

객체지향 개발 방법론

DVM Project 001

#4조



201710240

이해림



201811217

이해인



201711836

송호영

Index

Activity 2051. Implement Class & Methods Definitions

Activity 2055. Write Unit Test Code

Activity 2061. Unit Testing

Activity 2063. System Testing

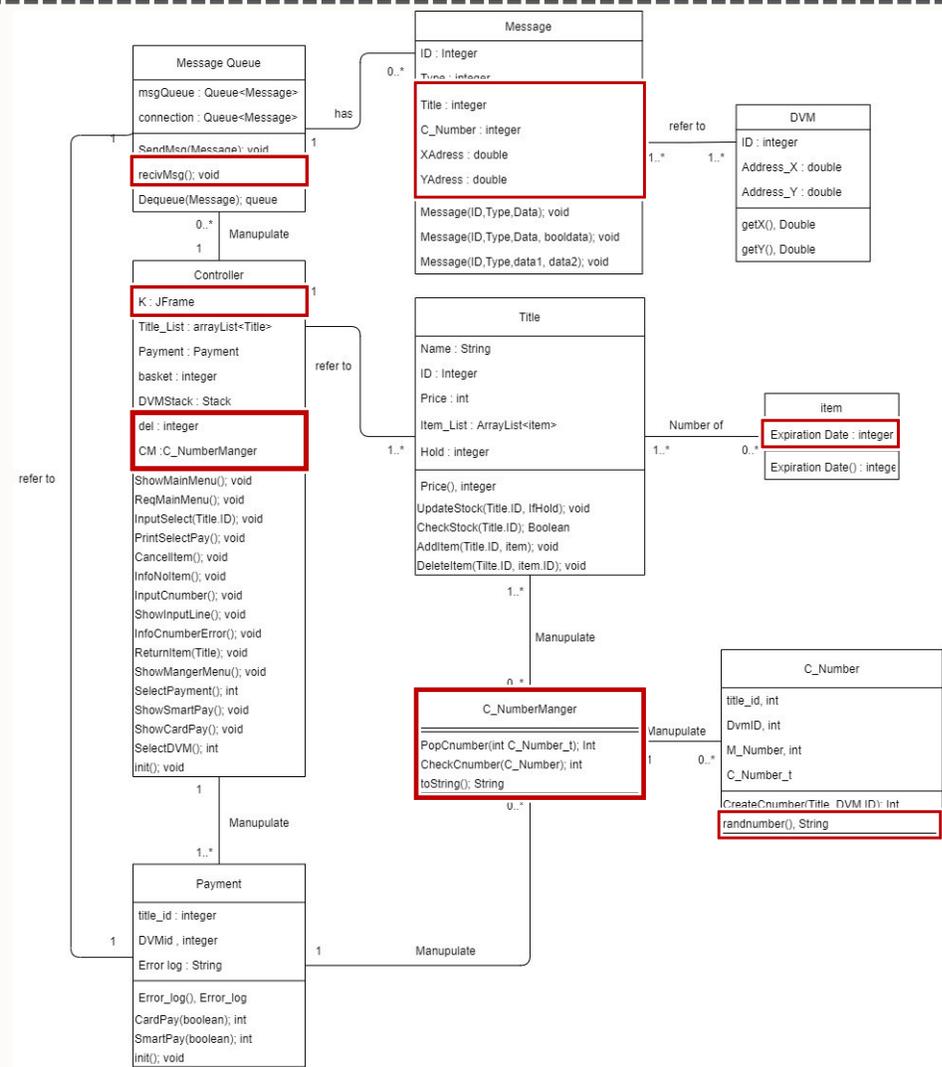
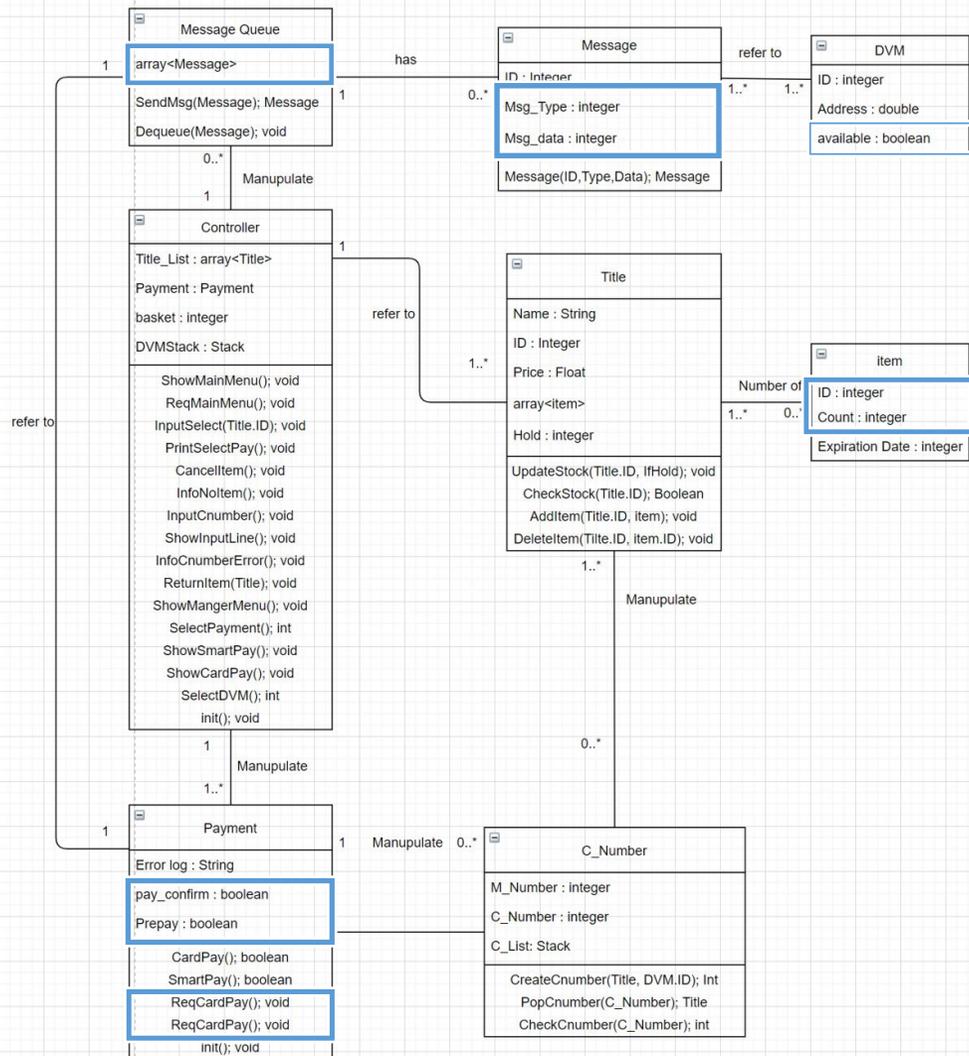
Activity 2066. Testing Traceability Analysis

Demonstration video

Activity 2051. Implement Class & Methods Definitions

수정 전

수정 후



Activity 2051. Implement Class & Methods Definitions

Name	Controller
Type	Class
Purpose	사용자에게 메뉴를 표시
Overview	N/A
Cross Reference	R1.1, R1.2.1, R1.2.2, R1.2.3, R1.3, R1.4, R1.5, R2.1, R2.4, R3.1.2 "자판기 음료 종류 출력", "구매할 음료 입력", "사용자 선택 취소", "사용자 인증번호/바코드 입력", "재고가 부족한 제품 안내", "구매가능한 자판기 안내", "사용자 자판기 선택", "결제 수단 목록 출력", "인증번호/바코드 출력"
Input	TitleID, DVMid
Output	Integer , void
Exceptional Courses of Events	N/A

Activity 2051. Implement Class & Methods Definitions

Name	Title
Type	Class
Purpose	재고를 관리하고 그에 대한 정보를 저장하는 클래스
Overview	N/A
Cross Reference	R1.1, R1.2.1, R1.3, R1.4, R2.3.1, R2.3.2, R3.1.1, R3.2, R4.2, R4.4 "자판기 음료 종류 출력", "구매할 음료 입력", "재고가 부족한 제품 안내", "구매가능한 자판기 안내", "음료 전달", "재고 수량 업데이트", "인증번호/바코드 생성", "재고변경 및관리"
Input	Id, IfHold, item, ilst
Output	Name, Item_List, boolean
Exceptional Courses of Events	N/A

Activity 2051. Implement Class & Methods Definitions

Name	Item
Type	Class
Purpose	재고 각각의 정보를 저장하는 클래스
Overview	N/A
Cross Reference	R1.1, R1.2.1, R1.3, R1.4, R3.1.1, R3.2, R4.2, R4.4 "자판기 음료 종류 출력", "구매할 음료 입력", "재고가 부족한 제품 안내", "구매가능한 자판기 안내", "음료 전달", "재고 수량 업데이트", "인증번호/바코드 생성", "재고변경 및관리"
Input	N/A
Output	Expiration_date
Exceptional Courses of Events	N/A

Activity 2051. Implement Class & Methods Definitions

Name	Payment
Type	Class
Purpose	진행되는 결제정보를 컨트롤러에 전달하는 클래스
Overview	N/A
Cross Reference	R2.1, R2.3.1, R2.3.2, R2.4, R3.1.1 "결제 수단 목록 출력", "간편 결제", "카드 결제", "결제 취소", "인증번호, 바코드 출력"
Input	Title_id, DVMid, Pay_confirm
Output	Error_log, CN, integer
Exceptional Courses of Events	N/A

Activity 2051. Implement Class & Methods Definitions

Name	C_NumberManager
Type	Class
Purpose	인증번호 구매 진행시 저장된 인증번호를 확인하는 클래스
Overview	N/A
Cross Reference	R1.2.3, R3.1.1, R3.1.2 "사용자 인증번호/바코드 입력", "인증번호/바코드 생성", 인증번호/바코드 출력"
Input	Title_id, id, C_Number_t
Output	R, integer, String
Exceptional Courses of Events	N/A

Activity 2051. Implement Class & Methods Definitions

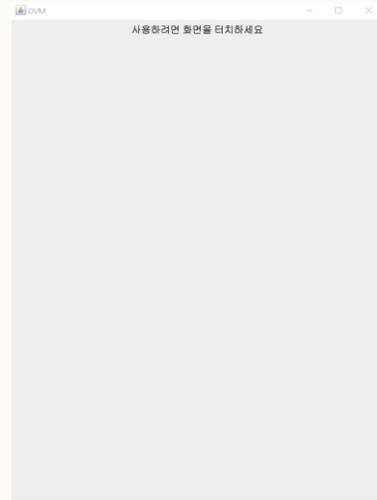
Name	C_Number
Type	Class
Purpose	인증번호를 생성하는 클래스
Overview	N/A
Cross Reference	R3.1.1, R3.1.2 "인증번호/바코드 생성", 인증번호/바코드 출력"
Input	Title_id, DvmID
Output	C_Number_t, numStr
Exceptional Courses of Events	N/A

Activity 2051. Implement Class & Methods Definitions

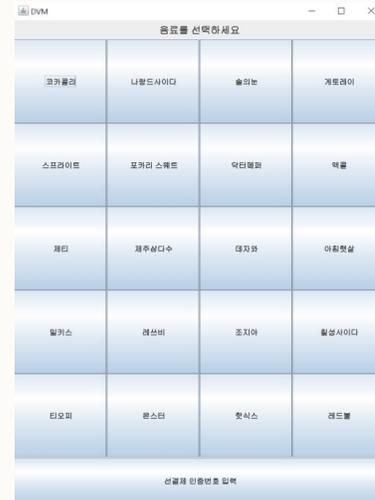
Name	Message_Queue
Type	Class
Purpose	메세지를 수신/발신하고 큐에 저장하는 클래스
Overview	N/A
Cross Reference	R1.4, R3.1.1, R4.3, R4.4, R4.5.1, R4.5.2 "구매가능한 자판기 안내", "인증번호/바코드 생성", "위치 정보 확인", "재고 정보 확인", "메시지 발신", "메시지 수신"
Input	Message, type
Output	temp1
Exceptional Courses of Events	N/A

Activity 2052. Implement Windows (사용자 매뉴얼)

첫 실행화면

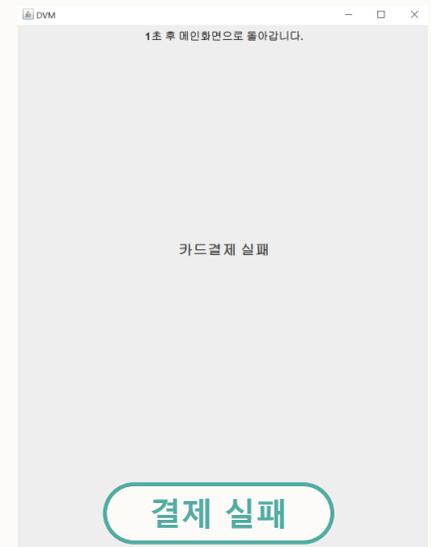


Touch!



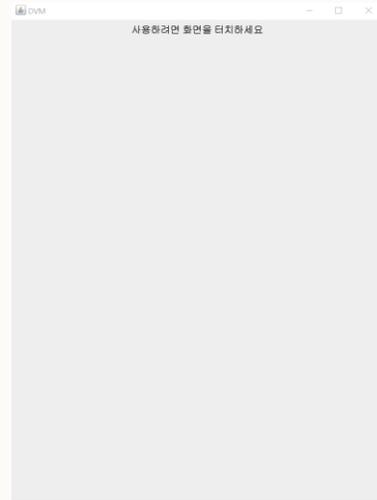
재고가 있는 제품 선택

카드결제

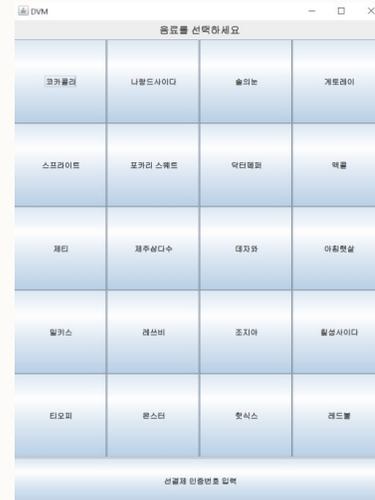


Activity 2052. Implement Windows (사용자 매뉴얼)

첫 실행화면



Touch!



재고가 있는 제품 선택

간편결제



남은 시간: 17

간편결제

취소

결제 성공

1초 후 메인화면으로 돌아갑니다.

코카콜라
음료가 나왔습니다.

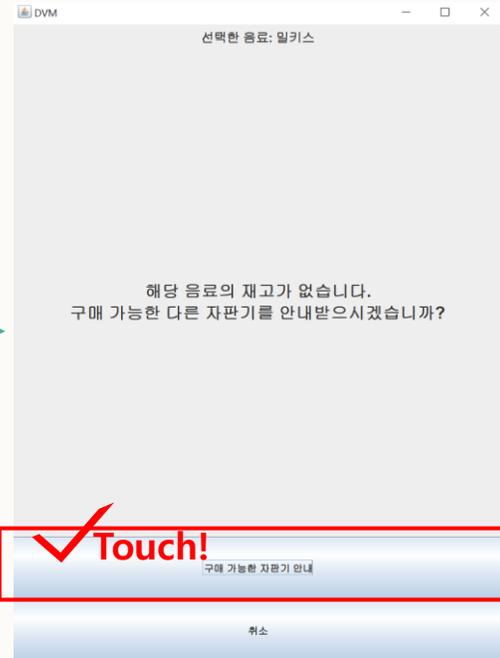
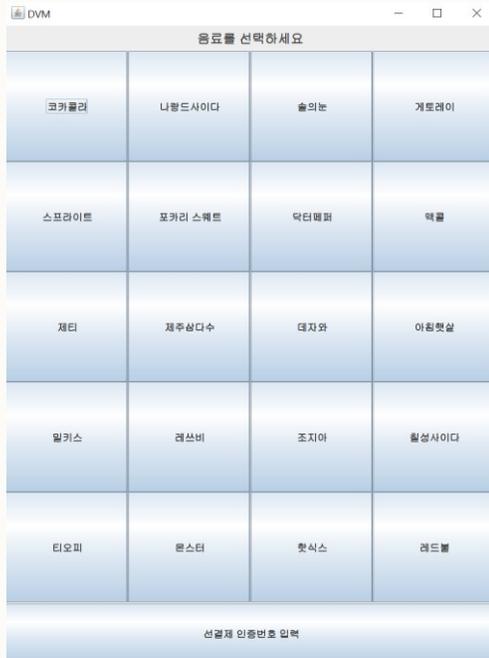
결제 실패

2초 후 메인화면으로 돌아갑니다.

간편결제 실패

Activity 2052. Implement Windows (사용자 매뉴얼)

재고가 존재하지 않는 제품 선택



Activity 2052. Implement Windows (사용자 매뉴얼)

재고가 존재하지 않는 제품 선택



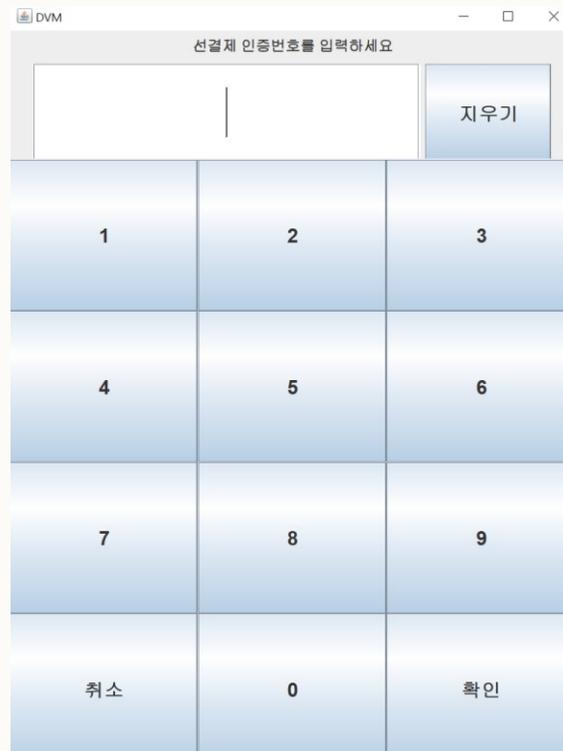
선결제 성공



선결제 실패

Activity 2052. Implement Windows (사용자 매뉴얼)

존재하지 않는 인증번호 입력



DVM 선결제 인증번호를 입력하세요

지우기

1	2	3
4	5	6
7	8	9
취소	0	확인



DVM 2초 후 이전화면으로 돌아갑니다.

해당 인증번호에 대한 선결제 정보를 확인할 수 없습니다.
다시 입력하세요

확인

Activity 2052. Implement Windows (사용자 매뉴얼)

관리자 모드

관리자 고유번호 입력

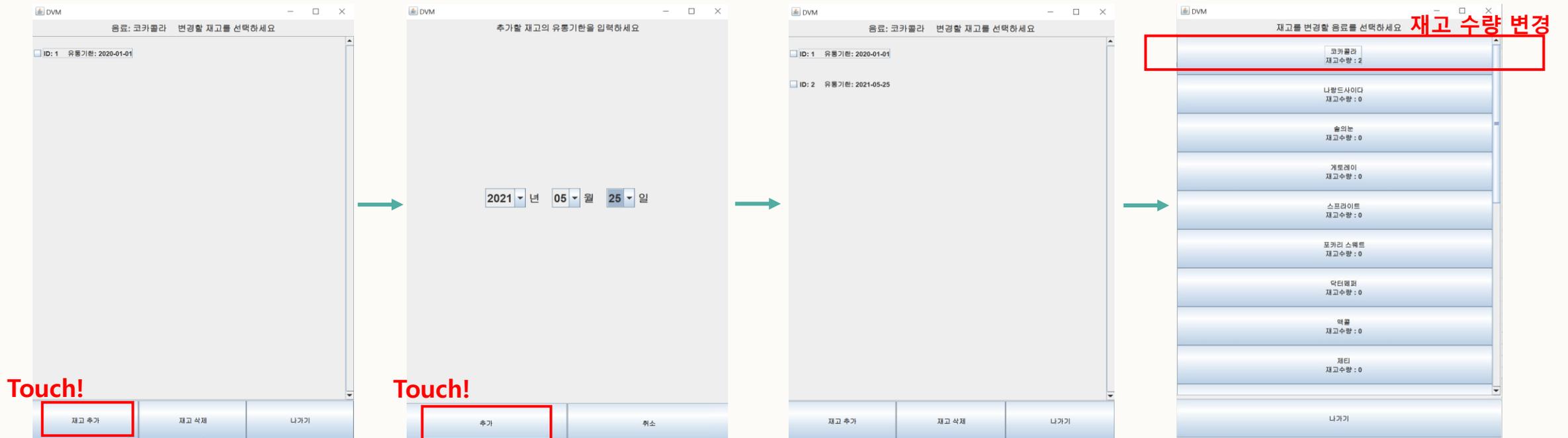
선결제 인증번호를 입력하세요		
111111		
1	2	3
4	5	6
7	8	9
취소	0	확인



코카콜라 재고수량 : 1
나랑드사이다 재고수량 : 0
솔의눈 재고수량 : 0
계토레이 재고수량 : 0
스프라이트 재고수량 : 0
포카리 스웨트 재고수량 : 0
덕터피퍼 재고수량 : 0
맥콜 재고수량 : 0
제티 재고수량 : 0
나가기

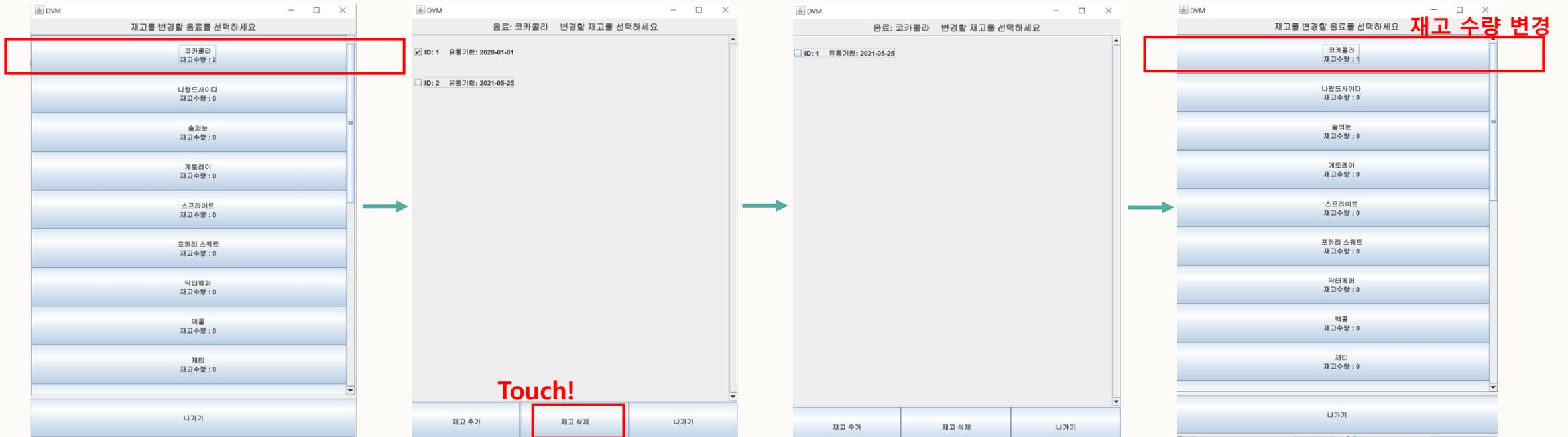
Activity 2052. Implement Windows (사용자 매뉴얼)

재고 추가



Activity 2052. Implement Windows (사용자 매뉴얼)

재고 삭제



Activity 2055. Write Unit Test Code

Class:Title

```
1 package Logic;
2
3 import org.junit.Assert;
4 import org.junit.Test;
5 import java.util.ArrayList;
6 import static org.junit.Assert.*;
7
8 public class TitleTest {
9     private Title t=new Title( Name: "코카콜라", Price: 700);
10    @Test
11    public void addItem() {
12        t.AddItem(new Item( Expiration_Date: 20201125));
13        t.AddItem(new Item( Expiration_Date: 20201125));
14        t.AddItem(new Item( Expiration_Date: 20201125));
15        t.AddItem(new Item( Expiration_Date: 20201125));
16        t.AddItem(new Item( Expiration_Date: 20201125));
17        Assert.assertEquals( expected: 5,t.Item_List().size());
18    }
19    @Test
20    public void deleteItem() {
21        t.AddItem(new Item( Expiration_Date: 20201125));
22        t.AddItem(new Item( Expiration_Date: 20201125));
23        t.AddItem(new Item( Expiration_Date: 20201125));
24        t.AddItem(new Item( Expiration_Date: 20201125));
25        t.AddItem(new Item( Expiration_Date: 20201125));
26        ArrayList<Integer> delitemlist = new ArrayList<>();
27        delitemlist.add(0);delitemlist.add(1);delitemlist.add(2);
28        t.DeleteItem(delitemlist);
29        Assert.assertEquals( expected: 2,t.Item_List().size());
30    }
}
```

```
31 @Test
32 public void item_List() {
33     t.AddItem(new Item( Expiration_Date: 20201025));
34     t.AddItem(new Item( Expiration_Date: 20201125));
35     t.AddItem(new Item( Expiration_Date: 20201125));
36     t.AddItem(new Item( Expiration_Date: 20201125));
37     t.AddItem(new Item( Expiration_Date: 20201125));
38     Assert.assertEquals( expected: 20201025,t.Item_List().get(0).Expiration_Date());
39 }
40 @Test
41 public void updateStock() {
42     t.AddItem(new Item( Expiration_Date: 20201025));
43     t.AddItem(new Item( Expiration_Date: 20201125));
44     t.AddItem(new Item( Expiration_Date: 20201125));
45     t.AddItem(new Item( Expiration_Date: 20201125));
46     t.AddItem(new Item( Expiration_Date: 20201125));
47     t.UpdateStock( id: 1, IfHold: true);
48     Assert.assertEquals( expected: 4,t.Item_List().size());
49     t.UpdateStock( id: 1, IfHold: false);
50     Assert.assertEquals( expected: 3,t.Item_List().size());
51     t.UpdateStock( id: 0, IfHold: true);
52     Assert.assertEquals( expected: 3,t.Item_List().size());
53     t.UpdateStock( id: 0, IfHold: false);
54     Assert.assertEquals( expected: 2,t.Item_List().size());
55 }
56 @Test
57 public void checkStock() {
58     t.AddItem(new Item( Expiration_Date: 20201025));
59     Assert.assertEquals( expected: true,t.CheckStock());
60     t.UpdateStock( id: 0, IfHold: false);
61     Assert.assertEquals( expected: false,t.CheckStock());
62 }
```

Activity 2055. Write Unit Test Code

Class:Item

```
1   package Logic;
2
3   import org.junit.Assert;
4   import org.junit.Test;
5
6   import static org.junit.Assert.*;
7
8   public class ItemTest {
9       Item item;
10      @Test
11      public void Testexpiration_Date() {
12          item=new Item( Expiration_Date: 20201125);|
13          Assert.assertEquals( expected: 20201125,item.Expiration_Date());
14      }
15  }
```

Activity 2055. Write Unit Test Code

Class:Payment

```
1 package Logic;
2 import ...
6 public class PaymentTest {
7     Payment payment;
8     @Test
9     public void cardPay() {
10        payment=new Payment( title_id: 1, DVMid: 1);
11        Assert.assertEquals( expected: true, actual: (payment.CardPay( Pay_confirm: true))>0);
12        Assert.assertEquals( expected: -3,payment.CardPay( Pay_confirm: false));
13        payment=new Payment( title_id: 1, DVMid: 0);
14        Assert.assertEquals( expected: 0,payment.CardPay( Pay_confirm: true));
15        Assert.assertEquals( expected: -3,payment.CardPay( Pay_confirm: false));
16    }
17    @Test
18    public void smartPay() {
19        payment=new Payment( title_id: 1, DVMid: 1);
20        Assert.assertEquals( expected: true, actual: (payment.SmartPay( Pay_confirm: true))>0);
21        Assert.assertEquals( expected: -3,payment.SmartPay( Pay_confirm: false));
22        payment=new Payment( title_id: 1, DVMid: 0);
23        Assert.assertEquals( expected: 0,payment.SmartPay( Pay_confirm: true));
24        Assert.assertEquals( expected: -3,payment.SmartPay( Pay_confirm: false));
25    }
26    @Test
27    public void init() {
28        payment=new Payment( title_id: 1, DVMid: 1);
29        payment.init();
30        Assert.assertEquals( expected: -1,payment.title_id);
31        Assert.assertEquals( expected: -1,payment.DVMid);
32        Assert.assertEquals( expected: null,payment.Error_log());
33    }
34 }
```

Activity 2055. Write Unit Test Code

Class:C_Number, Class: C_NumberManager

```
package Logic;

import org.junit.*;

import static org.junit.Assert.*;

public class C_NumberTest {
    private C_Number CN=new C_Number( title_id: 1, id: 1);
    @Test
    public void testCreateCnumber() {
        Assert.assertEquals(CN.CreateCnumber( title_id: 1, DvmlID: 1),CN.getC_Number_t());
    }
}
```

```
1 package Logic;
2
3 import ...
4
5
6
7
8 public class C_NumberManagerTest {
9     private C_NumberManager C_NumberManager;
10    private C_Number C_Number;
11    //
12    @Test
13    public void popCnumber() {
14        C_NumberManager = new C_NumberManager();
15        C_Number = new C_Number();
16        int num = C_Number.CreateCnumber( title_id: 5, DvmlID: 0);
17        C_NumberManager.C_List.put(0, C_Number);
18        int expectedResult = C_Number.title_id;
19        int actualResult = C_NumberManager.title_id;
20        Assert.assertEquals(expectedResult, actualResult);
21    }
22
23    @Test
24    public void checkCnumber() {
25        C_NumberManager = new C_NumberManager();
26        C_Number = new C_Number();
27        int num = C_Number.CreateCnumber( title_id: 5, DvmlID: 0);
28        C_NumberManager.C_List.put(0, C_Number);
29        int expectedResult = 1;
30        int actualResult = C_NumberManager.CheckCnumber(num);
31        Assert.assertEquals(expectedResult, actualResult);
32    }
33 }
```

Activity 2055. Write Unit Test Code

Class:Message_Queue

```
1 package Logic;
2
3
4 import org.junit.Assert;
5 import org.junit.Test;
6
7 public class Message_QueueTest {
8     private Message message;
9     private Message_Queue message_queue;
10
11     @Test
12     public void sendMsg() {
13         message = new Message( myid: 1);
14         message_queue = new Message_Queue();
15         message.setmsg( id: 1, type: 1, data: 1);
16         Message_Queue.sendMsg(message);
17         Assert.assertEquals(message, message_queue.Connection.poll());
18     }
19
20     @Test
21     public void dequeue() {
22         message = new Message( myid: 1);
23         message_queue = new Message_Queue();
24         message.setmsg( id: 1, type: 1, data: 1);
25         Message_Queue.sendMsg(message);
26         Message_Queue.recivMsg();
27         message = (Message) Message_Queue.dequeue( type: 1).poll();
28         Assert.assertEquals( expected: 1, message.Type);
29     }
30 }
```

Activity 2055. Write Unit Test Code

Class:Controller

```
1 package Logic;
2
3 import GUI.*;
4 import org.junit.Assert;
5 import org.junit.Test;
6
7 import java.util.Timer;
8
9 public class ControllerTest {
10     Controller c= new Controller();
11     Timer t= new Timer();
12
13
14     @Test
15     public void inputSelect() {
16         int del;
17         c.k=new MainMenu(c.Title_List);
18         del=c.InputSelect();
19         Assert.assertEquals( expected: -2,del);
20         c.k.setVisible(false);
21     }
22
23     @Test
24     public void showInputLine() {
25         int del;
26         c.k=new MainMenu(c.Title_List);
27         del=c.InputSelect();
28         Assert.assertEquals( expected: 0,del);
29         c.k.setVisible(false);
30     }
31 }
```

```
33     @Test
34     public void inputCnumber() {
35         int del;
36         c.k=new InputLine();
37         del= c.InputCnumber();
38         Assert.assertEquals( expected: 0,del);
39         c.k.setVisible(false);
40     }
41
42     @Test
43     public void infoCnumberError() {
44         int del;
45         c.k=new InputLine();
46         del= c.InputCnumber();
47         Assert.assertEquals( expected: 1,del);
48         c.k.setVisible(false);
49     }
50
51     @Test
52     public void manShowTitle() {
53         c.ManShowTitle();
54         Assert.assertEquals( expected: 0,c.del);
55         c.k.setVisible(false);
56     }
57
58     @Test
59     public void manSelectTitle() {
60         c.ManShowTitle();
61         Assert.assertEquals( expected: 0,c.del);
62         c.k.setVisible(false);
63     }
```

Activity 2055. Write Unit Test Code

Class:Controller

```
65     @Test
66     public void manShowItem() {
67         c.Title_List.get(0).AddItem(new Item( Expiration_Date: 20250101));
68         c.ManShowItem( TitleID: 1);
69         Assert.assertEquals( expected: 0,c.del);
70         c.k.setVisible(false);
71     }
72
73     @Test
74     public void manEditItem() {
75         c.Title_List.get(0).AddItem(new Item( Expiration_Date: 20250101));
76         c.ManShowItem( TitleID: 1);
77         Assert.assertEquals( expected: 2,c.Title_List.get(0).Item_List().size());
78         c.ManShowItem( TitleID: 1);
79         Assert.assertEquals( expected: 0,c.Title_List.get(0).Item_List().size());
80         c.k.setVisible(false);
81     }
82
83     @Test
84     public void printSelectPay() {
85         c.k=new MainMenu(c.Title_List);
86         c.Title_List.get(0).AddItem(new Item( Expiration_Date: 20201125));
87         c.InputSelect();
88         Assert.assertEquals( expected: 1,c.Title_List.get(0).Item_List().size());
89         c.k.setVisible(false);
90     }
91
```

```
92     @Test
93     public void selectPayment() {
94         c.k=new PaymentMenu(new Title( Name: "test", Price: 700));
95         c.basket=1;
96         c.SelectPayment( DVMid: 0);
97         Assert.assertEquals( expected: 0,c.del);
98         c.k.setVisible(false);
99     }
100
101     @Test
102     public void cancelItem() {
103         c.basket=1;
104         c.Payment=new Payment( title_id: 1, DVMid: 0);
105         c.CancelItem();
106         Assert.assertEquals( expected: -666,c.basket);
107         Assert.assertEquals( expected: true,c.DVMStack.isEmpty());
108         Assert.assertEquals( expected: null,c.Payment);
109         c.k.setVisible(false);
110     }
111
112     @Test
113     public void showCardPay() {
114         c.Payment=new Payment( title_id: 1, DVMid: 0);
115         c.ShowCardPay();
116         c.k.setVisible(false);
117     }
118
```

Activity 2055. Write Unit Test Code

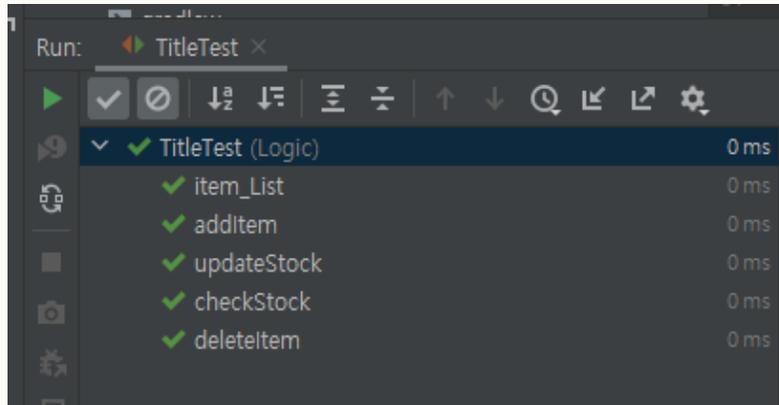
Class:Controller

```
118
119
120 @Test
121 public void cardPay() {
122     c.k=new CardPayVI();
123     c.Title_list.get(0).AddItem(new Item(ExpirationDate: 20201125));
124     c.basket=1;
125     c.Payment=new Payment(ItemId: 1, DVNid: 0);
126     c.CardPay();
127     Assert.assertEquals( expected: 0,c.Title_list.get(0).Item_list().size());
128     c.k.setVisible(false);
129 }
130
131 @Test
132 public void showSmartPay() {
133     c.Payment=new Payment(ItemId: 1, DVNid: 0);
134     c.ShowSmartPay();
135     c.k.setVisible(false);
136 }
137
138 @Test
139 public void smartPay() {
140     c.k=new SmartPayVI();
141     c.Title_list.get(0).AddItem(new Item(ExpirationDate: 20201125));
142     c.basket=1;
143     c.Payment=new Payment(ItemId: 1, DVNid: 0);
144     c.SmartPay();
145     Assert.assertEquals( expected: 0,c.Title_list.get(0).Item_list().size());
146     c.k.setVisible(false);
147 }
```

```
148
149
150 public void returnItem() {
151     c.Title_list.get(0).AddItem(new Item(ExpirationDate: 20201125));
152     c.basket=1;
153     c.Payment=new Payment(ItemId: 1, DVNid: 0);
154     c.ReturnItem(c.Title_list.get(0), DVNid: false);
155     Assert.assertEquals( expected: 0,c.Title_list.get(0).Item_list().size());
156     c.k.setVisible(false);
157 }
158
159 @Test
160 public void printNumber() {
161     c.basket=1;
162     c.Payment=new Payment(ItemId: 1, DVNid: 1);
163     c.PrintNumber(c.Title_list.get(0), DVNid: 1, Dumber: 00000);
164     c.k.setVisible(false);
165 }
166
167 @Test
168 public void infoNoItem() {
169     c.basket=1;
170     c.InfoNoItem();
171     c.k.setVisible(false);
172 }
173
174 @Test
175 public void reqFindDVN() {
176     c.basket=1;
177     c.InfoNoItem();
178     c.k.setVisible(false);
179 }
180
181 @Test
182 public void init() {
183     c.basket=1;
184     c.Payment=new Payment(ItemId: 1, DVNid: 0);
185     c.CancelItem();
186     Assert.assertEquals( expected: -000,c.basket);
187     Assert.assertEquals( expected: true,c.DVNStack.isEmpty());
188     Assert.assertEquals( expected: null,c.Payment);
189 }
190 }
```

Activity 2061. Unit Testing

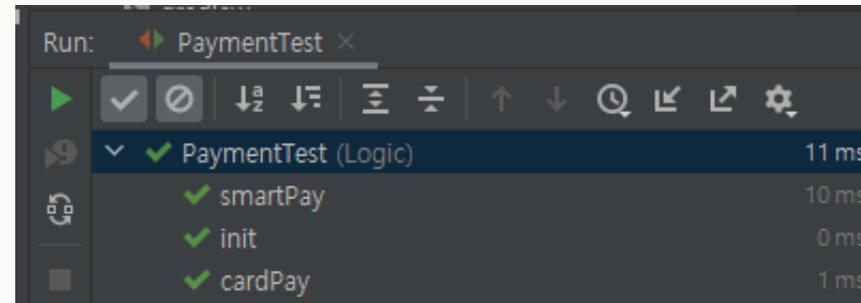
TitleTest



Run: TitleTest x

Test Name	Duration
✓ TitleTest (Logic)	0 ms
✓ item_List	0 ms
✓ addItem	0 ms
✓ updateStock	0 ms
✓ checkStock	0 ms
✓ deleteItem	0 ms

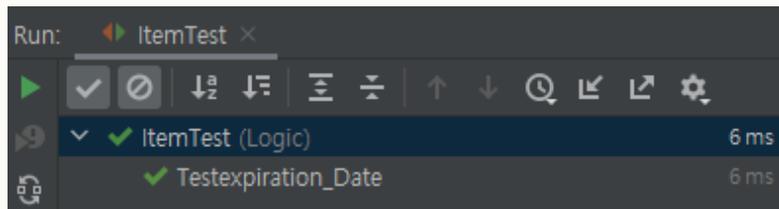
PaymentTest



Run: PaymentTest x

Test Name	Duration
✓ PaymentTest (Logic)	11 ms
✓ smartPay	10 ms
✓ init	0 ms
✓ cardPay	1 ms

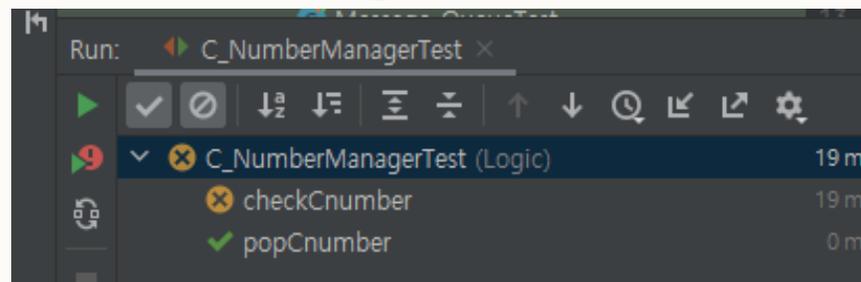
ItemTest



Run: ItemTest x

Test Name	Duration
✓ ItemTest (Logic)	6 ms
✓ Testexpiration_Date	6 ms

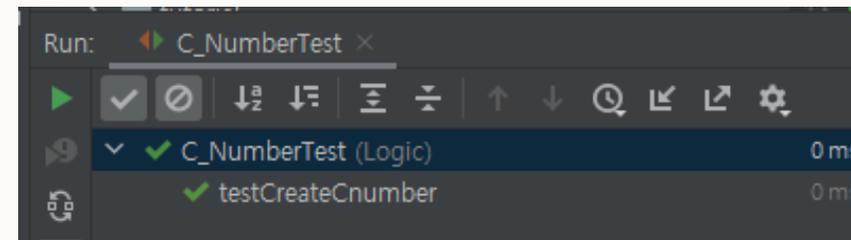
C_NumberManagerTest



Run: C_NumberManagerTest x

Test Name	Duration
✗ C_NumberManagerTest (Logic)	19 ms
✗ checkCNumber	19 ms
✓ popCNumber	0 ms

C_NumberTest

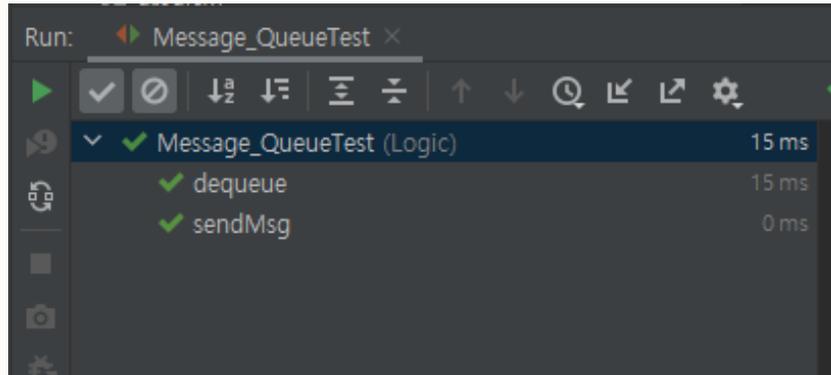


Run: C_NumberTest x

Test Name	Duration
✓ C_NumberTest (Logic)	0 ms
✓ testCreateCNumber	0 ms

Activity 2061. Unit Testing

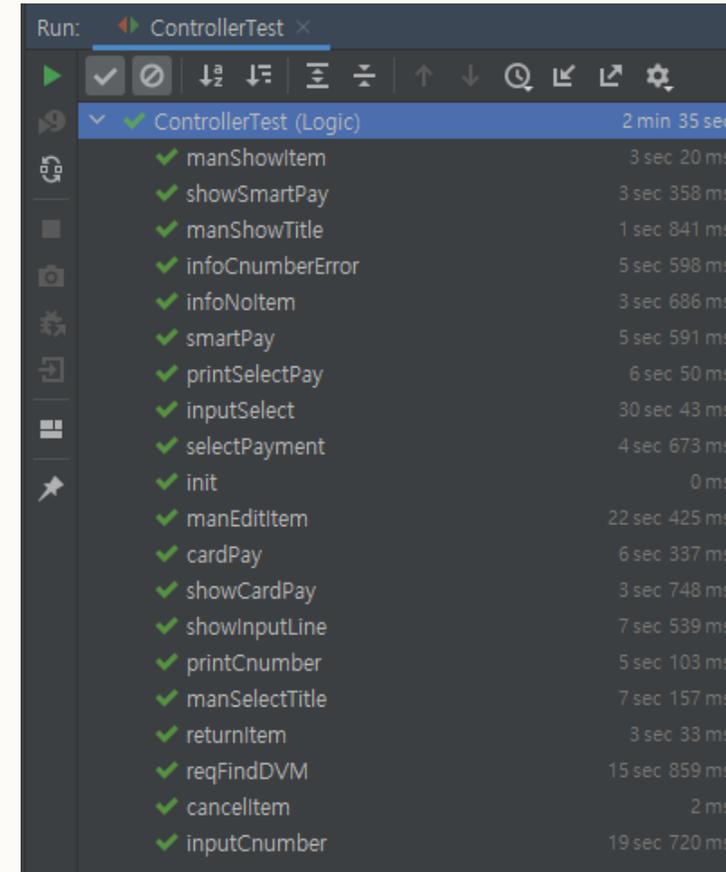
Message_QueueTest



Run: Message_QueueTest x

Test Name	Duration
Message_QueueTest (Logic)	15 ms
✓ dequeue	15 ms
✓ sendMsg	0 ms

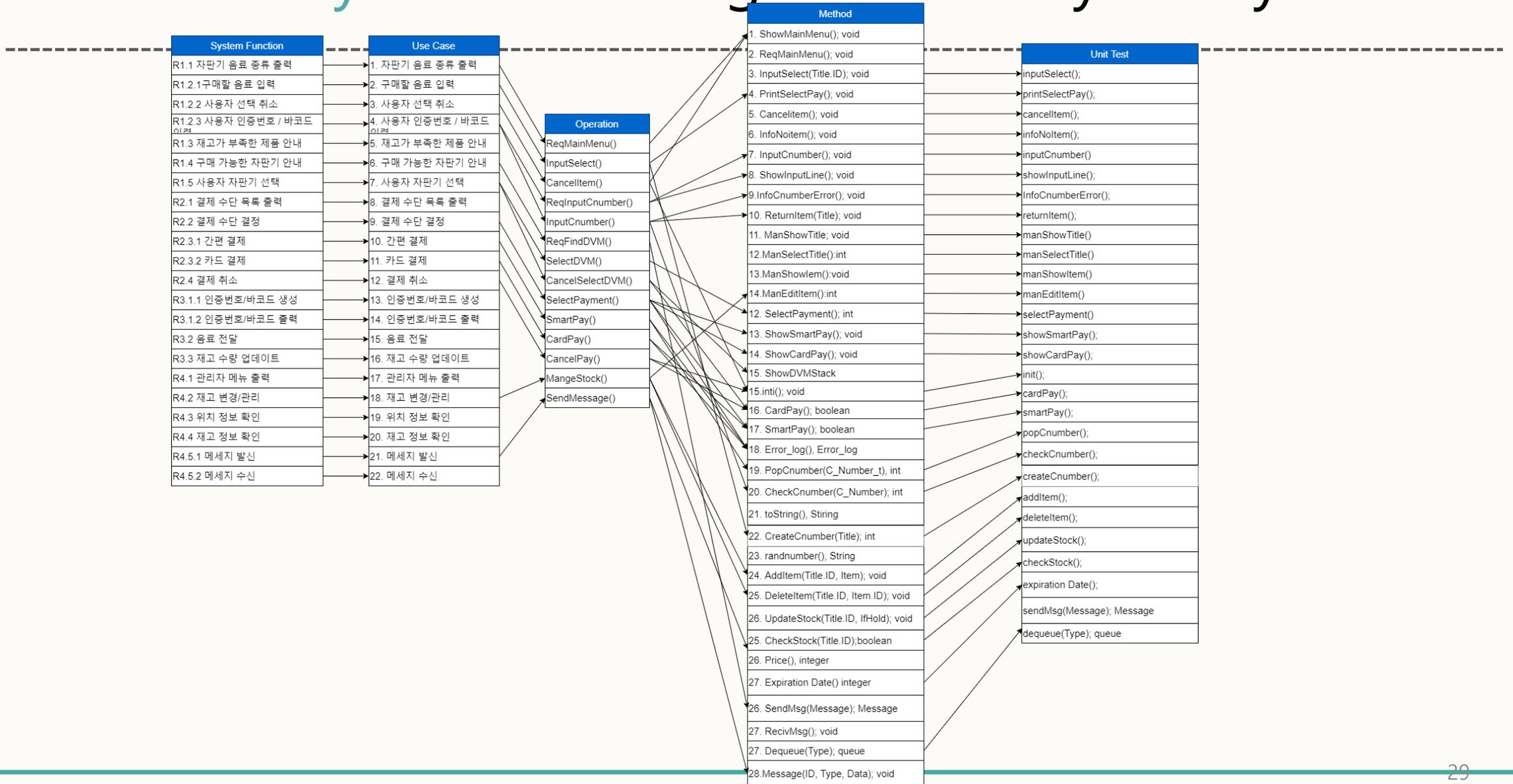
ControllerTest



Run: ControllerTest x

Test Name	Duration
ControllerTest (Logic)	2 min 35 sec
✓ manShowItem	3 sec 20 ms
✓ showSmartPay	3 sec 358 ms
✓ manShowTitle	1 sec 841 ms
✓ infoCnumberError	5 sec 598 ms
✓ infoNoItem	3 sec 686 ms
✓ smartPay	5 sec 591 ms
✓ printSelectPay	6 sec 50 ms
✓ inputSelect	30 sec 43 ms
✓ selectPayment	4 sec 673 ms
✓ init	0 ms
✓ manEditItem	22 sec 425 ms
✓ cardPay	6 sec 337 ms
✓ showCardPay	3 sec 748 ms
✓ showInputLine	7 sec 539 ms
✓ printCnumber	5 sec 103 ms
✓ manSelectTitle	7 sec 157 ms
✓ returnItem	3 sec 33 ms
✓ reqFindDVM	15 sec 859 ms
✓ cancelItem	2 ms
✓ inputCnumber	19 sec 720 ms

Activity 2067. Testing Traceability Analysis



Demonstration Video

https://drive.google.com/file/d/1M_4SKk2iUdU3v1aiQV3VzzW1H8dnFImg/view?usp=drivesdk