System Test & Spec Review

-5팀-
이지영 (팀장)
강민호
김정연
유경원

INDEX

01 Spec Review – Stage 1000

- 1003. Define Requirements
  - Performance Requirements
    - 자판기 간 통신 응답 시간이 1초 이내로 수행되어야 한다.
      데이터를 확인하는 시간이 1초 이내로 수행되어야 한다.

SRS 3.3 Performance Requirements 항목, 20p

👉 Q. 1초인가요? 0.1초인가요? SRS 문서에선 0.1초라고 되어있어 서로 말이 상충됩니다.하나로 통일 부탁드립니다.
A. 1초로 통일하겠습니다.

OOPT 1000문서와 SRS문서 비교 통일되지 않은 기준

- 1004.Record Term in Glossary(p.7)

Stock	재고
Count	Item의 수량

💡 Stock과 Count이 문서상 같은 표현으로 사용되고 있습니다. 같은 의미를 혼용하고 있는데, **Stock을 '품목(Item의 종류)'으로 고치거나** 혼동을 줄이기 위해 **한 가지 표현으로 통일**하는 것이 좋습니다.
또한 Stock을 '재고의 수량'을 표현한 것이라면 **"재고" 대신에 "재고량"으로 수정**하셔야 합니다. '재고'는 물건을 나타내는 단어지 수량을 표현하는 단어가 아닙니다.

Stock / Count 의미가 혼동되는 단어 사용

01 Spec Review – Stage 1000

문장의 표현이 불명확한 부분

OOPT 1000 pg12

- Describe use cases
  - 4. Show Item

Use Case	4. Show Item
Actor	None
Description	- 사용자가 현재 자판기의 재고 유무에 상관없이 구매 가능한 음료를 보여준다.

"사용자가 현재 자판기의 재고 유무에 상관없이 구매 가능한 **음료**를 보여준다." → "사용자가 현재 자판기의 재고 유무에 상관없이 구매 가능한 **모든 음료**를 보여준다." 로 구체화하여 수정 부탁드립니다.

OOPT 1000 pg13

- 12. Inform Location

Use Case	12. Inform Location
Actor	None
Description	- 다른 자판기로부터 응답 msg가 도착한 후 실행되는 use case. - 사용자에게 선택한 음료를 판매하고 있는 자판기의 위치를 알려준다.

"- 사용자에게 선택한 음료를 판매하고 있는 **자판기**의 위치를 알려준다" → "- 사용자에게, 선택한 음료를 판매하고 있는 **모든 자판기**의 위치를 알려준다" 로 수정 부탁드립니다.

OOPT 1000 pg19

Check Stock Count

egative Test Case

- 재고가 있는데 없다고 파악하는 경우, 다른 자판기의 재고 수를 요청한다

6	Check Stock Count	1. 기본: 실제 재고와 일치하는 값으로 저장해 다음단계를 진행한다.	1. 재고가 있는데 없다고 파악하는 경우 : 다른 자판기의 재고 수를 요청한다.

Q. 이게 무슨 뜻인가요?
A. local에 재고가 없고 remote에 재고가 있는 경우를 나타냅니다.

"자판기에 재고가 없지만, 다른 자판기에 재고가 있는 경우"로 수정 부탁드립니다.

문장의 표현이 모호하거나 정확하게 의미전달이 되지않는 경우 수정 요청

01 Spec Review – Stage 2030

이전문서와의 통일 및 모호한 표현 수정

- 13. Create Verification Code (p.10)

Cross Reference	System Functions: R2.10, R2.6 Use Case: “Advance Purchase”
Pre-Requisites	선택된 음료에 대한 결제가 완료됨
Typical Courses of Events	(A1): User, (A2): Manager, (A3): Other DVM, (S): System 1. (S): 새로운 선구매 코드를 생성한다.

- Pre-Requisites

  보다 정확하게 "선택된 음료에 대한 선결제가 완료됨."으로 수정 부탁드립니다.

- Typical Courses of Events

  1. (S): "새로운 선구매 코드를 생성한다." → "6자리 난수의 선구매 코드를 생성한다." 로 수정해야 합니다. OOPT 1000 문서에서 이미 언급된 내용이기 때문입니다.

현장구매와 선구매 결제의 명확한 구분 필요

- 12. Inform Location (p.9)

Cross Reference	System Functions: R2.9, R2.8, R4.1 Use Case: “Select Advance Payment”, “Message Request”
Pre-Requisites	음료가 선택됨

- Pre-Requisites

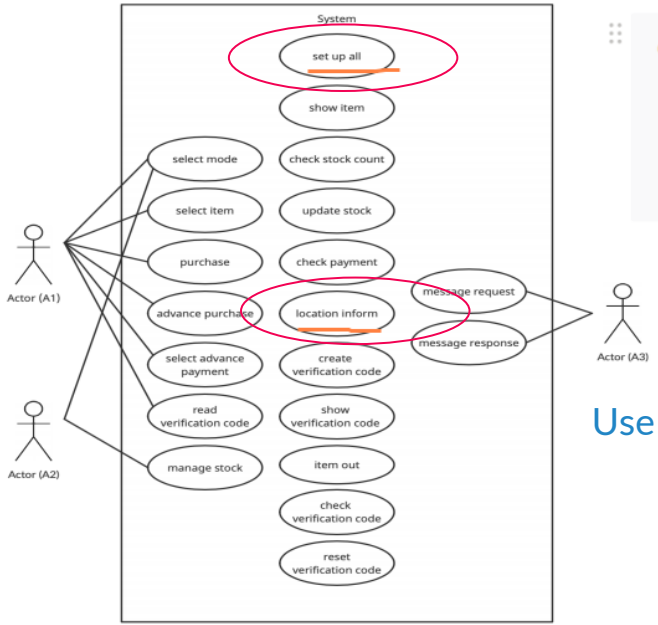
  위치보기 버튼을 누를 경우 실행되므로 "음료가 선택됨"이 아니라 "위치보기를 누름"으로 바뀌어야 합니다.

Pre-Requisites의 오류

01 Spec Review – Stage 2030

Use case 명칭오류

2132. Refine Use Case Diagrams



(1) 'set up all'은 function 명입니다. 해당 function에 대응되는 UseCase 명칭은 "set up"이므로 수정 부탁드립니다.

(2) 'location inform'은 function 명입니다. 해당 function에 대응되는 UseCase 명칭은 "inform location"이므로 수정 부탁드립니다.

Use case 명칭과 function의 이름이 혼동되어 사용되고 있는 부분

* (A1): User, (A2): Manager, (A3): Other DVMs

01 Spec Review – Stage 2040

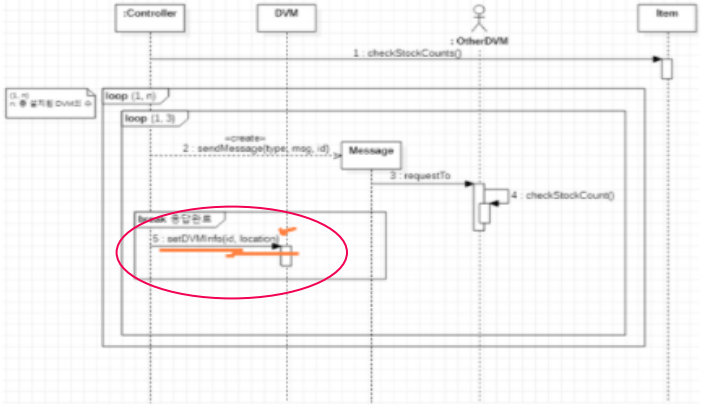
Interaction Diagram

2143. Define Interaction Diagrams

- (1) Set up(pg 19)

2143. Define Interaction Diagrams

(1) Set Up

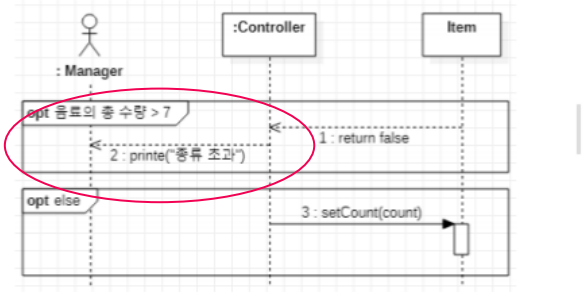


"setDVMInfo" 오퍼레이션은 DVM의 id, location 그리고 item의 count 항목을 업데이트하는 것이므로 매개변수로 item도 같이 넣어야 합니다.
⇒ "setDVMInfo(id, location)" → setDVMInfo(id, location, item)

매개변수 누락

- (7) Update Stock (pg 22)

(7) Update Stock



Q. 음료의 총 수량이 7초과일경우 종류 초과 라는 에러 메시지를 생성한다고 되어있는데 "음료의 총 수량"이라는 의미는 음료의 종류가 7가지를 넘어간 경우인가요/음료의 재고의 수가 7개를 넘어간 경우인가요?
A. 음료의 종류의 가짓수가 7개를 초과한 경우입니다.

음료의 종류의 가짓수가 7개 초과한 경우라고 의미가 분명하게 나타나게 수정바랍니다.

의미전달이 정확하지 않은 부분

01 Spec Review

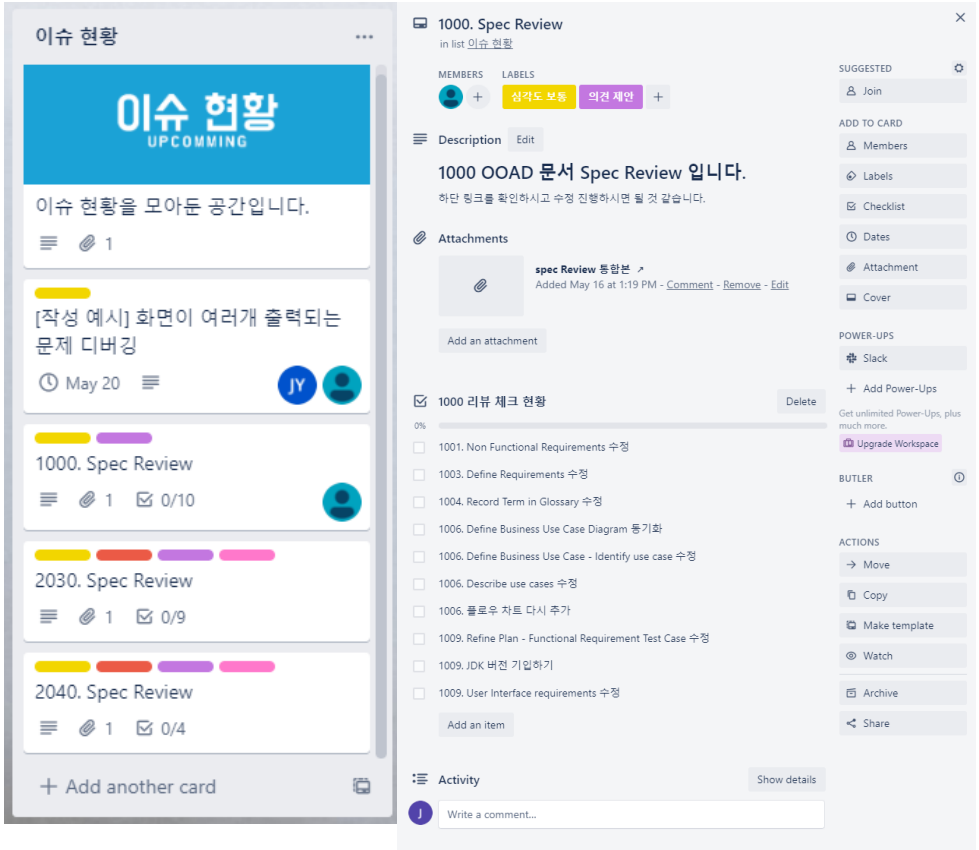
3학년 개발팀 피드백

기타 의문

- Q. 선결제 코드는 어떻게 관리되나요?
  A. 특정 시점에 한가지 음료에 대해서 선결제 코드가 여러개가 있는 경우도 존재하며 선결제 코드는 배열에 들어가 있습니다.
- Q. User는 1명으로 가정하나요, 여러 명으로 가정하나요?
  A. User는 여러명이며 식별할 필요는 없습니다.
- Q. PFR에는 VM당 음료의 종류의 개수가 7개로 고정인 것 같은데 B1팀은 유동적으로 음료의 종류가 변할 수 있도록 할 것인가요?
  A. 네 음료의 종류는 변합니다. 판매되는 음료의 종류의 가짓수가 7개 이하인 경우 새로운 음료가 추가될 수 있습니다.
- Q. VM ID 관리는 어떻게 하시나요?
  A. 현재 로컬 자판기의 ID는 0으로 고정되었고 VM ID가 IP주소를 대체합니다.
- Q. 네트워크 통신 라이브러리를 사용해서 실제로 분산 시스템을 만들 것인가요,
  아니면 네트워크 통신 없이 분산 시스템처럼 보이는 단일 프로그램을 만들 것인가요?
  A. 네트워크 통신을 사용하지 않습니다.
- Q. 자판기는 총 몇 대인가요?
  A. 총 5대입니다.
- Q. OOPT 1000 FR Test Case에서 어떤경우 positive/negative 가 되나요?
  A. use case의 목적에 맞으면 positive, 아니면 negative test case 가 됩니다

Spec Review를 진행하면서 생기는 의문에 대해서
3학년팀과 피드백을 주고받고 답변을 얻음

01 Spec Review - Trello



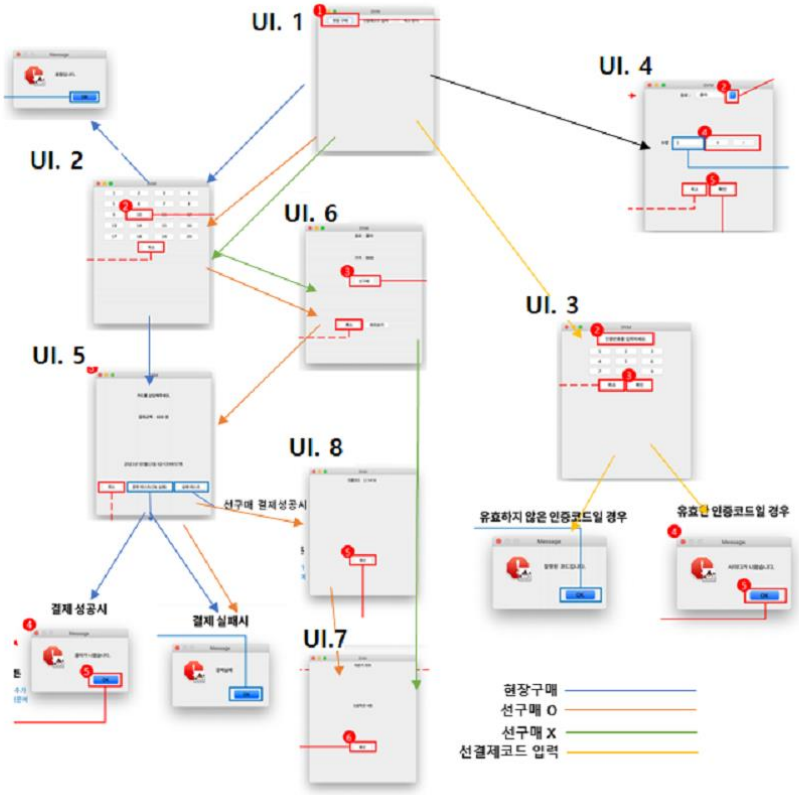
spec review 진행 후
수정이 필요한 부분
트렐로에 issue로 등록하여
체크리스트로 진행 사항을 관리

02 Test Stub

가상 Stub(매크로) 제작 방법

- 다음의 2가지 input event을 자동화 시키는 프로그램 작성
  1. 마우스로 버튼 클릭
  2. 콤보박스에서 마우스로 특정 항목 선택
- 결과적으로 다음 5가지 flow가 자동화 됨

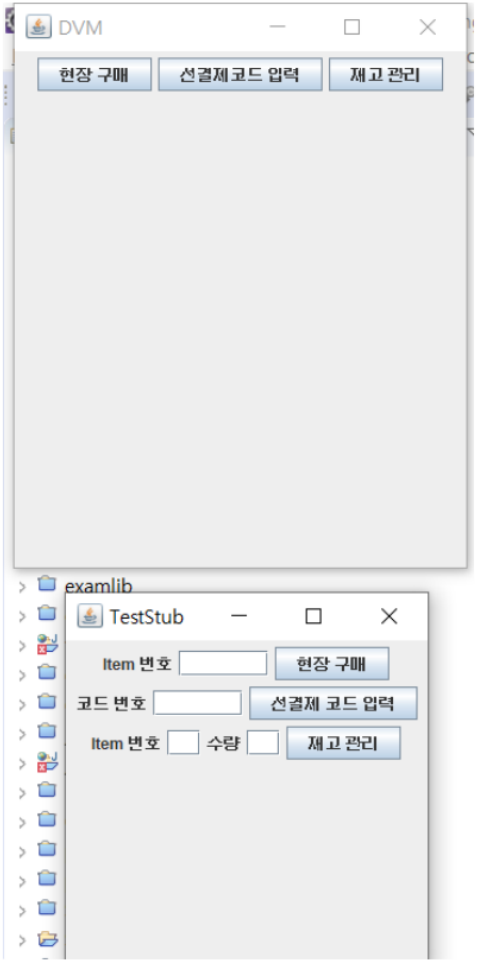
1. 파랑색 화살표에 해당하는 flow (현장구매)
2. 빨간색 화살표에 해당하는 flow (선구매)
3. 노란색 화살표에 해당하는 flow (선구매 코드입력)
4. 아래 그림에는 없는 flow (품절)
5. 아래 그림에는 없는 flow (재고 관리)



02 Test Stub

실행 방법

1. 현장 구매 버튼
   - 자동화 대상 : 현장 구매 / 선구매 / 품절
     - 'Item 번호' input 필드에 1~20 사이의 값 입력
     - '현장 구매 버튼' 클릭
2. 선결제 코드 입력
   - 자동화 대상 : 선결제 코드 입력
     - '코드 번호' input 필드에 선결제 코드 입력
     - '선결제 코드 입력' 버튼 클릭
3. 재고 관리
   - 자동화 대상 : 재고 관리
     - 'Item 번호' input 필드에 1~20 사이의 값 입력
     - '수량' input 필드에 정수 입력
     - '재고 관리' 버튼 클릭

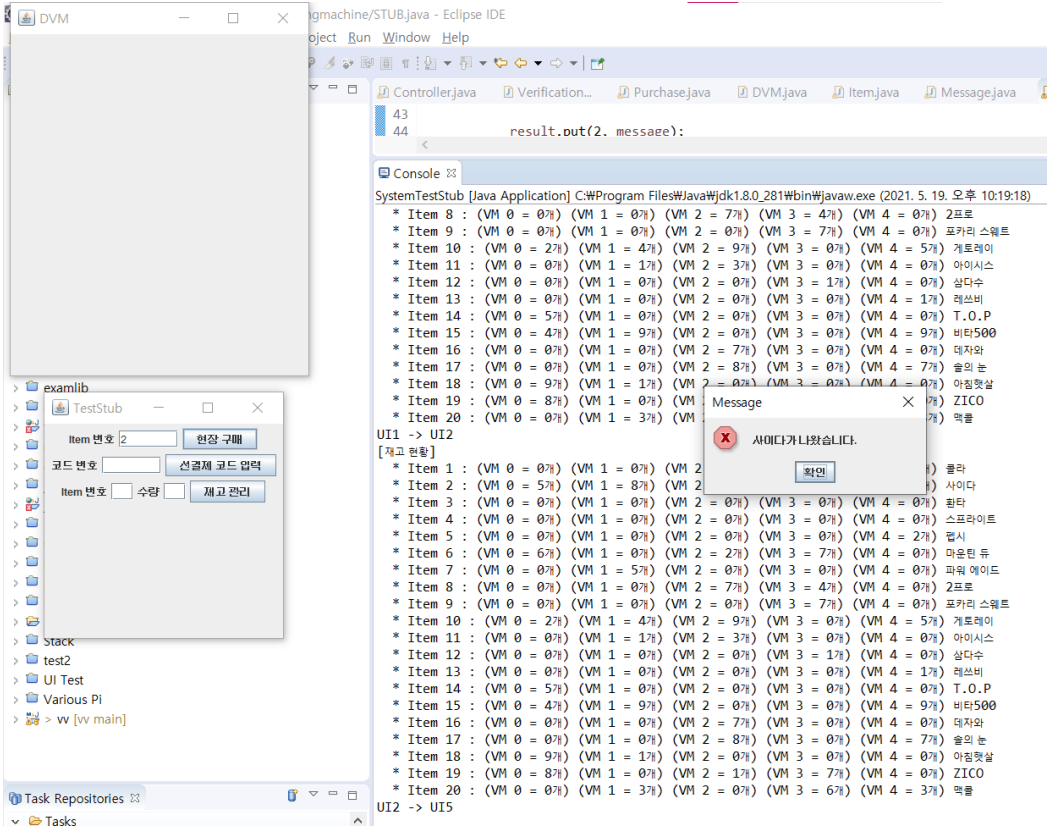


02 Test Stub

실제 실행 모습

1. 현장 구매 버튼

1) 현장 구매

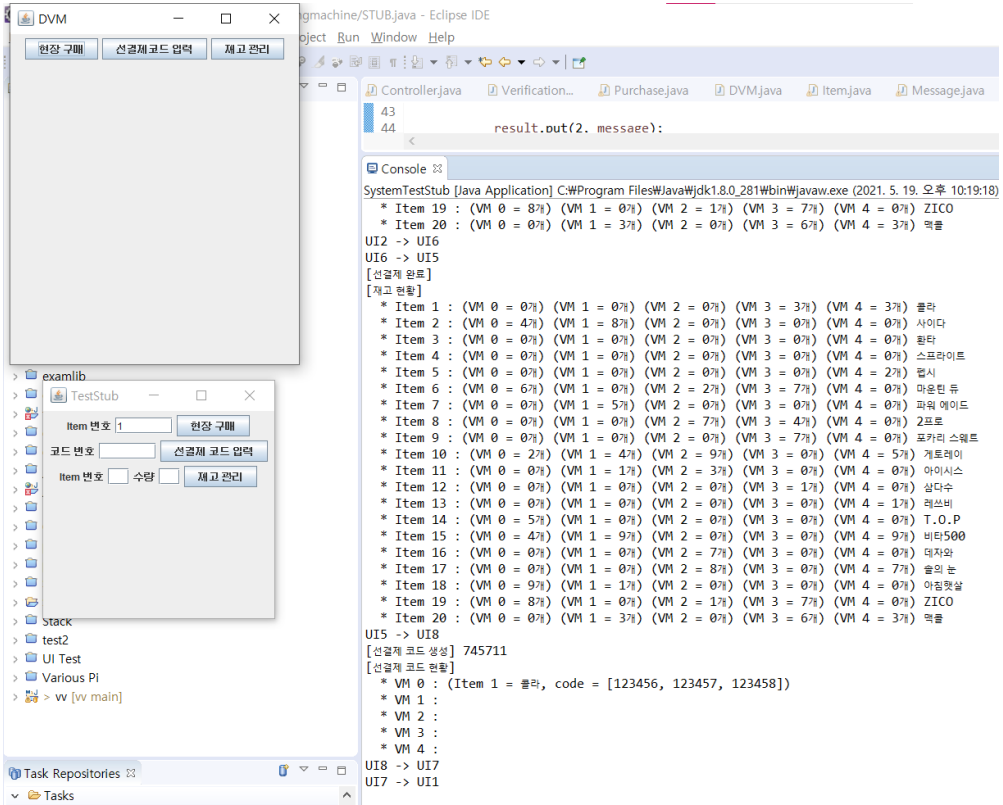


02 Test Stub

실제 실행 모습

1. 현장 구매 버튼

2) 선구매

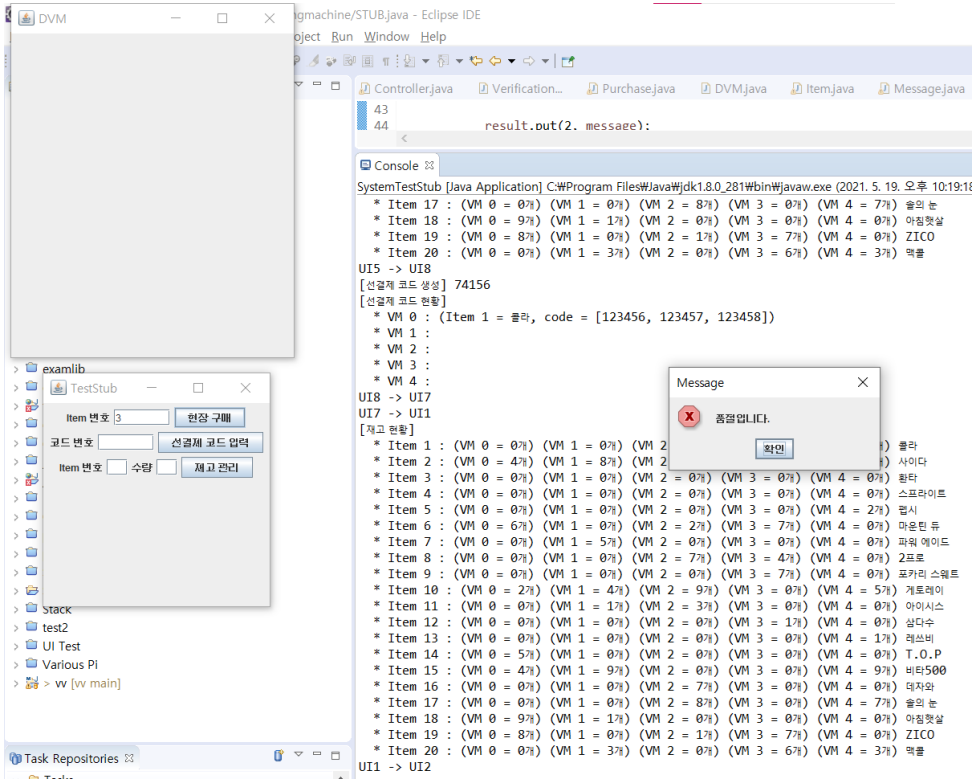


02 Test Stub

실제 실행 모습

1. 현장 구매 버튼

3) 품절

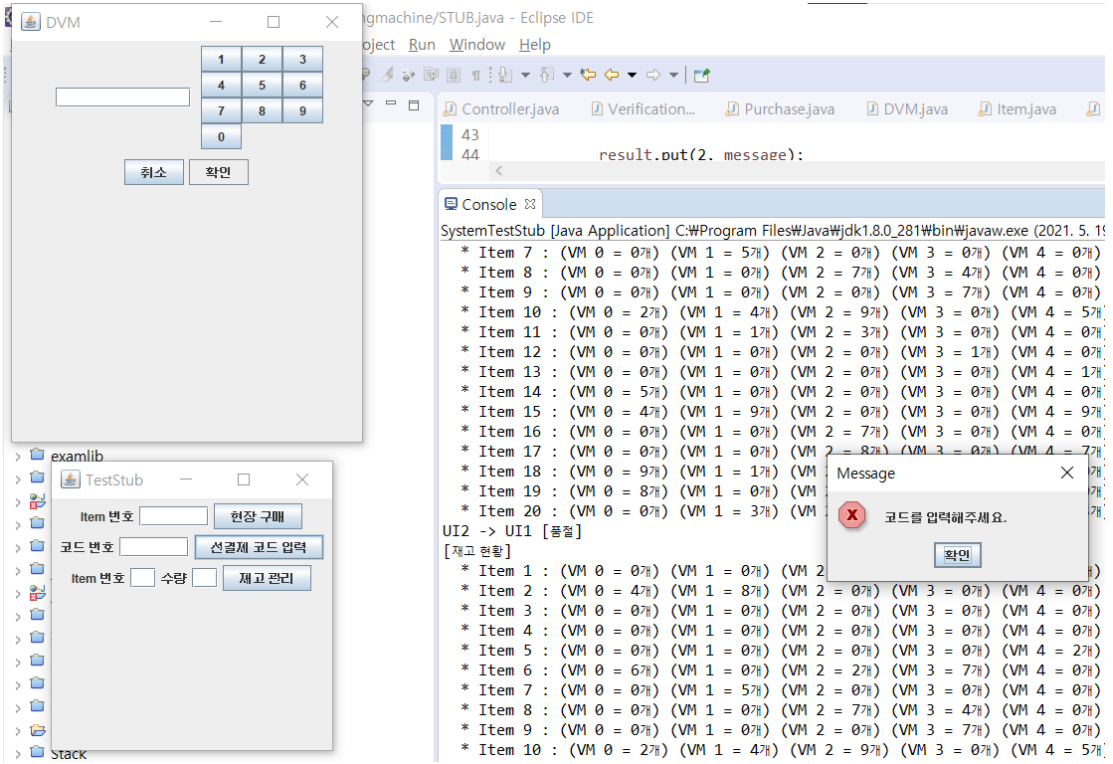


02 Test Stub

실제 실행 모습

2. 선결제 코드 입력

1) 빈칸 입력

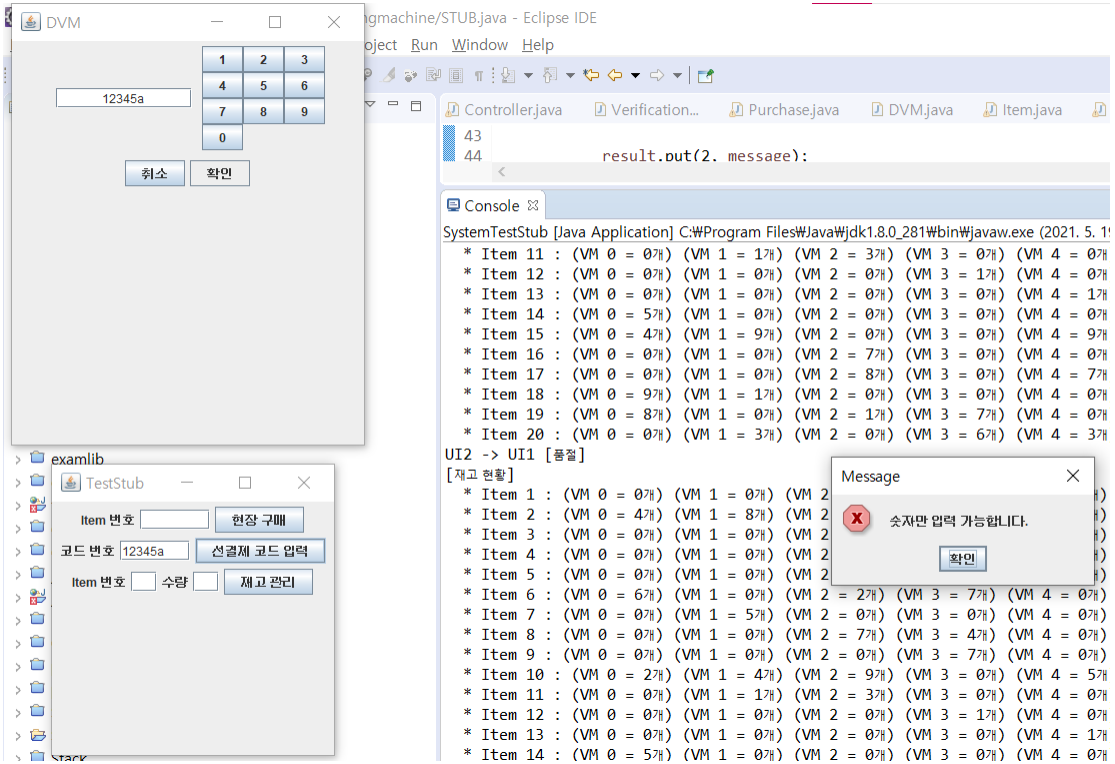


02 Test Stub

실제 실행 모습

2. 선결제 코드 입력

2) 숫자 외의 문자 입력

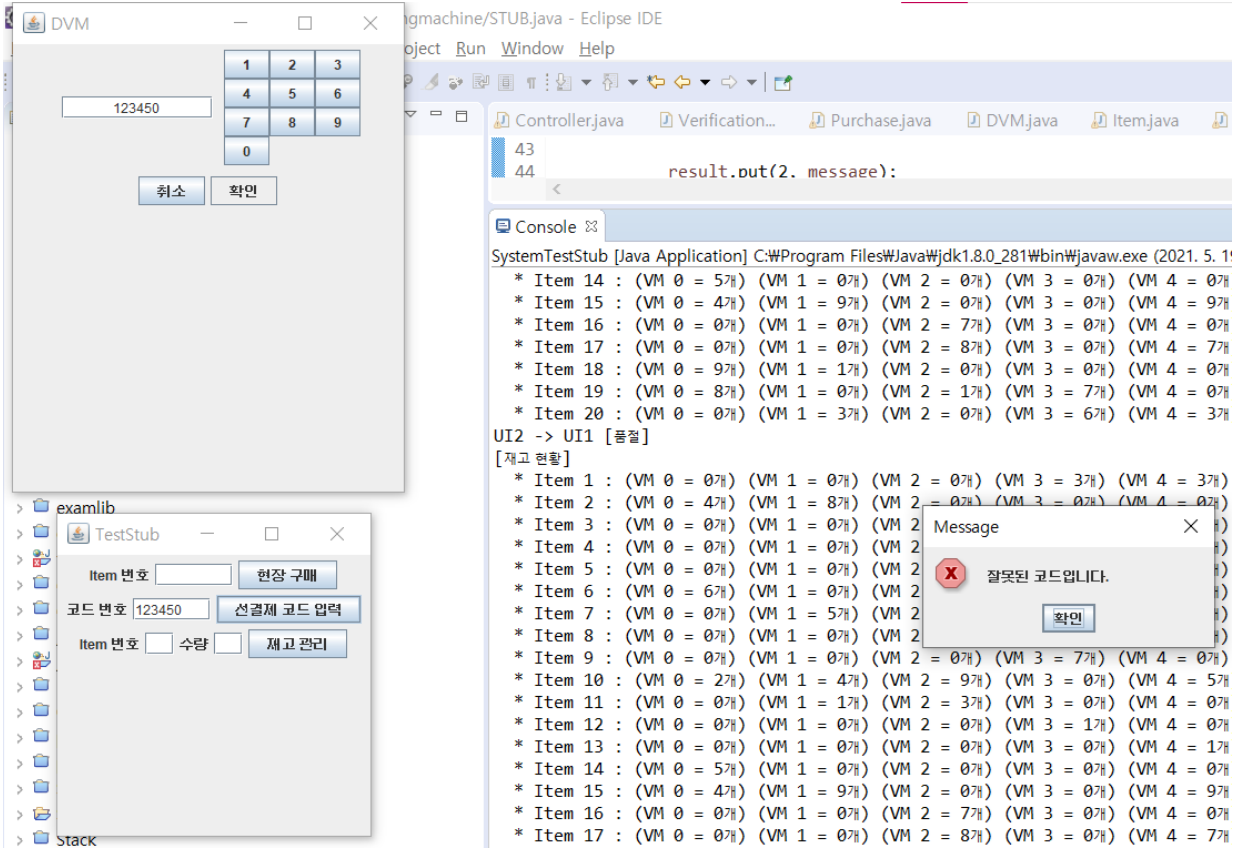


02 Test Stub

실제 실행 모습

2. 선결제 코드 입력

3) 유효하지 않은 코드

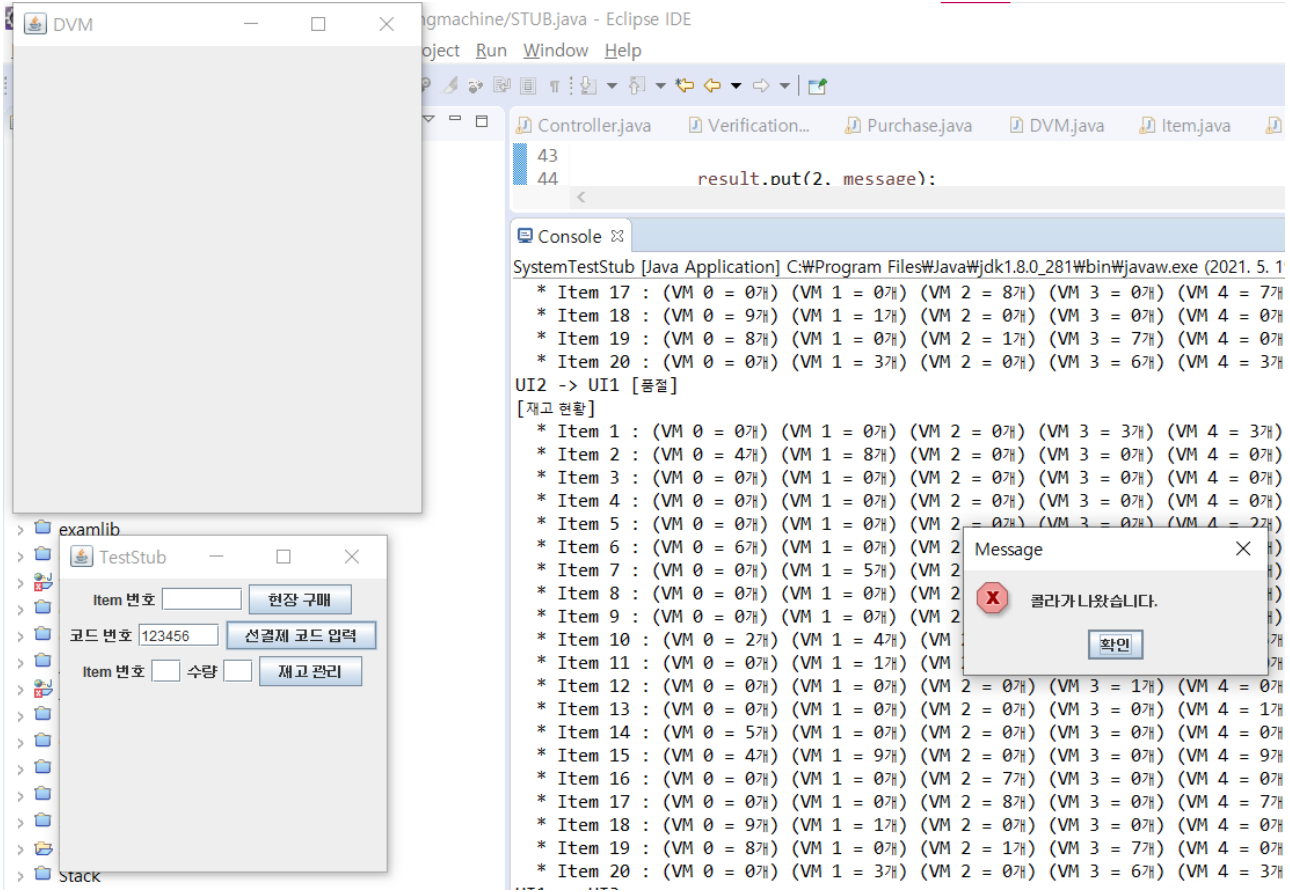


02 Test Stub

실제 실행 모습

2. 선결제 코드 입력

4) 유효한 코드

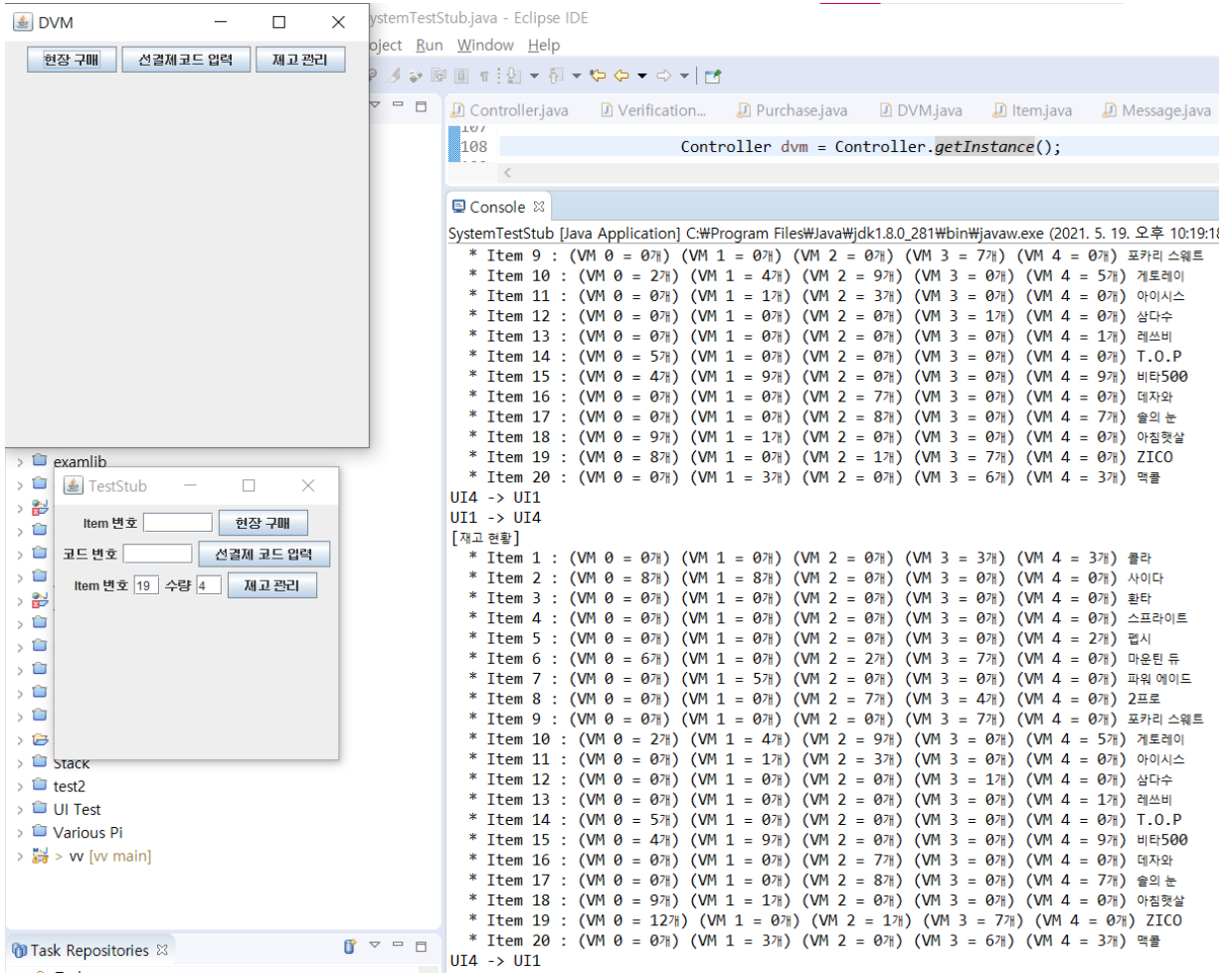


02 Test Stub

실제 실행 모습

3. 재고 관리

1) 정상 처리

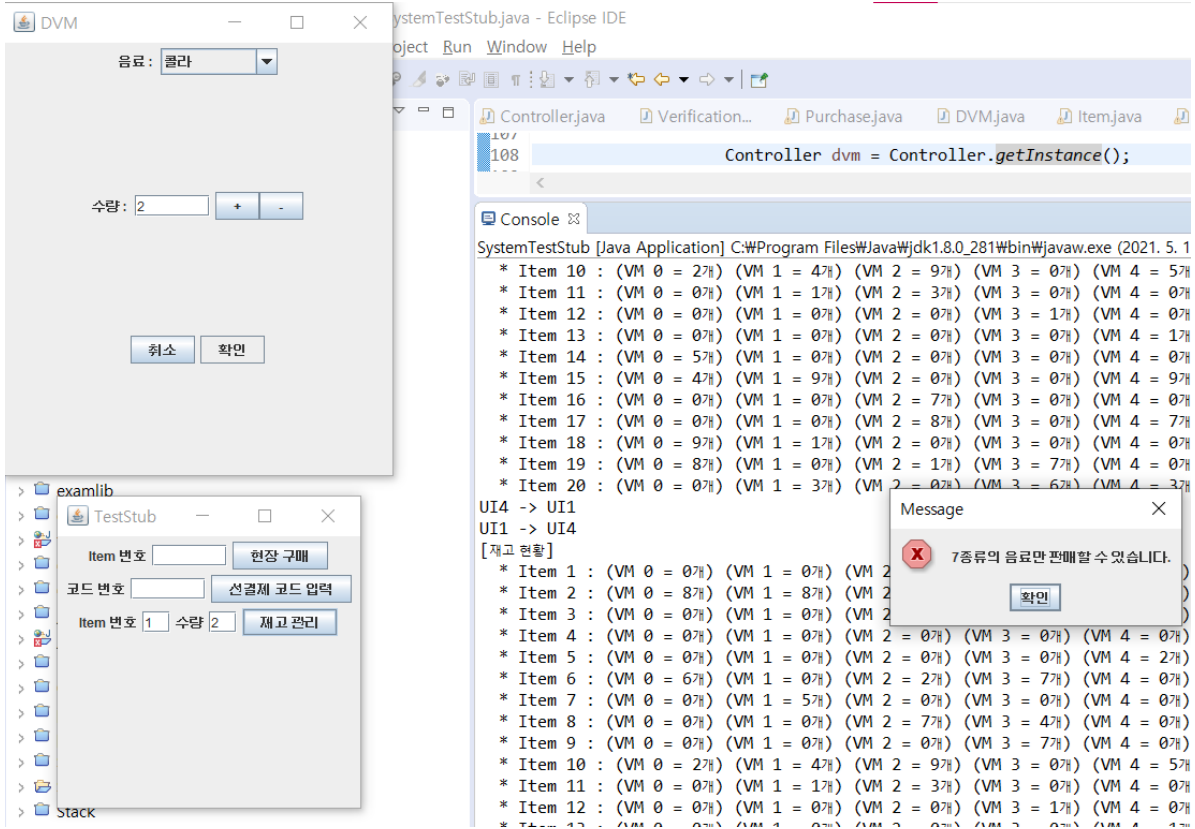


02 Test Stub

실제 실행 모습

3. 재고 관리

2) 예상 음료 종류가 7가지를 초과하는 경우

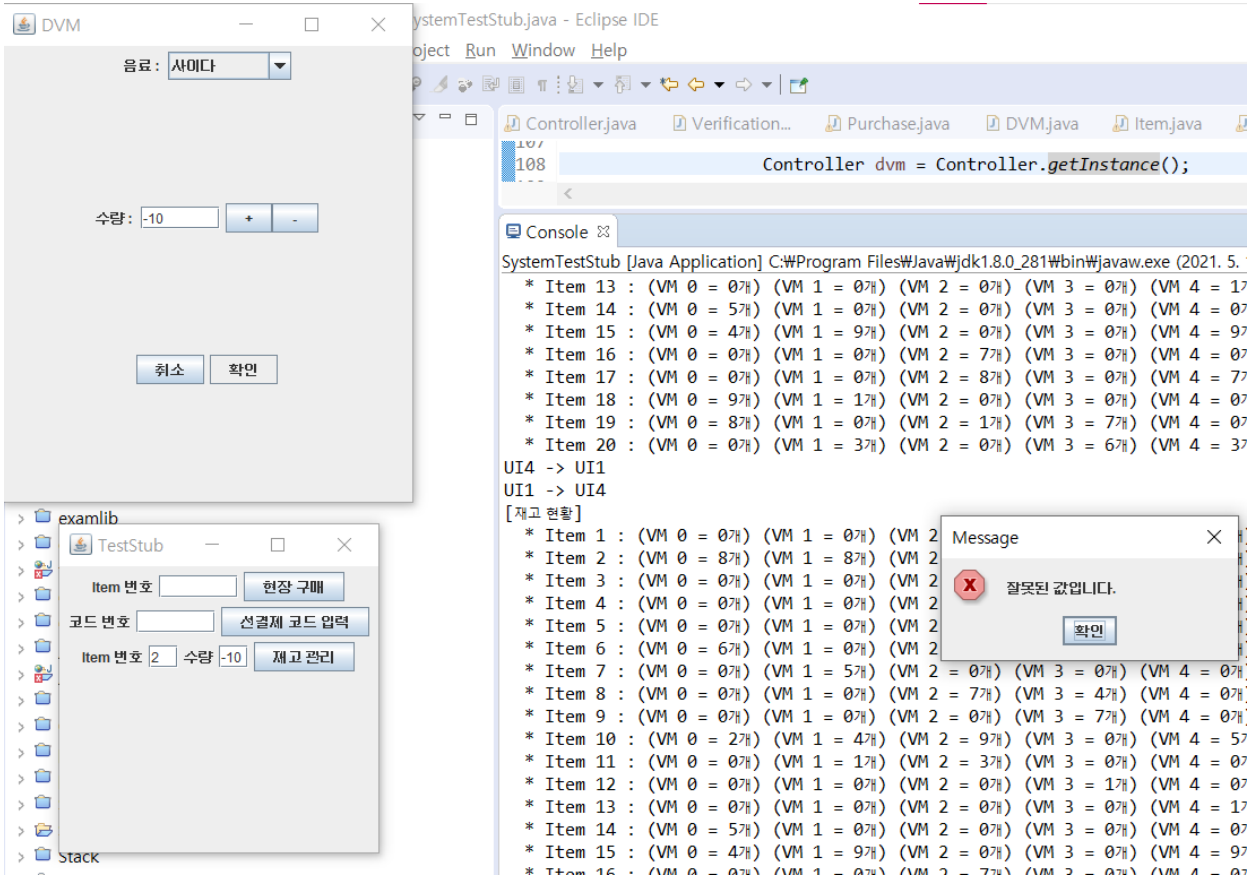


02 Test Stub

실제 실행 모습

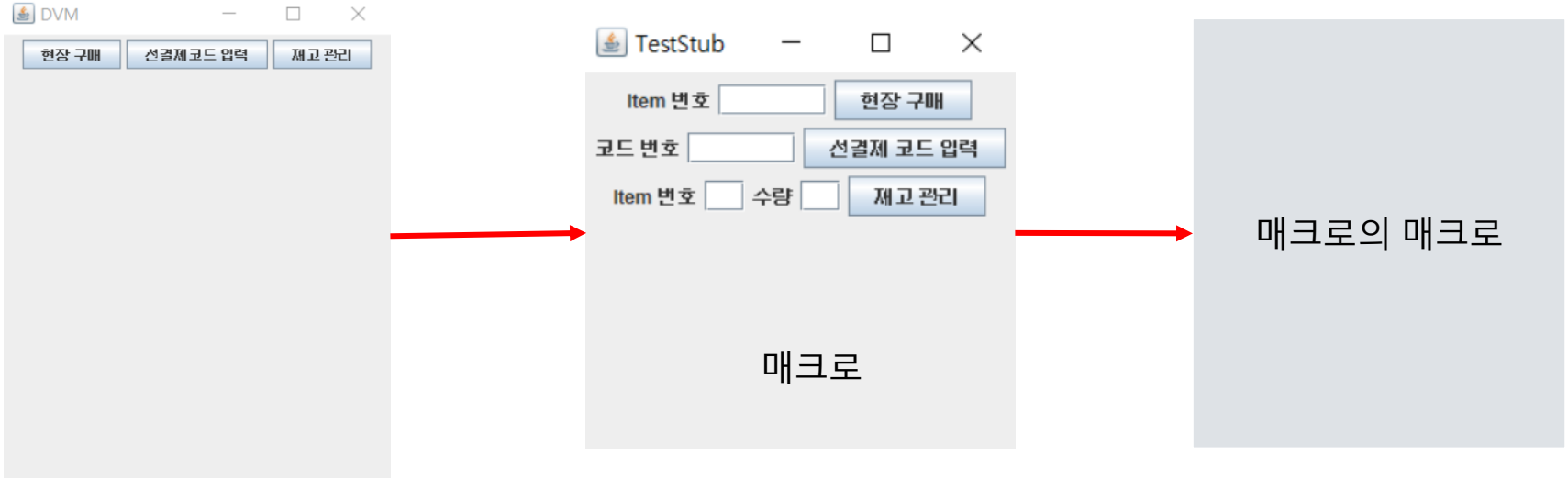
3. 재고 관리

3) 예상 재고가 음수가 되는 경우



02 Test Stub

매크로의 매크로



02 Test Stub

매크로의 매크로로 테스트하기 어려운 이유

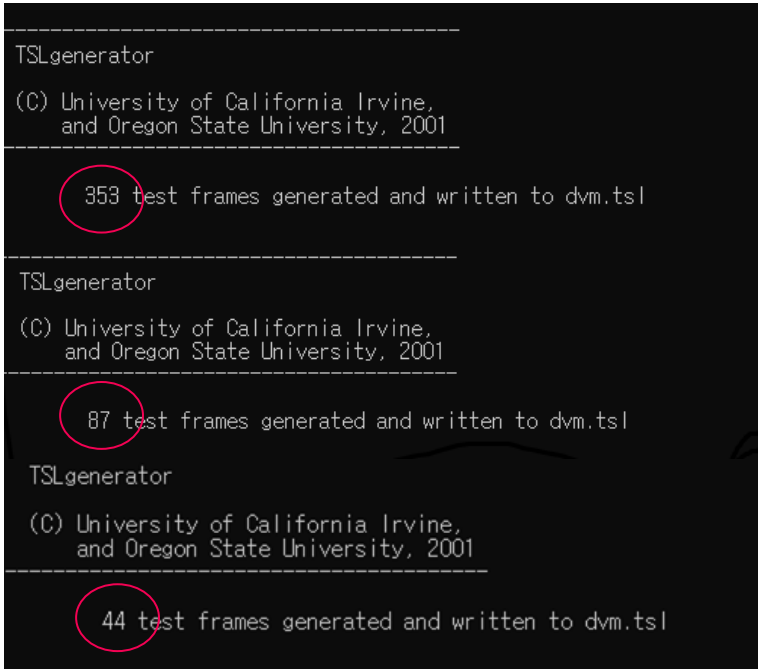
- 현장구매 하나만 가지고도 item out 순서에 따른 조합이 매우 많음
  - 예를 들어 특정 샘플에 대해 조합의 개수를 계산해보면
  - 15 x 4 x 8 x 9 x 4 x 7 x 19 x 5 x 5 x 21 x 8 x 9 x 11 x 18 x 12 x 12 x 9 x 7
  - 즉, 10,403,133,021,388,800 개의 조합을 테스트해야 함
  - 버튼을 클릭하여 연산이 수행되고 페이지가 넘어가는 delay를 무시한다고 해도 시간적으로 테스트 불가능
  - 샘플을 단순화 시켜서 item당 재고가 최대 2개만 가능하다고 해도 1,048,576개의 조합 생성
- 현장구매/선결제코드입력/재고관리의 수행 순서가 섞이는 경우
  - 그런데 그 1,048,576개의 조합이 '선결제코드입력'과 '재고관리'와 또 다시 조합을 이루게 되면, 역시 테스트 불가능한 개수가 나옴

03 Category-Partition Testing

```
#
# Test specification for DVM
#

Environments:

  Current UI:
    initial page.
                    [property ui1]
    item select.          [property ui2]
    input verification code.  [property ui3]
    modify stock.         [property ui4]
    payment.              [property ui5]
    inform prepurchase.   [property ui6]
    inform location.      [property ui7]
    inform verification code.     [property ui8]

  Mode:
    purchase.             [property mode1] [if ui5]
    prepurchase.          [property mode2] [if ui5]

  is Advance:
    false.                [if ui5]
    true.                 [if ui5]

  selected item stock:
    local stock zero and remote stock zero.
                    [if ui2]
    local stock zero and valid remote stock.
                    [if ui2]
    valid local stock.          [if ui2]

  expected changed local stock:
    less than zero.           [if ui4]
    more than zero.           [if ui4]

  expected changed local item type:
    less than seven.          [if ui4]
    more than seven.          [if ui4]


Parameters:

  Page move button:
    initial page-purchase button.
                    [if ui1]
    initial page-prepurcahse button.
                    [if ui1]
    initial page-stock management button.
                    [if ui1]
    select item button. [if ui2]
    cancel button.            [if ui2 || ui3 || ui4 || ui5 || ui6]
    confirm button.           [if ui3 || ui4 || ui7 || ui8]
    1% payment fail.          [if ui5]
    100% payment fail.        [if ui5]
    location button.          [if ui6]

  Manage Stock:
    select item zero.         [if ui4]
    select item less than zero.   [if ui4]
    select item more than zero. [if ui4]

  Verification Code:
    valid verification code.  [if ui3]
    invalud verification code.    [if ui3]
```

03 Category-Partition Testing

TSL Generator 사용하여 Test Case 생성



Test Case 생성

Environment constraint 적용-> 353개

Environment constraint 부분 삭제 후 적용 -> 87개

Property constraint 적용 -> 44개

03 Category-Partition Testing

Constraint 적용

1.1 Categorize & Property & Constraints

Group	Category	Value	Constraint [Property prop]
Environments	Current UI	initial page.	[property ui1]
Environments	Current UI	item select.	[property ui2]
Environments	Current UI	input verification code	[property ui3]
Environments	Current UI	modify stock	[property ui4]
Environments	Current UI	payment	[property ui5]
Environments	Current UI	inform prepurchase	[property ui6]
Environments	Current UI	inform location	[property ui7]
Environments	Current UI	inform verification code	[property ui8]
Environments	Mode	purchase	[property mode1] [if ui5]
Environments	Mode	prepurchase	[property mode2] [if ui5]
Environments	is Advance	false	[if ui5]
Environments	is Advance	true	[if ui5]
Environments	selected item stock	local stock zero and remote stock zero	[if ui2]
Environments	selected item stock	local stock zero and valid remote stock	[if ui2]
Environments	selected item stock	valid local stock	[if ui2]
Environments	expected changed local stock	less than zero	[if ui4]
Environments	expected changed local stock	more than zero	[if ui4]
Environments	expected changed local item type	less than seven	[if ui4]
Environments	expected changed local item type	more than seven	[if ui4]
Parameters	Page move button	initial page-purchase button.	[if ui1]
Parameters	Page move button	initial page-prepurcahse button.	[if ui1]
Parameters	Page move button	initial page-stock management button	[if ui1]
Parameters	Page move button	select item button	[if ui2]
Parameters	Page move button	cancel button	[if ui2 || ui3 || ui4 || ui5 || ui6]
Parameters	Page move button	confirm button	[if ui3 || ui4 || ui7 || ui8]
Parameters	Page move button	location button	[if ui6]
Parameters	Page move button	1% payment fail	[if ui5]
Parameters	Page move button	100% payment fail	[if ui5]
Parameters	Manage Stock	select item zero	[if ui4]
Parameters	Manage Stock	select item less than zero	[if ui4]
Parameters	Manage Stock	select item more than zero	[if ui4]
Parameters	Verification Code	valid verification code	[if ui3]
Parameters	Verification Code	invalud verification code	[if ui3]

Constraint(single, property)를 적용하여 test case 44개로 감소

03 Category-Partition Testing

CPT Result

TestCase	Result	Description
1	Pass	
2	Pass	
3	Pass	
4	Pass	
5	Pass	
6	Pass	
7	Pass	
8	Pass	
9	Pass	
10	Pass	
11	Pass	
12	Pass	
13	Pass	
14	Pass	
15	Pass	
16	Pass	
17	Pass	
18	Pass	
19	-	
20	-	
21	Pass	
22	-	
23	-	
24	Pass	
25	Pass	
26	Pass	
27	Pass	
28	Pass	
29	Pass	
30	Pass	
31	Pass	
32	Pass	
33	Pass	
34	Pass	
35	Pass	
36	Pass	
37	Pass	
38	Pass	
39	Pass	
40	Pass	
41	Pass	
42	Pass	
43	Pass	
44	Pass	
45	Pass	

TestCase 19, 20, 22, 23은 logical contradictory 한 테스트 케이스라서 제외함. ☞ 40 / 40 = 100%

04 Brute Force Testing

3학년 개발팀(B1)이 정의한 기능(OOPT 2063)을 구체화하고 실제 프로그램 실행시 기능이 잘 동작하는지 확인

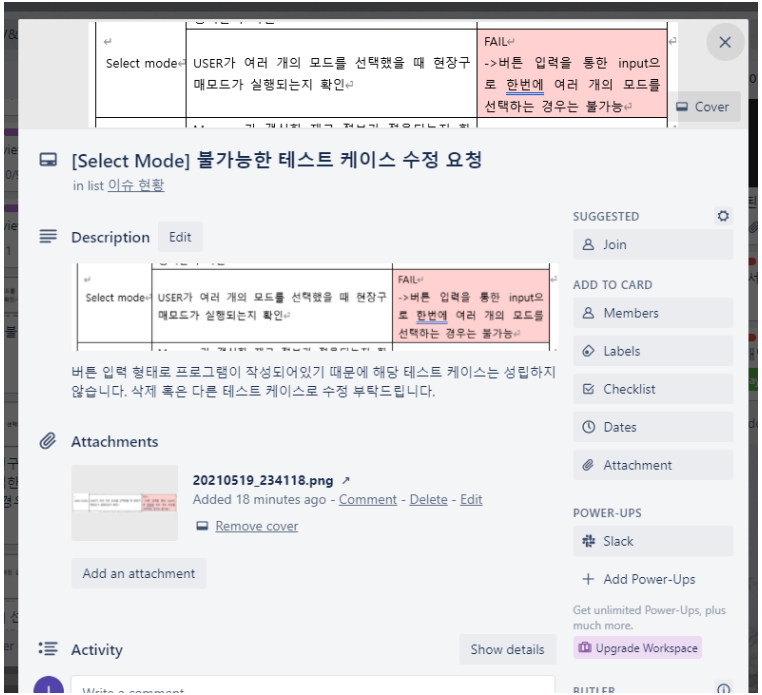
Use case	Number	Test case	Test Result
Set up	1-1	시스템이 시작되었을때 다른 DVM의 ID를 받아왔는지 확인	PASS
	1-2	시스템이 시작되었을때 다른 DVM의 재고를 받아왔는지 확인	PASS
	1-3	시스템이 시작되었을 때 자판기의 재고가 업데이트 되었는지 확인	PASS
Select mode	2-1	USER가 선택한 모드에 따라 정해진 메뉴가 실행되는지 확인	PASS
	2-2	USER가 여러 개의 모드를 선택했을때 현장 구매모드가 실행되는지 확인	FAIL ->버튼 입력을 통한 input으로 한번에 여러 개의 모드를 선택하는 경우는 불가능
Manage Stock	3-1	Manager가 갱신한 재고 정보가 적용되는지 확인	PASS
	3-2	Manager가 총 8가지 이상의 음료를 갱신할 경우 에러가 나오는지 확인	PASS
Show item	4-1	모든 음료가 나오는지 확인	PASS
	4-2	User가 선택한 음료가 선택되는지 확인	FAIL ->선구매 코드를 이용한 음료 구매시 선택한 음료와 다른 음료가 Item out되는 경우 발생
Check stock count	6-1	User가 선택한 음료에 대해 현재 자판기에서 판매 가능한지 확인	PASS
	6-2	현재 자판기에서 판매 가능하다면 Purchase를 정상적으로 실행하는지 확인	PASS
	6-3	동신을 통해 다른 자판기로부터 재고를 받아오는지 확인	PASS
	6-4	다른 자판기로부터 재고가 없다는 응답을 받으면, "Select Advance Payment"를 정상적으로 실행하는지 확인	PASS
Update stock	7-1	음료 재고정보를 주어진 값에 맞게 최신화하는지 확인	FAIL ->선구매 코드를 이용하여 item out 실행 후 다른 자판기에서 재고가 update되지 않는 상황 발생
Purchase	8-1	결제가 성공했을 때 음료 재고를 맞게 줄이는지 확인	PASS
	8-2	결제가 성공했을 때 해당 음료를 배출하는지 확인	PASS
	8-3	결제가 실패했을 때 결제 실패 메시지를 보여주는지 확인	PASS
Advance purchase	9-1	선구매결제에 성공했을 때, 선구매코드를 받아오는지 확인	PASS
	9-2	선구매 결제에 성공했을 때, 받아온 선구매코드를 "Message Request"통해 'Broadcast"하는지 확인	FAIL ->선구매 코드는 현재자판기에만 저장되고 다른 자판기에는 저장되지 않음
	9-3	선구매 결제에 성공했을 때, 해당 음료가 있는 자판기에 대해 "Inform Location"을 실행하는지 확인	PASS
	9-4	선구매 결제에 성공했을 때, 생성된 선구매코드를 보여주는지 확인	PASS
	9-5	선구매 결제에 실패했을 때, 결제 실패 메시지를 보여주는지 확인	PASS
	9-6	선구매 결제에 실패했을 때, 감소시켰던 재고 수량을 다시 증가시키는지 확인	FAIL -> 스펙과 다르게 코드를 구현 케이스인것 같음
	9-7	선구매 진행 후 선구매 코드를 현재 자판기에 저장하는지 확인	PASS
	9-8	선구매 코드 입력 결제 시, 해당 item의 재고가 줄어드는지 확인	FAIL -> 줄어들지 않음
Check payment	10-1	결제 결과를 받아오는지 확인	PASS
	10-2	정상적인 결제 카드가 아니라면 "잘못된 카드"라는 에러 메시지를 보여주는지 확인	PASS
Select advance payment	11-1	선구매를 선택한 음료에 대해 "Message Request"로 재고를 받아오는지 확인	PASS
	11-2	선구매결정한 음료의 재고가 남아있다면 "Message Request"로 재고를 감소시키는지 확인	FAIL -> 줄어들지 않음
	11-3	재고를 감소시킨 후 "Advance Purchase"를 실행하는지 확인	FAIL -> 재고가 감소되지 않는 상황 발생
	11-4	선구매 결정한 음료의 재고가 없다면 품절메시지를 보여주고, "Select Mode"로 돌아가는지 확인	PASS
	11-5	위치확인 모드를 선택한 음료에 대해 "Message Request"로 재고를 받아오는지 확인	PASS
	11-6	위치확인 모드를 선택한 음료의 재고가 남아있다면 해당 음료에 대해 "Inform Location"을 실행하는지 확인	PASS
	11-7	위치확인 모드를 선택한 음료의 재고가 남아있지 않다면 품절메시지를 보여주고, "Select Mode"로 돌아가는지 확인	PASS
	11-8	모드 선택과 취소를 눌다 누를 경우 사용자로부터 재입력 받는지 확인	FAIL ->버튼을 통한 Input을 받으므로 버튼을 중복해서 누를수가 없는 상황 발생
	11-9	선구매 진행을 위한 결제시 다른 자판기들의 재고가 바로 줄어드는지 확인	FAIL ->현재 자판기에서 Item out을 하기 전까지는 다른 자판기에는 재고가 줄어들지 않아 선결제가 무한정으로 가능함
	11-10	선결제 후 선구매 코드 입력시 해당 item의 재고가 줄어드는지 확인	FAIL -> 줄어들지 않음
Inform location	12-1	음료를 갖고있는 자판기에 대해 위치를 보여주는지 확인	PASS
Create verification code	13-1	음료가 선택되었을 때, 새로운 선구매 코드를 생성하는지 확인	PASS
	13-2	"Create Verification Code"를 통해 생성된 선구매 코드를 보여주는지 확인	PASS
	13-3	선결제 코드가 여러 종류의 음료에 중복되어 생성되지 않는지 확인	FAIL ->서로 다른 종류의 음료에 같은 선구매 코드가 할당되어 서로 다른 음료임에도 불구하고 item out되는 상황 발생
Item out	15-1	음료가 선택되어 있고, 결제가 완료되었거나 유효한 선구매 코드를 입력했을 때만 음료를 제공하는지 확인	FAIL -> 현재 자판기에 재고가 없는데도, 선구매 코드를 입력하면 item out이 되고 있음
	15-2	현재 자판기에 재고가 없을 때는, 선구매 코드를 입력해도 item out이 되지 않아야 함	FAIL -> item out 됨
Read verification code	16-1	User가 입력한 선구매 코드를 "Check Verification Code"에 전달하고 그 결과를 받는지 확인	PASS
	16-2	기록된 코드가 유효하다면 "Item Out"을 실행하는지 확인	PASS
	16-3	"Item Out"을 실행한 후에 "Reset Verification Code"를 실행하는지 확인	PASS
Check verification code	17-1	시스템에 입력된 코드가 존재하는지 확인	PASS
Reset verification code	18-1	시스템에서 입력된 코드를 정상적으로 삭제하는지 확인	PASS
Message request	19-1	보내고자하는 메시지 종류를 수신 대상에게 잘보내는지 확인	FAIL ->"Broadcast"메시지 실패로 다른 DVM에게 선구매 코드를 전송하지 못함
	19-2	수신된 메시지에 따라 명시된 응답을 전송한 DVM에 정상적으로 보내는지 확인	PASS
	19-3	알 수 없는 종류에 메시지를 수신한 경우 알 수 없음을 응답하는지 확인	PASS
전체 Test Case : 50 개			Pass : 35 개

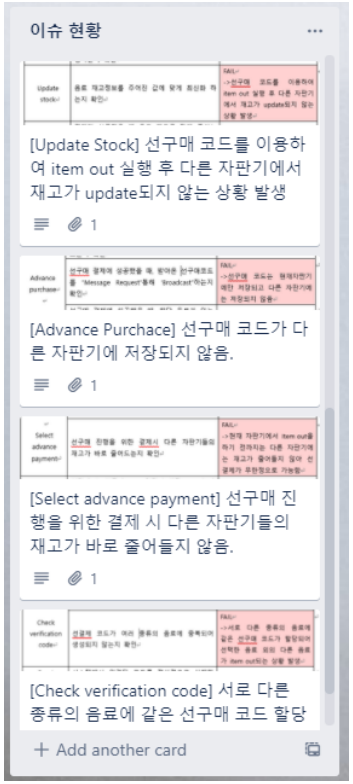
(35 / 50) = 70% Pass

50개의 test case 중에서 35개의 test case pass => 70% Pass

04 Brute Force Testing - Trello

BFT 결과 trello에 등록





BFT 진행 결과 FAIL된 Test case trello에 issue로 등록하여 3학년 개발팀에게 전달하여 관리

감사합니다