



DVM OOPT 2050

객체지향개발방법론 3팀

권민혁, 노성준, 이서준

목차

2151

Implement Class &
Methods Definitions

2152

Implement Windows

2153

Implement Reports

2154

Implement DB
Schema(SQL, etc.)

2155

Write Unit Test Code

2151. Implement Class & Methods Definitions

프로젝트 구조

```
CICD-TEAM3
> .github
> .vscode
> include
✓ src
  > Domain
  > Exception
  > main.dSYM
  > Services
  G+ main.cpp
> test
◆ .gitignore
≡ card_db.txt
M CMakeLists.txt
① README.md
⚙ sonar-project.properties
```

빌드 환경 및 의존성

- 컴파일러: g++ (C++ 17 이상)
- CMake 버전 3.16 이상
- 테스트 프레임워크:
GoogleTest

2151. Implement Class & Methods Definitions

빌드 명령어

```
cmake -S . -B build -DCMAKE_BUILD_TYPE=Debug  
cmake --build build
```

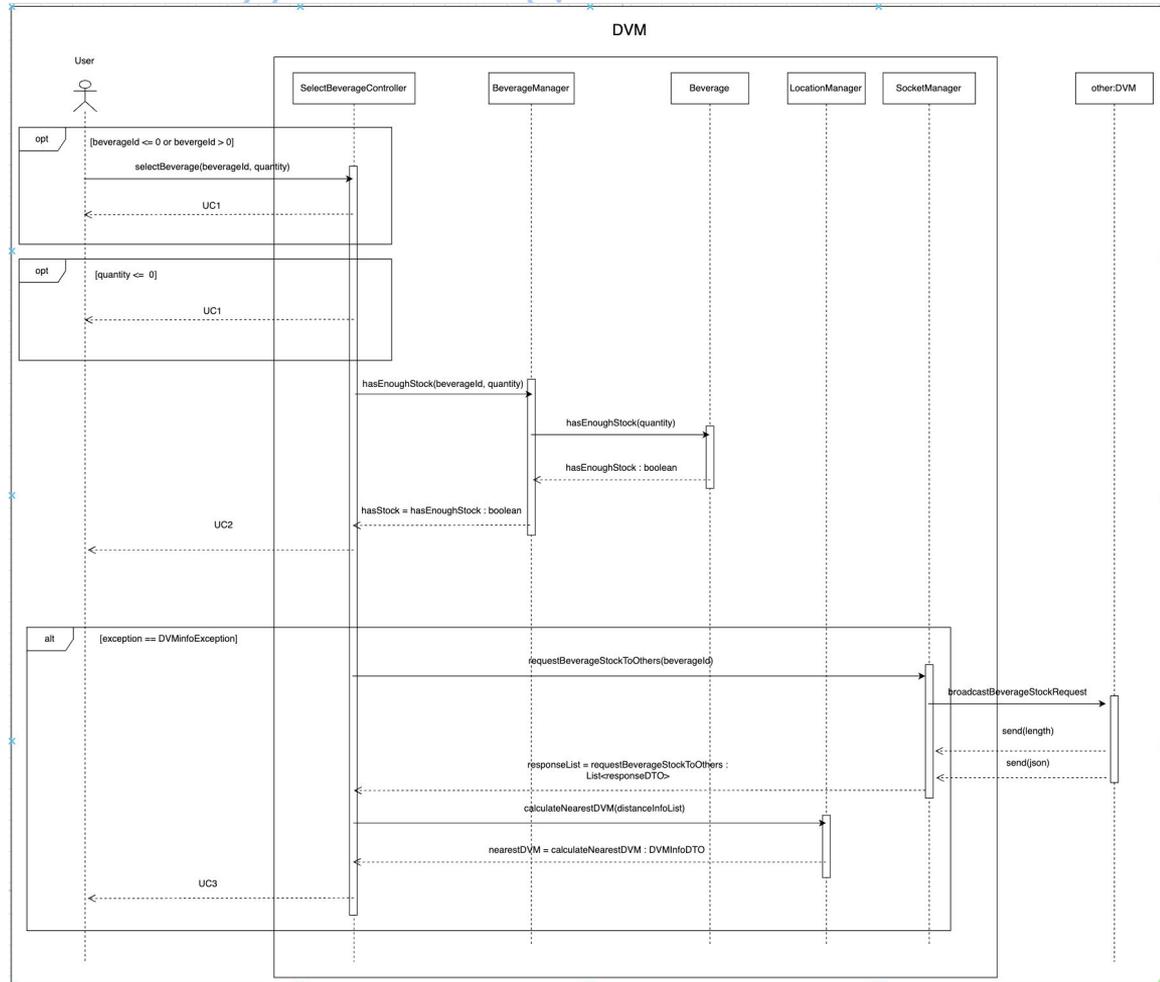
실행 명령어

```
cd build  
./sonar_scanner_example 1 9001  
(//argv[1] : srcId, argv[2] : server socket port)
```


2151. Implement Class & Methods Definitions

UC1. 음료 선택

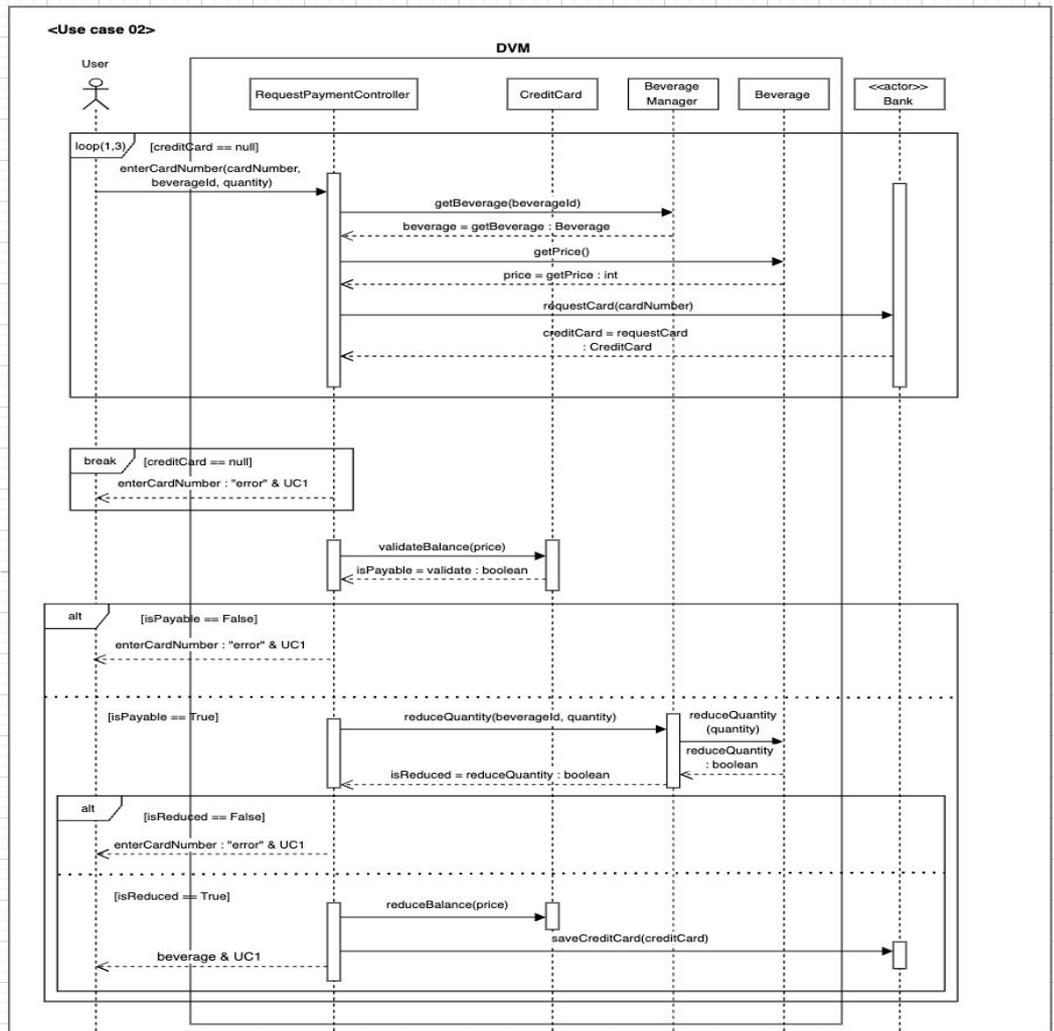
1. (U) 사용자는 원하는 음료의 [ID, 원하는 수량]을 입력한다.
2. (D) DVM은 사용자가 입력한 음료 ID가 유효한지 확인한다.
3. (D) DVM은 사용자가 입력한 음료 ID에 해당하는 음료의 재고가 사용자로부터 받은 수량보다 크거나 같은지 확인한다.
4. (D) 음료의 재고가 있다면 사용자에게 결제 요청 화면을 보여준다.
5. (D) 음료의 재고가 없다면 다른 모든 DVM에게 해당 음료의 재고를 요청하여 확인한다.
6. (D) DVM은 자신과 가장 가깝고, 사용자가 구매하고자 하는 음료의 재고가 있는 자판기의 위치를 계산한 후, 선결제 요청 화면을 보여준다.



2151. Implement Class & Methods Definitions

UC2. 결제 요청

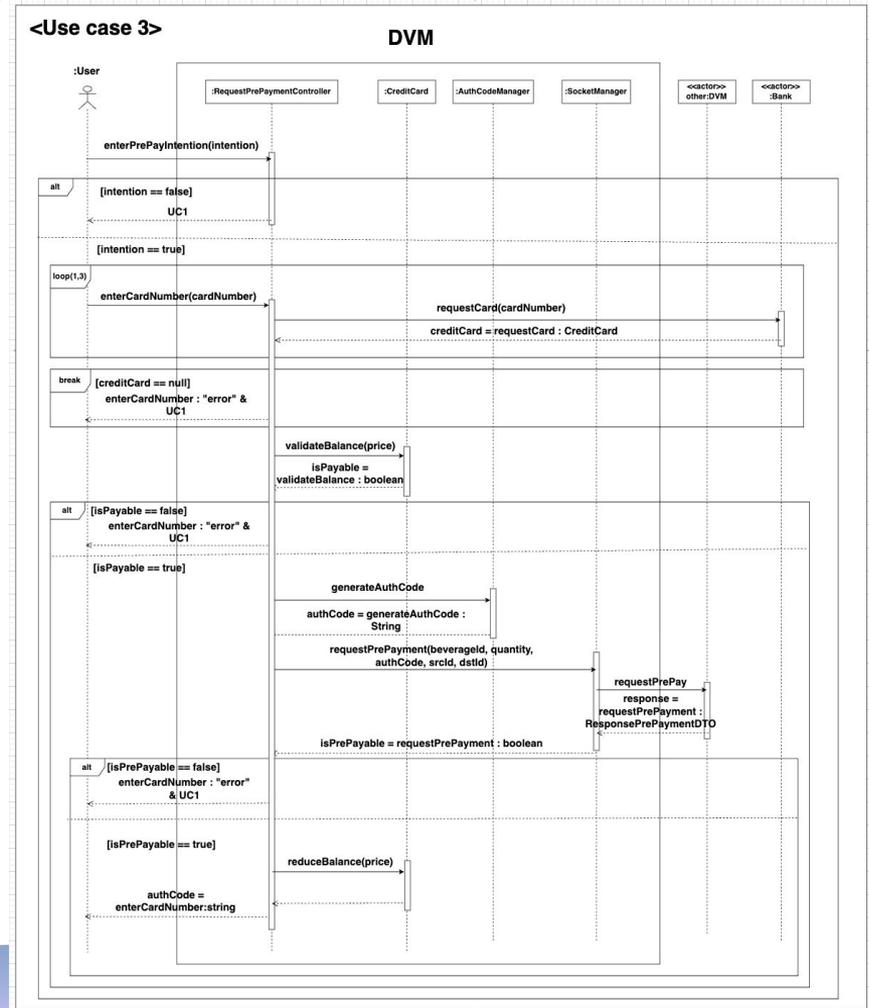
1. (U) 사용자는 카드 번호를 입력한다.
2. (D) DVM은 사용자가 입력한 카드 번호로 Bank 에게 카드를 요청한다.
3. (B) Bank는 카드 번호가 유효하다면, 카드 번호에 해당하는 카드를 반환한다.
4. (D) DVM은 카드의 잔액이 음료 구매가보다 크거나 같을 경우,음료의 재고를 반환하고 카드의 잔액을 음료 구매가만큼 차감한다.
5. (D) DVM은 사용자에게 음료를 제공한다.



2151. Implement Class & Methods Definitions

UC3. 선결제 요청

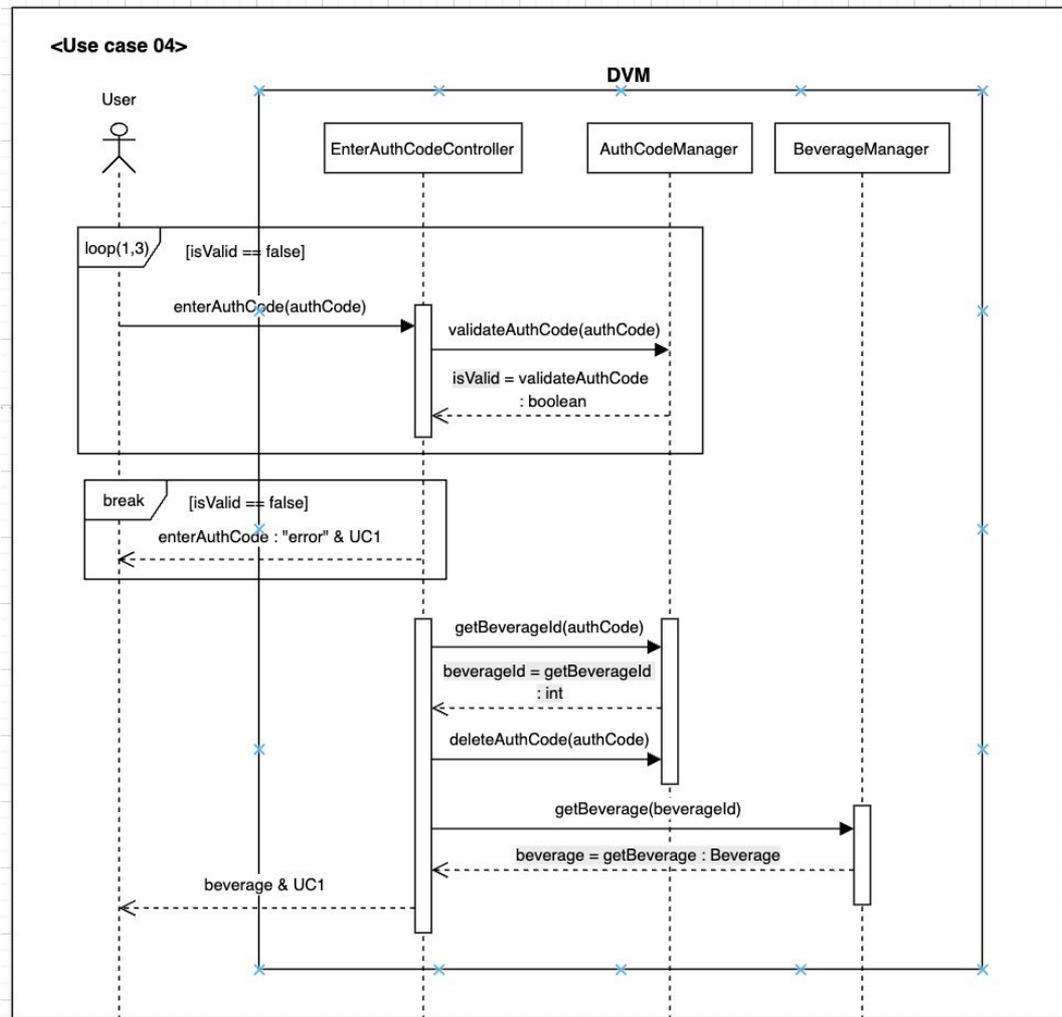
1. (U) 사용자는 선결제 의사를 입력한다.
2. (U) 선결제하고자 하는 사용자는 카드 번호를 입력한다.
3. (D) DVM은 사용자가 입력한 카드 번호로 Bank 에게 카드를 요청한다.
4. (B) Bank는 카드 번호가 유효하다면, 카드 번호에 해당하는 카드를 반환한다.
5. (D) DVM은 카드의 잔액이 음료 구매가보다 크거나 같을 경우, 인증코드를 생성한다.
6. (D) DVM은 사용자가 선결제하고자 하는 다른 DVM에게 생성한 인증코드와 선결제하고자 하는 음료 id, 수량 정보를 포함해서 선결제 요청을 보낸다.
7. (D) DVM은 다른 DVM에게 선결제 성공 응답을 받으면, 사용자에게 인증코드를 제공한다.



2151. Implement Class & Methods Definitions

UC4. 인증 코드 입력 후 음료 수령

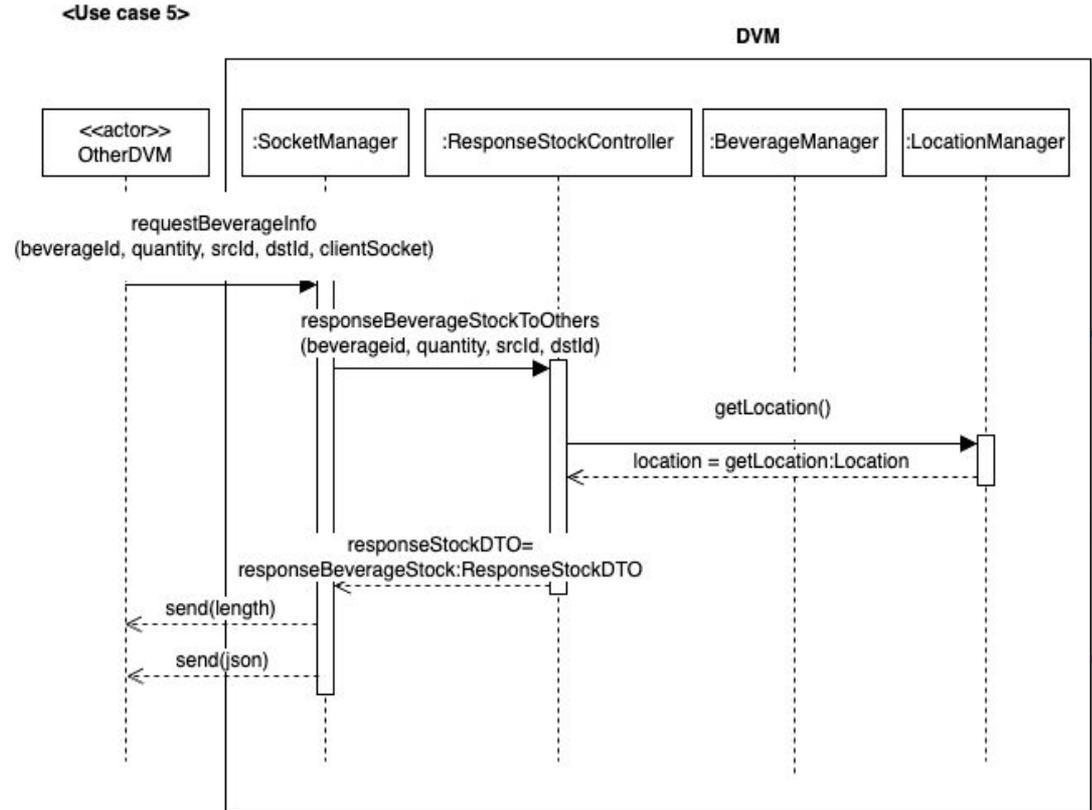
1. (U) 사용자는 인증 코드를 입력한다
2. (D) DVM은 사용자가 입력한 인증코드가 유효한지 검사한다.
3. (D) 인증코드가 유효하다면 DVM은 사용자에게 음료를 제공한다.



2151. Implement Class & Methods Definitions

UC5. 선결제시 재고 확인 응답

1. (O) 다른 DVM은 DVM에게 특정 음료에 대한 재고 확인 요청을 전송한다.
2. (D) DVM은 다른 DVM으로부터 재고 확인 요청 메시지를 수신한다.
3. (D) DVM은 재고 확인 요청에 포함된 음료의 재고를 확인한다.
4. (D) DVM은 음료의 재고와 자신의 위치 정보를 응답한다.

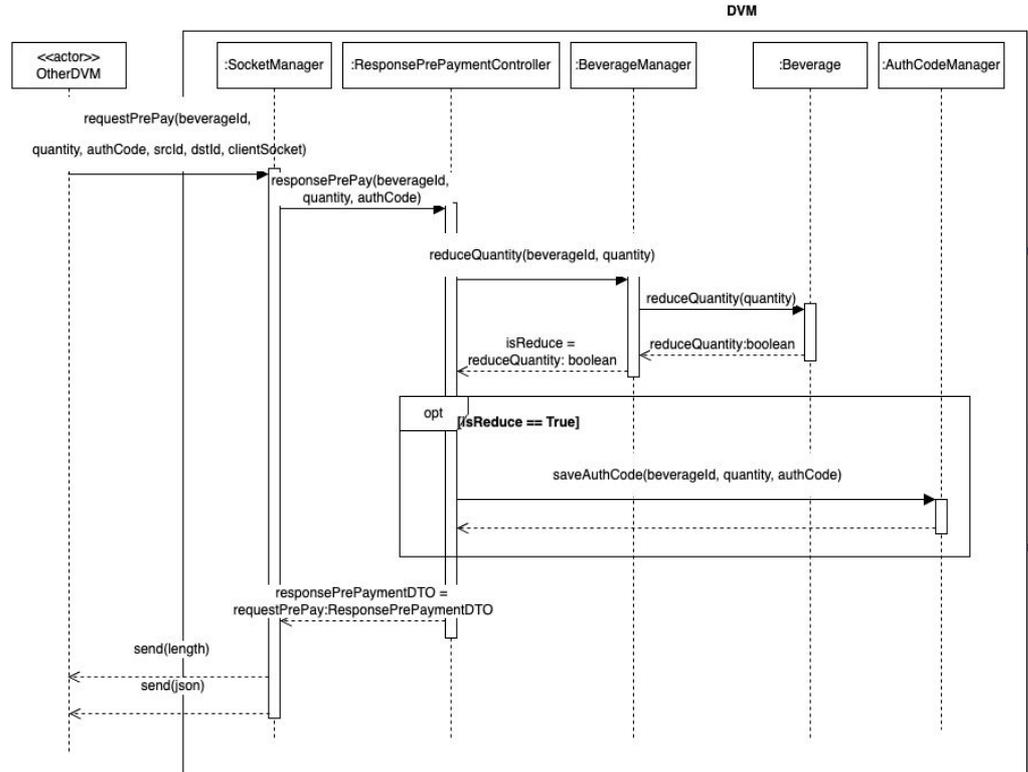


2151. Implement Class & Methods Definitions

UC6. 선결제 요청 응답

1. (O) 다른 DVM은 DVM에게 특정 음료에 대한 선결제를 요청한다.
2. (D) DVM은 선결제 요청에 포함된 음료의 재고를 확인한다.
3. (D) DVM은 해당 음료의 재고가 있다면 재고를 차감하고, 선결제 요청에 포함된 인증 코드를 저장한 후, 선결제 성공을 응답한다.

<Use case 6>



2152. Implement Windows

ID	상품명	가격
1	콜라	1200₩
2	사이다	1100₩
3	녹차	1600₩
4	홍차	1300₩
5	밀크티	1400₩
6	탄산수	2000₩
7	보리차	2500₩
8	캔커피	2600₩
9	물	1500₩
10	에너지드링크	1700₩
11	유자차	1800₩
12	식혜	1900₩
13	아이스티	2000₩
14	딸기주스	2100₩
15	오렌지주스	2200₩
16	포도주스	2300₩
17	이온음료	2400₩
18	아메리카노	2500₩
19	핫초코	2600₩
20	카페라떼	2700₩

메뉴를 선택하세요 (1: 음료 선택, 2: 선결제 코드 입력, 0: 종료): 1
음료 아이디를 입력하세요 (1~20): 2
수량을 입력하세요 (1~10): 1

UC1에 대한 CLI 화면

1. 음료 메뉴 출력
2. 메뉴 입력
3. 1번 메뉴 입력
4. 음료 아이디 입력
5. 음료 수량 입력

2152. Implement Windows

```
수당을 입력하세요 (1~10): 1  
음료 결제  
카드 번호를 입력하세요 : 1234-1234-1234  
[Bank] 해당 카드 번호 없음 : 1234-1234-1234  
[1번 실패] 올바르지 않은 카드번호입니다. 다시 입력해주세요 : 1111-2222-3333-4444  
[Bank] 카드 조회 성공 : 1111-2222-3333-4444 (잔액 : 185000)  
결제 성공 : 2
```

UC2 에 대한 CLI 화면

(현재 자판기에 재고가 있는 경우)

1. 카드번호 입력
2. 잘못된 경우 최대 3번 입력
3. 카드 조회 성공 출력
4. 결제 성공 출력

2152. Implement Windows

```
음료 선결제
가장 가까운 DVM 정보 : DvmId = 1, 위치 = (0, 0)
선결제 의사를 입력하세요 (1: 선결제, 0: 일반 결제) : 0
```

```
=====
```

ID	상품명	가격
1	콜라	1200W
2	사이다	1100W

```
음료 선결제
가장 가까운 DVM 정보 : DvmId = 1, 위치 = (0, 0)
선결제 의사를 입력하세요 (1: 선결제, 0: 일반 결제) : 1
선결제 의사를 확인했습니다.
선결제를 진행합니다.
카드 번호를 입력하세요 현재 1 : 1234-1234-1234-1234
CARD NUMBER1234-1234-1234-1234
[Bank] 해당 카드 번호 없음 : 1234-1234-1234-1234
Not Found: 해당 카드 번호 없음 : 1234-1234-1234-1234 다시 입력하세요.
카드 번호를 입력하세요 현재 2 : 1111-2222-3333-4444
CARD NUMBER1111-2222-3333-4444
[Bank] 카드 조회 성공 : 1111-2222-3333-4444 (잔액 : 10000)
선결제 성공 : SqmLt
음료 결제
```

UC3 에 대한 CLI 화면

(현재 자판기에 재고가 없는 경우)

1. 가까운 DVM 위치 출력
2. 선결제 의사 입력
3. 카드번호 입력
4. 잘못된 경우 최대 3번 입력
5. 카드 조회 여부 출력
6. 인증코드 출력

2152. Implement Windows

```
2
인증 코드를 입력하세요 : SqmLt
beverageId : 7
인증 코드 확인 성공! 음료를 받으세요 : 7
```

UC 4 에 대한 CLI 화면

1. 메뉴에서 2번 선택
2. 인증코드 입력
3. 인증 코드 확인 여부
출력
4. 음료 제공

2155. Write Unit Test Code

Number	Class	MethodName	testMethodName	Description	Result
1	AuthCodeManager	validateAuthCode	ValidateAuthCode_등록된 코드_True반환	미리 저장된 코드 "ABC12", "ABC34"에 대해 validateAuthCode가 true를 반환하는지 확인	true 반환
2	AuthCodeManager	validateAuthCode	ValidateAuthCode_없는코드_False반환	존재하지 않는 코드에 대해 NotFoundException이 발생하는지 확인	NotFoundException 발생
3	AuthCodeManager	getBeverageId	GetBeverageId_등록된코드_Id반환	미리 저장된 코드로 getBeverageId 호출 시 올바른 음료 ID를 반환하는지 확인	1, 2 반환
4	AuthCodeManager	getBeverageId	GetBeverageId_없는코드_exception_반환	존재하지 않는 코드로 getBeverageId 호출 시 NotFoundException이 발생하는지 확인	NotFoundException 발생
5	AuthCodeManager	generateAuthCode	GenerateAuthCode_생성된_인증코드_길이는_5	generateAuthCode로 생성된 코드의 길이가 5인지 확인	길이 5
6	AuthCodeManager	generateAuthCode	GenerateAuthCode_영어와_숫자만포함	생성된 코드가 영문 대소문자와 숫자로만 구성되어 있는지 확인	모두 영숫자 (isalnum)

2155. Write Unit Test Code

Number	Class	MethodName	testMethodName	Description	Result
7	AuthCodeManager	deleteAuthCode	DeleteAuthCode_등록된코드_성공	기존에 저장된 코드를 삭제한 뒤, 다시 조회하면 NotFoundException이 발생하는지 확인	삭제 후 조회 시 NotFoundException 발생
8	AuthCodeManager	deleteAuthCode	DeleteAuthCode_없는코드_exception	존재하지 않는 코드를 삭제하려 할 때 NotFoundException이 발생하는지 확인	NotFoundException 발생
9	AuthCodeManager	saveAuthCode	SaveAuthCode_새로운코드_저장후_조회성공	새로운 코드 "XYZ99"를 저장한 뒤 validateAuthCode와 getBeverageId가 정상 동작하는지 확인	true 반환, ID=7 반환
10	AuthCodeManager	deleteAuthCode	DeleteAuthCode_하나삭제_다른코드유효	"ABC12"만 삭제한 뒤 "ABC34"는 여전히 유효한지 확인	"ABC34" -> true 반환, ID=2
11	AuthCodeManager	saveAuthCode	SaveAuthCode_중복키_기존값유지	기존에 "ABC12"로 저장된 값이 덮어쓰이지 않고 유지되는지 확인	ID=1 반환 (덮어쓰기 없이 유지)

2155. Write Unit Test Code

Number	Class	MethodName	testMethodName	Description	Result
12	Beverage	getId	Beverage_getId_성공	생성자에 전달한 id(42)가 getId()로 반환되는지 확인	42 반환
13	Beverage	getPrice	Beverage_getPrice_성공	생성자에 전달한 price(1500)가 getPrice()로 반환되는지 확인	1500 반환
14	Beverage	hasEnoughStock	hasEnoughStock_0이면_TRUE반환	stock=5, 요청량 0일 때 true 반환	true 반환
15	Beverage	hasEnoughStock	hasEnoughStock_재고보다 작으면_TRUE반환	stock=5, 요청량 3일 때 true 반환	true 반환
16	Beverage	hasEnoughStock	hasEnoughStock_재고보다 1만큼 작아도_TRUE반환_경계값_테스트	stock=5, 요청량 4(경계값-1)일 때 true 반환	true 반환
17	Beverage	hasEnoughStock	hasEnoughStock_재고와같으면_TRUE반환	stock=5, 요청량 5(경계값)일 때 true 반환	true 반환

2155. Write Unit Test Code

Number	Class	MethodName	testMethodName	Description	Result
18	Beverage	hasEnoughStock	hasEnoughStock_재고보다 크면_FALSE반환	stock=5, 요청량 8일 때 false 반환	false 반환
19	Beverage	hasEnoughStock	hasEnoughStock_재고보다 _1만큼_커도_FALSE반환_ 경계값_테스트	stock=5, 요청량 6일 때 false 반환	false 반환
20	Beverage	hasEnoughStock	hasEnoughStock_음수요청 이면_TRUE반환	stock=5, 요청량 -1일 때 (현재 구현상) true 반환	true 반환
21	Beverage	reduceQuantity	reduceQuantity_0이면_TR UE반환	stock=7, 감소량 0일 때 true 반환	true 반환
22	Beverage	reduceQuantity	reduceQuantity_재고보다_ 작으면_TRUE반환	stock=7, 감소량 5일 때 true 반환	true 반환
23	Beverage	reduceQuantity	reduceQuantity_재고보다_ 1만큼_작아도_TRUE반환	stock=7, 감소량 6(경계값-1)일 때 true 반환	true 반환

2155. Write Unit Test Code

Number	Class	MethodName	testMethodName	Description	Result
24	Beverage	reduceQuantity	reduceQuantity_재고와_같으면_TRUE반환	stock=7, 감소량 7(경계값)일 때 true 반환	true 반환
25	Beverage	reduceQuantity	reduceQuantity_재고보다_크면_FALSE반환	stock=7, 감소량 10일 때 false 반환	false 반환
26	Beverage	reduceQuantity	reduceQuantity_재고보다_1만큼_커도_FALSE반환	stock=7, 감소량 8(경계값+1)일 때 false 반환	false 반환

2155. Write Unit Test Code

Number	Class	MethodName	testMethodName	Description	Result
27	BeverageManager	getBeverage	GetBeverage_성공_Beverage반환	유효한 ID(1)로 호출 시 Beverage 객체 반환	1 반환
28	BeverageManager	getBeverage	GetBeverage_실패_문자입력_예외	문자형 리터럴 'a'(=97) 입력 시 예외 발생	NotFoundException 던짐
29	BeverageManager	getBeverage	GetBeverage_실패_음수입력_예외	음수 ID(-1) 입력 시 예외 발생	NotFoundException 던짐
30	BeverageManager	getBeverage	GetBeverage_실패_0입력_예외	ID 0 입력 시 예외 발생	NotFoundException 던짐
31	BeverageManager	getBeverage	GetBeverage_실패_초과입력_예외	범위를 벗어난 ID(999) 입력 시 예외 발생	NotFoundException 던짐
32	BeverageManager	hasEnoughStock	HasEnoughStock_0인경우_TRUE반환	ID 1, 요청량 0일 때 true 반환	true 반환

2155. Write Unit Test Code

Number	Class	MethodName	testMethodName	Description	Result
33	BeverageManager	hasEnoughStock	HasEnoughStock_재고보다 작을경우_TRUE반환	ID 1, 요청량 3일 때 true 반환	true 반환
34	BeverageManager	hasEnoughStock	HasEnoughStock_재고와같을경우_TRUE반환	ID 2, 요청량 10일 때 true 반환	true 반환
35	BeverageManager	hasEnoughStock	HasEnoughStock_재고보다 클경우_FALSE반환	ID 2, 요청량 20일 때 false 반환	false 반환
36	BeverageManager	hasEnoughStock	HasEnoughStock_음수요청이면_TRUE반환	ID 1, 요청량 -5일 때 (현재 구현상) true 반환	true 반환
37	BeverageManager	hasEnoughStock	HasEnoughStock_존재하지 않는ID이면_예외	존재하지 않는 ID(999,0,-1) 요청 시 예외 발생	NotFoundException 던짐
38	BeverageManager	reduceQuantity	ReduceQuantity_0인경우_TRUE반환	ID 1, 감소량 0일 때 true 반환	true 반환

2155. Write Unit Test Code

Number	Class	MethodName	testMethodName	Description	Result
39	BeverageManager	reduceQuantity	ReduceQuantity_재고보다 작을경우_TRUE반환	ID 1, 감소량 3일 때 true 반환	true 반환
40	BeverageManager	reduceQuantity	ReduceQuantity_재고와같은경우_TRUE반환	ID 2, 감소량 10일 때 true 반환	true 반환
41	BeverageManager	reduceQuantity	ReduceQuantity_음수요청이면_FALSE반환	ID 1, 감소량 -2일 때 false 반환	false 반환
42	BeverageManager	reduceQuantity	ReduceQuantity_존재하지 않는ID이면_예외	존재하지 않는 ID(999,0,-1) 요청 시 예외 발생	NotFoundException 던짐

2155. Write Unit Test Code

Number	Class	MethodName	testMethodName	Description	Result
43	Bank	requestCard	RequestCard_있는번호_조회 성공	DB에 존재하는 카드 번호로 호출 시 유효한 CreditCard 객체 반환	유효한 카드인 card->getCardNumber() == "1234-5678-9012-3456" 반환
44	Bank	requestCard	RequestCard_없는번호_조회 실패	DB에 없는 카드 번호로 호출 시 예외 발생	NotFoundException 던짐
45	Bank	saveCreditCard	SaveCreditCard_기존카드_ 잔액업데이트	기존 DB 카드의 잔액 변경 후 저장하고, 조회 시 업데이트된 잔액 반환	requestCard()->getBalance() == 20000 반환
46	Bank	saveCreditCard	SaveCreditCard_신규카드_ 추가후조회	DB에 없는 신규 카드 저장 후 requestCard 가능 확인	card->getCardNumber() == "1111-2222-3333-4444", getBalance() == 3000 반환
47	Bank	requestCard	RequestCard_파일열기실패_ 예외발생	DB 파일이 없을 때 requestCard 호출 시 예외 발생	FileNotFoundException 던짐

2155. Write Unit Test Code

Number	Class	MethodName	testMethodName	Description	Result
48	CreditCard	validateBalance	ValidateBalance_0_성공	price=0일 때 validateBalance(0)가 true 반환되는지 확인	true 반환
49	CreditCard	validateBalance	ValidateBalance_유효한 값_성공	price=3000일 때 validateBalance(3000)가 true 반환되는지 확인	true 반환
50	CreditCard	validateBalance	ValidateBalance_경계값 _max_성공	price=5000(잔액과 동일)일 때 validateBalance(5000)가 true 반환되는지 확인	true 반환
51	CreditCard	validateBalance	ValidateBalance_경계값 _5001_예외	price=5001(잔액 초과)일 때 validateBalance(5001)가 false 반환되는지 확인	false 반환
52	CreditCard	validateBalance	ValidateBalance_초과값 _예외	price=99999(잔액 훨씬 초과)일 때 validateBalance(99999)가 false 반환되는지 확인	false 반환
53	CreditCard	reduceBalance	ReduceBalance_잔액감 소	reduceBalance(1000) 호출 후 잔액이 4000으로 감소했는지 validateBalance(4000)로 확인	validateBalance(4000) → true 반환

2155. Write Unit Test Code

Number	Class	MethodName	testMethodName	Description	Result
54	CreditCard	isValid	IsValid_있는번호_성공	유효한 카드번호일 때 isValid()가 true 반환되는지 확인	true 반환
55	CreditCard	isValid	IsValid_없는번호_실패	빈 문자열 카드번호일 때 isValid()가 false 반환되는지 확인	false 반환

2155. Write Unit Test Code

Number	Class	MethodName	testMethodName	Description	Result
56	Location	getX	getX_정상작동	생성자에 전달한 x(7)가 getX()로 반환되는지 확인	7 반환
57	Location	getY	getY_정상작동	생성자에 전달한 y(-3)가 getY()로 반환되는지 확인	-3 반환
58	Location	distanceTo	distanceTo_같은좌표_0 반환	동일 좌표(5,5) 간 거리가 0.0인지 확인	0.0 반환
59	Location	distanceTo	distanceTo_가로이동_거리계산	(10,2) → (4,2) 수평 이동 시 거리 6.0인지 확인	6.0 반환
60	Location	distanceTo	distanceTo_세로이동_거리계산	(-1,8) → (-1,3) 수직 이동 시 거리 5.0인지 확인	5.0 반환
61	Location	distanceTo	distanceTo_대각선이동_거리계산	(0,0) → (3,4) 대각선 이동 시 피타고라스 5.0인지 확인	5.0 반환

2155. Write Unit Test Code

Number	Class	MethodName	testMethodName	Description	Result
62	Location	distanceTo	distanceTo_음수좌표_거리계산	(-2,-3) → (1,1) 음수 좌표 간 거리 $\sqrt{(9+16)}=5.0$ 인지 확인	5.0 반환
63	Location	distanceTo	distanceTo_큰차이_거리계산	(1000,-1000) → (-500,200) 간 거리($\sqrt{(1500^2+(-1200)^2)})$ 인지 확인	expected 변수 값 반환 (≈ 1920.937)
64	LocationManager	calculateNearest	calculateNearest_단일요소_반환	responseList에 단일 요소 전달 시 해당 요소의 DVMLInfoDTO 반환 확인	X=3, Y=4, prePaymentDvmId=100 반환
65	LocationManager	calculateNearest	calculateNearest_여러요소_가장가까운반환	여러 요소 중 최단 거리인 요소 반환 확인	X=1, Y=1, prePaymentDvmId=2 반환
66	LocationManager	calculateNearest	calculateNearest_음수좌표_반환	음수 좌표를 가진 요소 반환 확인	X=-3, Y=-4, prePaymentDvmId=42 반환



감사합니다.