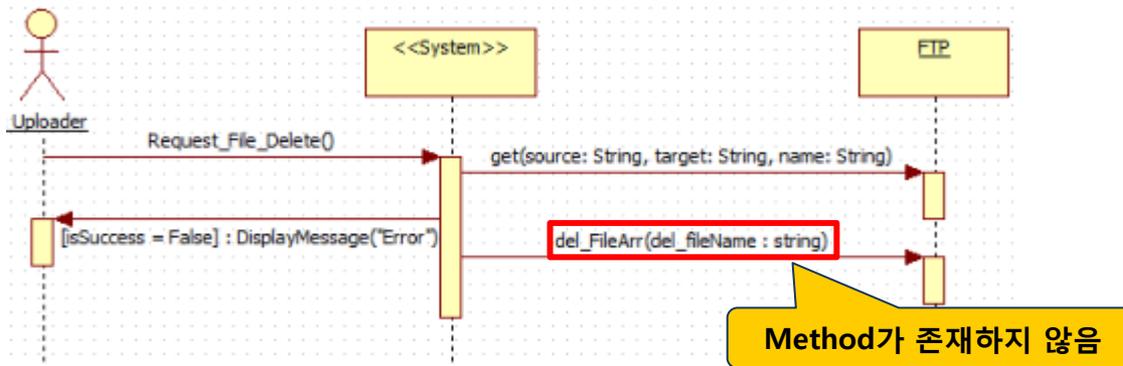


```
FTP
  S server : String
  S port : int
  S id : String
  S password : String
  S ftpClient : FTPClient
  C FTP(String, int, String, String)
  S main(String[]) : void
  C createDirFile() : void
  S isVeriEnabled() : boolean
  C login(String, String) : boolean
  C logout() : boolean
  C connect() : int
  C createDirectory(String) : File
  C delFile(String) : File
  C list() : FTPFile[]
  S get(String, String, String) : boolean
  S autoclassify(String, Category) : String
  S put(String, String, String) : boolean
  C cd(String) : void
  C disconnect() : void
```

FTP 클래스 안에 FTP_Access, Check_User() 메소드가 존재하지 않음

Inspection – Document & Code

#15. 파일 다운로드



```
FTP
  S server : String
  S port : int
  S id : String
  S password : String
  S ftpClient : FTPClient
  C FTP(String, int, String, String)
  S main(String[]) : void
  createDirFile() : void
  isVeriEnabled() : boolean
  login(String, String) : boolean
  logout() : boolean
  connect() : int
  createDirectory(String) : File
  delFile(String) : File
  list() : FTPFile[]
  get(String, String, String) : boolean
  autclassify(String, Category) : String
  put(String, String, String) : boolean
  cd(String) : void
  disconnect() : void
```

Testing [Team 5]

“ 자동분류 자료관리 시스템 ”



Parameterized Test

Parameterized Test

◆ Parameterized Test

테스트에 사용되는 각종 데이터를 파라미터로 뽑아내고, 뽑아낸 파라미터 값을 매번 바꿔가며 테스트하는 Junit Testing 기법

→ Pairwise가 여러 개인 경우, 테스트 케이스를 따로따로 만들지 않고, Parameterized Test를 이용해 하나의 테스트 케이스를 여러 번 반복 실행

```
@RunWith(Parameterized.class)
public class PasswordTest_BooleanMethod {
    private String result;
    private boolean returnval;
    Password PasswordTest;

    @Parameters
    public static Collection createN10Data(){
        return Arrays.asList(new Object[][] {
            {"1234", true},
            {"aa", false},
            {null, false}, // null 처리 안함거 error
        });
    }

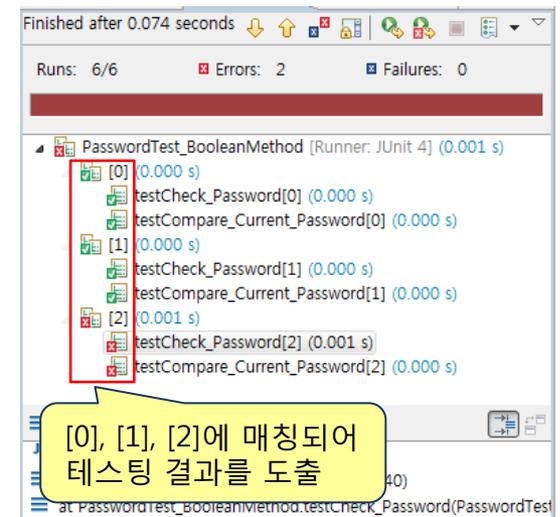
    public PasswordTest_BooleanMethod(String result, boolean returnval) {
        this.result = result;
        this.returnval = returnval;
    }

    @Test
    public void testCheck_Password() throws IOException {
        PasswordTest = new Password();
        assertThat( PasswordTest.Check_Password(this.result), is(returnval));
    }
}
```

RunWith(Parameterized.class)
어노테이션 부여

→ 생성자가 파라미터를 받아 파라미터 값을 받음

파라미터값을 input으로 하여 assert문에 삽입



[0], [1], [2]에 매칭되어 테스트 결과를 도출

Parameterized Test

◆ *Parameterized Test in our testing*

```
@RunWith(Parameterized.class)
public class PasswordTest_BooleanMethod {
    private String result;
    private boolean returnval;
    Password PasswordTest;

    @Parameters
    public static Collection createN10Data(){
        return Arrays.asList(new Object[][] {
            {"1234", true},
            {"aa", false},
            {null, false}, // null
        });
    }

    public PasswordTest_BooleanMethod(String result, boolean returnval) {
        this.result = result;
        this.returnval = returnval;
    }

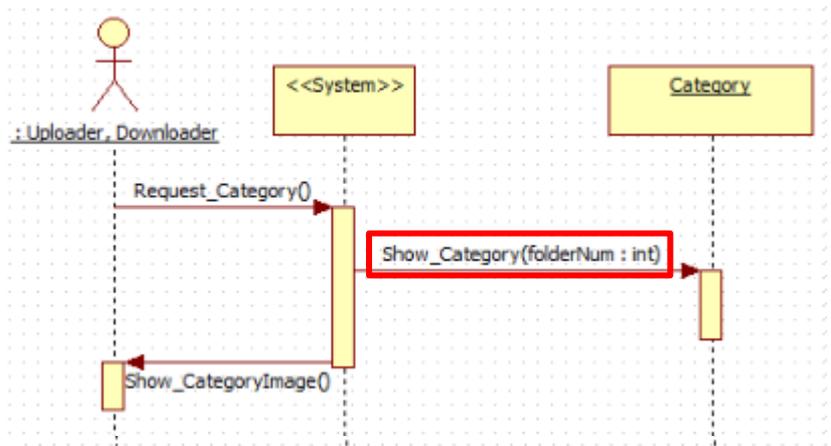
    @Test
    public void testCheck_Password() throws IOException {
        PasswordTest = new Password();
        assertTrue(PasswordTest.Check_Password(this.result), is(returnval));
    }
}
```

Pairwise해서 나온 input값을 입력하고,
마지막 값에는 input값에 대한 기대
output값을 입력

Assert문을 이용해 입력한 input에 대해
기대 output값이 일치하는지 testing

Unit Testing – JUnit Code

#2. 카테고리별 확장자 보기



Output이 null 값으로 나올 경우에 대해 exception 처리를 하지 않음

Show_Category(folderNum : int)

값	Input	output
[0]	Int.Max_Val	Null
[1]	1	"jpg bmp "
[2]	0	"mp3 "
[3]	-1	Null
[4]	Int.Min_Val	Null

Finished after 0.079 seconds

Runs: 5/5 Errors: 3 Failures: 0

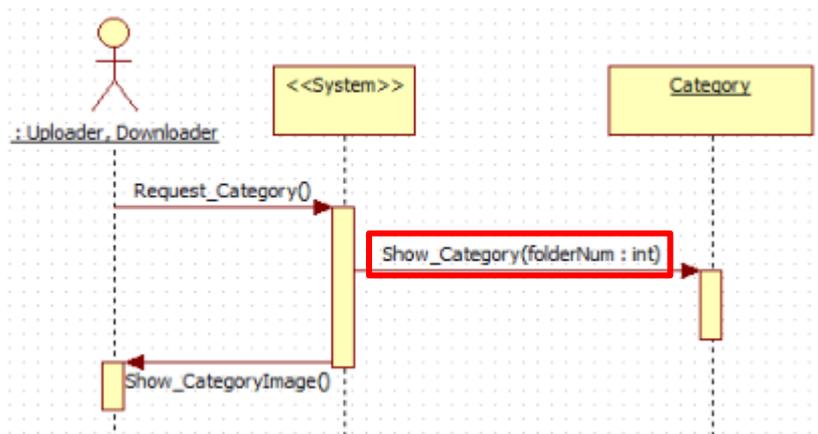
```
n02RequestCategoryTest [Runner: JUnit 4] (0.001 s)
├─ [0] (0.000 s)
│   └─ testShow_Category[0] (0.000 s)
├─ [1] (0.000 s)
├─ [2] (0.000 s)
├─ [3] (0.000 s)
│   └─ testShow_Category[3] (0.000 s)
└─ [4] (0.001 s)
```

Failure Trace

```
java.lang.ArrayIndexOutOfBoundsException: 2147483647
    at Category.Show_Category(Category.java:37)
    at n02RequestCategoryTest.testShow_Category(n02RequestCategoryTest.java:37)
```

Unit Testing – JUnit Code

#2. 카테고리별 확장자 보기



Show_Category(folderNum : int)

값	Input	output
[0]	Int.Max_Val	Null
[1]	1	"jpg bmp "
[2]	0	"mp3 "
[3]	-1	Null
[4]	Int.Min_Val	Null

Finished after 0.079 seconds

Runs: 5/5 Errors: 3 Failures: 0

```

n02RequestCategoryTest [Runner: JUnit 4] (0.001 s)
├─ [0] (0.000 s)
│   └─ testShow_Category[0] (0.000 s)
├─ [1] (0.000 s)
├─ [2] (0.000 s)
├─ [3] (0.000 s)
│   └─ testShow_Category[3] (0.000 s)
└─ [4] (0.001 s)
    
```

Failure Trace

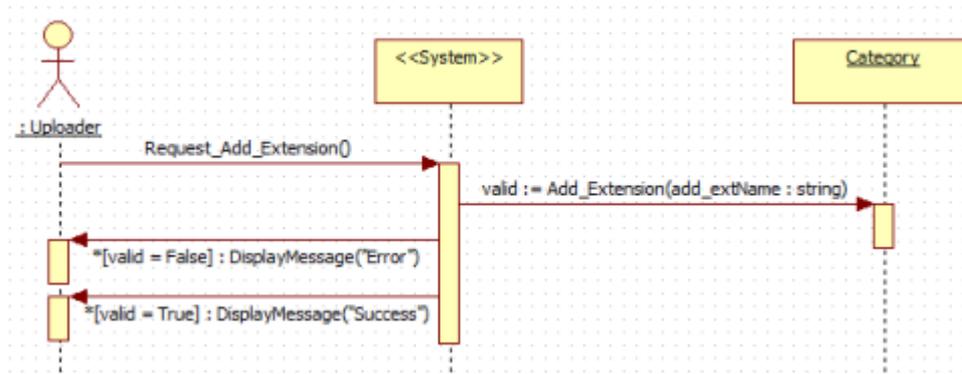
```

java.lang.ArrayIndexOutOfBoundsException: 2147483647
    at Category.Show_Category(Category.java:37)
    at n02RequestCategoryTest.testShow_Category(n02RequestCategoryTest.java:37)
    
```

Output이 null 값으로 나올 경우에 대해 exception 처리를 하지 않음

Unit Testing – JUnit Code

#3. 확장자 추가



Use case	3. 확장자 추가
Actor	업로더
Purpose	등록되지 않은 파일에 대한 확장자를 추가시킨다.
Overview	업로더는 이 기능을 통해 새로운 파일형식에 대해서 이 파일이 어떤 카테고리로 분류 될 수 있는지 정할 수 있다.
Type	Primary
Cross Reference	Functions : R1, R2.1, R2.2.1, R2.2.2, R2.3 Use cases : 확장자 기반 파일 분류, 카테고리 별 확장자 보기, 중복 확장자 검사, 확장자 삭제
Pre-Requisites	업로더 모드로 로그인 후 카테고리 별 확장자 보기가 활성화 되어 있어야 한다.
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) : 확장자 추가 버튼을 누른다. 2. (A) : 추가될 확장자를 입력한다. 3. (S) : 확장자가 추가된다.
Alternative Courses of Events	None
Exceptional Courses of Events	None

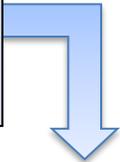
- 1 : 이상한 문자열을 받을 경우
- 2 : 이미 있는 문자열일 경우
- 0 : 제대로 될 경우

Unit Testing – JUnit Code

#3. 확장자 추가

※ Pairwise 값 생성

	A	B
1	add_extName	FolderNum
2	aa	Integer.MAX_VALUE
3	mp3	1
4	null	0
5		-1
6		Integer.MIN_VALUE



	A	B	C	D
8	TEST CASES			
9	case	add_extName	FolderNum	pairings
10	1	aa	Integer.MAX_VALUE	1
11	2	mp3	Integer.MAX_VALUE	1
12	3	null	Integer.MAX_VALUE	1
13	4	aa		1 1
14	5	mp3		1 1
15	6	null		1 1
16	7	aa		0 1
17	8	mp3		0 1
18	9	null		0 1
19	10	aa		-1 1
20	11	mp3		-1 1
21	12	null		-1 1
22	13	aa	Integer.MIN_VALUE	1
23	14	mp3	Integer.MIN_VALUE	1
24	15	null	Integer.MIN_VALUE	1

Add_Extension(add_extName : String)

값	Input1	Input2	Output
[0]	"aa"	Int.Max_Val	0
[1]	"mp3"	Int.Max_Val	0
[2]	null	Int.Max_Val	1
[3]	"aa"	1	0
[4]	null	1	1
...			
[12]	Null	Int.Min_Val	1v

Method가 올바르게 동작하나, Null, 범위를 넘어가는 int에 대한 exception 처리가 되어있지 않음

Finished after 0.097 seconds

Runs: 16/16 Errors: 4 Failures: 0

```

PasswordTest_Compare_New_Password [Runner: JUnit 4] (0.045 s)
  [0] (0.005 s)
  [1] (0.002 s)
  [2] (0.002 s)
  [3] (0.002 s)
  [4] (0.009 s)
  [5] (0.002 s)
  [6] (0.001 s)
  [7] (0.001 s)
  [8] (0.002 s)
  [9] (0.001 s)
  [10] (0.001 s)
  [11] (0.002 s)
  [12] (0.007 s)
    testCompare_New_Password[12] (0.007 s)
  [13] (0.001 s)
    testCompare_New_Password[13] (0.001 s)
  [14] (0.002 s)
  [15] (0.003 s)
  
```

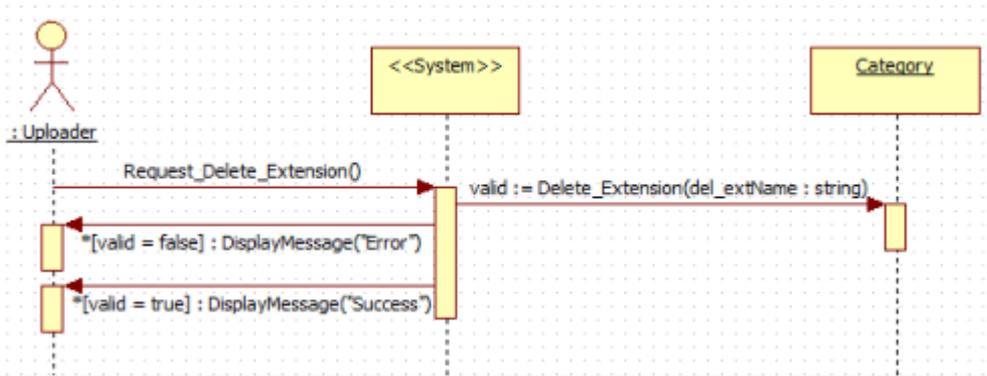
Failure Trace

```

java.lang.NullPointerException
  at Password.Compare_New_Password(Password.java:67)
  at PasswordTest_Compare_New_Password.testCompare_New_Password
  
```

Unit Testing – JUnit Code

#5. 확장자 삭제



Use case	3. 확장자 추가
Actor	업로더
Purpose	등록되지 않은 파일에 대한 확장자를 추가시킨다.
Overview	업로더는 이 기능을 통해 새로운 파일형식에 대해서 이 파일이 어떤 카테고리로 분류 될 수 있는지 정할 수 있다.
Type	Primary
Cross Reference	Functions : R1, R2.1, R2.2.1, R2.2.2, R2.3 Use cases : 확장자 기반 파일 분류, 카테고리 별 확장자 보기, 중복 확장자 검사, 확장자 삭제
Pre-Requisites	업로더 모드로 로그인 후 카테고리 별 확장자 보기가 활성화 되어 있어야 한다.
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) : 확장자 추가 버튼을 누른다. 2. (A) : 추가될 확장자를 입력한다. 3. (S) : 확장자가 추가된다.
Alternative Courses of Events	None
Exceptional Courses of Events	None

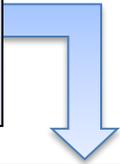
- 1 : 이상한 문자열을 받을 경우
- 2 : 이미 있는 문자열일 경우
- 0 : 제대로 될 경우

Unit Testing – JUnit Code

#5. 확장자 삭제

※ Pairwise 값 생성

	A	B
1	add_extName	FolderNum
2	aa	Integer.MAX_VALUE
3	mp3	1
4	null	0
5		-1
6		Integer.MIN_VALUE



	A	B	C	D
8	TEST CASES			
9	case	add_extName	FolderNum	pairings
10	1	aa	Integer.MAX_VALUE	1
11	2	mp3	Integer.MAX_VALUE	1
12	3	null	Integer.MAX_VALUE	1
13	4	aa		1 1
14	5	mp3		1 1
15	6	null		1 1
16	7	aa		0 1
17	8	mp3		0 1
18	9	null		0 1
19	10	aa		-1 1
20	11	mp3		-1 1
21	12	null		-1 1
22	13	aa	Integer.MIN_VALUE	1
23	14	mp3	Integer.MIN_VALUE	1
24	15	null	Integer.MIN_VALUE	1

Delete_Extension(del_extName : String)

값	Input1	Input2	Output
[0]	"aa"	Int.Max_Val	False
[1]	"mp3"	Int.Max_Val	False
...			
[6]	"aa"	0	False
[7]	"mp3"	0	True
...			
[11]	"aa"	Int.Min_Val	False
[12]	"mp3"	Int.Min_Val	False

Finished after 0.113 seconds

Runs: 15/15 Errors: 0 Failures: 0

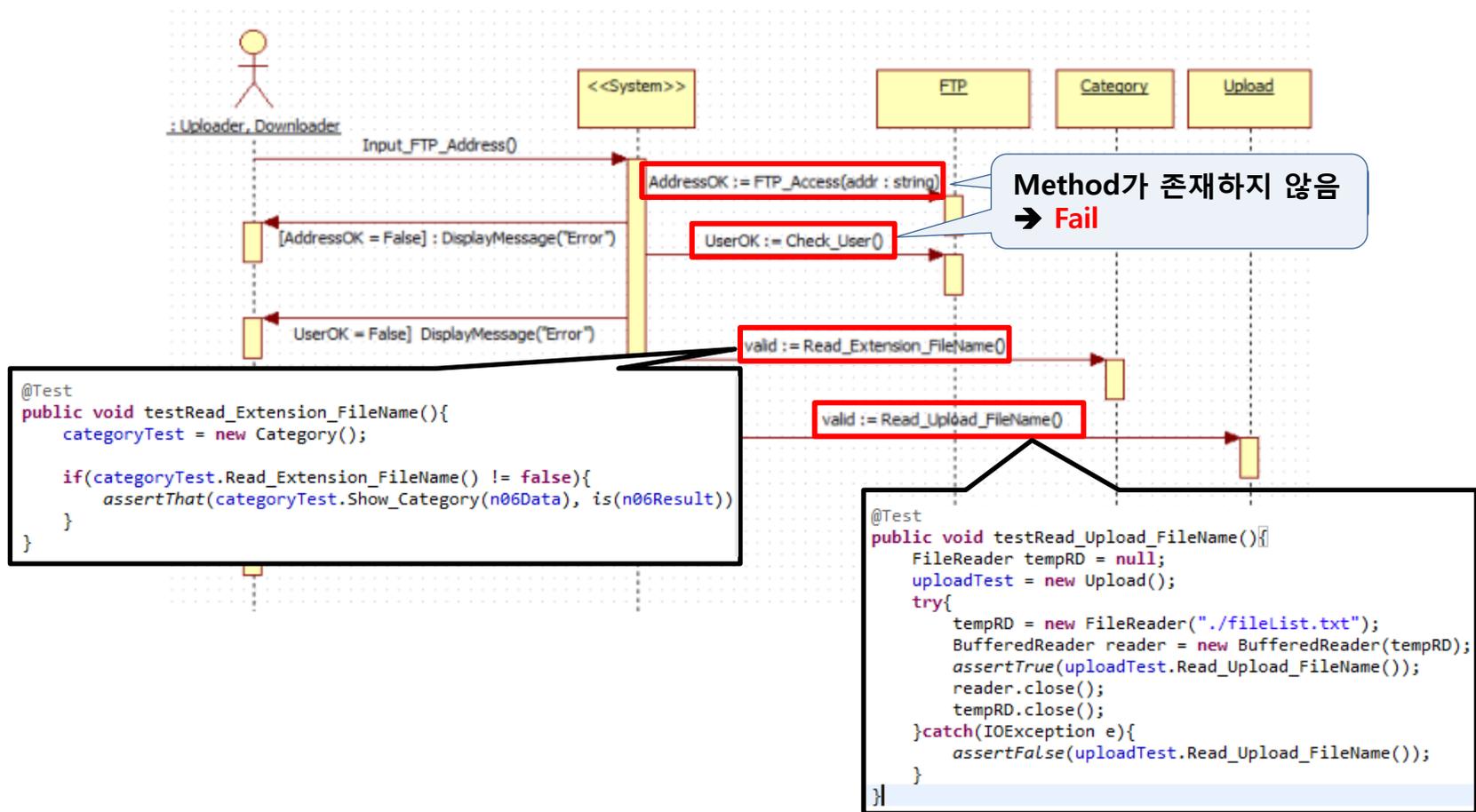
```

n05RequestDeleteExtension [Runner: JUnit 4] (0.001 s)
  [0] (0.000 s)
  [1] (0.000 s)
  [2] (0.000 s)
  [3] (0.000 s)
  [4] (0.000 s)
  [5] (0.000 s)
  [6] (0.000 s)
  [7] (0.000 s)
  [8] (0.000 s)
  [9] (0.000 s)
  [10] (0.000 s)
  [11] (0.001 s)
  [12] (0.000 s)
  [13] (0.000 s)
  [14] (0.000 s)
    
```

Success!

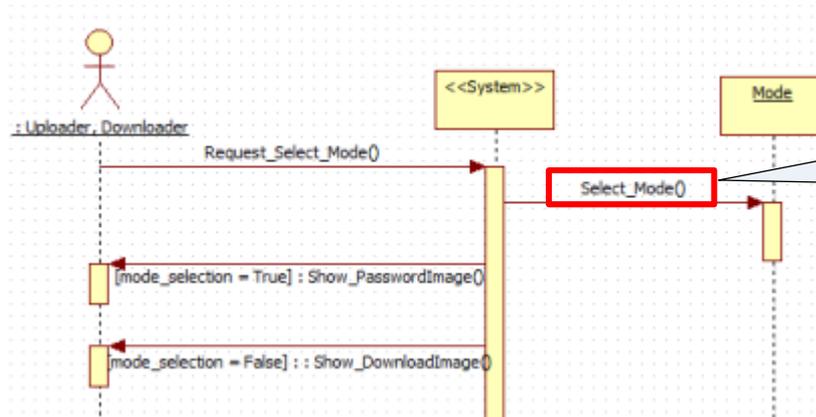
Unit Testing – JUnit Code

#6. FTP 연결



Unit Testing – JUnit Code

#7. 모드선택



False였던 mode_selection를 true로 바꿈.

Select_Mode()

```
Mode modeTest;|
@Test
public void testSelect_Mode() {
    modeTest = new Mode();
    assertFalse(modeTest.mode_selection);
    modeTest.Select_Mode();
    assertTrue(modeTest.mode_selection);
}
```

Select_Mode()를 실행 전, 실행 후로 테스트.

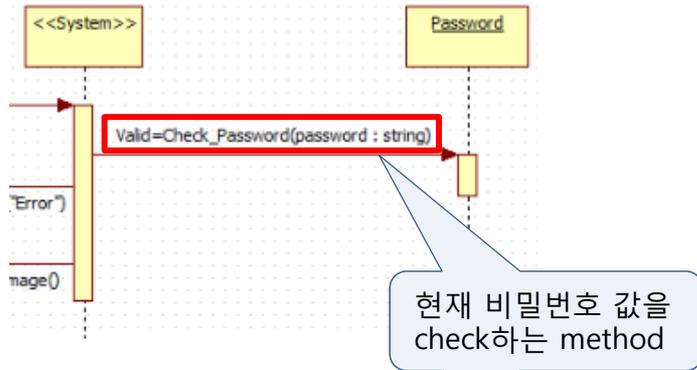
Finished after 0.027 seconds

Runs: 1/1 Errors: 0 Failures: 0

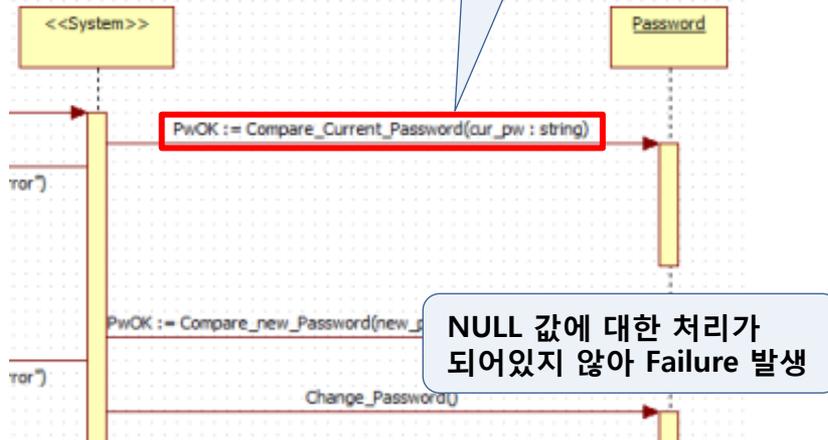
n07ModeTest [Runner: JUnit 4] (0.000 s)

Unit Testing – JUnit Code

#8. 업로더 모드 로그인



#10. 비밀번호 변경



Check_Password(), Compare_Current_Password()

값	Input	기대 output
[0]	"1234"	True
[1]	"abcd"	False
[2]	Null	False

Finished after 0.074 seconds

Runs: 6/6 Errors: 2 Failures: 0

```

PasswordTest_BooleanMethod [Runner: JUnit 4] (0.001 s)
├─ [0] (0.000 s)
│   ├── testCheck_Password[0] (0.000 s)
│   └── testCompare_Current_Password[0] (0.000 s)
├─ [1] (0.000 s)
│   ├── testCheck_Password[1] (0.000 s)
│   └── testCompare_Current_Password[1] (0.000 s)
└─ [2] (0.001 s)
    ├── testCheck_Password[2] (0.001 s)
    └── testCompare_Current_Password[2] (0.000 s)
    
```

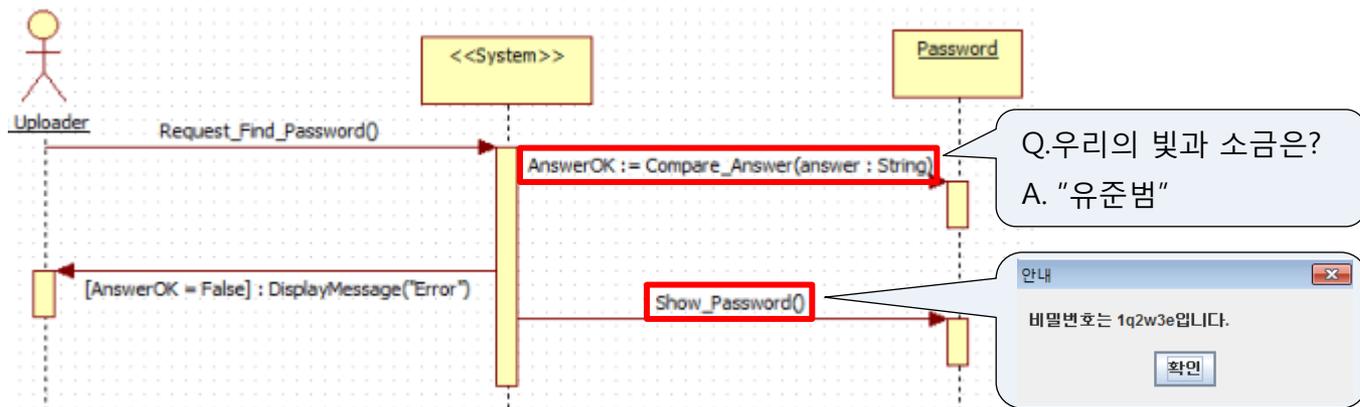
Failure Trace

```

java.lang.NullPointerException
at Password.Check_Password(Password.java:40)
at PasswordTest_BooleanMethod.testCheck_Password(PasswordTes
    
```

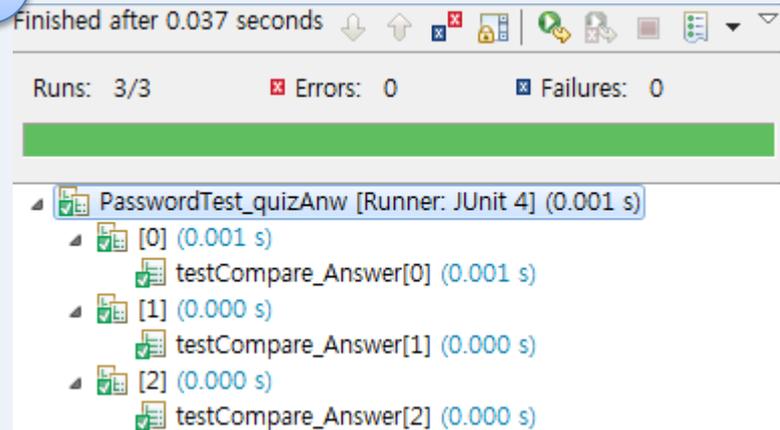
Unit Testing – JUnit Code

#9. 비밀번호 찾기



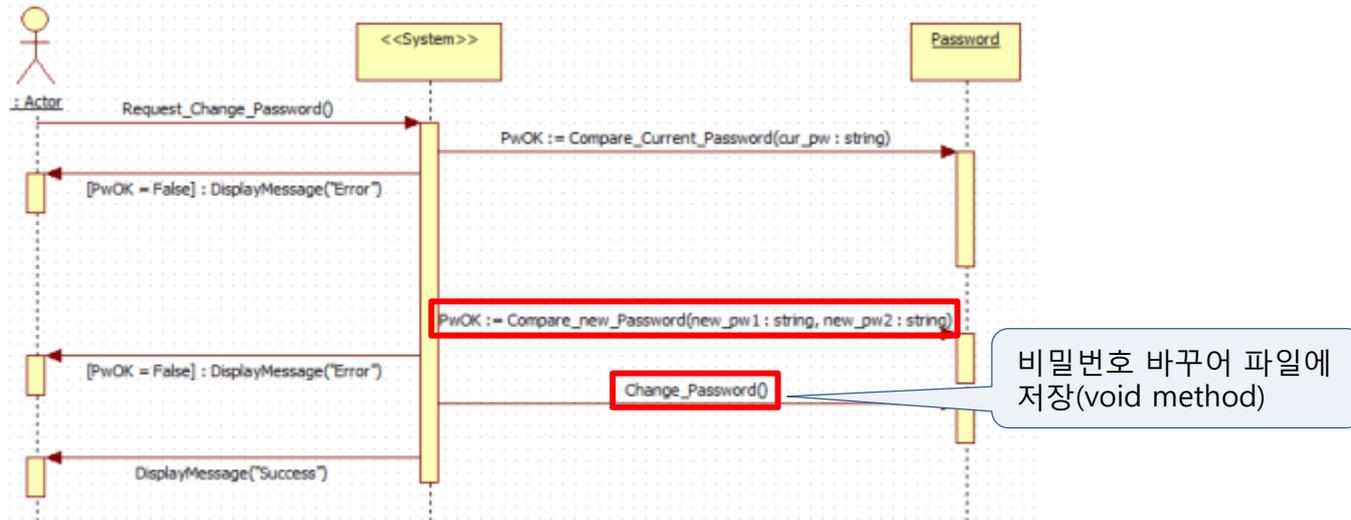
Compare_Answer(answer : String)

값	Input	기대 output
[0]	"유준범"	True
[1]	"abcd"	False
[2]	Null	False



Unit Testing – JUnit Code

#10. 비밀번호 변경



Compare_Answer(answer : String)

- i) $Str1 < 4$ or $Str1 > 8$
→ return 1
- ii) $4 < str1 < 8$
 - 현재패스워드와 비교
→ return 0
 - else → return 2

Unit Testing – JUnit Code

#10. 비밀번호 변경

※ Pairwise 값 생성

	A	B
1	string1	string2
2	1234	1234
3	123456	123456
4	123456789	123456789
5	null	null

1	TEST CASES			
3	case	string1	string2	pairings
4	1	1234	1234	1
5	2	1234	123456	1
6	3	1234	123456789	1
7	4	1234	null	1
8	5	123456	1234	1
9	6	123456	123456	1
10	7	123456	123456789	1
11	8	123456	null	1
12	9	123456789	1234	1
13	10	123456789	123456	1
14	11	123456789	123456789	1
15	12	123456789	null	1
16	13	null	1234	1
17	14	null	123456	1
18	15	null	123456789	1
19	16	null	null	1

Compare_Answer(answer : String)

값	Input1	Input2	Output
[0]	"1234"	"1234"	1
[1]	"1234"	"123456"	1
[2]	"1234"	"123456789"	1
[3]	"1234"	null	1
[4]	"123456"	"1234"	2
...			
[12]	Null	Null	1

Method가 올바르게 동작하나, Null에 대한 exception 처리가 되어있지 않음

Finished after 0.097 seconds

Runs: 16/16 Errors: 4 Failures: 0

```

PasswordTest_Compare_New_Password [Runner: JUnit 4] (0.045 s)
  [0] (0.005 s)
  [1] (0.002 s)
  [2] (0.002 s)
  [3] (0.002 s)
  [4] (0.009 s)
  [5] (0.002 s)
  [6] (0.001 s)
  [7] (0.001 s)
  [8] (0.002 s)
  [9] (0.001 s)
  [10] (0.001 s)
  [11] (0.002 s)
  [12] (0.007 s)
    testCompare_New_Password[12] (0.007 s)
  [13] (0.001 s)
    testCompare_New_Password[13] (0.001 s)
  [14] (0.002 s)
  [15] (0.003 s)
  
```

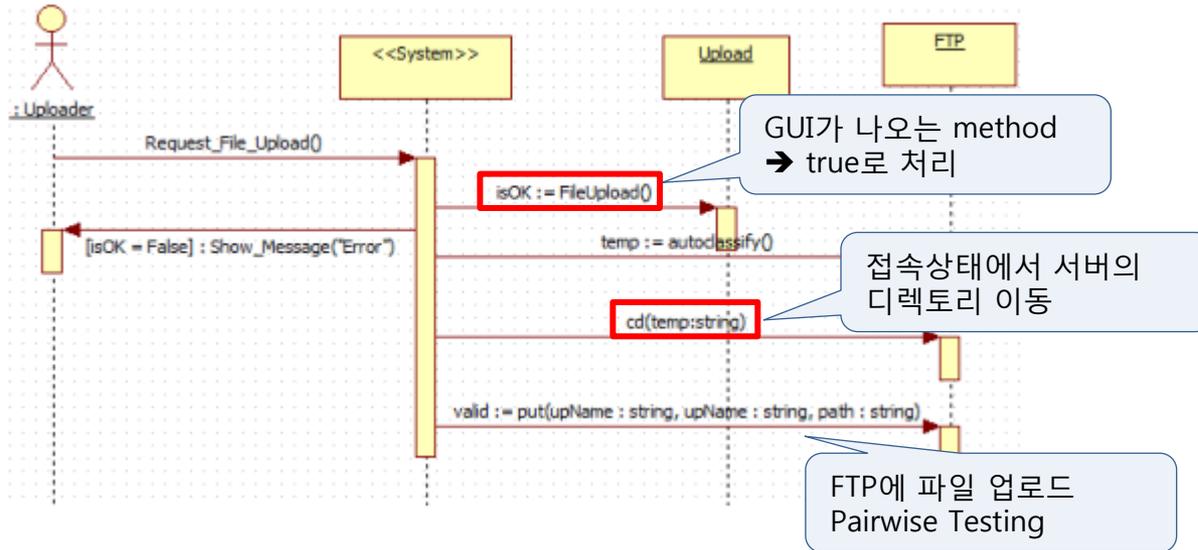
Failure Trace

```

java.lang.NullPointerException
  at Password.Compare_New_Password(Password.java:67)
  at PasswordTest_Compare_New_Password.testCompare_New_Password
  
```

Unit Testing – JUnit Code

#11. 파일 업로드

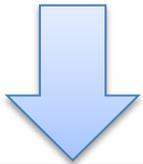


Unit Testing – JUnit Code

#11. 파일 업로드

※ Pairwise 값 생성

	A	B	C
57	n11FileNameDa	n11TargetName	n11UpDirData
58	build.xml	build.xml	./
59	null	null	null
60	"aa"	"aa"	"aa"



build.xml	build.xml	./
build.xml	null	null
build.xml	aa	aa
null	build.xml	null
null	null	./
null	aa	./
aa	build.xml	aa
aa	null	./
aa	aa	null
null	null	aa

`put(upName : String, upName : String, path : String)`

값	Input1	Input2	Input3	Output
[0]	"build.xml"	"build.xml"	./	True
[1]	"build.xml"	Null	Null	True
[2]	"build.xml"	"aa"	"aa"	True
[3]	Null	Null	./	True
[4]	Null	"aa"	./	True
...				
[8]	aa	Null	"aa"	true

파일 입출력의 경우 I/O Exception 이 걸리는 경우 프로그램이 죽는다. 파일 I/O Exception은 되어있는데, 디렉토리 경로가 Null로 오는데에 대한 error처리는 되어있지 않음

Finished after 0.577 seconds

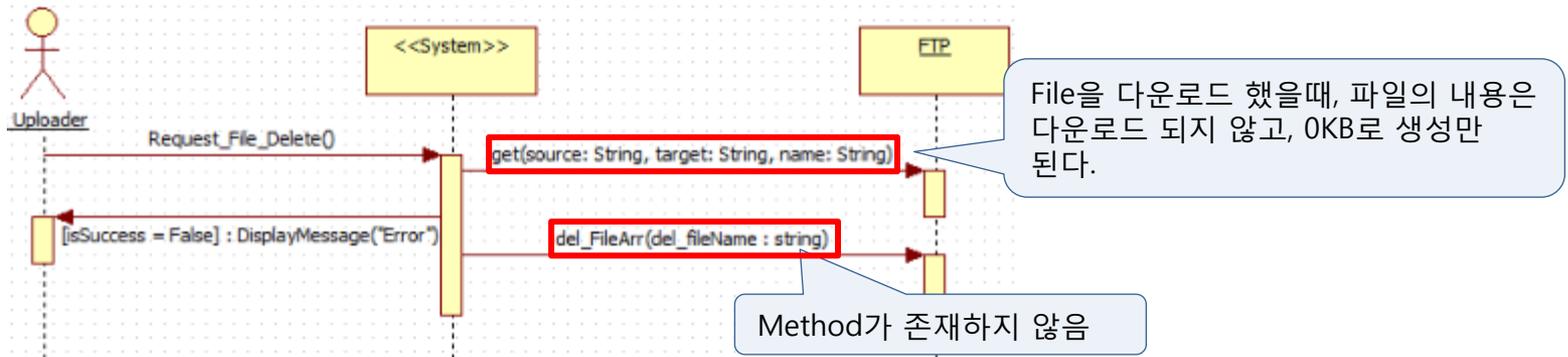
Runs: 30/30 Errors: 4 Failures: 6

```

n11RequestFileUploadTest [Runner: JUnit4]
  [0] (0.327 s)
  [1] (0.000 s)
  [2] (0.016 s)
  [3] (0.016 s)
  [4] (0.000 s)
  [5] (0.015 s)
  [6] (0.016 s)
  [7] (0.015 s)
  [8] (0.000 s)
    testFileUpload[8] (0.000 s)
    testCd[8] (0.000 s)
    testPut[8] (0.000 s)
  [9] (0.000 s)
    testFileUpload[9] (0.000 s)
    testCd[9] (0.000 s)
    testPut[9] (0.000 s)
    
```

Unit Testing – JUnit Code

#15. 파일 다운로드

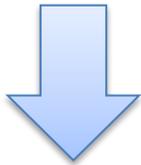


Unit Testing – JUnit Code

#15. 파일 다운로드

※ Pairwise 값 생성

	A	B	C
74	n15SourceData	n15TargetData	n15NameData
75	"n15temp.txt"	"/."	"13.mp3"
76	null	null	null
77	"aa"	"aa"	"aa"



	A	B	C	D
81	case	n15SourceData	n15TargetData	n15NameData
82	1	n15temp.txt	/.	13.mp3
83	2	n15temp.txt	null	null
84	3	n15temp.txt	aa	aa
85	4	null	/.	null
86	5	null	null	13.mp3
87	6	null	aa	13.mp3
88	7	aa	/.	aa
89	8	aa	null	13.mp3
90	9	aa	aa	null
91	10	null	null	aa

`put(upName : String, upName : String, path : String)`

값	Input1	Input2	Input3	Output
[0]	"n15temp.txt"	"/."	"13.mp3"	True
[1]	"n15temp.txt"	Null	Null	False
[2]	Null	"/."	Null	False
[3]	Null	Null	"13.mp3"	False
[4]	"aa"	"/."	"13.mp3"	True
...				
[9]	Null	Null	"aa"	False

경로에 대한 error처리가 되어있지 않고, success라고 나왔더라도 파일 다운로드의 기능이 제대로 수행이 되지 않음!

Finished after 0.825 seconds

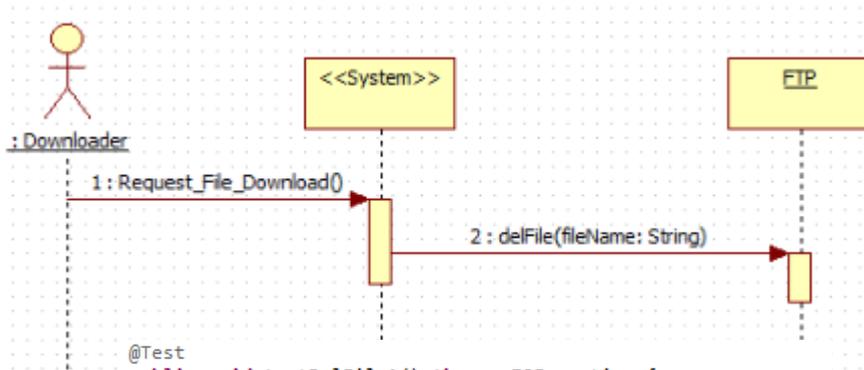
Runs: 20/20 Errors: 4 Failures: 12

```

n15RequestFileDownloadTest [Runner: JUnit 4]
  [x] [0] (0.323 s)
  [x] [1] (0.127 s)
  [x] [2] (0.003 s)
  [x] [3] (0.125 s)
  [x] [4] (0.000 s)
  [x] [5] (0.001 s)
  [x] [6] (0.030 s)
  [x] testGet[6] (0.029 s)
  [x] testdel_FileArr[6] (0.001 s)
  [x] [7] (0.025 s)
  [x] [8] (0.015 s)
  [x] testGet[8] (0.015 s)
  [x] testdel_FileArr[8] (0.000 s)
  [x] [9] (0.006 s)
  [x] testGet[9] (0.005 s)
  [x] testdel_FileArr[9] (0.001 s)
    
```

Unit Testing – JUnit Code

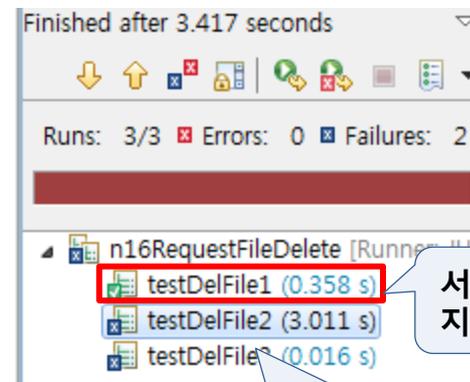
#16. 파일 삭제



```
@Test
public void testDelFile1() throws IOException {
    ftpTest = new FTP("smat5all.no-ip.org",2221,"admin",null);
    ftpTest.connect();
    assertNull(ftpTest.delFile("abc.txt"));
}

@Test
public void testDelFile2() throws IOException {
    ftpTest = new FTP("smat5all.no-ip.org",2221,"admin",null);
    ftpTest.connect();
    assertNotNull(ftpTest.delFile("aa"));
}

@Test
public void testDelFile3() throws IOException {
    ftpTest = new FTP("smat5all.no-ip.org",2221,"admin",null);
    ftpTest.connect();
    assertNotNull(ftpTest.delFile(null));
}
```



서버에 있는 abc.txt파일이 지워졌는지.. 알수 없음

“aa.txt”, null값 모두 기대값을 null이 나오지 않을 것으로 assert문을 사용하였는데, null이 나와서 fail로 처리된다.

Testing [Team 5]

“ 자동분류 자료관리 시스템 ”



Requirement Coverage

Requirement Coverage Summary

Summary (26)	9 (34.62%)	14 (53.85%)	3 (11.54%)
--------------	------------	-------------	------------

Requirement Coverage Details

Sr#	Category	Coverage
1	3.확장자 추가 (1)	1 (100%)
2	7.모드선택 (1)	1 (100%)
3	14.파일 덮어쓰기 (1)	1 (100%)
4	16.파일 삭제 (1)	1 (100%)
5	4.중복확장자 검사 (1)	1 (100%)
6	11.파일 업로드 (4)	2 (50%) 2 (50%)
7	9.비밀번호 찾기 (2)	1 (50%) 1 (50%)
8	10.비밀번호 변경 (3)	1 (33.33%) 2 (66.67%)
9	2.카테고리 별 확장자 보기 (1)	1 (100%)
10	15.파일 다운로드 (2)	2 (100%)
11	13.파일 용량 검사 (1)	1 (100%)
12	8.업로더 모드 로그인 (1)	1 (100%)
13	1.확장자 기반 파일 분류 (1)	1 (100%)
14	5.확장자 삭제 (1)	1 (100%)

Unique Test Methods 109
Requirements:Test Methods Ratio 1:4
Missing Test Methods 1
Unmapped Test Methods 39

All Requirements

- testAutoClassify... (10%)
- testShow_Category (40%)
- testAdd_Extension (20%)
- testCheck_Extens... (0%)
- testDelete_Exten... (100%)
- testFTP_Access (0%)
- testCheck_User (0%)
- testRead_Extensi... (40%)
- testRead_Upload_... (100%)
- testSelect_Mode (100%)
- testCheck_Passwo... (66.67%)
- testCompare_Answ... (100%)
- testShow_Passwor... (0%)
- testCompare_Curr... (66.67%)
- testCompare_New_... (75%)
- testChange_Passw... (100%)
- testFileUpload (100%)
- testAutoClassify (0%)
- testcd (100%)
- testput (0%)

지금까지 작성한 Test Case에 대해 Coverage를 산출

Requirement Coverage

지금까지 작성한 Test Case에 대해 Coverage를 산출

Requirement Coverage Report

Requirement Coverage Summary

Number of Requirements	4
Unique Test Methods	21
Requirements:Test Methods Ratio	1:5
Missing Test Methods	1

11.파일 업로드 (4) 2 (50%) 2 (50%)

Requirement Coverage Details

Sr#	Coverage Item	Coverage
1.	FileUpload()_Test (1)	1 (100%)
2.	AutoClassify()_Test (1)	1 (100%)
3.	cd()_Test (1)	1 (100%)
4.	put()_Test (1)	1 (100%)

Report generated on 금, 25 5월 2012 02:38:04 KST

Coverage Report

Requirement Coverage Summary

testRead_Extension_FileName_1 2 (40%) 3 (60%) (5)

Requirement Coverage Details

Sr#	Test Methods	Coverage
1.	n06InputFtpAddress.testRead_Extension_FileName[0]	0%
2.	n06InputFtpAddress.testRead_Extension_FileName[1]	100%
3.	n06InputFtpAddress.testRead_Extension_FileName[2]	100%
4.	n06InputFtpAddress.testRead_Extension_FileName[3]	0%
5.	n06InputFtpAddress.testRead_Extension_FileName[4]	0%

Report generated on 금, 25 5월 2012 02:38:04 KST

Testing [Team 5]

“ 자동분류 자료관리 시스템 ”



Hudson Build

Hudson search

Hudson » Team5 2ndTest 자동 재실행 크기

[Back to Dashboard](#)
[Status](#)
[Changes](#)
[Workspace](#)
[Build Now](#)
[Delete Project](#)
[Configure](#)

Project Team5 2ndTest

[Workspace](#)
[Recent Changes](#)
[Latest Test Result \(74 failures / +44\)](#)
[Latest Console output](#)

Test Result Trend

count

#9 #10 #11 #12 #13 #14 #15 #16 #17 #18 #19 #20

(just show failures) enlarge

[소개 내용 입력](#)
[Disable Project](#)

Build History (trend)

#20	2012. 5. 25 오전 2:39:27	▼
#19	2012. 5. 24 오후 11:18:25	▼
#18	2012. 5. 24 오후 10:29:18	▼
#17	2012. 5. 24 오후 9:31:50	▼
#16	2012. 5. 24 오후 9:30:52	▼
#15	2012. 5. 23 오후 8:51:12	▼
#14	2012. 5. 23 오후 8:49:50	▼
#13	2012. 5. 23 오후 8:45:00	▼
#12	2012. 5. 22 오후 8:46:58	▼

Permalinks

- [Last build \(#20\), 26 sec ago](#)
- [Last successful build \(#20\), 26 sec ago](#)
- [Last failed build \(#8\), 2 days 9 hr ago](#)
- [Last unstable build \(#20\), 26 sec ago](#)
- [Last unsuccessful build \(#20\), 26 sec ago](#)

Usecase #6에서 FTP에서 서버 주소에 이상한 값을 넣었을 때, 연결 안되는 것 때문에 Junit이 멈춰버림.

결과 보고서

테스트 보고서 - Team5

1. 프로그램 및 인터페이스

- ① 홈 버튼 클릭 시 동일한 파일이 놓여나서 보임.
- ② 한글 사용 시 파일 이름 깨짐.
- ③ 파일 업로드 기능 시 기존의 파일명이 아니라 직접 입력해야 하는 불편함이 있음.
- ④ 종료 버튼으로 제대로 종료가 안 됨.
- ⑤ 파일 다운로드시 원본과 다른 이름으로 저장할 시 원본 파일명의 파일과 다른 이름의 파일 두 개가 생성됨.
- ⑥ 파일의 전송이 완전하지 않음.

2. Use-Case별 Failed Test Case

Use-Case #1 확장자 기반 파일 분류

Test Method	Failed Test Case	원인분석
n01AutoClassifyTest.testAutoclassify	서버에 적절하지 않은 값이 들어간 경우 포트에 적절하지 않은 값이 들어간 경우	• 예외 처리 필요

Use-Case #2 카테고리 별 확장자 보기

Test Method	Failed Test Case
n03RequestCategoryTest.testShow_Category	네명의 범위를 넘어가는 경우 네명에 음수값을 넣는 경우

Use-Case #3 '장비구니'에 추가된 물품 결제

Test Method	Failed Test Case	원인분석
n03RequestAddExtension.testAdd_Extension	네명의 범위를 넘어가는 경우 네명에 음수값을 넣는 경우 확장자에 null 값을 넣는 경우	• 예외 처리 필요

Use-Case #4 중복확장자 검사

Test Method	Failed Test Case	원인분석
		• 문서에 명시 없음.

Use-Case #5 확장자 삭제

Test Method	Failed Test Case	원인분석
n05RequestDeleteExtension.testDelete_Extension	테스트 성공	

후기 및 Tip

◆ 테스트 케이스 작성의 어려움

① 동시성 문제

- 멀티쓰레드의 경우, 테스트케이스가 모두 실행되기 전에 스레드가 종료되어 버리는 경우가 있음
- 네트워크의 경우, 서버의 값이 변하는 경우 클라이언트 측에서 결과를 확인하기 어려움

② 접근 제한자

- Private method는 Junit code를 작성할 수 없음.
- 일반적으로 public으로 되어있는 method만 테스트하는 경향이 있음

③ GUI

- 사용자의 동작에 따라 화면에 영향이 나타나기 때문에 단순히 결과 예상 → 수행 → 확인 의 테스트가 곤란함
- UI영역에 비즈니스 로직이 들어가지 않게 코드를 작성해야함

④ 의존성 모듈 테스트

- 테스트가 되는 대상이 기타 method나 class를 참조하고 있을 경우, test에 문제가 발생할 수 있음
- 임의의 값을 가진 mock객체를 생성하여 의존성 제거를 위해 노력해야함

후기 및 Tip

◆ *Void Method의 사용*

- 리턴값이 있는 메소드 → 기능(function)
- Void method → 절차(procedure)

→ 리턴값이 없다는 의미는 로직의 절차를 기술했거나,
프로그램 로직 트리의 맨 끝단의 작업(화면출력, 데이터 등록 등)

◆ *Void Method의 테스트 : 행위기반 테스트*

- 상태(state)를 확인할 수 있는 다른 메소드로 결과를 테스트
- Mock 객체를 이용하여 가짜 값 생성

원본 프로그램의 코드를 수정할 수 없기 때문에
void 메소드를 테스트하는데 어려움이 있었다.