

#5 - the 2nd Cycle Presentation

컴퓨터공학부 김진솔
컴퓨터공학부 백현준
컴퓨터공학부 이현수

1. OOD에서 수정된 부분 정리

Activity 2033. 기존 Define Domain Model

Drink	
name	String
price	Float
stock	Integer

AuthCode	
code	String
expiry	Date

VendingMachine	
id	String
location	String

CardReader	
readerID	String

CardSystem	
SystemName	String

otherVendingMachine	
id	String
location	String

BroadcastMessage	
messageId	String
targetDrink	String

InventoryStatus	
status	String
availableQty	Integer

Order	
id	String
SelectedDrink	Drink
time	Date

Payment	
paymentId	String
status	String
method	String

1. OOD에서 수정된 부분 정리

Activity 2033. Define Domain Model

Drink	
drinkCode	String
name	String
price	Int

Inventory	
drink	Drink
qty	Int

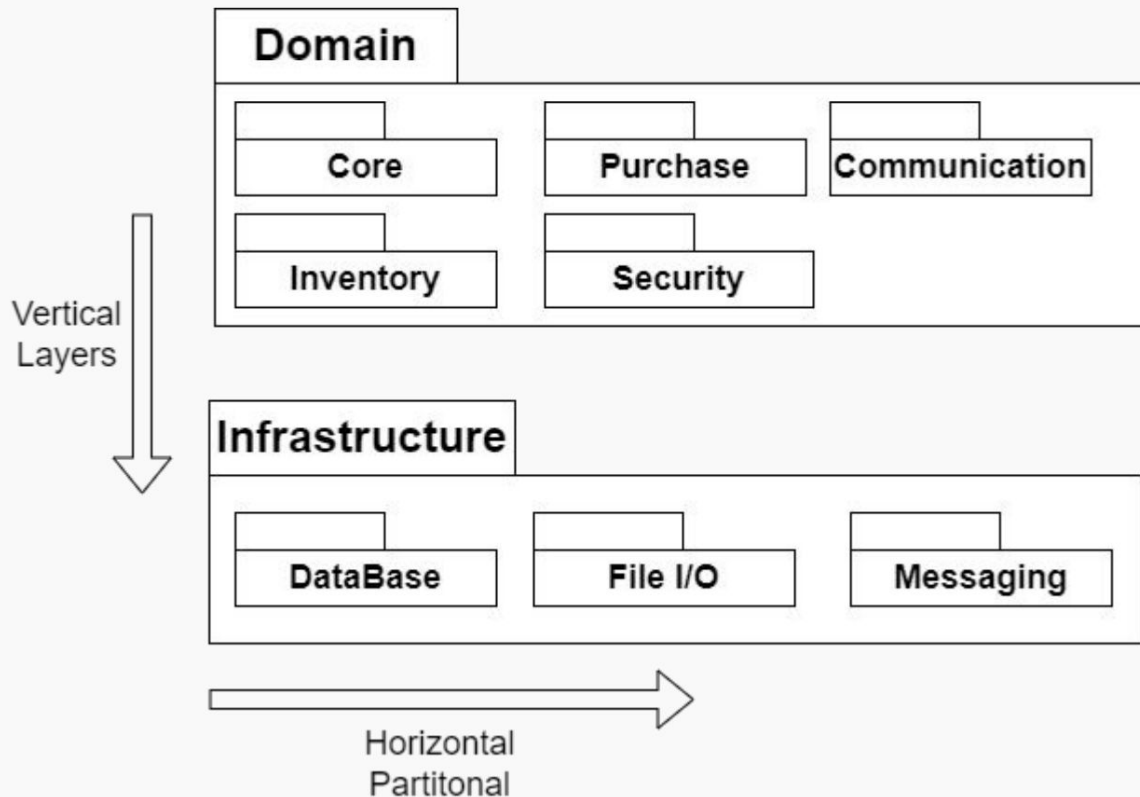
Order	
vmid	String
drink	Drink
qty	Int
certcode	String
paystatus	String

PrePaymentCode	
code	String
status	CodeStatus
heldorder	Order

VendingMachine	
id	String
location	<int,int>
port	String

1. OOD에서 수정된 부분 정리

Activity 2043. 기존 Refine System Archi tecture



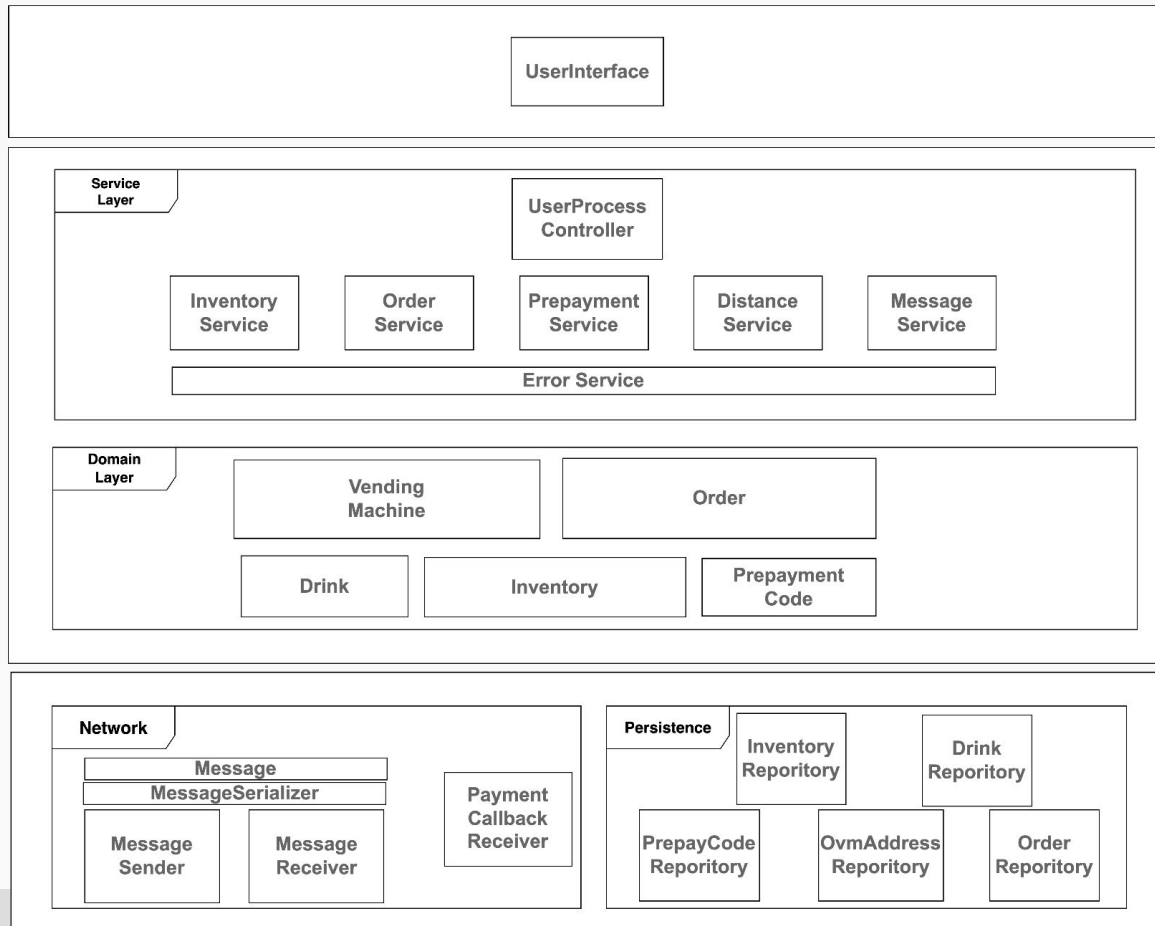
1. OOD에서 수정된 부분 정리

Activity 2043. Refine System Archi tecture

Presentation
Tier

Application Logic
Tier

Infrastructure
Tier



2. System Test Case 시나리오 및 결과

1. 사용자가 음료 선택 → 재고 확인 (현재 자판기에 있음) → 카드 결제 → 음료 배출

& [현재 자판기 음료 재고 감소되는지 확인]

```
--- 음료 목록 (총 20종) ---
+-----+
| 음료코드 | 음료명       | 가격  |
+-----+
| 01       | 콜라         | 1000원 |
| 02       | 사이다       | 1000원 |
| 03       | 녹차         | 1000원 |
| 04       | 홍차         | 1000원 |
| 05       | 밀크티       | 1000원 |
| 06       | 탄산수       | 1000원 |
| 07       | 보리차       | 1000원 |
| 08       | 캔커피       | 1000원 |
| 09       | 물           | 1000원 |
| 10       | 에너지드링크 | 1000원 |
| 11       | 유자차       | 1000원 |
| 12       | 식혜         | 1000원 |
| 13       | 아이스티     | 1000원 |
| 14       | 딸기주스     | 1000원 |
| 15       | 오렌지주스   | 1000원 |
| 16       | 포도주스     | 1000원 |
| 17       | 이온음료     | 1000원 |
| 18       | 아메리카노   | 1000원 |
| 19       | 핫초코       | 1000원 |
| 20       | 카패라떼     | 1000원 |
+-----+

음선을 선택하세요:
1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료
선택 (숫자 입력): 1
구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'c' 입력): 01
콜라 선택됨. 가격: 1000원. (재고: 3개)
콜라 구매. 가격: 1000원.

결제할 금액: 1000원 입니다.
카드를 삽입해주세요. (시뮬레이션: 실제 카드 처리 없음)
결제를 계속 진행하시겠습니까? (Y/N): Y
결제 승인 중입니다. 잠시만 기다려주세요...
[결제 성공] 결제가 성공적으로 완료되었습니다!
[T1] DEBUG: state_processingPayment - Payment success. isPrepay: 0
[T1] DEBUG: selectedTargetVmForPrepayment_ is NOT set.
```

```
[콜라] 음료가 배출되고 있습니다. 잠시만 기다려주세요...
[콜라] 음료가 나왔습니다. 이용해주시어 감사합니다!
거래가 완료되었습니다. 감사합니다.

===== Vending Machine Menu =====
--- 음료 목록 (총 20종) ---
+-----+
| 음료코드 | 음료명       | 가격  |
+-----+
| 01       | 콜라         | 1000원 |
| 02       | 사이다       | 1000원 |
| 03       | 녹차         | 1000원 |
| 04       | 홍차         | 1000원 |
| 05       | 밀크티       | 1000원 |
| 06       | 탄산수       | 1000원 |
| 07       | 보리차       | 1000원 |
| 08       | 캔커피       | 1000원 |
| 09       | 물           | 1000원 |
| 10       | 에너지드링크 | 1000원 |
| 11       | 유자차       | 1000원 |
| 12       | 식혜         | 1000원 |
| 13       | 아이스티     | 1000원 |
| 14       | 딸기주스     | 1000원 |
| 15       | 오렌지주스   | 1000원 |
| 16       | 포도주스     | 1000원 |
| 17       | 이온음료     | 1000원 |
| 18       | 아메리카노   | 1000원 |
| 19       | 핫초코       | 1000원 |
| 20       | 카패라떼     | 1000원 |
+-----+

음선을 선택하세요:
1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료
선택 (숫자 입력): 1
구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'c' 입력): 01
콜라 선택됨. 가격: 1000원. (재고: 2개)
콜라 구매. 가격: 1000원.

결제할 금액: 1000원 입니다.
카드를 삽입해주세요. (시뮬레이션: 실제 카드 처리 없음)
결제를 계속 진행하시겠습니까? (Y/N): Y
```

재고 3개에서 2개로
감소되었음을 확인

2. System Test Case 시나리오 및 결과

2. 사용자가 음료 선택 → 재고 확인 (현재 자판기에 없음) → 다른 자판기 재고 확인 → 현재 자판기에서 선결제 요청 → 카드 결제 → 인증 코드 발급 → 다른 자판기에서 인증코드 입력 → 음료 배출

```
정보: 자판기 시스템을 시작합니다. 현재 자판기 ID: T2
T2의 io_context 스레드 시작.

===== Vending Machine Menu =====

--- 음료 목록 (총 20종) ---
+-----+-----+-----+
| 음료코드 | 음료명       | 가격   |
+-----+-----+-----+
| 01       | 콜라         | 1000원 |
| 02       | 사이다       | 1000원 |
| 03       | 녹차         | 1000원 |
| 04       | 홍차         | 1000원 |
| 05       | 밀크티       | 1000원 |
| 06       | 탄산수       | 1000원 |
| 07       | 보리차       | 1000원 |
| 08       | 캔커피       | 1000원 |
| 09       | 물           | 1000원 |
| 10       | 에너지링크   | 1000원 |
| 11       | 유자차       | 1000원 |
| 12       | 식혜         | 1000원 |
| 13       | 아이스티     | 1000원 |
| 14       | 딸기주스     | 1000원 |
| 15       | 오렌지주스   | 1000원 |
| 16       | 포도주스     | 1000원 |
| 17       | 이온음료     | 1000원 |
| 18       | 아메리카노   | 1000원 |
| 19       | 핫초코       | 1000원 |
| 20       | 카페라떼     | 1000원 |
+-----+-----+-----+

음선을 선택하세요:
1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료
선택 (숫자 입력): 1
구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'C' 입력): 01
알림: 선택하신 음료 [콜라]는 현재 자판기에 재고가 없습니다.
```

현재 자판기 T2에 콜라를 주문하려고 했으나 재고가 없어 가장 가까운 T1으로 이동

옵션을 선택하세요:

- 음료 선택 (구매/다른 자판기 조회)
- 인증 코드로 음료 받기
- 시스템 종료

선택 (숫자 입력): 1

구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'C' 입력): 01

알림: 선택하신 음료 [콜라]는 현재 자판기에 재고가 없습니다.

콜라 음료의 재고를 주변 자판기에 문의합니다...

```
[Sender DEBUG] Sending to 127.0.0.1:12345: {"msg_type":0,"src_id":"T2","dst_id":"0","msg_content":{"item_code":"01","item_num":"1"}}

Accepting connections on port 12346
[Receiver DEBUG] Received line: {"msg_type":1,"src_id":"T1","dst_id":"T2","msg_content":{"item_code":"01","coord_x":"10","item_num":"5","coord_y":"10"}}
[Receiver DEBUG] Parsed msg type: 1 from T1
[T2] onRespStockReceived: 메시지 수신됨 (from: T1, type: 1)
- item_code: 01
- 유효한 재고 응답. availableOtherVmsForDrink_ 크기: 1/1
- 모든 다른 VM으로부터 응답 수신 완료 또는 지정된 수만큼 받음. 타이머 취소 및 상태 변경 시도.
```

음료 [콜라]를 구매 가능한 가장 가까운 자판기 정보입니다:

- > 자판기 ID: T1
- > 위치: (X: 10, Y: 10)

[콜라] 음료에 대해 다른 자판기에서 선결제 하시겠습니까? (Y/N): Y

콜라 선결제 (대상 자판기: T1). 가격: 1000원.

2. System Test Case 시나리오 및 결과

2. 사용자가 음료 선택 → 재고 확인 (현재 자판기에 없음) → 다른 자판기 재고 확인 → 현재 자판기에서 선결제 요청 → 카드 결제 → 인증 코드 발급 → 다른 자판기에서 인증코드 입력 →

음료 [콜라]를 구매 가능한 가장 가까운 자판기 정보입니다:

> 자판기 ID: T1

> 위치: (X: 10, Y: 10)

[콜라] 음료에 대해 다른 자판기에서 선결제 하시겠습니까? (Y/N): Y

콜라 선결제 (대상 자판기: T1). 가격: 1000원.

결제할 금액: 1000원 입니다.

카드를 삽입해주세요. (시뮬레이션: 실제 카드 처리 없음)

결제를 계속 진행하시겠습니까? (Y/N): Y

결제 승인 중입니다. 잠시만 기다려주세요...

[결제 성공] 결제가 성공적으로 완료되었습니다!

[T2] DEBUG: state_processingPayment - Payment success. isPrepay: 1

[T2] DEBUG: selectedTargetVmForPrepayment ID: T1

[T2] DEBUG: Changing state to ISSUING_AUTH_CODE_AND_REQUESTING_RESERVATION

T1에 콜라 재고 확인 요청 (인증코드: QZrzv)

[Sender DEBUG] Sending to 127.0.0.1:12345: {"msg_type":2,"src_id":"T2","dst_id":"T1","msg_content":{"item_code":"01","cert_code":"QZrzv","item_num":"1"}}

Accepting connections on port 12346

[Receiver DEBUG] Received line: {"msg_type":3,"src_id":"T1","dst_id":"T2","msg_content":{"item_code":"01","item_num":"1","availability":"T"}}

[Receiver DEBUG] Parsed msg type: 3 from T1

==== 선결제 완료 ====

음료명: 콜라

수령 자판기 ID: T1

수령 자판기 위치: (X: 10, Y: 10)

★인증 코드: QZrzv ★

=====

위 자판기에서 발급된 인증 코드를 입력하여 음료를 수령하세요.

거래가 완료되었습니다. 감사합니다.

Accepting connections on port 12345

[Receiver DEBUG] Received line: {"msg_type":2,"src_id":"T2","dst_id":"T1","msg_content":{"item_code":"01","cert_code":"QZrzv","item_num":"1"}}

[Receiver DEBUG] Parsed msg type: 2 from T2

[T1] DEBUG: onReqPrepayReceived 진입 (from: T2)

[Sender DEBUG] Sending to 127.0.0.1:12346: {"msg_type":3,"src_id":"T1","dst_id":"T2","msg_content":{"item_code":"01","item_num":"1","availability":"T"}}

2

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴: c 또는 C): QZrzv

인증 코드 (QZrzv) 확인 완료: 콜라

[콜라] 음료가 배출되고 있습니다. 잠시만 기다려주세요...

[콜라] 음료가 나왔습니다. 이용해주셔서 감사합니다!

거래가 완료되었습니다. 감사합니다.

**T2 자판기에서 선결제 성공 뒤
인증코드 QZrzv 발급
인증코드는 T1 자판기에서 입력**

2. System Test Case 시나리오 및 결과

3. 사용자가 음료 선택 → 재고 확인 (현재 자판기에 있음) → 카드 결제 → 카드 결제 N 하는 경우 → Main 화면으로 돌아감

```
===== Vending Machine Menu =====
--- 음료 목록 (총 20종) ---
+-----+-----+-----+
| 음료코드 | 음료명       | 가격   |
+-----+-----+-----+
| 01       | 콜라        | 1000원 |
| 02       | 사이다      | 1000원 |
| 03       | 녹차        | 1000원 |
| 04       | 홍차        | 1000원 |
| 05       | 밀크티      | 1000원 |
| 06       | 탄산수      | 1000원 |
| 07       | 보리차      | 1000원 |
| 08       | 캔커피      | 1000원 |
| 09       | 물          | 1000원 |
| 10       | 에너지드링크 | 1000원 |
| 11       | 유자차      | 1000원 |
| 12       | 식혜        | 1000원 |
| 13       | 아이스티    | 1000원 |
| 14       | 딸기주스    | 1000원 |
| 15       | 오렌지주스  | 1000원 |
| 16       | 포도주스    | 1000원 |
| 17       | 이온음료    | 1000원 |
| 18       | 아메리카노 | 1000원 |
| 19       | 핫초코      | 1000원 |
| 20       | 카페라떼    | 1000원 |
+-----+-----+-----+

음선을 선택하세요:
1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료
선택 (숫자 입력): 1
구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'C' 입력): 01
콜라 선택됨. 가격: 1000원. (재고: 5개)
콜라 구매. 가격: 1000원.

결제할 금액: 1000원 입니다.
카드를 삽입해주세요. (시뮬레이션: 실제 카드 처리 없음)
결제를 계속 진행하시겠습니까? (Y/N): N
구매가 취소되었습니다.
```

```
===== Vending Machine Menu =====
--- 음료 목록 (총 20종) ---
+-----+-----+-----+
| 음료코드 | 음료명       | 가격   |
+-----+-----+-----+
| 01       | 콜라        | 1000원 |
| 02       | 사이다      | 1000원 |
| 03       | 녹차        | 1000원 |
| 04       | 홍차        | 1000원 |
| 05       | 밀크티      | 1000원 |
| 06       | 탄산수      | 1000원 |
| 07       | 보리차      | 1000원 |
| 08       | 캔커피      | 1000원 |
| 09       | 물          | 1000원 |
| 10       | 에너지드링크 | 1000원 |
| 11       | 유자차      | 1000원 |
| 12       | 식혜        | 1000원 |
| 13       | 아이스티    | 1000원 |
| 14       | 딸기주스    | 1000원 |
| 15       | 오렌지주스  | 1000원 |
| 16       | 포도주스    | 1000원 |
| 17       | 이온음료    | 1000원 |
| 18       | 아메리카노 | 1000원 |
| 19       | 핫초코      | 1000원 |
| 20       | 카페라떼    | 1000원 |
+-----+-----+-----+

음선을 선택하세요:
1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료
선택 (숫자 입력):
```

2. System Test Case 시나리오 및 결과

4. 사용자가 음료 선택 → 재고 확인 (현재 자판기에 있음) → 카드 결제 → 카드 결제 실패하는 경우 → Main 화면으로 돌아감

```
===== Vending Machine Menu =====
--- 음료 목록 (총 20종) ---
+-----+-----+-----+
| 음료코드 | 음료명       | 가격   |
+-----+-----+-----+
| 01       | 콜라        | 1000원 |
| 02       | 사이다      | 1000원 |
| 03       | 녹차        | 1000원 |
| 04       | 홍차        | 1000원 |
| 05       | 말크티      | 1000원 |
| 06       | 탄산수      | 1000원 |
| 07       | 보리차      | 1000원 |
| 08       | 캔커피      | 1000원 |
| 09       | 물          | 1000원 |
| 10       | 에너지드링크 | 1000원 |
| 11       | 유자차      | 1000원 |
| 12       | 식혜        | 1000원 |
| 13       | 아이스티    | 1000원 |
| 14       | 딸기주스    | 1000원 |
| 15       | 오렌지주스  | 1000원 |
| 16       | 포도주스    | 1000원 |
| 17       | 이온음료    | 1000원 |
| 18       | 아메리카노  | 1000원 |
| 19       | 핫초코      | 1000원 |
| 20       | 카페라떼    | 1000원 |
+-----+-----+-----+

음료를 선택하세요:
1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료
선택 (숫자 입력): 1
구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'C' 입력): 01
콜라 선택됨. 가격: 1000원. (재고: 5개)
콜라 구매. 가격: 1000원.

결제할 금액: 1000원 입니다.
카드를 삽입해주세요. (시뮬레이션: 실제 카드 처리 없음)
결제를 계속 진행하시겠습니까? (Y/N): Y
결제 승인 중입니다. 잠시만 기다려주세요...
[결제 실패] 결제에 실패했습니다.
```

```
===== Vending Machine Menu =====
--- 음료 목록 (총 20종) ---
+-----+-----+-----+
| 음료코드 | 음료명       | 가격   |
+-----+-----+-----+
| 01       | 콜라        | 1000원 |
| 02       | 사이다      | 1000원 |
| 03       | 녹차        | 1000원 |
| 04       | 홍차        | 1000원 |
| 05       | 말크티      | 1000원 |
| 06       | 탄산수      | 1000원 |
| 07       | 보리차      | 1000원 |
| 08       | 캔커피      | 1000원 |
| 09       | 물          | 1000원 |
| 10       | 에너지드링크 | 1000원 |
| 11       | 유자차      | 1000원 |
| 12       | 식혜        | 1000원 |
| 13       | 아이스티    | 1000원 |
| 14       | 딸기주스    | 1000원 |
| 15       | 오렌지주스  | 1000원 |
| 16       | 포도주스    | 1000원 |
| 17       | 이온음료    | 1000원 |
| 18       | 아메리카노  | 1000원 |
| 19       | 핫초코      | 1000원 |
| 20       | 카페라떼    | 1000원 |
+-----+-----+-----+

음료를 선택하세요:
1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료
선택 (숫자 입력): 1
구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'C' 입력): 01
콜라 선택됨. 가격: 1000원. (재고: 5개)
콜라 구매. 가격: 1000원.

결제할 금액: 1000원 입니다.
카드를 삽입해주세요. (시뮬레이션: 실제 카드 처리 없음)
결제를 계속 진행하시겠습니까? (Y/N): Y
결제 승인 중입니다. 잠시만 기다려주세요...
[결제 실패] 결제에 실패했습니다.
```

2. System Test Case 시나리오 및 결과

5. 사용자가 음료 선택 → 재고 확인 (현재 자판기에 없음) → 다른 자판기 재고 확인 → 현재 자판기에서 선결제 요청 → 카드 결제 → 인증 코드 발급 → 다른 자판기에서 인증코드 입력 → 인증코드 잘못 입력한 경우 → Main 화면으로 돌아감

현재 자판기 T1에 콜라를 주문하려고 했으나 재고가 없어 T2, T3 중 가장 가까운 T3으로 이동

음료코드	음료명	가격
01	콜라	1000원
02	사이다	1000원
03	녹차	1000원
04	홍차	1000원
05	밀크티	1000원
06	탄산수	1000원
07	보리차	1000원
08	커피	1000원
09	물	1000원
10	에너지드링크	1000원
11	유자차	1000원
12	식혜	1000원
13	아이스티	1000원
14	딸기주스	1000원
15	오렌지주스	1000원
16	포도주스	1000원
17	이온음료	1000원
18	아메리카노	1000원
19	핫초코	1000원
20	카파라페	1000원

```
음료를 선택하세요:
1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료
선택 (숫자 입력): 1
구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'c' 입력): 01
알림: 선택하신 음료 [콜라]는 현재 자판기에 재고가 없습니다.
콜라 음료의 재고를 주변 자판기에 문의합니다...
[Sender DEBUG] Sending to 127.0.0.1:12346: {"msg_type":0,"src_id":"T1","dst_id":"0","msg_content":{"item_code":"01","item_num":"1"}}
[Sender DEBUG] Sending to 127.0.0.1:12347: {"msg_type":0,"src_id":"T1","dst_id":"0","msg_content":{"item_code":"01","item_num":"1"}}
Accepting connections on port 12345
```

```
선택 (숫자 입력): 1
구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'c' 입력): 01
알림: 선택하신 음료 [콜라]는 현재 자판기에 재고가 없습니다.
콜라 음료의 재고를 주변 자판기에 문의합니다...
[Sender DEBUG] Sending to 127.0.0.1:12346: {"msg_type":0,"src_id":"T1","dst_id":"0","msg_content":{"item_code":"01","item_num":"1"}}
[Sender DEBUG] Sending to 127.0.0.1:12347: {"msg_type":0,"src_id":"T1","dst_id":"0","msg_content":{"item_code":"01","item_num":"1"}}
Accepting connections on port 12345
[Receiver DEBUG] Received line: {"msg_type":1,"src_id":"T2","dst_id":"T1","msg_content":{"item_code":"01","coord_x":"20","item_num":"3","coord_y":"20"}}
[Receiver DEBUG] Parsed msg type: 1 from T2
Accepting connections on port 12345
[T1] onRespStockReceived: 메시지 수신됨 (from: T2, type: 1)
- item_code: 01
[Receiver DEBUG] Received line: {"msg_type":1,"src_id":"T3","dst_id":"T1","msg_content":{"item_code":"01","coord_x":"10","item_num":"3","coord_y":"30"}}
[Receiver DEBUG] Parsed msg type: 1 from T3
[T1] onRespStockReceived: 메시지 수신됨 (from: T3, type: 1)
- item_code: 01
- 유효한 재고 응답. availableOtherVmsForDrink 크기: 1/2
- 모든 다른 VM으로부터 응답 수신 완료 또는 지정된 수만큼 받음. 타이머 취소 및 상태 변경 시도.
- 유효하지 않은 재고 응답 또는 조건 불일치 (요청 음료: 01, 수신 음료: 01, 재고: 3)
음료 [콜라]를 구매 가능한 가장 가까운 자판기 정보입니다:
> 자판기 ID: T3
> 위 치: (X: 10, Y: 30)
[콜라] 음료에 대해 다른 자판기에서 선결제 하시겠습니까? (Y/N): Y
콜라 선결제 (대상 자판기: T3). 가격: 1000원.
결제할 금액: 1000원 입니다.
카드를 삽입해주세요. (시뮬레이션: 실제 카드 처리 없음)
결제를 계속 진행하시겠습니까? (Y/N): Y
결제 승인 중입니다. 잠시만 기다려주세요...
[결제 성공] 결제가 성공적으로 완료되었습니다!
[T1] DEBUG: state_processingPayment - Payment success. isPrepay: 1
[T1] DEBUG: selectedTargetVmForPrepayment ID: T3
```

2. System Test Case 시나리오 및 결과

5. 사용자가 음료 선택 → 재고 확인 (현재 자판기에 없음) → 다른 자판기 재고 확인 → 현재 자판기에서 선결제 요청 → 카드 결제 → 인증 코드 발급 → 다른 자판기에서 인증코드 입력 → 인증코드 잘못 입력한 경우 → Main 화면으로 돌아감

음료 「콜라」를 구매 가능한 가장 가까운 자판기 정보입니다:

> 자판기 ID: T3

> 위치: (X: 10, Y: 30)

[콜라] 음료에 대해 다른 자판기에서 선결제 하시겠습니까? (Y/N): Y

콜라 선결제 (대상 자판기: T3), 가격: 1000원.

결제할 금액: 1000원 입니다.

카드를 삽입해주세요. (시뮬레이션: 실제 카드 처리 없음)

결제를 계속 진행하시겠습니까? (Y/N): Y

결제 승인 중입니다. 잠시만 기다려주세요...

[결제 성공] 결제가 성공적으로 완료되었습니다!

```
[T1] DEBUG: state processingPayment - Payment success. isPrepay: 1
```

```
[T1] DEBUG: selectedTargetVmForPrepayment ID: T3
```

```
[T1] DEBUG: Changing state to ISSUING_AUTH_CODE_AND_REQUESTING_RESERVATION
```

T3에 콜라 재고 확보 요청 (인증코드: BN6bg)

```
[Sender DEBUG] Sending to 127.0.0.1:12347: {"msg_type":2,"src_id":"T1","dst_id":"T3","msg_content":{"item code":"01","cert code":"BN6bg","item num":"1"}}
```

```
Accepting connections on port 12345
```

```
[Receiver DEBUG] Received line: {"msg_type":3,"src_id":"T3","dst_id":"T1","msg_content":{"i
tem code":"01","item num":"1","availability":"T"}}

```

```
[Receiver DEBUG] Parsed msg type: 3 from T3
```

==== 선결제 완료 ====

ನಾ
ಫಿ
..
ಹ
ನ
೦

수령 자판기 ID: T3

수령 자판기 위치: (X: 10, Y: 30)

★인증 코드: BN6bg ★

=====

위 자판기에서 발급된 인증 코드를 입력하여 음료를 수령하세요.

거래가 완료되었습니다. 감사합니다.

인증코드 : BN6bg를 BN6bG로 입력한 경우

옵션을 선택하세요:

1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료

```
선택 (숫자 입력): Accepting connections on port 12347
```

```
[Receiver DEBUG] Received line: {"msg_type":0,"src_id":"T1","dst_id":"0","msg_content":{"item code":"01","item num":"1"}}
```

```
[Receiver DEBUG] Parsed msg type: 0 from T1
```

```
[Sender DEBUG] Sending to 127.0.0.1:12345: {"msg_type":1,"src_id":"T3","dst_id":"T1","msg_content":{"item code":"01","coord x":"10","item num":"3","coord y":"30"}}
```

```
Accepting connections on port 12347
```

```
[Receiver DEBUG] Received line: {"msg_type":2,"src_id":"T1","dst_id":"T3","msg_content":{"item code":"01","cert code":"BN6be","item num":"1"}}
```

```
[Receiver DEBUG] Parsed msg type: 2 from T1
```

```
[T3] DEBUG: onRegPrepayReceived 진입 (from: T1)
```

```
[Sender DEBUG] Sending to 127.0.0.1:12345: {"msg_type":3,"src_id":"T3","dst_id":"T1","msg_content":{"item_code":"01","item_num":"1","availability":"I"}}
```

2

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴: c 또는 C): BN6bG

[시스템 오류] ⚠️⚠️⚠️ ⚠️帯 あ⚠️⚠️⚠️⚠️⚠️. (입력 코드 BN6bG는 유효하지 않거나 이미 사용되었습니다.)

메인 메뉴로 돌아갑니다.

3. CP Test Resolution

Test Num	P/F	Expected Output	Test Output
CP01	P	유효하지 않음	유효하지 않음
CP02	F	음료 종류 관련 에러 메시지	결제 관련 에러메시지
CP03	F	에러 메시지 출력	미출력
CP04	-	에러 메시지 출력	(음수 재고 입력 불가)
CP05	F	에러 메시지 출력	재고 감소 확인 실패
CP06	-	에러 메시지 출력	(음수 재고 입력 불가)
CP07	F	에러 메시지 출력	선결제 거절 불가
CP08	F	에러 메시지 출력	다른 입력 불가
CP09	F	에러 메시지 출력	10초간 입력 대기 불가
CP10	F	결제 거절 메시지 출력	카드 입력 부분 미구현
CP11	F	결제 거절 메시지 출력	카드 입력 부분 미구현
CP12	-	결제 거절 메시지 출력	카드 금액 0 미만 설정 불가
CP13	-	(Invalid)	(출력 정의 불가)
CP14	F	에러 메시지 호출	선결제 코드 유효
CP15	F	에러 메시지 호출	선결제 코드 처리
CP16	F	에러 메시지 호출	선결제 코드 처리
CP17	F	음료 정상 출고	타 자판기 재고 요청 수행
CP18	F	에러 메시지 호출	(음료 재고 확인 불가)
CP19	F	음료 정상 출고	비정상 출력
CP20	F	음료 정상 출고	결제 거절 및 비정상 출력
CP21	F	음료 정상 출고	(음료 재고 확인 불가)

- PASS: 4
- INVALID: 8
- TOTAL: 16
- PASS/VALID: 4/9
- CPT Pass Percentage = 44.44%

3. CP Test Resolution

| 자판기가 1개 있을 때, 1번째 자판기에서 판매하지 않는 음료수

- 테스트 순서
 - b. (음료 선택 및 결제)
 - 음료수를 선택하세요: 박카스
- expected output: 에러메시지 출력 (재고 조회 브로드캐스트 E1)
- real output: 결제가 거절되었습니다.

```
===== Main Menu =====
1. 음료 목록 보기
2. 음료 선택 및 결제
3. 선결제 코드 입력
0. 종료
=====
선택 : 2
음료수를 선택하세요 : 박카스
결제가 거절되었습니다. -> UC6
```

test result: Fail

===== Vending Machine Menu =====

--- 음료 목록 (총 20종) ---

음료 코드	음료 명	가 격
01	콜라	1000원
02	사 이 다	1000원
03	녹차	1000원
04	홍차	1000원
05	밀크티	1000원
06	탄산수	1000원
07	보리차	1000원
08	캔커피	1000원
09	물	1000원
10	에너지드링크	1000원
11	유자차	1000원
12	식혜	1000원
13	아이스티	1000원
14	딸기주스	1000원
15	오렌지주스	1000원
16	포도주스	1000원
17	이온음료	1000원
18	아메리카노	1000원
19	핫초코	1000원
20	카페라떼	1000원

음선을 선택하세요 :

1. 음료 선택 (구매/다른 자판기 조회)

2. 인증 코드로 음료 받기

3. 시스템 종료

선택 (숫자 입력) : 1

구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'C' 입력) : 02

알림 : 선택하신 음료 [사이다]는 현재 자판기에 재고가 없습니다.

사이다 음료의 재고를 주변 자판기에 문의합니다...

[시스템 오류] 다른 자판기 (주변 자판기 재고 조회)로부터 응답을 받지 못했습니다.
메인 메뉴로 돌아갑니다.

3. CP Test Resolution

| 원격 자판기의 음료 개수가 0개인 경우

- 테스트 순서
 - 자판기 개수: 2개
 - 1번 자판기에서 아래 sequence를 반복 (박카스 음료 개수가 0개가 될때까지)
 - 음료: 박카스 (원격 자판기 음료)
 - 현재 자판기 음료 개수: 0개
 - 원격 자판기 음료 개수: 0개 초과
 - 선결제: 승인
 - 카드 번호: 유효
 - 카드 잔고: 음료 금액보다 많음
 - 선결제 코드: 정상적으로 생성
 - 자판기 안내: 가까운 자판기
 - 선결제 코드 검증: 정상
- expected output: 에러 메시지 호출 (재고조회 브로드 캐스트, 재고가 없기 때문에 아무도 메시지를 안보낼 것임)
- real output: 재고 감소 확인 실패
- test result: fail

옵션을 선택하세요 :

1. 음료 선택 (구매 /다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료

선택 (숫자 입력): 1

구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'C' 입력): 02

알림: 선택하신 음료 [사이다]는 현재 자판기에 재고가 없습니다.

사이다 음료의 재고를 주변 자판기에 문의합니다...

```
[T3] onRespStockReceived: 메시지 수신됨 (from: T2, type: 1)
```

```
- item_code: 02
```

```
- 현재 상태 AWAITING_STOCK_RESPONSES 아님. 메시지 무시.
```

```
[Sender] warning: failed to send to 127.0.0.1:12352: connect: Connection refused in function 'connect']
```

```
[시스템 오류] 다른 자판기 (주변 자판기 재고 조회)로부터 응답을 받지 못했습니다.  
메인 메뉴로 돌아갑니다.
```

3. CP Test Resolution

| 선결제를 거절할 경우

- 테스트 순서
 - 자판기 개수 2개
 - 1번 자판기에서 2. (음료 선택 및 결제)
 - 음료: 박카스 (원격 자판기 음료)
 - 현재 자판기 음료 개수: 0개
 - 원격 자판기 음료 개수: 0개 초과
 - 선결제: 거절
- expected output: 에러 메시지 출력
- real output: 선결제 메뉴 미구현으로 인한 선결제 거절 불가

음료 [사이다]를 구매 가능한 가장 가까운 자판기 정보입니다 :

> 자판기 ID: T2

> 위치 : (X: 20, Y: 20)

[사이다] 음료에 대해 다른 자판기에서 선결제 하시겠습니까? (Y/N): n
선결제가 취소되었습니다.

옵션을 선택하세요 :

1. 음료 선택 (구매 /다른 자판기 조회)

2. 인증 코드로 음료 받기

3. 시스템 종료

선택 (숫자 입력): Accepting connections on port 12346

[Receiver DEBUG] Received line: {"msg_type":0,"src_id":"T3","dst_id":"0","msg_content":{"item_num":"1","item_code":"02"}}

[Receiver DEBUG] Parsed msg type: 0 from T3

[Sender DEBUG] Sending to 127.0.0.1:12347: {"msg_type":1,"src_id":"T2","dst_id":"T3","msg_content":{"coor_y":"20","item_num":"12","coor_x":"20","item_code":"02"}}

3. CP Test Resolution

| 선결제 승인/거절 선택지에서 승인 거절 외의 다른 입력

- 테스트 순서

- 자판기 개수 2개
- 1번 자판기에서 2. (음료 선택 및 결제)
- 음료: 박카스 (원격 자판기 음료)
- 현재 자판기 음료 개수: 0개
- 원격 자판기 음료 개수: 0개 초과
- 선결제: ANDKEIN

- expected output: 에러 메시지 출력

- real output: 선결제 메뉴 미구현으로 인한 선결제 승인/거절과 무관한 입력 불가

- test result: fail

```
[Sender] warning: failed to send to 127.0.0.1:12348: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_serv
in function 'connect']
[Sender] warning: failed to send to 127.0.0.1:12349: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_serv
in function 'connect']
[Sender] warning: failed to send to 127.0.0.1:12350: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_serv
in function 'connect']
[Sender] warning: failed to send to 127.0.0.1:12351: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_serv
in function 'connect']
[Sender] warning: failed to send to 127.0.0.1:12352: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_serv
in function 'connect']
Accepting connections on port 12347
[Receiver DEBUG] Received line: {"msg_type":1,"src_id":"T2","dst_id":"T3","msg_content":{"coor_y":"20","item_num":"12","coor_x":"20","item_code":"02"}}
[Receiver DEBUG] Parsed msg type: 1 from T2
[T3] onRespStockReceived: 메시지 수신됨 (from: T2, type: 1)
- item_code: 02
- 유효한 재고 응답. availableOtherVmsForDrink_ 크기: 1/7
- 모든 다른 VM으로부터 응답 수신 완료 또는 지정된 수만큼 받음. 타이머 취소 및 상태 변경 시도.
- 유효하지 않은 재고 응답 또는 조건 불일치 (요청 음료: 02, 수신 음료: 02, 재고: 12)

음료 [사이다]를 구매 가능한 가장 가까운 자판기 정보입니다:
> 자판기 ID: T2
> 위 치: (X: 20, Y: 20)
[사이다] 음료에 대해 다른 자판기에서 선결제 하시겠습니까? (Y/N): y
사이다 선결제 (대상 자판기: T2). 가격: 1000원.

결제할 금액: 1000원 입니다.
카드를 삽입해주세요. (시뮬레이션: 실제 카드 처리 없음)
결제를 계속 진행하시겠습니까? (Y/N): ANDKEIN
잘못된 입력입니다. Y 또는 N을 입력해주세요.
```

3. CP Test Resolution

| 선결제 승인/거절 선택지에서 승인 거절 외의 다른 입력

- 테스트 순서

- 자판기 개수 2개
- 1번 자판기에서 2. (음료 선택 및 결제)
- 음료: 박카스 (원격 자판기 음료)
- 현재 자판기 음료 개수: 0개
- 원격 자판기 음료 개수: 0개 초과
- 선결제: ANDKEIN

- expected output: 예러 메시지 출력

- real output: 선결제 메뉴 미구현으로 인한 선결제 승인/거절과 무관한 입력 불가

- test result: fail

```
발행: 선택하신 품목 [사이다]은 현재 자판기에 재고가 없습니다.
사이다 음료의 재고를 주변 자판기에 문의합니다...
[Sender] warning: failed to send to 127.0.0.1:12345: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_serv:
in function 'connect']
[Sender DEBUG] Sending to 127.0.0.1:12346: {"msg_type":0,"src_id":"T3","dst_id":"","msg_content":{"item_num":1,"item_code":"02"}}

[Sender] warning: failed to send to 127.0.0.1:12348: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_serv:
in function 'connect']
[Sender] warning: failed to send to 127.0.0.1:12349: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_serv:
in function 'connect']
[Sender] warning: failed to send to 127.0.0.1:12350: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_serv:
in function 'connect']
Accepting connections on port 12347
[Sender] warning: failed to send to 127.0.0.1:12351: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_serv:
in function 'connect']
[Receiver DEBUG] Received line: {"msg_type":1,"src_id":"T2","dst_id":"T3","msg_content":{"coor_y":"20","item_num":"12","coor_x":"20","item_code":"02"}}
[Receiver DEBUG] Parsed msg type: 1 from T2
[Sender] warning: failed to send to 127.0.0.1:12352: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_serv:
in function 'connect']
[T3] onRespStockReceived: 메시지 수신됨 (from: T2, type: 1)
- item_code: 02
- 유효한 재고 응답. availableOtherVmsForDrink_ 크기: 1/7
- 모든 다른 VM으로부터 응답 수신 완료 또는 지정된 수만큼 받을. 타이머 취소 및 상태 변경 시도.
- 유효하지 않은 재고 응답 또는 조건 불일치 (요청 음료: 02, 수신 음료: 02, 재고: 12)

음료 [사이다]를 구매 가능한 가장 가까운 자판기 정보입니다:
> 자판기 ID: T2
> 위치: (X: 20, Y: 20)
[사이다] 음료에 대해 다른 자판기에서 선결제 하시겠습니까? (Y/N): ANDKEIN
잘못된 입력입니다. Y 또는 N을 입력해주세요.
[사이다] 음료에 대해 다른 자판기에서 선결제 하시겠습니까? (Y/N):
```

3. CP Test Resolution

| 잘못된 자판기에 선결제 코드를 입력하는 경우

- 테스트 순서
 - 자판기 개수 3개
 - 1번 자판기에서 2. (음료 선택 및 결제)
 - 음료: 박카스 (2번 자판기 음료)
 - 현재 자판기 음료 개수: 0개
 - 원격 자판기 음료 개수: 0개 초과
 - 선결제: 승인
 - 카드 번호 입력: 5555-4555-4444-3333
 - 카드 잔액: price <
 - 3번 자판기
 - 선결제 코드 입력
- expected output: 에러 메시지 호출

```
[T3] DEBUG: state_processingPayment - Payment success. isPrepay: 1
[T3] DEBUG: selectedTargetVmForPrepayment ID: T1
[T3] DEBUG: Changing state to ISSUING_AUTH_CODE_AND_REQUESTING_RESERVATION
T1에 콜라 재고 확보 요청 (인증코드: qITlIz)
[Sender DEBUG] Sending to 127.0.0.1:12345: {"msg_type":2,"src_id":"T3","dst_id":"T1","msg_content":{"item_code":"01","cert_code":"qITlIz","item_num":"1"}}
```

```
Accepting connections on port 12347
[Receiver DEBUG] Received line: {"msg_type":3,"src_id":"T1","dst_id":"T3","msg_content":{"item_code":"01","item_num":"1","availability":"T"}}
[Receiver DEBUG] Parsed msg type: 3 from T1
```

==== 선결제 완료 ====

음료명: 콜라

수량 자판기 ID: T1

수량 자판기 위치: (X: 10, Y: 10)

★인증 코드: qITlIz ★

=====

위 자판기에서 발급된 인증 코드를 입력하여 음료를 수령하세요.
거래가 완료되었습니다. 감사합니다.

```
[Receiver DEBUG] Received line: {"msg_type":0,"src_id":"T3","dst_id":"0","msg_content":{"item_code":"01","item_num":"1"}}
```

```
[Receiver DEBUG] Parsed msg type: 0 from T3
```

```
[Sender DEBUG] Sending to 127.0.0.1:12347: {"msg_type":1,"src_id":"T2","dst_id":"T3","msg_content":{"item_code":"01","coord_x":"20","item_num":"0","coord_y":"20"}}
```

2

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴: c 또는 C): dgy25

[시스템 오류]

되었습니다.)

메인 메뉴로 돌아갑니다.

(입력 코드 dgy25는 유효하지 않거나 이미 사용

3. CP Test Resolution

| 이미 사용된 선결제 코드를 입력하는 경우

- 테스트 순서
 - 자판기 개수 3개
 - 1번 자판기에서 2. (음료 선택 및 결제)
 - 음료: 박카스 (2번 자판기 음료)
 - 현재 자판기 음료 개수: 0개
 - 원격 자판기 음료 개수: 0개 초과
 - 선결제: 승인
 - 카드 번호 입력: 5555-4555-4444-3333
 - 카드 잔액: price <
 - 2번 자판기
 - 선결제 코드 입력 (AAAAA)
 - 선결제 코드 처리
 - 2번 자판기
 - 선결제 코드 재입력 (AAAAA)
- expected output: 에러 메시지 호출

2

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴 : c 또는 C): mNF0s

인증 코드 (mNF0s) 확인 완료 : 사이다

[사이다] 음료가 배출되고 있습니다. 잠시만 기다려주세요...

[사이다] 음료가 나왔습니다. 이용해주셔서 감사합니다!

거래가 완료되었습니다. 감사합니다.

옵션을 선택하세요 :

1. 음료 선택 (구매 /다른 자판기 조회)

2. 인증 코드로 음료 받기

3. 시스템 종료

선택 (숫자 입력): 2

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴 : c 또는 C): mNF0s

[시스템 오류] 입력하신 인증 코드를 찾을 수 없습니다. (입력 코드 mNF0s는 유효하지 않거나 이미 사용되었습니다.)

메인 메뉴로 돌아갑니다.

3. CP Test Resolution

- 테스트 순서
 - 자판기 개수 1개
 - 1번 자판기에서 2. (음료 선택 및 결제)
 - 음료: 펍시
 - 현재 자판기 음료 개수: 0개
- expected output: 에러 메시지 출력
- real output: (음료 재고 확인 불가)
- test result: fail

06	단 산 수	1000원
07	보 리 차	1000원
08	캔 커 피	1000원
09	물	1000원
10	에 너 지 드 링 크	1000원
11	유 자 차	1000원
12	식 해	1000원
13	아 이 스티	1000원
14	딸 기 주 스	1000원
15	오 렌 지 주 스	1000원
16	포 도 주 스	1000원
17	이 온 음 료	1000원
18	아 메 리 카 노	1000원
19	핫 초 코	1000원
20	카 페 라 테	1000원

음선을 선택하세요 :

1. 음료 선택 (구매 / 다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료

선택 (숫자 입력): 1

구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'C' 입력) : 02

알림 : 선택하신 음료 [사이다]는 현재 자판기에 재고가 없습니다 .

사이다 음료의 재고를 주변 자판기에 문의합니다 ...

[Sender] warning: failed to send to 127.0.0.1:12345: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_service.h pp:587:33 in function 'connect']

[Sender] warning: failed to send to 127.0.0.1:12346: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_service.h pp:587:33 in function 'connect']

[Sender] warning: failed to send to 127.0.0.1:12348: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_service.h pp:587:33 in function 'connect']

[Sender] warning: failed to send to 127.0.0.1:12349: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_service.h pp:587:33 in function 'connect']

[Sender] warning: failed to send to 127.0.0.1:12350: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_service.h pp:587:33 in function 'connect']

[Sender] warning: failed to send to 127.0.0.1:12351: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_service.h pp:587:33 in function 'connect']

[Sender] warning: failed to send to 127.0.0.1:12352: connect: Connection refused [system:61 at /opt/homebrew/include/boost/asio/detail/reactive_socket_service.h pp:587:33 in function 'connect']

[시스템 오류] 다른 자판기 (주변 자판기 재고 조회)로부터 응답을 받지 못했습니다 .
메인 메뉴로 돌아갑니다 .

3. CP Test Resolution

● 테스트 순서

- 자판기 개수: 2개
- 음료: 박카스
- 현재 자판기 음료 개수: 0개
- 원격 자판기 음료 개수: 0개 초과
- 선결제: 승인
- 카드 번호: 유효
- 카드 잔고: 음료 금액보다 많음
- 선결제 코드: 정상적으로 생성
- 자판기 안내: 가까운 자판기
- 선결제 코드 검증: 정상

- expected output: 가까운 거리 안내 및 음료 출고

```
-----
- item_code: 02
- 유효한 재고 응답 . availableOtherVmsForDrink, 크기: 1/7
- 모든 다른 VM으로부터 응답 수신 완료 또는 지정된 수만큼 발송, 타임어 채소 및 상태 변경 시도.
- 유효하지 않은 재고 응답 또는 조건 불일치 (요청 음료: 02, 수신 음료: 02, 재고: 12)

음료 [사이다]를 구매 가능한 가장 가까운 자판기 정보입니다:
> 자판기 ID: T2
> 위치: (X: 20, Y: 20)
[사이다] 음료에 대해 다른 자판기에서 선결제 하시겠습니까? (Y/N): y
사이다 선결제 (대상 자판기: T2). 가격: 1000원.

결제할 금액: 1000원 입니다.
카드를 삽입해주세요. (시뮬레이션: 실제 카드 처리 없음)
결제를 계속 진행하시겠습니까? (Y/N): y
결제 승인 중입니다. 잠시만 기다려주세요...
[결제 성공] 결제가 성공적으로 완료되었습니다!
[T3] DEBUG: state_processingPayment - Payment success. isPrepay: 1
[T3] DEBUG: selectedTargetVmForPrepayment ID: T2
[T3] DEBUG: Changing state to ISSUING_AUTH_CODE_AND_REQUESTING_RESERVATION
T2에 사이다 재고 확보 요청 (인증코드: TdyoL)
[Sender DEBUG] Sending to 127.0.0.1:12346: {"msg_type":2,"src_id":"T3","dst_id":"T2","msg_content":{"cert_code":"TdyoL","item_num":"1","item_code":"02"}}

Accepting connections on port 12347
[Receiver DEBUG] Received line: {"msg_type":3,"src_id":"T2","dst_id":"T3","msg_content":{"availability":"T","item_num":"1","item_code":"02"}}
[Receiver DEBUG] Parsed msg type: 3 from T2

==== 선결제 완료 ====
음료명: 사이다
수령 자판기 ID: T2
수령 자판기 위치: (X: 20, Y: 20)
★ 인증 코드: TdyoL ★
=====
위 자판기에서 발급된 인증 코드를 입력하여 음료를 수령하세요.
거래가 완료되었습니다. 감사합니다.
```

2

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴 : c 또는 C): TdyoL
인증 코드 (TdyoL) 확인 완료 : 사이다

[사이다] 음료가 배출되고 있습니다. 잠시만 기다려주세요...
[사이다] 음료가 나왔습니다. 이용해 주셔서 감사합니다!
거래가 완료되었습니다. 감사합니다.

3. CP Test Resolution

- 테스트 순서

- 자판기 개수: 3개
- 음료: 포카리스웨트 (3번 자판기 음료)
- 현재 자판기 음료 개수: 0개
- 원격 자판기 음료 개수: 0개 초과
- 선결제: 승인
- 카드 번호: 유효
- 카드 잔고: 음료 금액보다 많음
- 선결제 코드: 정상적으로 생성
- 자판기 안내: 가까운 자판기
- 선결제 코드 검증: 정상

- expected output: 음료 정상 출고

==== 선결제 완료 ====

음료명 : 사이다

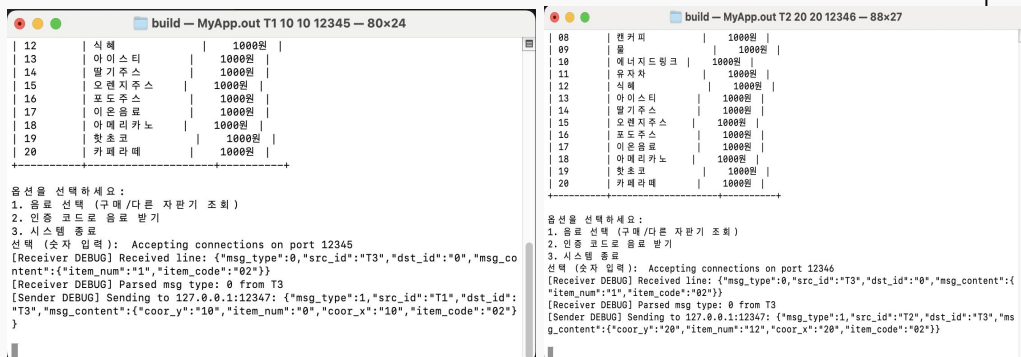
수량 자판기 ID: T2

수량 자판기 위치 : (X: 20, Y: 20)

★ 인증 코드 : 9KPc7 ★

=====

위 자판기에서 발급된 인증 코드를 입력하여 음료를 수령하세요.
거래가 완료되었습니다. 감사합니다.



2

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴 : c 또는 C): 9KPc7
인증 코드 (9KPc7) 확인 완료 : 사이다

[사이다] 음료가 배출되고 있습니다. 잠시만 기다려주세요...

[사이다] 음료가 나왔습니다. 이용해주셔서 감사합니다!

거래가 완료되었습니다. 감사합니다.

3. CP Test Resolution

| 음료 선택 메뉴에서 30초 이상 대기

- 테스트 순서
 - b. (음료선택 및 결제)
 - 30초 이상 대기
- expected output: 에러메시지 출력
- real output: 에러메시지 없음

```
===== Main Menu =====  
1. 음료 목록 보기  
2. 음료 선택 및 결제  
3. 선결제 코드 입력  
0. 종료  
=====
```

선택: 2
음료수를 선택하세요: █

- test result: fail

| 선결제 승인/거절 선택지에서 10초 대기

- 테스트 순서
 - 자판기 개수 2개
 - 1번 자판기에서 2. (음료 선택 및 결제)
 - 음료: 박카스 (원격 자판기 음료)
 - 현재 자판기 음료 개수: 0개
 - 원격 자판기 음료 개수: 0개 초과
 - 선결제: (입력 없음, 10초 간 대기)
- expected output: 에러 메시지 출력
- real output: 선결제 메뉴 미구현으로 인한 선결제 승인/거절 선택지 대기 불가

3. CP Test Resolution

| 잘못된 카드 번호 입력

- 테스트 순서
 - 자판기 개수 1개
 - 1번 자판기에서 2. (음료 선택 및 결제)
 - 음료: 펍시
 - 현재 자판기 음료 개수: 2개
 - 카드 번호 입력: ANDKLFIEINFOQIOQEFNOIEQ
- expected output: 결제 거절 메시지 출력
- real output: 카드 번호 입력 부분 미구현

시스템 설계상 카드 정보 입력 없으므로 대응 불가능함

3. CP Test Resolution

| 잘못된 형식의 선결제 코드 입력

- 테스트 순서
 - 자판기 개수 3개
 - 1번 자판기에서 2. (음료 선택 및 결제)
 - 음료: 박카스 (2번 자판기 음료)
 - 현재 자판기 음료 개수: 0개
 - 원격 자판기 음료 개수: 0개 초과
 - 선결제: 승인
 - 카드 번호 입력: 5555-4555-4444-3333
 - 카드 잔액: price <
 - 2번 자판기
 - 선결제 코드 입력 (모우림ㄴ위ㅏ르ㄴ)
- expected output: 에러 메시지 호출

터미널 강제종료

3. CP Test Resolution

| 카드 잔액이 음료 가격보다 작을 경우

- 테스트 순서
 - 자판기 개수 1개
 - 1번 자판기에서 2. (음료 선택 및 결제)
 - 음료: 펍시
 - 현재 자판기 음료 개수: 2개
 - 카드 번호 입력: 5555-4555-4444-3333
 - 카드 잔액: < 1500
- expected output: 결제 거절 메시지 출력
- real output: 카드 번호 입력 부분 미구현

시스템 설계상 카드 잔액 관련 설정이 없으므로
대응 불가능함

4. BF Test Resolution

Test	Test Num	Test Result	Detail
Select Menu	1-1	FAIL	프로그램 종료
	1-2	FAIL	미출력
	1-3	PASS	에러메시지 출력
	1-4	FAIL	입력값 유효
Select Drink	2-1	FAIL	미출력
	2-2	FAIL	결제 거절 메시지 출력
	2-3	PASS	에러 메시지 출력
	2-4	FAIL	미출력
	2-5	FAIL	문자가 깨져 확인 불가
Prepayment	3-1	FAIL	문자가 깨져 확인 불가
	3-2	FAIL	미구현
	3-3	FAIL	미출력
	3-4	FAIL	문자가 깨져 확인 불가
	3-5	FAIL	다수의 인증코드
	3-6	FAIL	문자가 깨져 확인 불가
	3-7	FAIL	문자가 깨져 확인 불가
	3-8	FAIL	문자가 깨져 확인 불가
Provide Drink	4-1	FAIL	비정상 출력
Card System	5-1	FAIL	미구현
	5-2	FAIL	미구현
	5-3	FAIL	미구현
DVMs location	6-1	FAIL	문자가 깨져 확인 불가
	6-2	FAIL	위치 좌표 확인 불가로 인한 확인 실패

PASS: 2

INVALID: 0

TOTAL: 23

PASS/VALID: 2/23

BFT Pass Percentage
= 8.69%

4. BF Test Resolution

(1-1)

| 메뉴선택에서 한글, 영문, 특수문자 입력

- 테스트 순서
메뉴 선택화면에서 한글: 테스트, 영문: test, 특수문자: % 입력
- expected output: 에러메시지 출력
- real output: 프로그램 종료

자판기 시스템을 시작합니다. 현재 자판기 ID: T2

===== Vending Machine Menu =====

--- 음료 목록 (총 20종) ---

음료코드	음료명	가격
01	콜라	1000원
02	사이다	1000원
03	녹차	1000원
04	홍차	1000원
05	밀크티	1000원
06	탄산수	1000원
07	보리차	1000원
08	캔커피	1000원
09	물	1000원
10	에너지링크	1000원
11	유자차	1000원
12	식혜	1000원
13	아이스티	1000원
14	딸기주스	1000원
15	오렌지주스	1000원
16	포도주스	1000원
17	이온음료	1000원
18	아메리카노	1000원
19	핫초코	1000원
20	카페라떼	1000원

음료를 선택하세요 :

1. 음료 선택 (구매 / 다른 자판기 조회)

2. 인증 코드로 음료 받기

3. 시스템 종료

선택 (숫자 입력): 테스트

잘못된 입력입니다. 1부터 3 사이의 숫자를 입력해주세요.

선택 (숫자 입력): test

잘못된 입력입니다. 1부터 3 사이의 숫자를 입력해주세요.

선택 (숫자 입력): %

잘못된 입력입니다. 1부터 3 사이의 숫자를 입력해주세요.

선택 (숫자 입력):

4. BF Test Resolution

(1-2)

| 메뉴선택에서 공란 입력

- 테스트 순서
메뉴 선택화면에서 아무것도 입력하지 않은 후 엔터
- expected output: 에러메시지 출력
- real output: 미출력

```
= 18 (12332): 통신 과정에서 제외됨 (자정 목록에 없음)
Accepting connections on port 12345
정보: T5의 io_context 스레드 시작.
자판기 시스템을 시작합니다. 현재 자판기 ID: T5

===== Vending Machine Menu =====

--- 음료 목록 (총 20종) ---
+-----+-----+-----+
| 음료코드 | 음료명       | 가격   |
+-----+-----+-----+
| 01       | 콜라        | 1000원 |
| 02       | 사이다      | 1000원 |
| 03       | 녹차        | 1000원 |
| 04       | 홍차        | 1000원 |
| 05       | 밀크티      | 1000원 |
| 06       | 탄산수      | 1000원 |
| 07       | 보리차      | 1000원 |
| 08       | 캔커피      | 1000원 |
| 09       | 물          | 1000원 |
| 10       | 에너지드링크 | 1000원 |
| 11       | 유자차      | 1000원 |
| 12       | 식혜        | 1000원 |
| 13       | 아이스티    | 1000원 |
| 14       | 딸기주스    | 1000원 |
| 15       | 오렌지주스  | 1000원 |
| 16       | 포도주스    | 1000원 |
| 17       | 이온음료    | 1000원 |
| 18       | 아메리카노  | 1000원 |
| 19       | 핫초코      | 1000원 |
| 20       | 카페라떼    | 1000원 |
+-----+-----+-----+

옵션을 선택하세요 :
1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료
선택 (숫자 입력):
입력이 비어있습니다. 1부터 3 사이의 숫자를 입력해주세요.
```

4. BF Test Resolution

(1-4)

| 메뉴 선택에서 여러 메뉴 입력

- 테스트 순서
메뉴 선택 화면에서 1 2 3 입력
- expected output: 에러 메시지 출력
- real output: 버퍼에 입력된 순서대로 입력값이 유효하게 적용됨

```
정보: T5 자판기의 초기 채고 설정 완료.
정보: 다른 자판기 정보 로드 중...
- T1 (12345): 통신 대상에서 제외됨 (지정 목록에 없음)
+ T2 (Port: 12346, Endpoint: 127.0.0.1:12346) 통신 대상으로 추가됨.
- T3 (12347): 통신 대상에서 제외됨 (지정 목록에 없음)
- T4 (12348): 통신 대상에서 제외됨 (지정 목록에 없음)
- T6 (12350): 통신 대상에서 제외됨 (지정 목록에 없음)
- T7 (12351): 통신 대상에서 제외됨 (지정 목록에 없음)
- T8 (12352): 통신 대상에서 제외됨 (지정 목록에 없음)
Accepting connections on port 12345
정보: T5의 io_context 스레드 시작.
자판기 시스템을 시작합니다. 현재 자판기 ID: T5

===== Vending Machine Menu =====

--- 음료 목록 (총 20종) ---
+-----+-----+-----+
| 음료코드 | 음료명       | 가격   |
+-----+-----+-----+
| 01       | 콜라        | 1000원 |
| 02       | 사이다      | 1000원 |
| 03       | 녹차        | 1000원 |
| 04       | 홍차        | 1000원 |
| 05       | 밀크티      | 1000원 |
| 06       | 탄산수      | 1000원 |
| 07       | 보리차      | 1000원 |
| 08       | 캔커피      | 1000원 |
| 09       | 물          | 1000원 |
| 10       | 에너지드링크 | 1000원 |
| 11       | 유자차      | 1000원 |
| 12       | 식혜        | 1000원 |
| 13       | 아이스티    | 1000원 |
| 14       | 딸기주스    | 1000원 |
| 15       | 오렌지주스  | 1000원 |
| 16       | 포도주스    | 1000원 |
| 17       | 이온음료    | 1000원 |
| 18       | 아메리카노  | 1000원 |
| 19       | 핫초코      | 1000원 |
| 20       | 카페라떼    | 1000원 |
+-----+-----+-----+

음선을 선택하세요:
1. 음료 선택 (구매 / 다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료
선택 (숫자 입력):
입력이 비어있습니다. 1부터 3 사이의 숫자를 입력해주세요.
선택 (숫자 입력): 1 2 3
잘못된 입력입니다. 1부터 3 사이의 숫자를 정확히 입력해주세요.
선택 (숫자 입력):
```

4. BF Test Resolution (2-1)

| 음료 선택화면에서 선택하지 않고 기다리기

- 테스트 순서
메뉴 2의 음료선택 화면에서 음료를 선택하지 않고 30초 이상 대기
- expected output: 에러 메시지 출력
- real output: 미출력

자판기 시스템을 시작합니다. 현재 자판기 ID: T5

===== Vending Machine Menu =====

--- 음료 목록 (총 20종) ---

음료 코드	음료명	가격
01	콜라	1000원
02	사이다	1000원
03	녹차	1000원
04	홍차	1000원
05	밀크티	1000원
06	탄산수	1000원
07	보리차	1000원
08	캔커피	1000원
09	물	1000원
10	에너지드링크	1000원
11	유자차	1000원
12	식혜	1000원
13	아이스티	1000원
14	딸기주스	1000원
15	오렌지주스	1000원
16	포도주스	1000원
17	이온음료	1000원
18	아메리카노	1000원
19	핫초코	1000원
20	카페라떼	1000원

음선을 선택하세요 :

1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료

선택 (숫자 입력):

- Use Case 1. 음료 목록 조회 및 표시와
- Use Case 2. 사용자 음료선택 처리가
같은 화면에서 처리되어 대응 불가

4. BF Test Resolution (2-2)

| 음료 선택에서 여러 음료 동시에 선택

- 테스트 순서
음료 선택 화면에서 펍시 게토레이 칸타타 선택
- expected output: 결제 승인
- real output: 결제 음료들 개수보다 초과 승인 후, 마지막에 거절

```
- T8 (12352): 통신 대상에서 제외됨 (지정 목록에 없음)
Accepting connections on port 12345
정보: T5의 io_context 스레드 시작.
자판기 시스템을 시작합니다. 현재 자판기 ID: T5

===== Vending Machine Menu =====

--- 음료 목록 (총 20종) ---
+-----+-----+-----+
| 음료코드 | 음료명   | 가격   |
+-----+-----+-----+
| 01      | 콜라    | 1000원 |
| 02      | 사이다  | 1000원 |
| 03      | 녹차    | 1000원 |
| 04      | 홍차    | 1000원 |
| 05      | 밀크티  | 1000원 |
| 06      | 탄산수  | 1000원 |
| 07      | 보리차  | 1000원 |
| 08      | 캔커피  | 1000원 |
| 09      | 물      | 1000원 |
| 10      | 에너지드링크 | 1000원 |
| 11      | 유자차  | 1000원 |
| 12      | 식혜    | 1000원 |
| 13      | 아이스티 | 1000원 |
| 14      | 딸기주스 | 1000원 |
| 15      | 오렌지주스 | 1000원 |
| 16      | 포도주스 | 1000원 |
| 17      | 이온음료 | 1000원 |
| 18      | 아메리카노 | 1000원 |
| 19      | 핫초코  | 1000원 |
| 20      | 카페라떼 | 1000원 |
+-----+-----+-----+

음선을 선택하세요:
1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료
선택 (숫자 입력):
입력이 비어있습니다. 1부터 3 사이의 숫자를 입력해주세요.
선택 (숫자 입력): 1 2 3
잘못된 입력입니다. 1부터 3 사이의 숫자를 정확히 입력해주세요.
선택 (숫자 입력): 1
구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'C' 입력): 01 02 03
잘못된 음료 코드입니다. 목록에 있는 2자리 숫자 코드를 입력하거나 'c'를 입력하여 취소하세요.
구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'C' 입력):
```

- 1 주문 1 음료로 구현해서 대응
불가

4. BF Test Resolution (2-4)

| 음료 선택에서 공란 입력

- 테스트 순서
음료 선택 화면에서 공란으로 엔터
- expected output: 에러 메시지 출력
- real output: 미출력

자판기 시스템을 시작합니다. 현재 자판기 ID: T5

===== Vending Machine Menu =====

--- 음료 목록 (총 20종) ---

음료코드	음료명	가격
01	콜라	1000원
02	사이다	1000원
03	녹차	1000원
04	홍차	1000원
05	밀크티	1000원
06	탄산수	1000원
07	보리차	1000원
08	캔커피	1000원
09	물	1000원
10	에너지링크	1000원
11	유자차	1000원
12	식혜	1000원
13	아이스티	1000원
14	딸기주스	1000원
15	오렌지주스	1000원
16	포도주스	1000원
17	이온음료	1000원
18	아메리카노	1000원
19	핫초코	1000원
20	카페라떼	1000원

음료를 선택하세요 :

1. 음료 선택 (구매/다른 자판기 조회)

2. 인증 코드로 음료 받기

3. 시스템 종료

선택 (숫자 입력):

입력이 비어있습니다. 1부터 3 사이의 숫자를 입력해주세요.

선택 (숫자 입력): 1 2 3

잘못된 입력입니다. 1부터 3 사이의 숫자를 정확히 입력해주세요.

선택 (숫자 입력): 1

구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'C' 입력): 01 02 03
잘못된 음료 코드입니다. 목록에 있는 2자리 숫자 코드를 입력하거나 'c'를 입력하여 취소하세요.

구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'C' 입력):
잘못된 음료 코드입니다. 목록에 있는 2자리 숫자 코드를 입력하거나 'c'를 입력하여 취소하세요.

4. BF Test Resolution (2-5)

| 현재 자판기 재고 확인

- 테스트 순서
메뉴 1의 음료 목록 보기 선택
- expected output: 현재 자판기 재고 출력
- real output:

```
===== Vending Machine Menu =====  
--- 음료 목록 (총 20종) ---  
+-----+  
| 음료 코드 | 음료명           | 가격   |  
+-----+  
| 01        | 콜라             | 1000원 |  
| 02        | 사 이 다         | 1000원 |  
| 03        | 녹 차            | 1000원 |  
| 04        | 홍 차            | 1000원 |  
| 05        | 밀크 티          | 1000원 |  
| 06        | 탄 산 수         | 1000원 |  
| 07        | 보 리 차         | 1000원 |  
| 08        | 캔 커 피         | 1000원 |  
| 09        | 물               | 1000원 |  
| 10        | 에 너 지 드 링크 | 1000원 |  
| 11        | 유 자 차         | 1000원 |  
| 12        | 식 혜           | 1000원 |  
| 13        | 아 이 스 티     | 1000원 |  
| 14        | 딸 기 주 스     | 1000원 |  
| 15        | 오 렌 지 주 스   | 1000원 |  
| 16        | 포 도 주 스     | 1000원 |  
| 17        | 이 온 음 료     | 1000원 |  
| 18        | 아 메 리 카 노   | 1000원 |  
| 19        | 핫 초 코        | 1000원 |  
| 20        | 카 페 라 떼     | 1000원 |  
+-----+  
  
옵션을 선택하세요 :  
1. 음료 선택 (구매 / 다른 자판기 조회)  
2. 인증 코드로 음료 받기  
3. 시스템 종료
```

현재 재고 출력 하지 않고 전체 음료
목록만 출력

4. BF Test Resolution (3-1)

| 선결제 코드 입력에서 유효하지 않은 인증코드 입력

- 테스트 순서

선결제 코드 입력 화면에서 유효하지 않은 코드 dfsf342 입력

- expected output: 에러 메시지 출력

Accepting connections on port 12349

==== 선결제 완료 ====

음료명 : 콜라

수량 자판기 ID: T1

수량 자판기 위치 : (X: 10, Y: 10)

★ 인증 코드 : U7Bir ★

=====

위 자판기에서 발급된 인증 코드를 입력하여 음료를 수령하세요 .

거래가 완료되었습니다 . 감사합니다 .

옵션을 선택하세요 :

1. 음료 선택 (구매 /다른 자판기 조회)

2. 인증 코드로 음료 받기

3. 시스템 종료

선택 (숫자 입력) : Accepting connections on port 12345

Accepting connections on port 12345

2

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴 :

c 또는 C): U7BiR

[시스템 오류] 입력하신 인증 코드를 찾을 수 없습니다 . (입력 코드 U7BiR는 유효하지 않거나 이미 사용되었습니다 .)

메인 메뉴로 돌아갑니다 .

4. BF Test Resolution

(3-2)

| 선결제 코드 입력에서 사용한 인증 코드 재입력

- 테스트 순서

선결제 코드 입력 화면에서 유효하지 않은 코드 dfsf342 입력

- expected output: 에러 메시지 출력

```
음선을 선택하세요 :
1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료
선택 (숫자 입력): 2
수정할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴: c 또는 C): U7B1r
인증 코드 (U7B1r) 확인 완료: 클라
```

```
[클라] 음료가 배송되고 있습니다. 잠시만 기다려주세요...
[클라] 음료가 나왔습니다. 이용해주시어서 감사합니다!
거래가 완료되었습니다. 감사합니다.
```

===== Vending Machine Menu =====

--- 음료 목록 (총 20종) ---		
음료코드	음료명	가격
01	클라	1000원
02	사이다	1000원
03	녹차	1000원
04	홍차	1000원
05	밀크티	1000원
06	탄산수	1000원
07	보리차	1000원
08	캔커피	1000원
09	물	1000원
10	에너지드링크	1000원
11	유자차	1000원
12	식혜	1000원
13	아이스티	1000원
14	딸기주스	1000원
15	오렌지주스	1000원
16	포도주스	1000원
17	이온음료	1000원
18	아메리카노	1000원
19	참초코	1000원
20	카베라메	1000원

```
음선을 선택하세요 :
1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료
선택 (숫자 입력): 2
수정할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴: c 또는 C): U7B1r
[시스템 오류] 입력하신 인증 코드를 찾을 수 없습니다. (입력 코드 U7B1r는 유효하지 않거나 이미 사용되었습니다.)
메인 메뉴로 돌아갑니다.
```

4. BF Test Resolution (3-3)

| 선결제 코드 입력에서 공란 입력

- 테스트 순서
선결제 코드 입력 화면에서 공란 입력
- expected output: 에러 메시지 출력

옵션을 선택하세요 :

1. 음료 선택 (구매 /다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료

선택 (숫자 입력) : 2

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴 : c 또는 C):

잘못된 형식의 인증 코드입니다. 5자리 영숫자를 입력해주세요.

4. BF Test Resolution (3-4)

| 선결제 코드 입력에서 특수문자 입력

- 테스트 순서
선결제 코드 입력 화면에서 % 입력
- expected output: 에러 메시지 출력

옵션을 선택하세요 :

1. 음료 선택 (구매 / 다른 자판기 조회)

2. 인증 코드로 음료 받기

3. 시스템 종료

선택 (숫자 입력) : 2

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴 : c 또는 C):

잘못된 형식의 인증 코드입니다. 5자리 영숫자를 입력해주세요.

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴 : c 또는 C): 2

잘못된 형식의 인증 코드입니다. 5자리 영숫자를 입력해주세요.

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴 : c 또는 C): @#\$

잘못된 형식의 인증 코드입니다. 5자리 영숫자를 입력해주세요.

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴 : c 또는 C): ###

잘못된 형식의 인증 코드입니다. 5자리 영숫자를 입력해주세요.

4. BF Test Resolution (3-5)

| 선결제 시 유일한 인증코드 생성 되는지 확인

- 테스트 순서
음료 선택 화면에서 선결제 실시
- expected output: 하나의 인증코드 ◆?

```
결제 할 금액 : 1000원 입니다 .  
카드를 삽입 해주세요 . (시뮬레이션 : 실제 카드 처리 없음 )  
결제를 계속 진행 하시겠습니까 ? (Y/N): y  
결제 승인 중입니다 . 잠시만 기다려주세요 ...  
[결제 성공] 결제가 성공적으로 완료되었습니다 !  
T1에 콜라 재고 확보 요청 (인증코드 : U7Bir)  
Accepting connections on port 12349
```

```
==== 선 결제 완료 ====
```

```
음료명 : 콜라
```

```
수량 자판기 ID: T1
```

```
수량 자판기 위치 : (X: 10, Y: 10)
```

```
★ 인증 코드 : U7Bir ★
```

```
=====
```

```
위 자판기에서 발급된 인증 코드를 입력하여 음료를 수령하세요 .  
거래가 완료되었습니다 . 감사합니다 .
```

4. BF Test Resolution (3-6)

| 선결제 코드 입력에서 한글 입력

- 테스트 순서
선결제 코드 입력 화면에서 '테스트' 입력
- expected output: 에러 메시지 출력

옵션을 선택하세요 :

1. 음료 선택 (구매 / 다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료

선택 (숫자 입력) : 2

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴 : c 또는 C): 테스트
잘못된 형식의 인증 코드입니다. 5자리 영숫자를 입력해주세요.

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴 : c 또는 C):

4. BF Test Resolution

(3-7)

| 안내되지 않은 VM에 인증코드 입력

- 테스트 순서

다른 터미널에서 실행한 VM에 인증코드 HXWYS 입력

- expected output: 에러 메시지 출력

```
==== 선결제 완료 ====
음료명 : 콜라
수령 자판기 ID: T1
수령 자판기 위치 : (X: 10, Y: 10)
★ 인증 코드 : LngTG ★
=====
위 자판기에서 발급된 인증 코드를 입력하여 음료를 수령하세요.
거레가 완료되었습니다. 감사합니다.

===== Vending Machine Menu =====

-- 음료 목록 (총 20종) --
+-----+
| 음료코드 | 음료명       | 가격   |
+-----+
| 01       | 콜라        | 1000원 |
| 02       | 사이다      | 1000원 |
| 03       | 녹차        | 1000원 |
| 04       | 홍차        | 1000원 |
| 05       | 밀크티      | 1000원 |
| 06       | 탄산수      | 1000원 |
| 07       | 브리차      | 1000원 |
| 08       | 캔커피       | 1000원 |
| 09       | 물           | 1000원 |
| 10       | 에너지드링크 | 1000원 |
| 11       | 유자차      | 1000원 |
| 12       | 식혜        | 1000원 |
| 13       | 아이스티    | 1000원 |
| 14       | 딸기주스    | 1000원 |
| 15       | 오렌지주스  | 1000원 |
| 16       | 포도주스    | 1000원 |
| 17       | 이온음료    | 1000원 |
| 18       | 아메리카노  | 1000원 |
| 19       | 핫초코      | 1000원 |
| 20       | 카페라떼    | 1000원 |
+-----+

음선을 선택하세요 :
1. 음료 선택 (구매/다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료
선택 (숫자 입력): 2
수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴: c 또는 C): LngTG
[시스템 오류] 입력하신 인증 코드를 찾을 수 없습니다. (입력 코드 LngTG는 유효하지 않거나 이미 사용되었습니다.)
메인 메뉴로 돌아갑니다.
```

4. BF Test Resolution

(3-8)

| 선결제 코드 입력에서 사용한 인증 코드 재입력

- 테스트 순서

서로 다른 VM에서 같은 코드 동시 입력

- expected output: 에러 메시지 출력

==== 선결제 완료 ====

음료명 : 콜라

수량 자판기 ID: T1

수량 자판기 위치 : (X: 10, Y: 10)

★ 인증 코드 : LngTG ★

위 자판기에서 발급된 인증 코드를 입력하여 음료를 수령하세요.
거래가 완료되었습니다. 감사합니다.

===== Vending Machine Menu =====

--- 음료 목록 (총 20종) ---

음료 코드	음료명	가격
01	콜라	1000원
02	사이다	1000원
03	녹차	1000원
04	홍차	1000원
05	밀크티	1000원
06	탄산수	1000원
07	보리차	1000원
08	커피	1000원
09	물	1000원
10	에너지드링크	1000원
11	유자차	1000원
12	식혜	1000원
13	아이스티	1000원
14	딸기주스	1000원
15	오렌지주스	1000원
16	포도주스	1000원
17	이온음료	1000원
18	아메리카노	1000원
19	핫초코	1000원
20	카페라떼	1000원

음션을 선택하세요 :

1. 음료 선택 (구매/다른 자판기 조회)

2. 인증 코드로 음료 받기

3. 시스템 종료

선택 (숫자 입력): 2

수령할 음료의 5자리 인증 코드를 입력하세요 (메인 메뉴 : c 또는 C): LngTG

[시스템 오류] 입력하신 인증 코드를 찾을 수 없습니다. (입력 코드 LngTG는 유효하지 않거나 이미 사용되었습니다.)

메인 메뉴로 돌아갑니다.

4. BF Test Resolution

(4-1)

| 음료 주문 완료후 정확한 음료가 제공되는지 확인

- 테스트 순서

음료 선택에서 음료 주문 완료 후 정확한 음료가 제공되는지 테스트

- expected output: 음료제공 확인 출력

===== Vending Machine Menu =====

--- 음료 목록 (총 20종) ---

음료코드	음료명	가격
01	콜라	1000원
02	사이다	1000원
03	녹차	1000원
04	홍차	1000원
05	밀크티	1000원
06	탄산수	1000원
07	보리차	1000원
08	캔커피	1000원
09	물	1000원
10	에너지드링크	1000원
11	유자차	1000원
12	식혜	1000원
13	아이스티	1000원
14	딸기주스	1000원
15	오렌지주스	1000원
16	포도주스	1000원
17	이온음료	1000원
18	아메리카노	1000원
19	핫초코	1000원
20	카페라떼	1000원

음선을 선택하세요 :

1. 음료 선택 (구매/다른 자판기 조회)

2. 인증 코드로 음료 받기

3. 시스템 종료

선택 (숫자 입력) : 1

구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'C' 입력) : 02

사이다 선택됨 . 가격 : 1000원 . (재고 : 5개)

사이다 구매 . 가격 : 1000원 .

결제할 금액 : 1000원 입니다 .

카드를 삽입해주세요 . (시뮬레이션 : 실제 카드 처리 없음)

결제를 계속 진행하시겠습니까? (Y/N) : y

결제 승인 중입니다 . 잠시만 기다려주세요 ...

[결제 성공] 결제가 성공적으로 완료되었습니다 !

[사이다] 음료가 배출되고 있습니다 . 잠시만 기다려주세요 ...

[사이다] 음료가 나왔습니다 . 이용해주셔서 감사합니다 !

거래가 완료되었습니다 . 감사합니다 .

4. BF Test Resolution

(5-1)

| 카드 정보 입력시 유효하지 않은 카드 정보 입력

- 테스트 순서
카드 정보 입력란에 유효하지 않은 카드 정보를 입력
- expected output: 에러 메시지 출력

```
결제할 금액 : 1000원 입니다 .  
카드를 삽입해 주세요 . (시뮬레이션 : 실제 카드 처리 없음 )  
결제를 계속 진행하시겠습니까 ? (Y/N): y  
결제 승인 중입니다 . 잠시만 기다려주세요 ...  
[결제 성공] 결제가 성공적으로 완료되었습니다 !
```

- 시스템 설계상 카드 정보 입력 없으므로 대응 불가능함

4. BF Test Resolution (5-2)

| 카드 정보 입력시 공란 입력

- 테스트 순서
카드 정보 입력 란에 공란 입력
- expected output: 에러 메시지 출력

```
결제 할 금액 : 1000원 입니다 .  
카드를 삽입해 주세요 . (시뮬레이션 : 실제 카드 처리 없음 )  
결제를 계속 진행하시겠습니까 ? (Y/N): y  
결제 승인 중입니다 . 잠시만 기다려주세요 ...  
[결제 성공] 결제가 성공적으로 완료되었습니다 !
```

- 시스템 설계상 카드 정보 입력 없으므로 대응 불가능함

4. BF Test Resolution

(5-3)

| 카드 정보 입력시 영문, 한글, 특수문자 입력

- 테스트 순서
카드 정보 입력란에 영문, 한글, 특수문자 각각 입력
- expected output: 에러 메시지 출력

```
결제할 금액 : 1000원 입니다 .  
카드를 삽입해 주세요 . (시뮬레이션 : 실제 카드 처리 없음 )  
결제를 계속 진행하시겠습니까 ? (Y/N): y  
결제 승인 중입니다 . 잠시만 기다려주세요 ...  
[결제 성공] 결제가 성공적으로 완료되었습니다 !
```

- 시스템 설계상 카드 정보 입력 없으므로 대응 불가능함

4. BF Test Resolution (6-1)

| DVMs 들의 위치가 유효하게 출력되는지 확인

- 테스트 순서
선결제 시 재고가 있는 자판기의 유효한 좌표가 출력되는지 확인
- expected output: 다른 DVM의 위치 좌표 출력

Accepting connections on port 12345

==== 선결제 완료 ====

음료명 : 콜라

수령 자판기 ID: T1

수령 자판기 위치 : (X: 10, Y: 10)

★ 인증 코드 : QFVG7 ★

=====

위 자판기에서 발급된 인증 코드를 입력하여 음료를 수령하세요.

각각의 자판기마다 인증 코드가 다릅니다.

4. BF Test Resolution (6-2)

| 다른 DVM의 전원 off 후 위치 출력 확인

- 테스트 순서

다른 DVM의 전원 off 후 해당 DVM의 위치가 출력되는지 확인

- expected output: 다른 DVM의 위치 좌표 출력

옵션을 선택하세요 :

1. 음료 선택 (구매 /다른 자판기 조회)
2. 인증 코드로 음료 받기
3. 시스템 종료

선택 (숫자 입력): 1

구매할 음료의 2자리 코드를 입력하세요 (메뉴로 돌아가려면 'c' 또는 'C' 입력): 01

알림: 선택하신 음료 [콜라]는 현재 자판기에 재고가 없습니다.

콜라 음료의 재고를 주변 자판기에 문의합니다...

[시스템 오류] 다른 자판기 (주변 자판기 재고 조회)로부터 응답을 받지 못했습니다.

메인 메뉴로 돌아갑니다.