

# **Distributed Vending Machine**

## **: MS129 Vending Machine**

**2021 객체지향개발방법론 1조**  
**OOPT Stage 2040**  
**[Ver 1.0]**

**201914173 김현웅**  
**201714164 박서영**  
**201711315 신원세**  
**201710560 정의재**

## Contents

**Activity 2141. Design Real Use Cases**

**Activity 2143. Define Interaction Diagram**

**Activity 2144. Define Design Class Diagrams**

**Activity 2145. Refine System Architecture**

**Activity 2147. Perform 2040 Traceability Analysis**

# Activity 2141. Design Real Use Cases

## 1. DVM들을 보여준다

|                                      |   |
|--------------------------------------|---|
| <b>Use Case</b>                      | 1. DVM들을 보여준다.  |
| <b>Actor</b>                         | System  |
| <b>Type</b>                          | Hidden  |
| <b>Cross Reference</b>               | Functions: R 1.1, R 1.2<br>Use cases: “음료 선택을 진행한다.”  |
| <b>Pre-Requisites</b>                | N/A   |
| <b>Typical Courses of Events</b>     | (A): Actor, (S): System<br>1. (S) 다른 자판기에게 On/Off 및 작동 상태 확인을 요청한다<br>2. (S) Screen 에 사용 가능한 작동상태의DVM 목록을 보여준다. |
| <b>Alternative Courses of Events</b> | N/A   |
| <b>Exceptional Courses of Events</b> | N/A   |

## 2. 음료 선택을 진행한다.

|                                      |   |
|--------------------------------------|---|
| <b>Use Case</b>                      | 2. 음료 선택을 진행한다.   |
| <b>Actor</b>                         | Customer  |
| <b>Type</b>                          | Evident   |
| <b>Cross Reference</b>               | Functions: R 1.2, R 1,1<br>Use cases : “DVM들을 보여준다”   |
| <b>Pre-Requisites</b>                | N/A   |
| <b>Typical Courses of Events</b>     | (A): Actor, (S): System<br>1.(A) Customer 가 DVM 을 dialButton 을 통해,선택한다.<br>2.(S) 선택된 DVM 시스템을 사용한다.<br>3.(S) Screen을 통해, 선택된 자판기의 음료 메뉴를 Screen을 통해 보여준다.<br>이 때 0번부터 6번dialButton 이 현재 자판기 판매 음료이며, 6번부터 20번dialButton 이 현재 자판기에서 판매하지 않는 음료 메뉴이다. |
| <b>Alternative Courses of Events</b> | N/A   |
| <b>Exceptional Courses of Events</b> | N/A   |

### 3. 현재 자판기에 판매하는 음료수를 선택한다

|                                      |   |
|--------------------------------------|---|
| <b>Use Case</b>                      | 3. 현재 자판기에서 판매하는 음료수를 선택한다.   |
| <b>Actor</b>                         | Customer  |
| <b>Type</b>                          | Evident   |
| <b>Cross Reference</b>               | Functions: R 2.1, R 1.2, R 2.3, R 3.1, R 3.2<br>Use cases :”음료 선택을 진행한다.”,”결제 방법 중 하나를 선택한다.”,”현재 자판기의 재고를 확인한다.”,”다른 자판기의 재고를 확인한다.”   |
| <b>Pre-Requisites</b>                | - 전체 20가지의 음료 종류를 알아야 한다.   |
| <b>Typical Courses of Events</b>     | (A1): Actor, (A2): Other DVMs, (S): System<br>1. (A1) Customer 가 현재 자판기에 판매하는 음료수를 dialButton 을 통해, 선택한다.<br>2. (S) “현재 자판기의 재고를 확인한다” 를 invoke 한다.<br>3. (S) 만약 현재 자판기에 재고가 있다면, “결제 방법 중 하나를 선택한다”를 invoke 한다.<br>4. (S) 만약 현재 자판기에 재고가 없다면, “다른 자판기의 재고 상태 확인을 요청한다”를 invoke 한다. |
| <b>Alternative Courses of Events</b> | N/A   |
| <b>Exceptional Courses of Events</b> | N/A   |

### 4. 현재 자판기에서 판매하지 않는 음료수를 선택한다

|                                      |  |
|--------------------------------------|--|
| <b>Use Case</b>                      | 4. 현재 자판기에서 판매하지 않는 음료수를 선택한다.   |
| <b>Actor</b>                         | Customer   |
| <b>Type</b>                          | Evident  |
| <b>Cross Reference</b>               | Functions: R 2.2, R 1.2, R 3.2, R5.1<br>Use cases :”음료 선택을 진행한다.”,”결제 방법 선택 메뉴를 보여준다.”,”다른 자판기에게 재고 상태 확인을 요청한다.”,”선결제를 진행한다.”   |
| <b>Pre-Requisites</b>                | - 전체 20가지의 음료 종류를 알아야 한다.  |
| <b>Typical Courses of Events</b>     | (A1): Actor, (A2): Other DVMs, (S): System<br>1. (A1) Customer 가 현재 자판기에서 판매하지 않는 음료수를 dialButton 을 통해, 선택한다.<br>2. (A2) “다른 자판기의 재고 상태 확인을 요청한다”를 invoke한다.<br>3. (S) 만약 다른 자판기에 재고가 있다면 “선결제를 진행한다”를 invoke한다. |
| <b>Alternative Courses of Events</b> | N/A  |
| <b>Exceptional Courses of Events</b> | N/A  |

## 5. 결제 방법 중 하나를 선택한다

|                                      |  |
|--------------------------------------|--|
| <b>Use Case</b>                      | 5. 결제방법 중 하나를 선택한다.  |
| <b>Actor</b>                         | Customer   |
| <b>Type</b>                          | Evident  |
| <b>Cross Reference</b>               | Functions: R 2.4, R 2.5<br>Use cases : “카드 결제를 진행한다”, “인증코드 결제를 진행한다”  |
| <b>Pre-Requisites</b>                | - 음료수 선택이 완료되어야한다<br>- 선택한 음료수의 재고가 존재해야한다   |
| <b>Typical Courses of Events</b>     | (A): Actor, (S): System<br>1. (A) 카드결제는 0번 dialButton을, 코드결제는 1번 dialButton 임이 Screen에 표시되어, 사용자가 카드결제, 코드결제 중 하나를 선택한다.<br>2. (S) 카드결제를 선택할 경우 “카드결제를 진행한다” 를 invoke한다.<br>3. (S) 코드결제를 선택할 경우 “인증코드 결제를 진행한다”를 invoke한다. |
| <b>Alternative Courses of Events</b> | N/A  |
| <b>Exceptional Courses of Events</b> | N/A  |

## 6. 카드 결제를 진행한다

|                                      |  |
|--------------------------------------|--|
| <b>Use Case</b>                      | 6. 카드 결제를 진행한다.  |
| <b>Actor</b>                         | Customer   |
| <b>Type</b>                          | Evident  |
| <b>Cross Reference</b>               | Functions: R 2.5, R 2.7, R 5.1<br>Use cases : “결제를 취소한다.”, “선결제를 진행한다.”  |
| <b>Pre-Requisites</b>                | - Customer 가 결제 방법 중 카드 결제를 선택해야 한다.   |
| <b>Typical Courses of Events</b>     | (A): Actor, (S): System<br>1. (A) Customer 는 카드 결제를 위해, 카드를 카드 단말기에 꽂는다.<br>2. (S) Customer 가 카드단말기에 사용한 카드가 결제를 할 수 있는 종류의 카드인지 확인한다.<br>3. (S) 결제할 음료의 가격을 Screen을 통해, 전달받는다.<br>4. (S) 결제 단말기에 사용한 카드의 잔액이 음료의 가격을 결제할 수 있는지 확인한다.<br>5. (S) 카드 결제를 완료한다.<br>6. (S) 재고를 업데이트한다. |
| <b>Alternative Courses of Events</b> | N/A  |
| <b>Exceptional Courses of Events</b> | Line 2. 카드가 결제를 할 수 없는 종류의 카드인 경우, “결제를  |

|  |   |
|--|---|
|  | 취소한다.”를 invoke 한다.<br>Line 3. 카드의 잔액이 음료의 가격을 결제할 수 없을 경우, “결제를 취소한다”를 invoke 한다. |
|--|---|

## 7. 인증코드 결제를 진행한다

|                                      |   |
|--------------------------------------|---|
| <b>Use Case</b>                      | 7. 인증코드 결제를 진행한다.   |
| <b>Actor</b>                         | Customer  |
| <b>Type</b>                          | Evident   |
| <b>Cross Reference</b>               | Functions: R 2.5, R 2.6, R 6.1<br>Use cases : “결제를 취소한다.”, “입력한 인증코드가 유효한지 확인한다.”   |
| <b>Pre-Requisites</b>                | - Customer 가 결제 방법 중 인증코드 결제를 선택해야 한다.  |
| <b>Typical Courses of Events</b>     | (A): Actor, (S): System<br>1. (A) Customer 는 인증코드를 dialButton을 사용하여, 입력한 뒤, 확인 버튼을 누른다.<br>2. (S) 입력된 인증코드에 대해, “입력한 인증코드가 유효한지 확인한다”를 invoke한다.<br>3. (S) 인증코드가 일치하면 인증코드 결제를 완료하고 음료를 제공한다. |
| <b>Alternative Courses of Events</b> | N/A   |
| <b>Exceptional Courses of Events</b> | Line 3. 인증코드가 일치하지 않는다면 사용자에게 일치하지 않는다고 Screen에 알려준 후 초기 화면으로 돌아간다.   |

## 8. 결제를 취소한다

|                                      |  |
|--------------------------------------|--|
| <b>Use Case</b>                      | 8. 결제를 취소한다.   |
| <b>Actor</b>                         | System   |
| <b>Type</b>                          | Hidden   |
| <b>Cross Reference</b>               | Functions: R 2.6, R 2.4, R 2.5<br>Use cases : “카드 결제를 진행한다.”, “인증코드 결제를 진행한다.” |
| <b>Pre-Requisites</b>                | - 카드 결제, 인증코드 결제 과정 중 결제 취소가 발생한다.   |
| <b>Typical Courses of Events</b>     | (A): Actor, (S): System<br>1. (S) 초기화면으로 돌아가, Screen에 나타낸다.                    |
| <b>Alternative Courses of Events</b> | N/A  |
| <b>Exceptional Courses of Events</b> | N/A  |

## 9. 현재 자판기의 재고를 확인한다

|                                      |  |
|--------------------------------------|--|
| <b>Use Case</b>                      | 9. 현재 자판기의 재고를 확인한다.   |
| <b>Actor</b>                         | System   |
| <b>Type</b>                          | Hidden   |
| <b>Cross Reference</b>               | Functions: R 3.1, R 2.1<br>Use cases : “현재 자판기에서 판매하는 음료수를 선택한다.”  |
| <b>Pre-Requisites</b>                | - 현재 자판기에서 판매하는 음료를 선택하거나 다른 자판기에게 재고 상태 확인 요청을 받았을 때만 실행된다.   |
| <b>Typical Courses of Events</b>     | (A1): Actor, (A2): DVMs, (S): System<br>1. (A2) 현재 자판기의 재고를 시스템에게 알려준다.<br>2. (A2) 요청받은 음료의 재고가 없다면, 재고가 없다고 알려준다. |
| <b>Alternative Courses of Events</b> | 1(a). 만약 현재 자판기가 작동 불능상태라면 재고가 없다고 간주하고 재고 없음을 알려준다.   |
| <b>Exceptional Courses of Events</b> | N/A  |

## 10. 다른 자판기의 재고를 확인한다.

|                                      |   |
|--------------------------------------|---|
| <b>Use Case</b>                      | 10. 다른 자판기의 재고를 확인한다  |
| <b>Actor</b>                         | System  |
| <b>Type</b>                          | Hidden  |
| <b>Cross Reference</b>               | Functions: R 3,2, R 2.2<br>Use cases : “현재 자판기에서 판매하지 않는 음료수를 선택한다.”  |
| <b>Pre-Requisites</b>                | - 현재 자판기의 재고가 부족하거나, 현재 자판기에서 판매하지 않는 음료를 사용자가 선택할 경우에만 실행된다.<br>-음료가 선택된 상태여야한다.   |
| <b>Typical Courses of Events</b>     | (A): Controller, (S): System<br>1. (A) 다른 자판기들에게 해당음료의 재고정보를 요청한다.<br>2. (S) 재고가 없다는 응답을 받을 경우, 재고가 없다는 것을 사용자에게 알린 뒤 초기화면으로 돌아간다.<br>3. (S) 재고가 존재할 경우, 재고가 존재하는 DVM 리스트를 전달한다.. |
| <b>Alternative Courses of Events</b> | 2 (a). 만약 요청한 DVM의 상태가 Off 혹은 작동 불능일 경우, 재고가 없다고 간주한다.<br>2 (b). 만약 일정 시간 이상 응답이 없다면, 재고가 없다고 간주한다.   |
| <b>Exceptional Courses of Events</b> | N/A   |

## 11. 자판기의 위치를 보여준다.

|                                      |   |
|--------------------------------------|---|
| <b>Use Case</b>                      | 11. 자판기의 위치를 보여준다.  |
| <b>Actor</b>                         | System  |
| <b>Type</b>                          | Hidden  |
| <b>Cross Reference</b>               | Functions: R 4.1,R 5.1<br>Use cases : “선결제를 진행한다.”  |
| <b>Pre-Requisites</b>                | - Customer 가 선결제를 완료한 후 실행되어야 한다.   |
| <b>Typical Courses of Events</b>     | (A): Actor, (S): System<br>1. (S) 사용자가 선택한 음료를 판매하는 자판기들의 위치를 사용자에게 Screen을 통해,보여준다.<br>2. (S) 사용자가 선택한 음료를 판매하지 않는 자판기들의 위치는 표시하지 않는다. |
| <b>Alternative Courses of Events</b> | N/A   |
| <b>Exceptional Courses of Events</b> | N/A   |

## 12. 선결제를 진행한다.

|                                      |  |
|--------------------------------------|--|
| <b>Use Case</b>                      | 12. 선결제를 진행한다.   |
| <b>Actor</b>                         | System   |
| <b>Type</b>                          | Hidden   |
| <b>Cross Reference</b>               | Functions: R 5.1, R 2.4<br>Use cases : “카드 결제를 진행한다.”                                      |
| <b>Pre-Requisites</b>                | - 현재 자판기에서 판매하지 않는 음료 또는 현재 자판기에 재고가 없는 음료가 선택되어야 한다.                                      |
| <b>Typical Courses of Events</b>     | (A): Actor, (S): System<br>1. (S) 카드 결제를 진행한다.<br>2. (S) 인증코드를 생성해, Screen을 통해 사용자에게 전달한다. |
| <b>Alternative Courses of Events</b> | N/A  |
| <b>Exceptional Courses of Events</b> | N/A  |

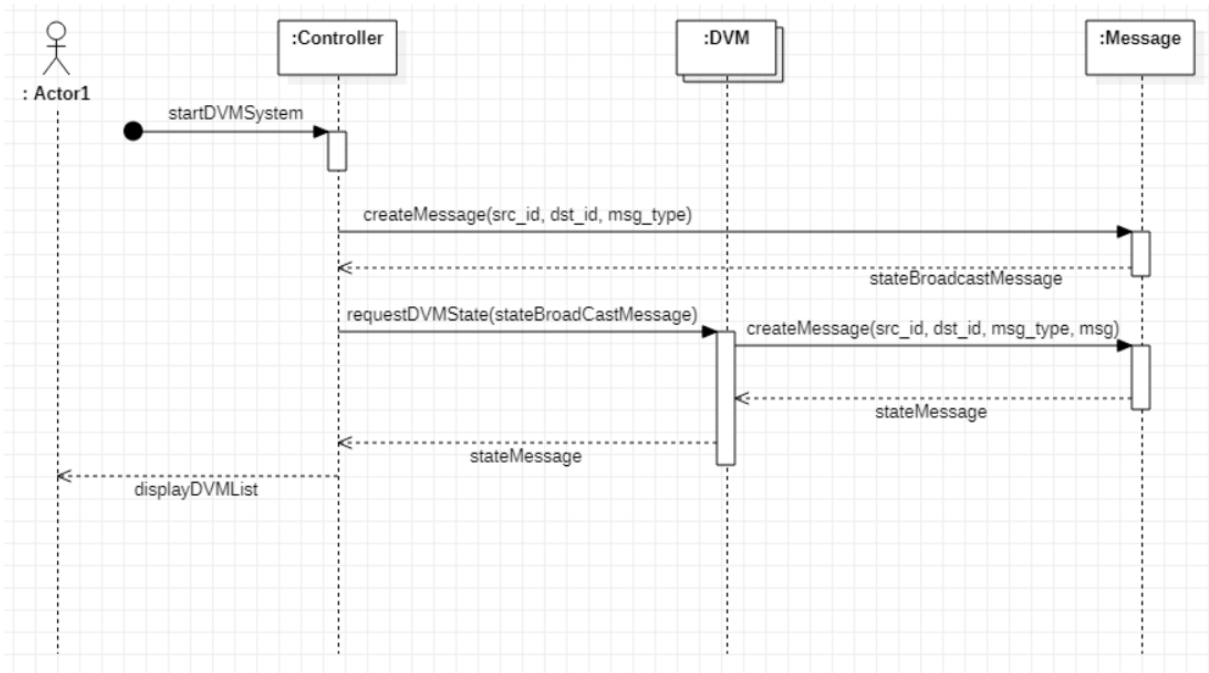
## 13. 입력한 인증코드가 유효한지 확인한다.



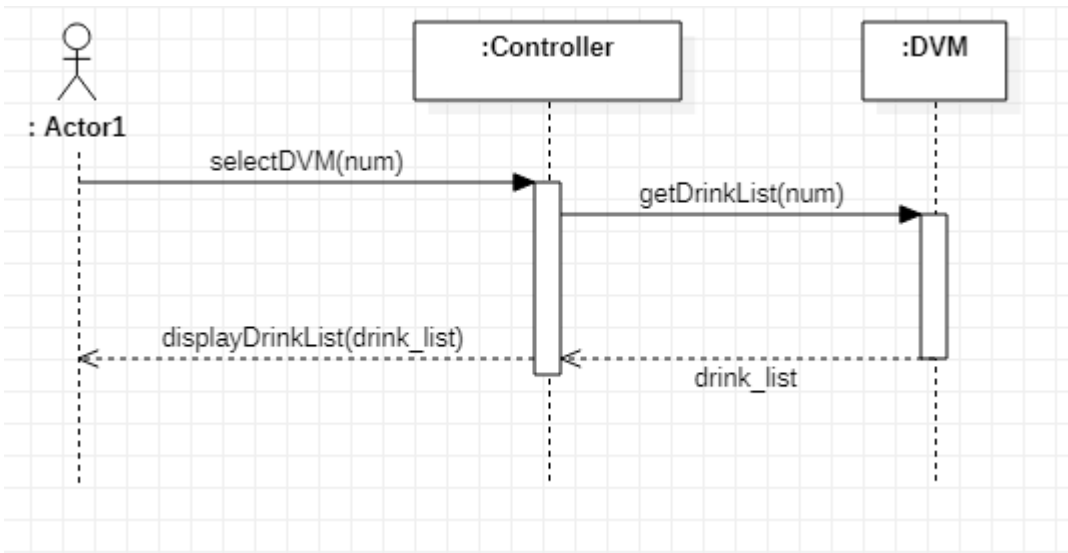
|                                      |   |
|--------------------------------------|---|
| <b>Use Case</b>                      | 13. 입력한 인증코드가 유효한지 확인한다.  |
| <b>Actor</b>                         | System  |
| <b>Type</b>                          | Hidden  |
| <b>Cross Reference</b>               | Functions: R 6.1,R 2.5<br>Use cases : "인증코드 결제를 진행한다."  |
| <b>Pre-Requisites</b>                | - Customer가 선결제에서 발급받은 인증코드를 입력한다.  |
| <b>Typical Courses of Events</b>     | (A): Actor, (S): System<br>1. (S) 사용자가 dialButton을 사용하여 입력한 인증코드와 발급된 인증코드가 일치하지 않는다면, 일치하지 않는 인증코드라고 Screen을 통해 사용자에게 알려준다.<br>2. (S) 사용자가 입력한 인증코드와 발급한 인증코드가 일치하다면 일치하는 인증코드라고 알려준다. |
| <b>Alternative Courses of Events</b> | 1(a). 만약 인증코드를 발급한 내역이 없는 상태에서 사용자가 인증코드를 입력한다면 일치하지 않는 인증코드로 간주한다.   |
| <b>Exceptional Courses of Events</b> | N/A   |

## Activity 2143. Define Interaction Diagram

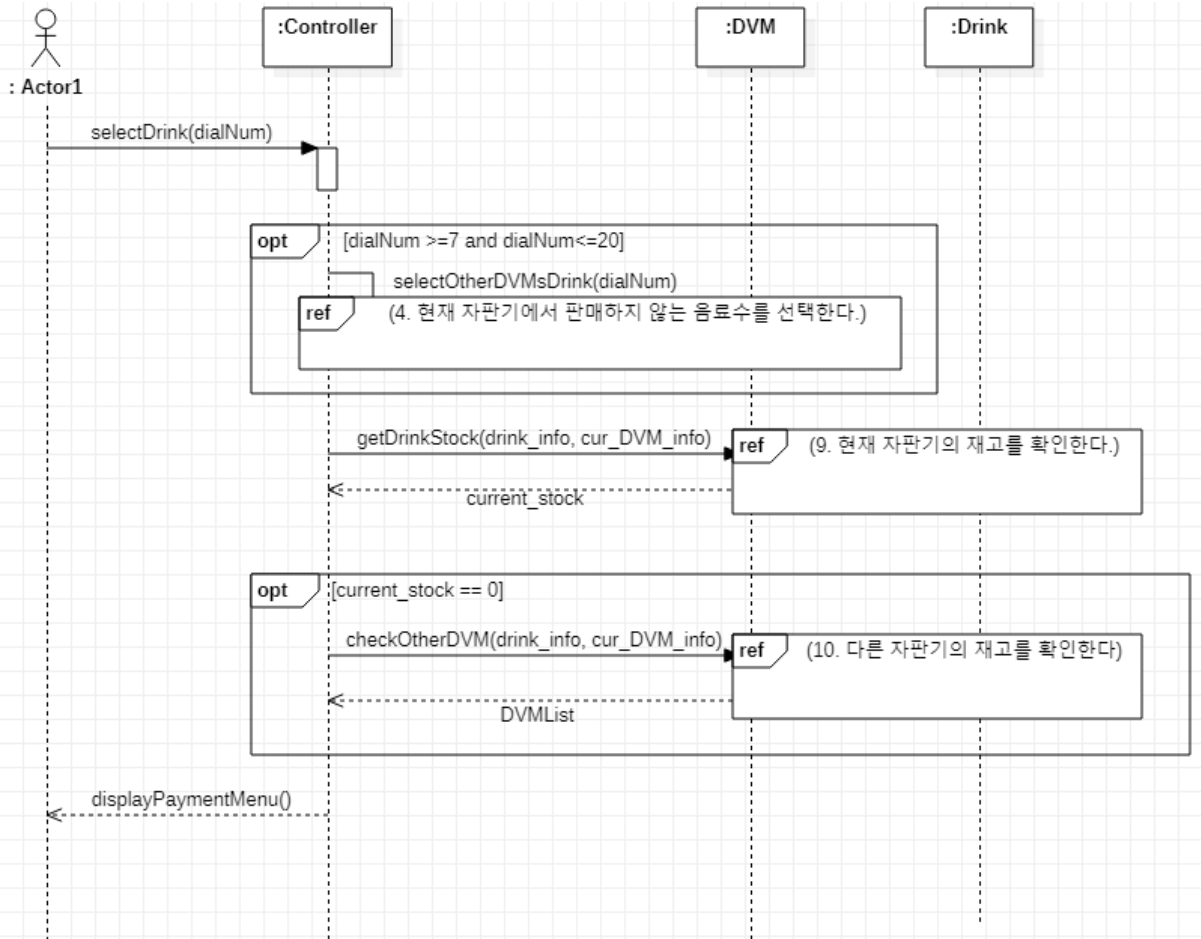
## 1. DVM들을 보여준다



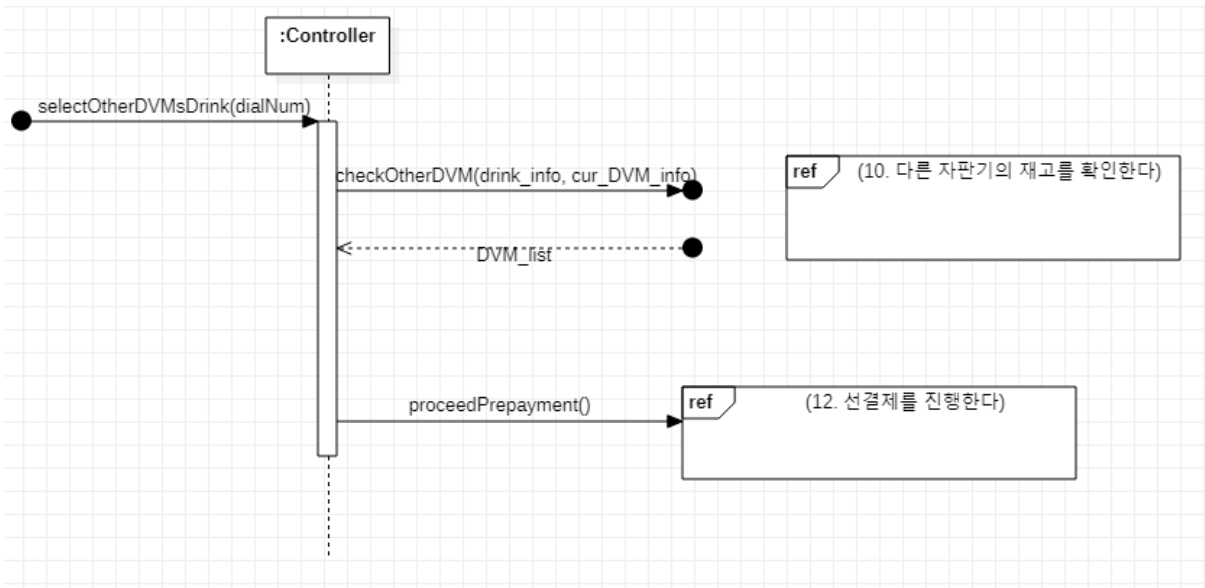
## 2. 음료 선택을 진행한다.



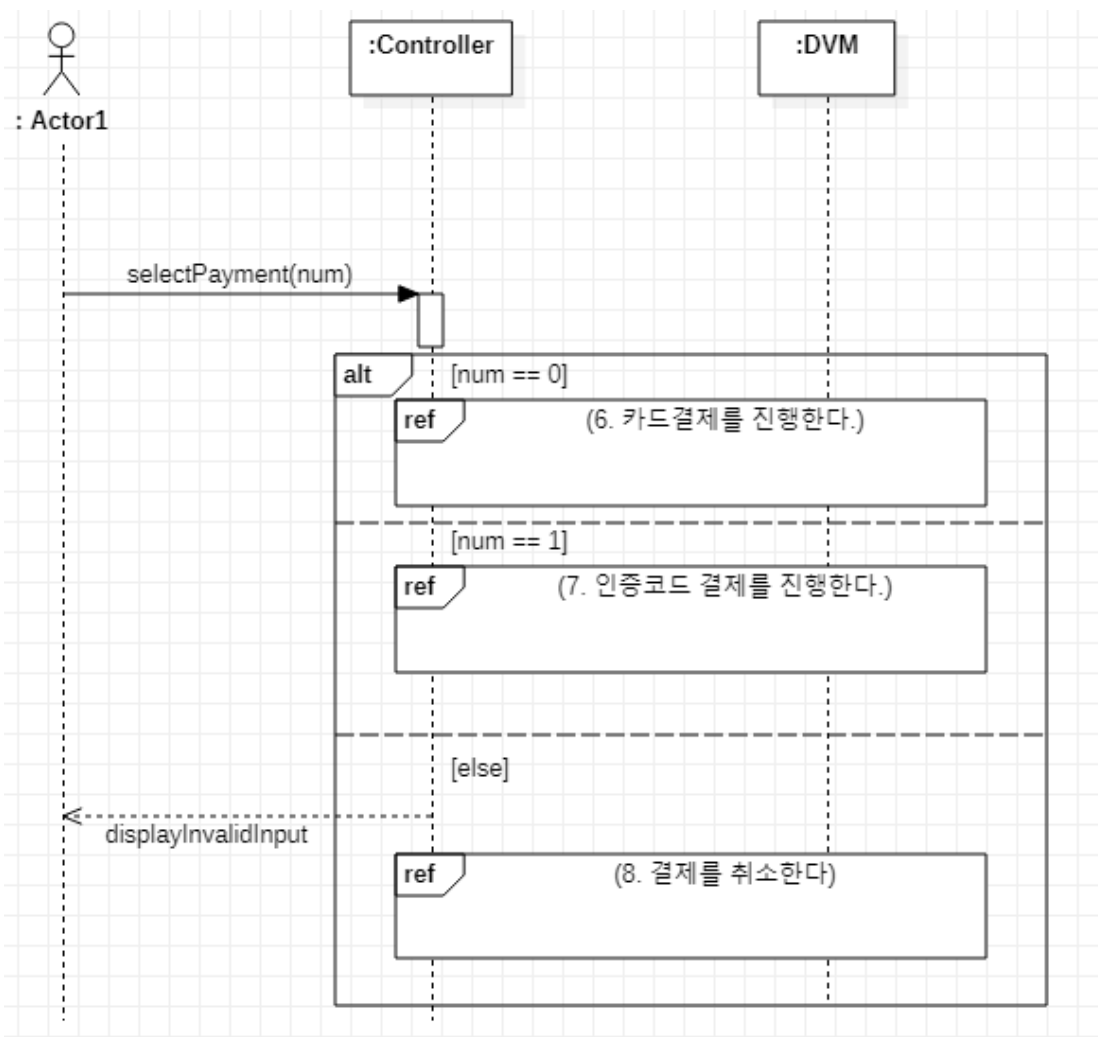
### 3. 현재 자판기에 판매하는 음료수를 선택한다.



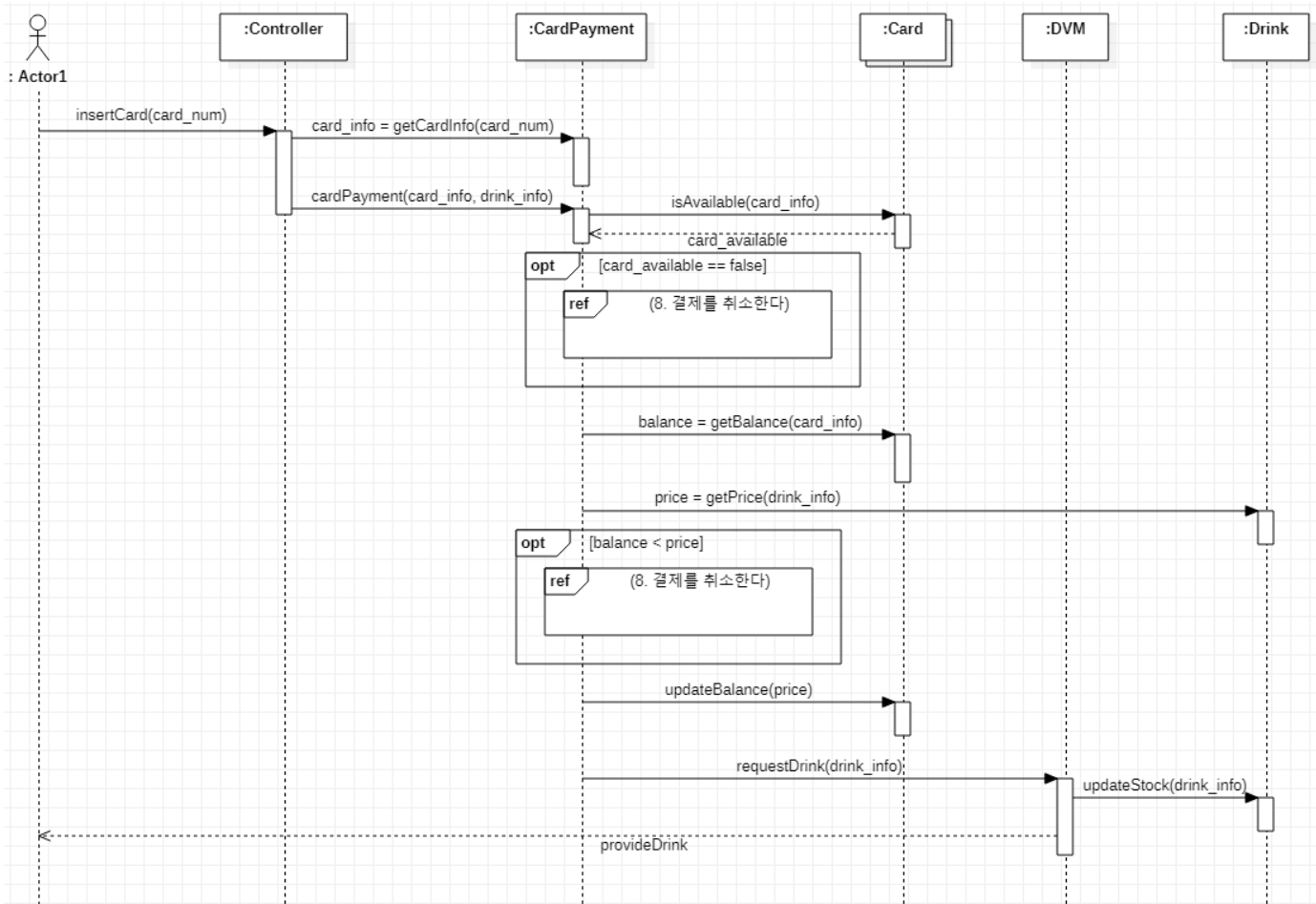
### 4. 현재 자판기에서 판매하지 않는 음료수를 선택한다.



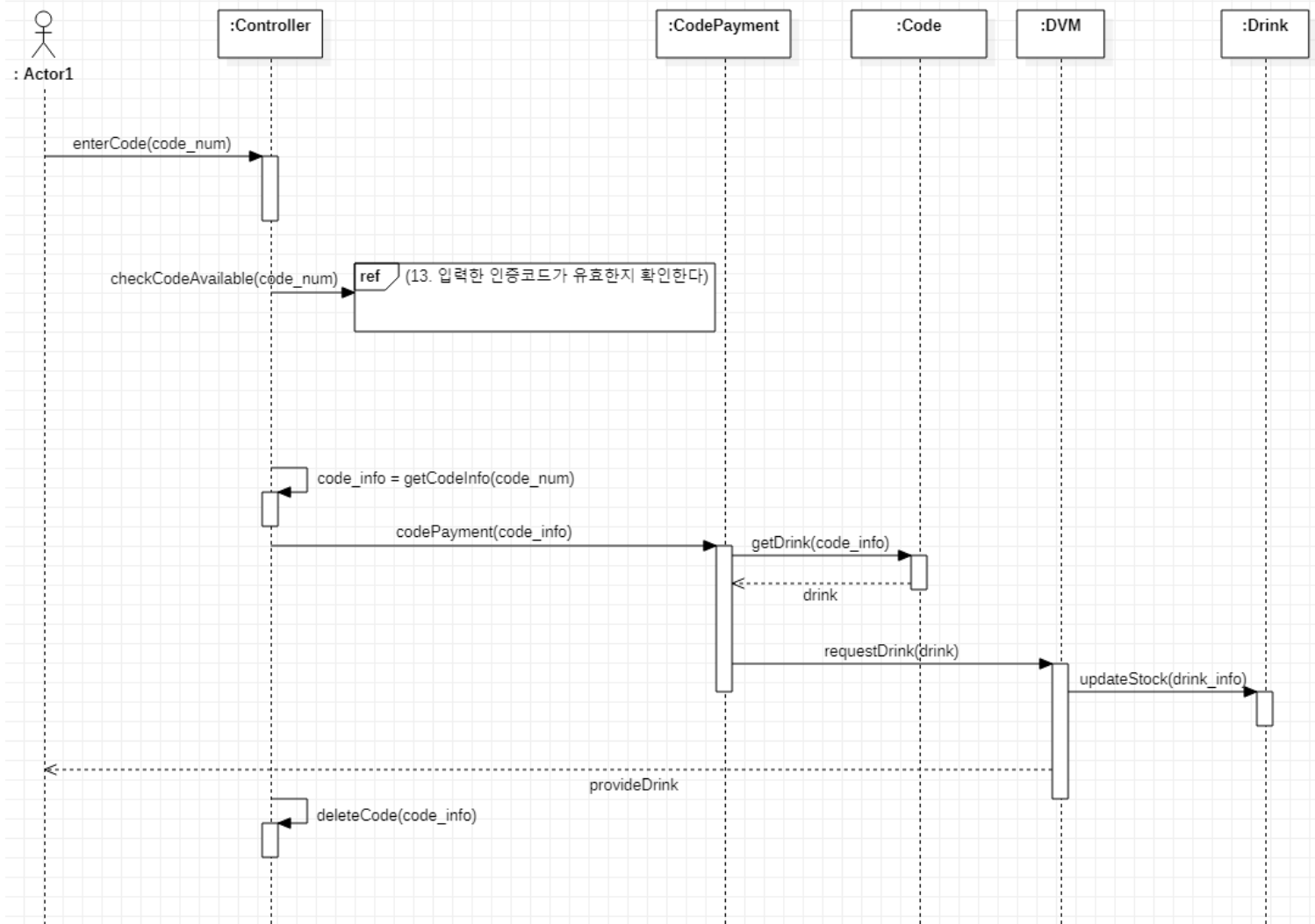
### 5. 결제 방법 중 하나를 선택한다.



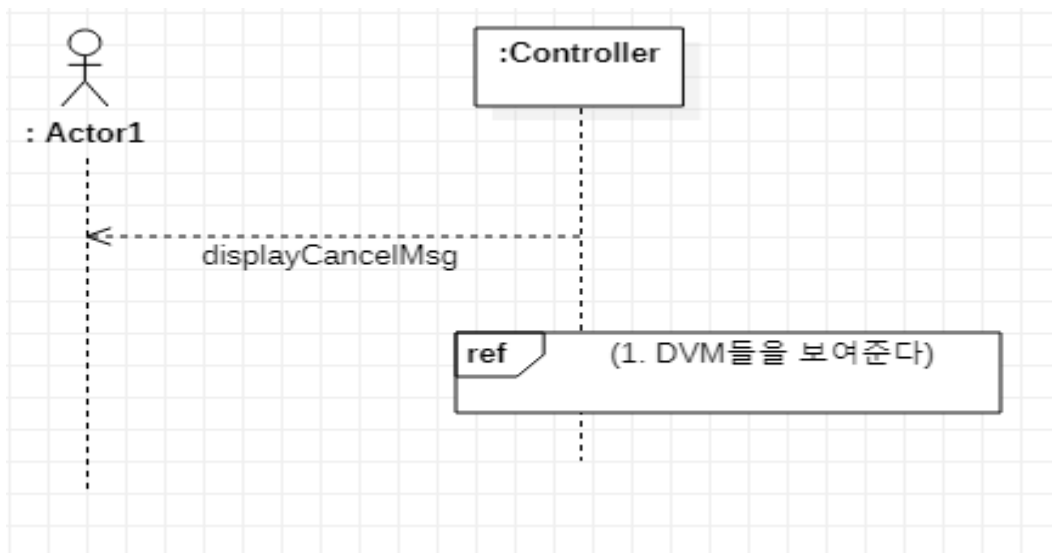
## 6. 카드 결제를 진행한다.



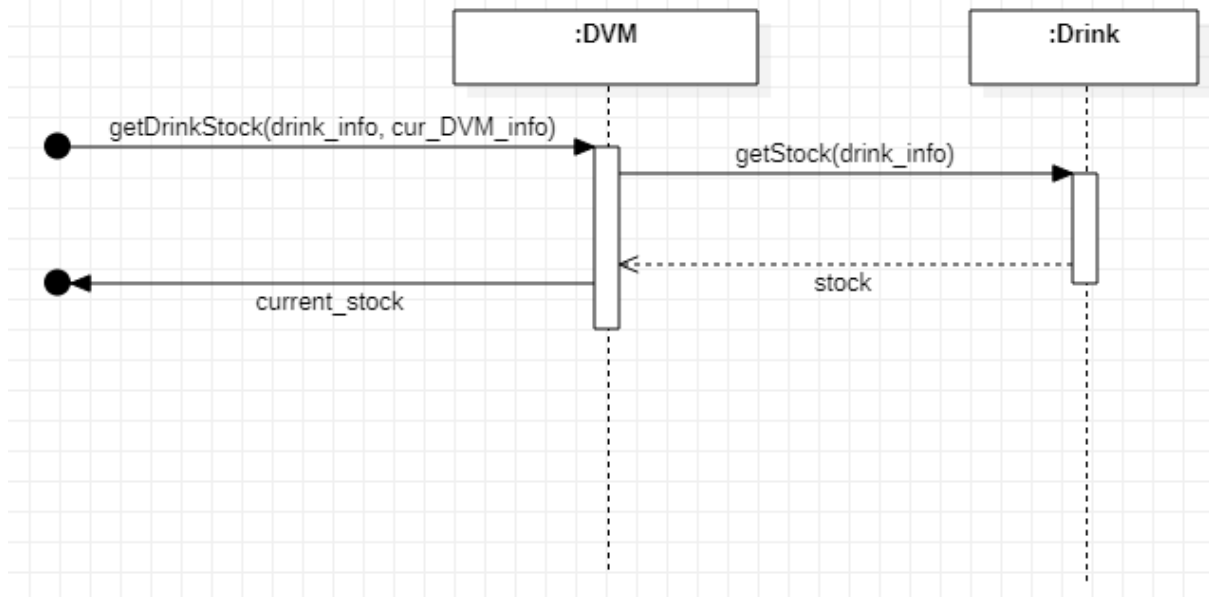
## 7. 인증코드 결제를 진행한다.



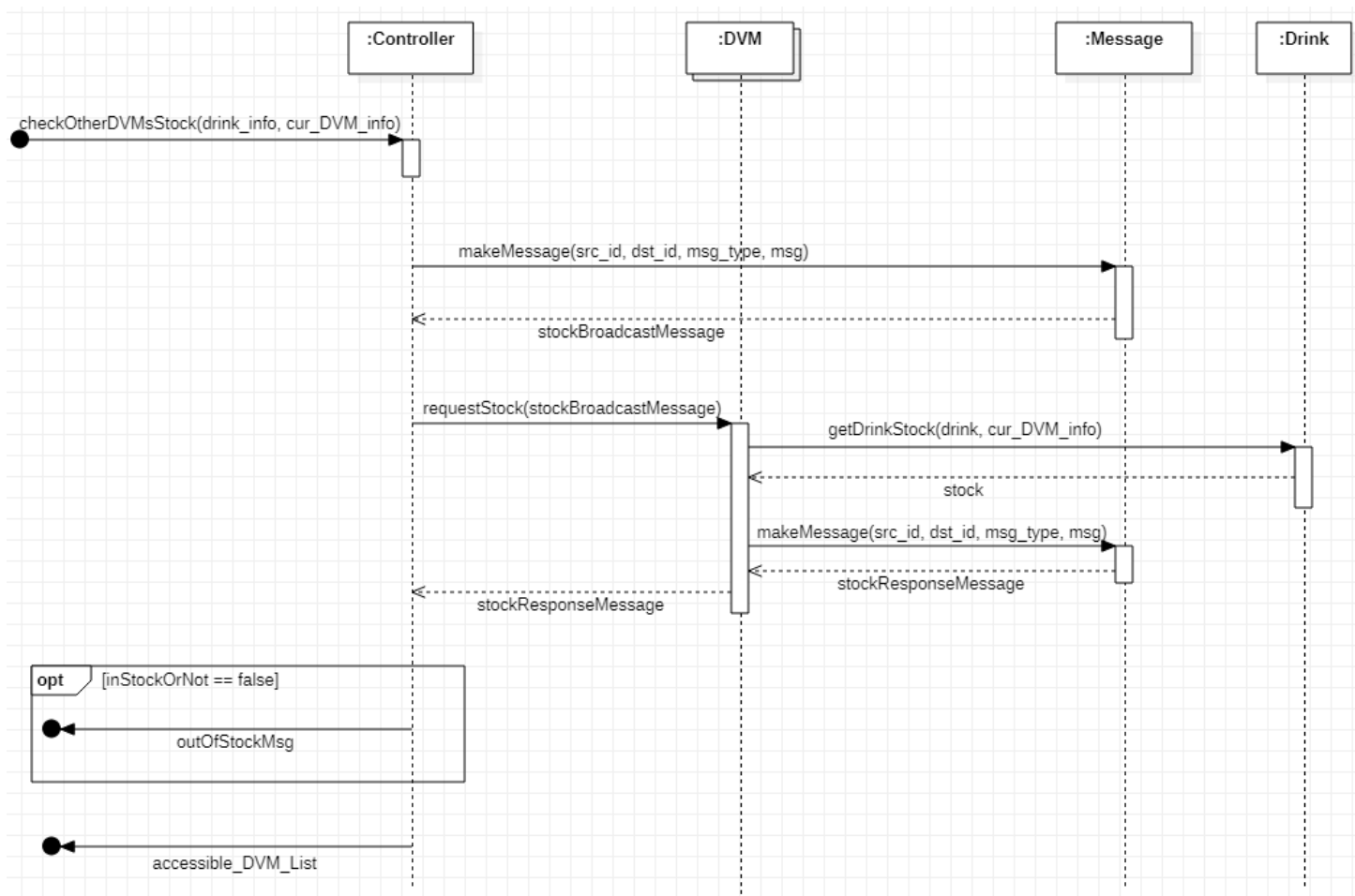
## 8. 결제를 취소한다.



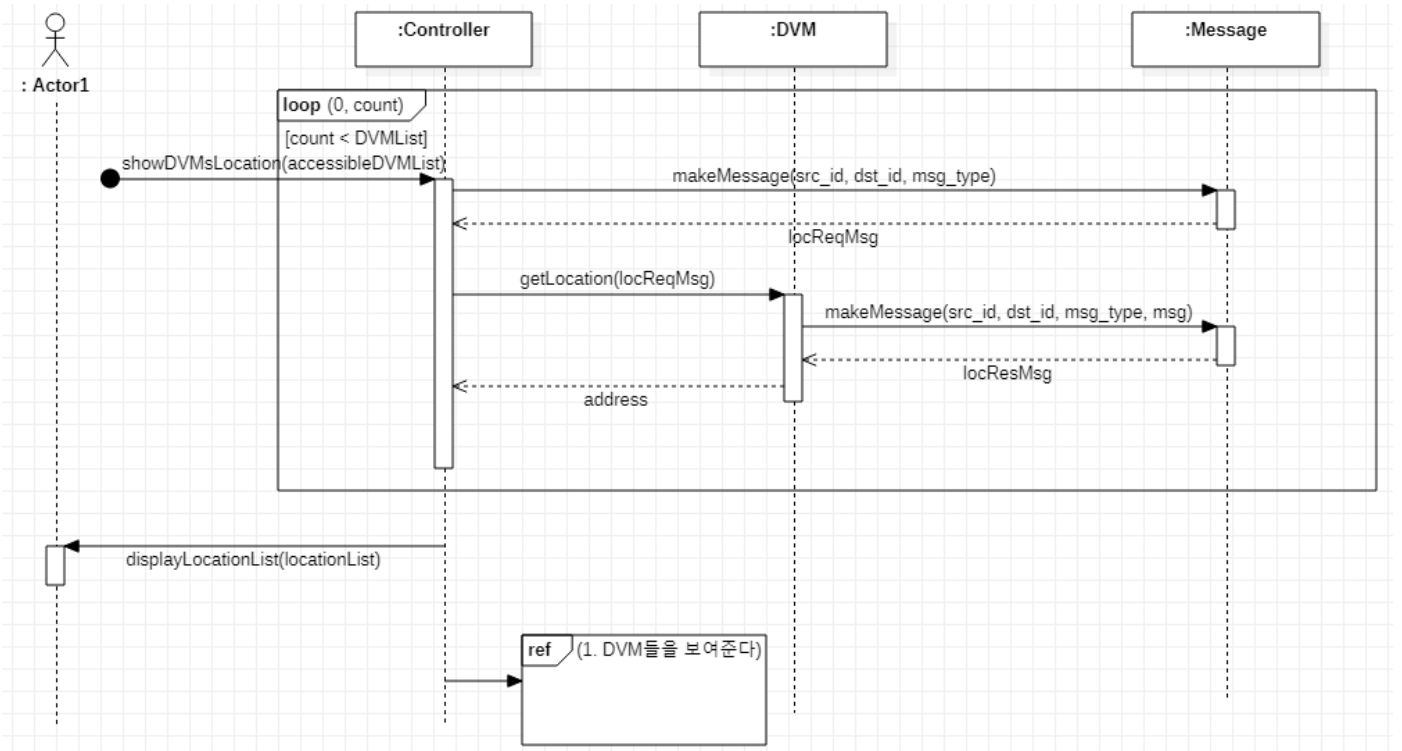
## 9. 현재 자판기의 재고를 확인한다.



## 10. 다른 자판기의 재고를 확인한다.

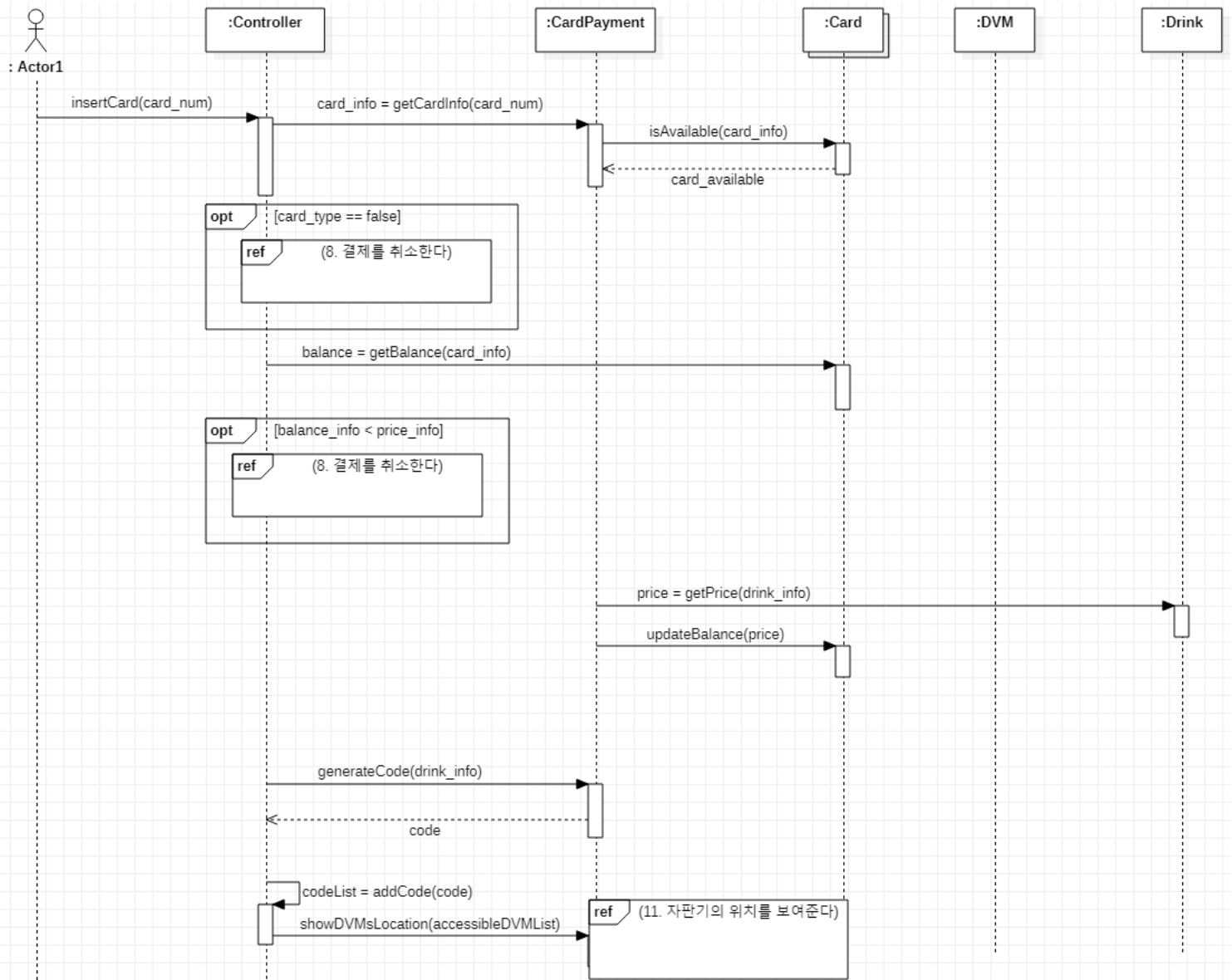


## 11. 자판기의 위치를 보여준다.

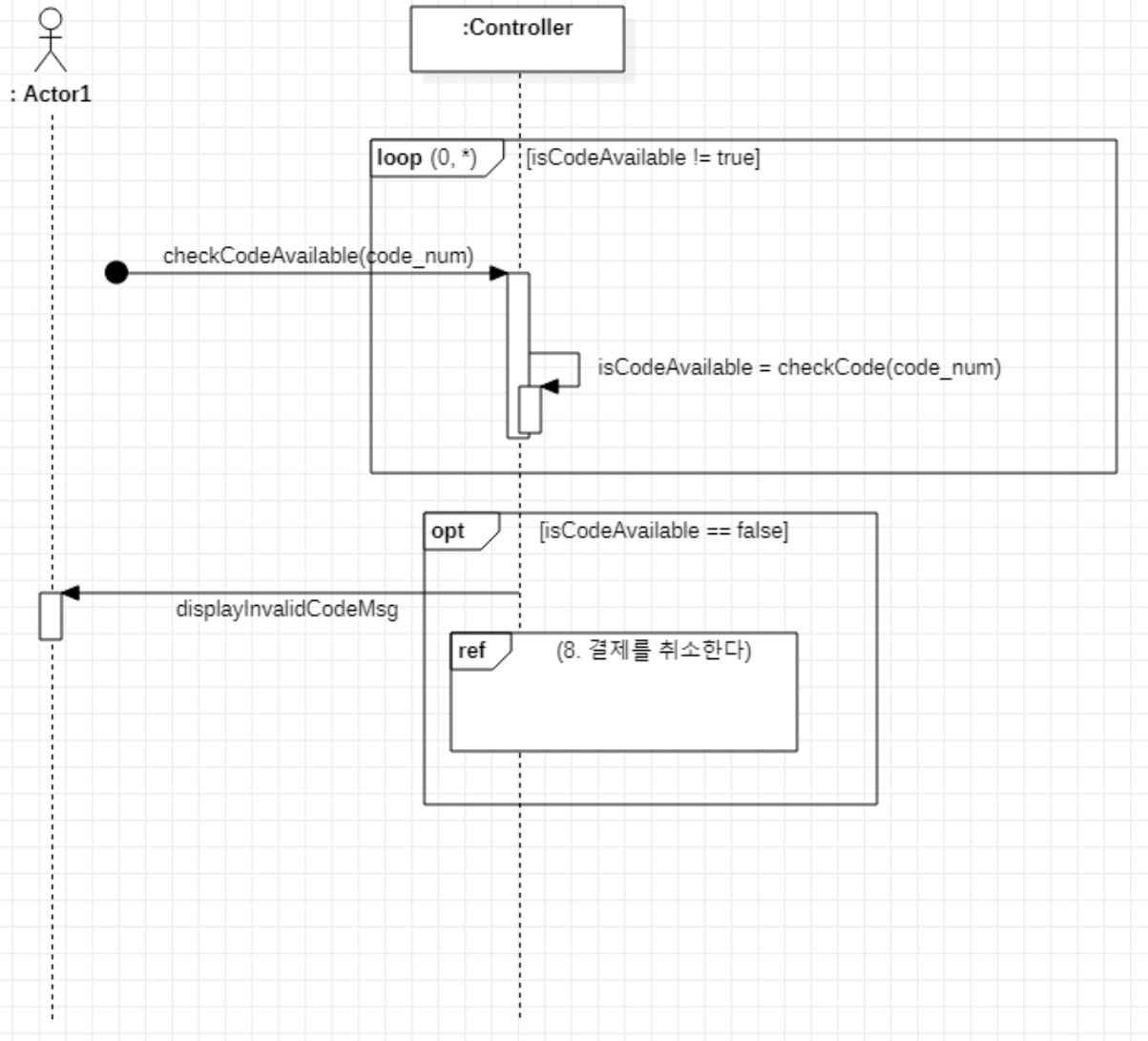




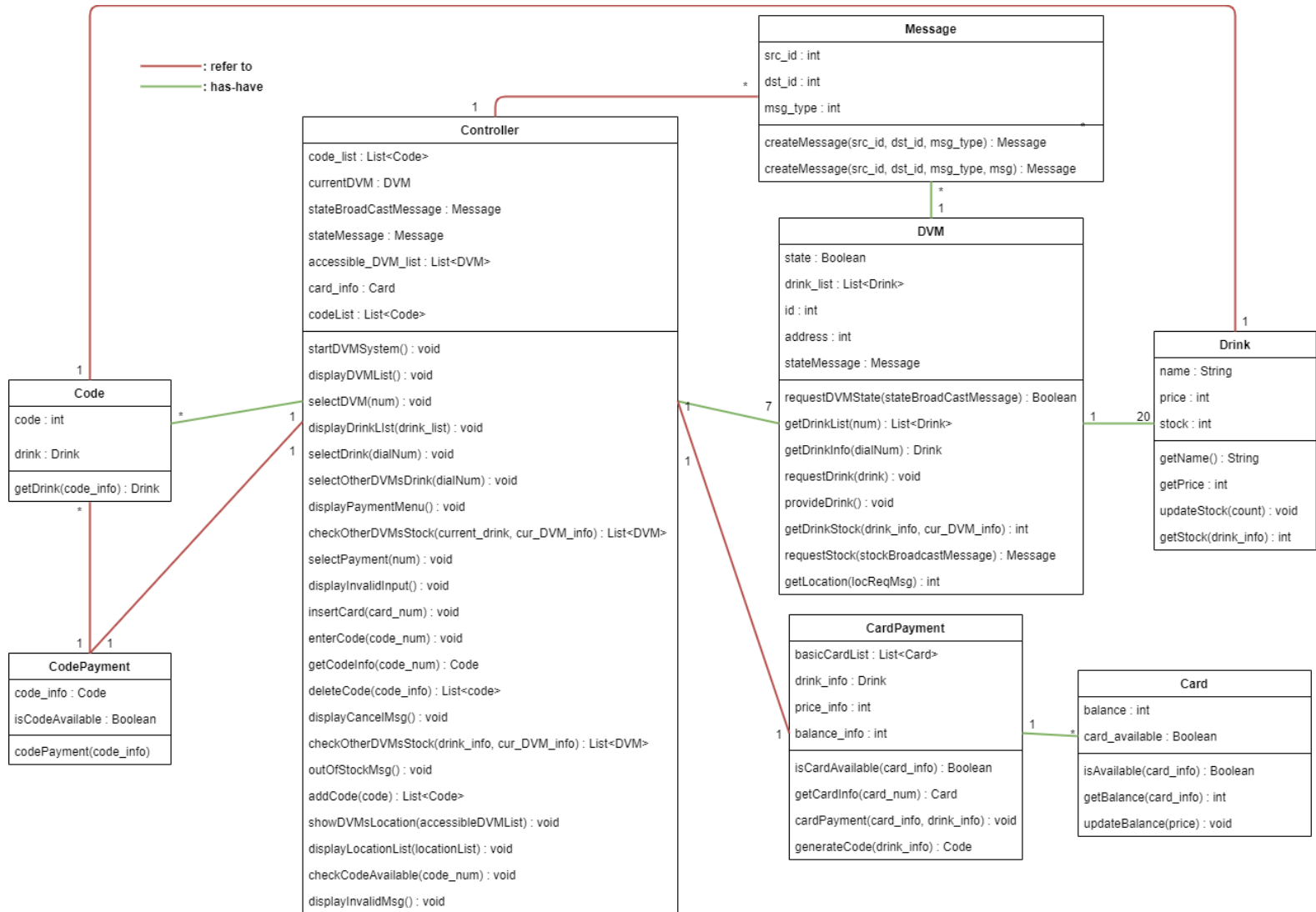
## 12. 선결제를 진행한다.



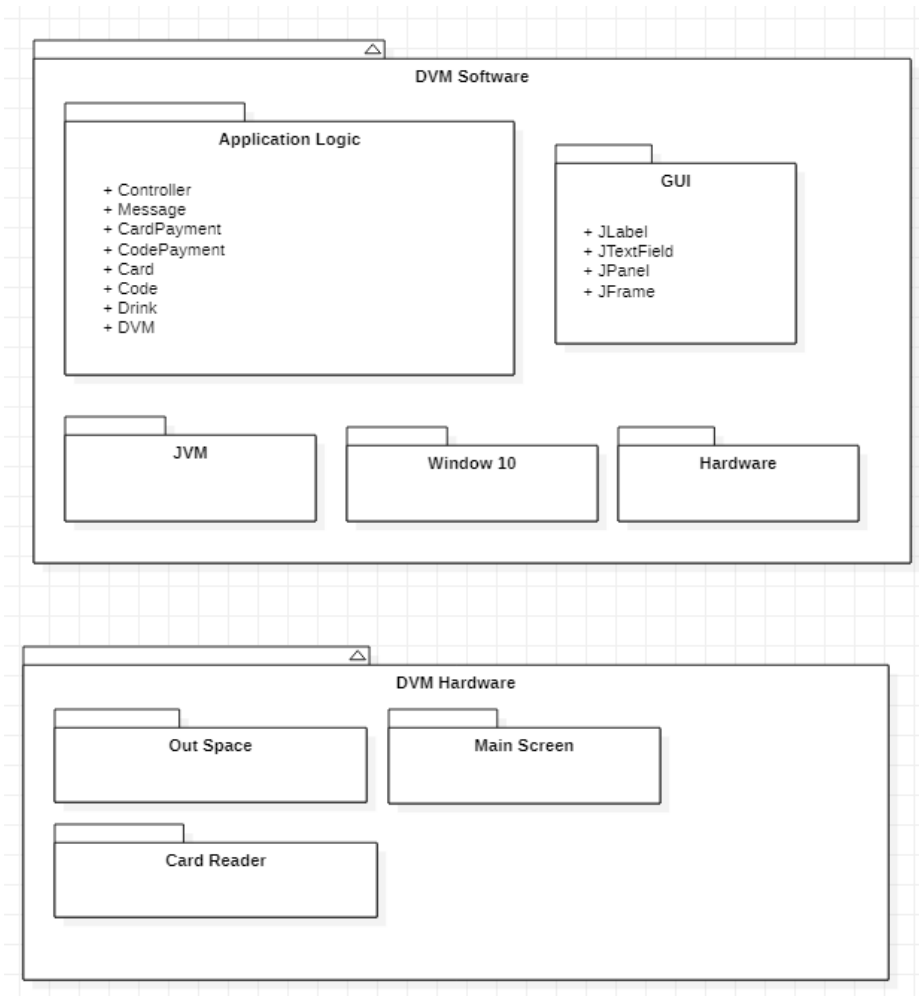
### 13. 입력한 인증코드가 유효한지 확인한다.



# Activity 2144. Define Design Class Diagrams



## Activity 2145. Refine System Architecture



# Activity 2147. Perform 2040 Traceability Analysis

| Essential Use-case         | Operation            | Method   | Class       |
|----------------------------|----------------------|--|-------------|
| DVM들을 보여준다                 | 1: Response My State | void startDVMSystem()                                      | Controller  |
| 음료 선택을 진행한다                | 2: Select DVM        | void displayDVMList()                                      |             |
| 현재 자판기에서 판매하는 음료수들 선택한다    | 3: Select Drink      | void selectDVM(num)  |             |
| 현재 자판기에서 판매하지 않는 음료수들 선택한다 | 4: Select Payment    | void displayDrinkList(drink_list)                          |             |
| 결제 방법 중 하나를 선택한다           | 5: Insert Card       | void selectDrink(dialNum)                                  |             |
| 카드 결제를 진행한다                | 6: Enter Code        | void selectOtherDVMsDrink(dialNum)                         |             |
| 인증코드 결제를 진행한다              | 7: Get Location      | List<DVM> checkOtherDVMsStock(current_drink, cur_DVM_info) |             |
| 자판기의 위치를 보여준다              |                      | void displayPaymentMenu()                                  |             |
|                            |                      | void proceedPrepayment()                                   |             |
|                            |                      | void selectPaument(num)                                    |             |
|                            |                      | void displayInvalidInput()                                 |             |
|                            |                      | void insertCard(card_num)                                  |             |
|                            |                      | void enterCode(code_num)                                   |             |
|                            |                      | Code getCodeInfo(code_num)                                 |             |
|                            |                      | List<Code> deleteCode(code_info)                           |             |
|                            |                      | void outOfStockMsg()                                       |             |
|                            |                      | List<Code> addCode(code)                                   |             |
|                            |                      | void showDVMsLocation(accessibleDVMList)                   |             |
|                            |                      | void displayLocationList(locationList)                     |             |
|                            |                      | void checkCodeAvallible(code_num)                          |             |
|                            |                      | void displayInvalidMsg()                                   |             |
|                            |                      | Boolean requestDVMState(stateBroadCastMessage)             |             |
|                            |                      | List<Drink> getDrinkList(num)                              |             |
|                            |                      | Drink getDrinkInfo(dialNum)                                |             |
|                            |                      | void requestDrink(drink)                                   |             |
|                            |                      | void provideDrink()  |             |
|                            |                      | int getDrinkStock(drink_info, cur_DVM_info)                |             |
|                            |                      | Message requestStock(stockBroadcastMessage)                |             |
|                            |                      | int getLocation(locReqMsg)                                 |             |
|                            |                      | Message createMessage(src_id, dst_id, msg_type)            | Message     |
|                            |                      | Message createMessage(src_id, dst_id, msg_type, msg)       |             |
|                            |                      | Boolean isCardAvallible(card_info)                         | CardPayment |
|                            |                      | Card getCardInfo(card_num)                                 |             |
|                            |                      | void cardPayment(card_info, drink_info)                    |             |
|                            |                      | Code generateCode(drink_info)                              | Card        |
|                            |                      | Boolean isAvallible(card_info)                             |             |
|                            |                      | int getBalance(card_info)                                  |             |
|                            |                      | void updateBalance(price)                                  | Drink       |
|                            |                      | String getName()   |             |
|                            |                      | int getPrice()   |             |
|                            |                      | void updateStock(count)                                    | Code        |
|                            |                      | int getStock(drink_info)                                   |             |
|                            |                      | Drink getDrink(code_info)                                  | CodePayment |
|                            |                      | codePayment(code_info)                                     |             |