



WWW.MIRICOMPANY.COM



편의점 POS 기능 구현



팀 이름 : Team 12

팀원 : 박정현 박현진 서은서 최수아 황인영

Level L1



Contents

01

02

03

서론

본론

결론

주제
프로젝트의 목표
설계 범위 지정

POS기 설계
POS기의 기능

시작화면
기능별 실행

01 서론

"편의점 POS기의 기능 구현"

손님



편의점



01 서론

판매

환불

상품



재고

회계

01 서론



01 서론

판매

편의점

CU



환불

재고

변동!

회계

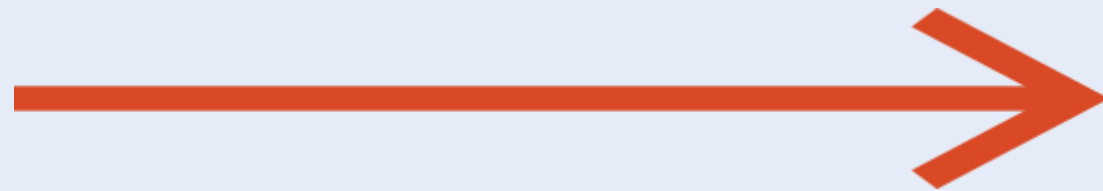
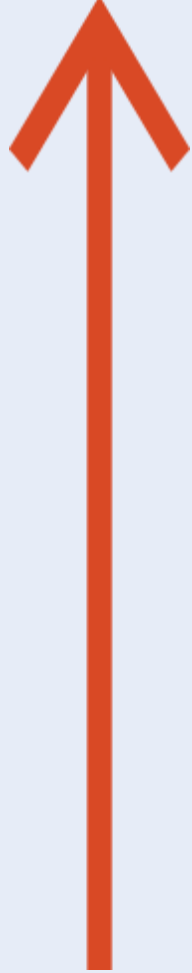
01 서론

판매

환불

재고

회계



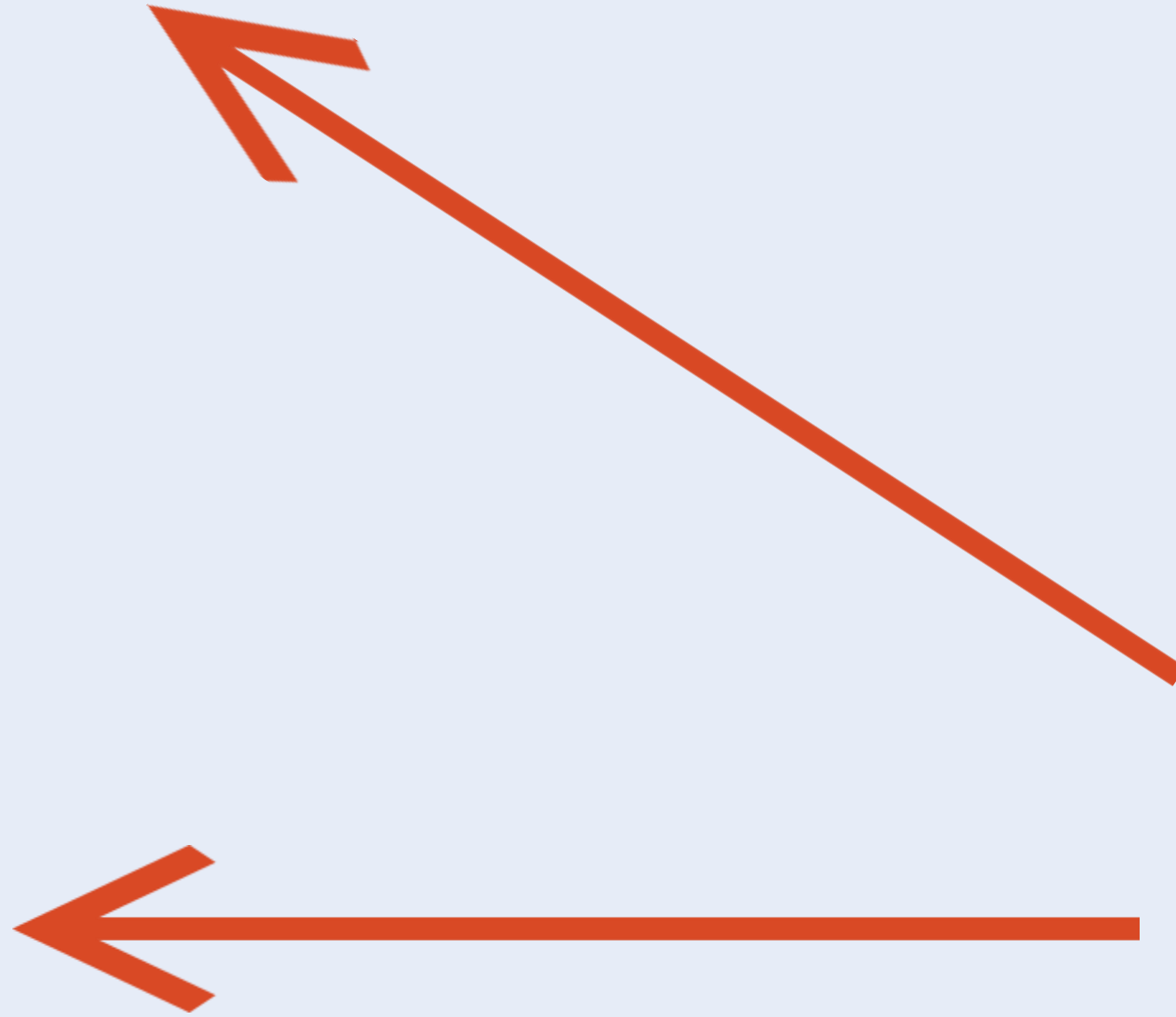
01 서론

판매

환불

재고

회계



01 서론

2) 프로젝트 목표

실제 편의점의 POS에서의 기능을 OOP 개념을 이용하여 cpp로 구현

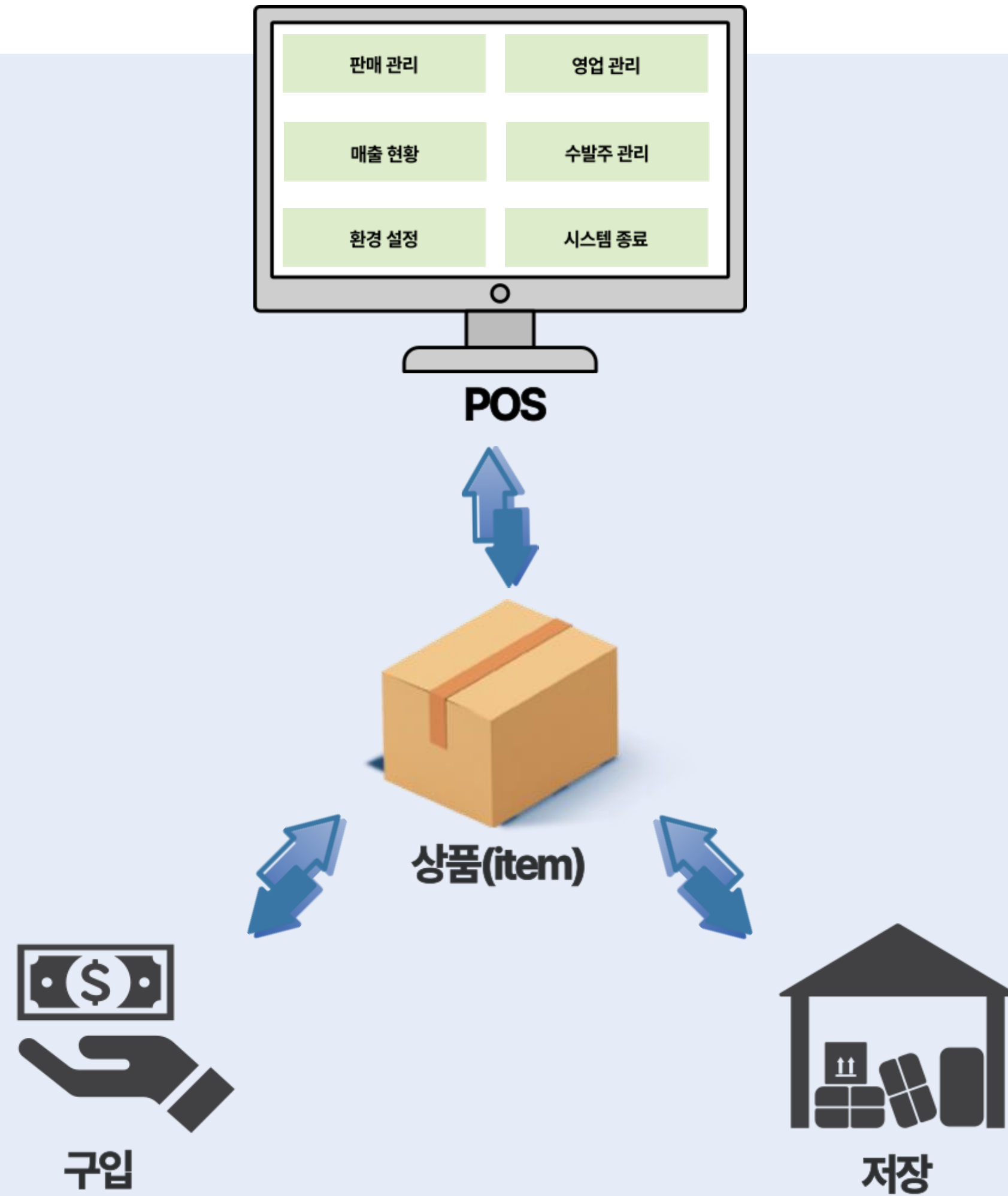
3) 설계 범위 지정

판매 관리 + 영업 관리(마감 정산, 준비금 등록, 수발주 관리)

1) POS기의 설계

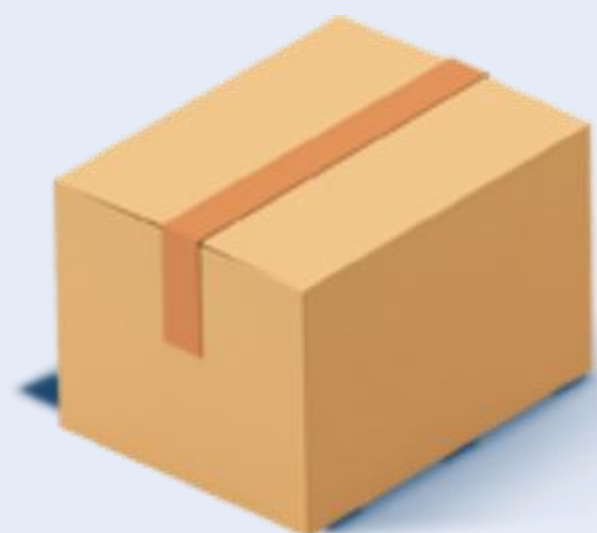
Q1) How to "기능" 분리?

02 본론



02 본론

상품 class의 추상화



상품(item)

유통기한이 있는 상품

ex) 과일, 과자, 유제품 등...

유통기한이 없는 상품

ex) 주류, 생필품 등...



공통된 속성 & 행위

- 상품명(name)
- 가격(price)
- 총 개수(total_count)
- 19세 미만 판매 가능여부(is_over_19)
- 폐기 처분

고유 속성

identifier_array(폐기 처분을 위한 날짜를 담음)

- 유통기한을 담은 array
- 입고날짜를 담은 array

02 본론

상품 class의 추상화

abstract_item class ◀

<i>abstract_item</i> { <i>abstract</i> }
name : string
price : int
total_count : int
is_over_19 : bool
identifier_array[int] : list
+ (virtual) get_name() : string
+ (virtual) get_price() : int
+ (virtual) get_total_count() : int
+ (virtual) get_is_over_19() : bool
+ (virtual) get_identifier_list() : list
+ (virtual) set_name(string) : void
+ (virtual) set_price(int) : void
+ (virtual) set_total_count(int) : void
+ (virtual) set_is_over_19(char) : void
+ (virtual) set_identifier_array(list<int> &) : void
+ (virtual) update_total_count(int): void
+ (virtual) update_total_count(): void
+ (virtual) item_disposal(): void
+ (virtual) item_pop(): void

공통된 property

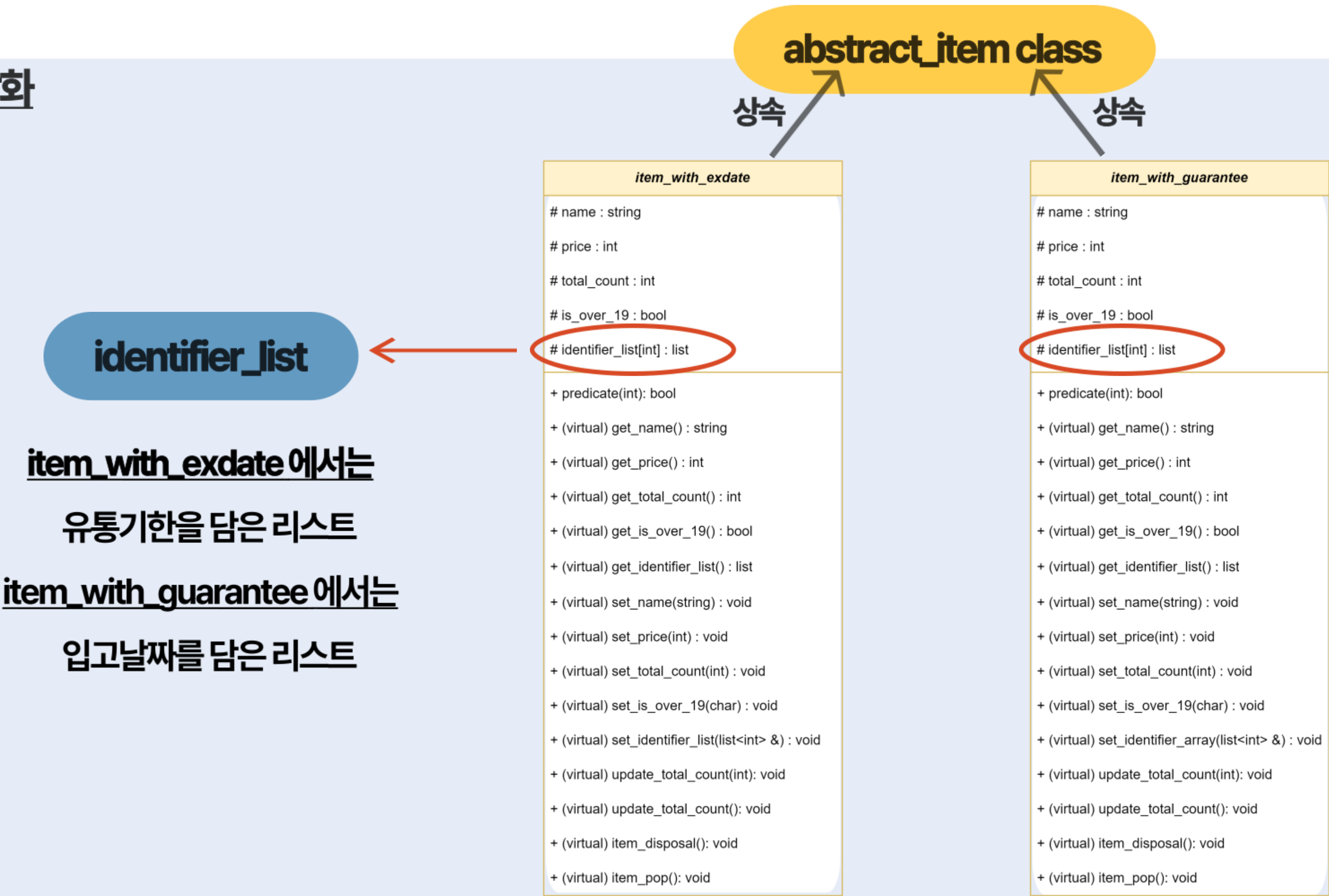
→ 다른 property

공통된 method

→ overriding의 적용

02 본론

상품 class의 추상화

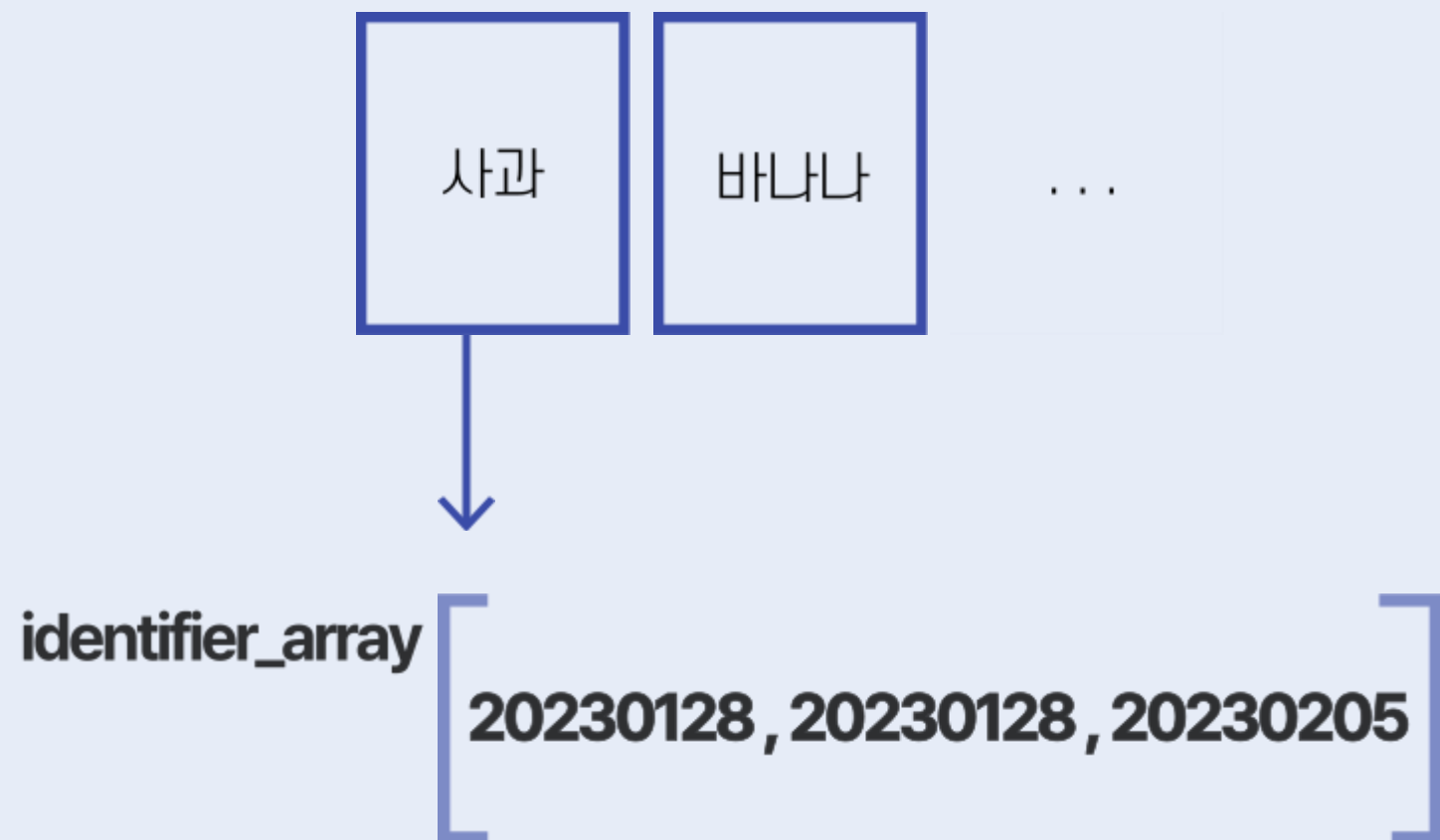


Q2)How to "상품 구조 설계"?

02 본론

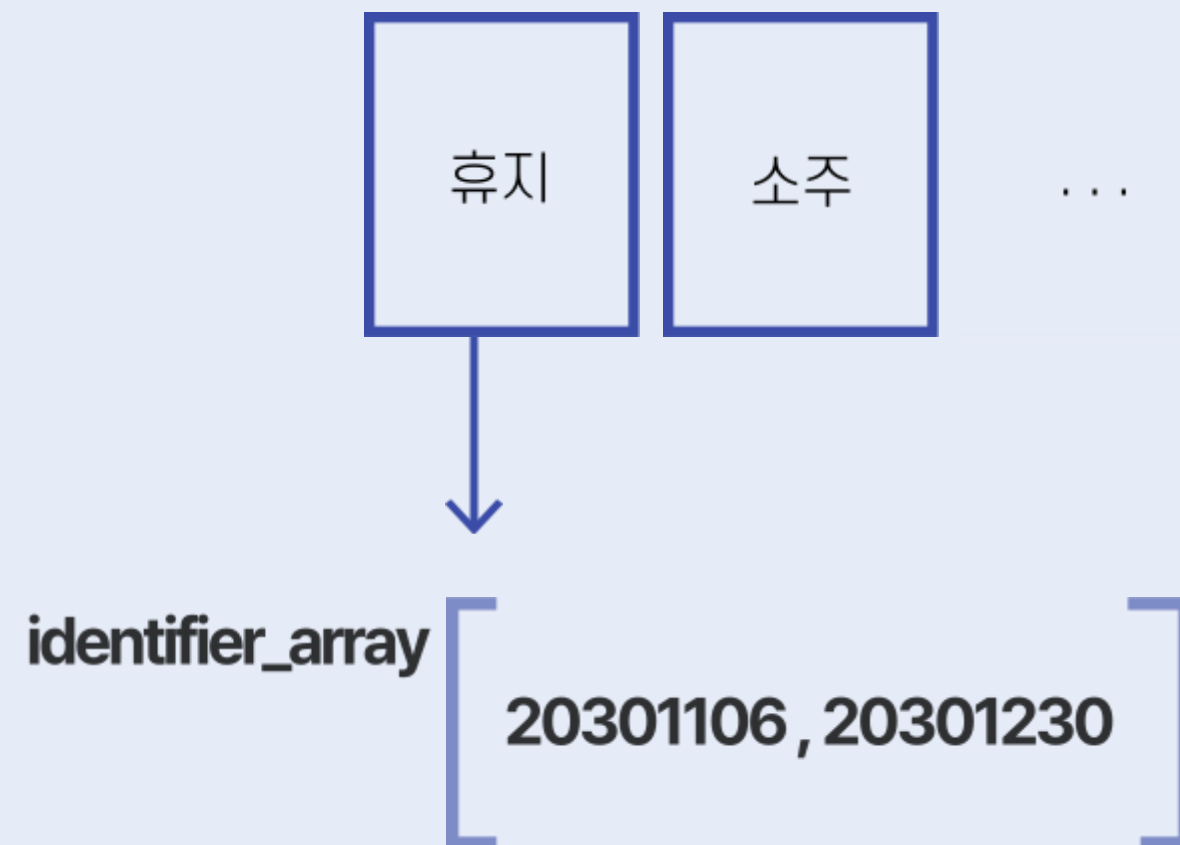
상품의 관리

유통기한이 있는 상품 (item_with_exdate)



* 사과, 바나나는 item_with_exdate로 찍어낸 객체

유통기한이 없는 상품 (item_with_guarantee)

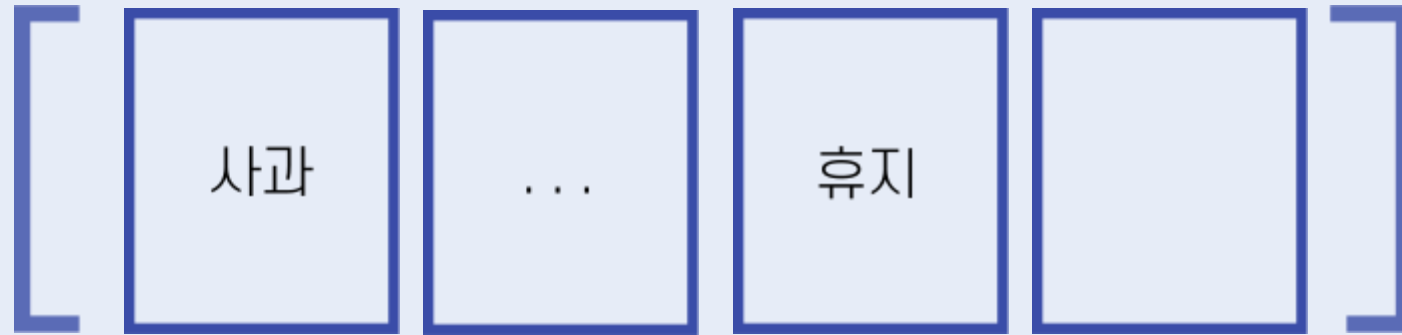


* 휴지, 소주는 item_with_guarantee로 찍어낸 객체

02 본론

상품의 저장

item_list



type: item_with_exdate != item_with_gaurantee



**abstract_item 포인터형을 이용
→ 두 종류의 아이템 서브클래스 접근가능**

Q3)How to "상품 구입"?

02 본론

상품의 구입

손님



한번에 한 (상품명, 개수)을 입력

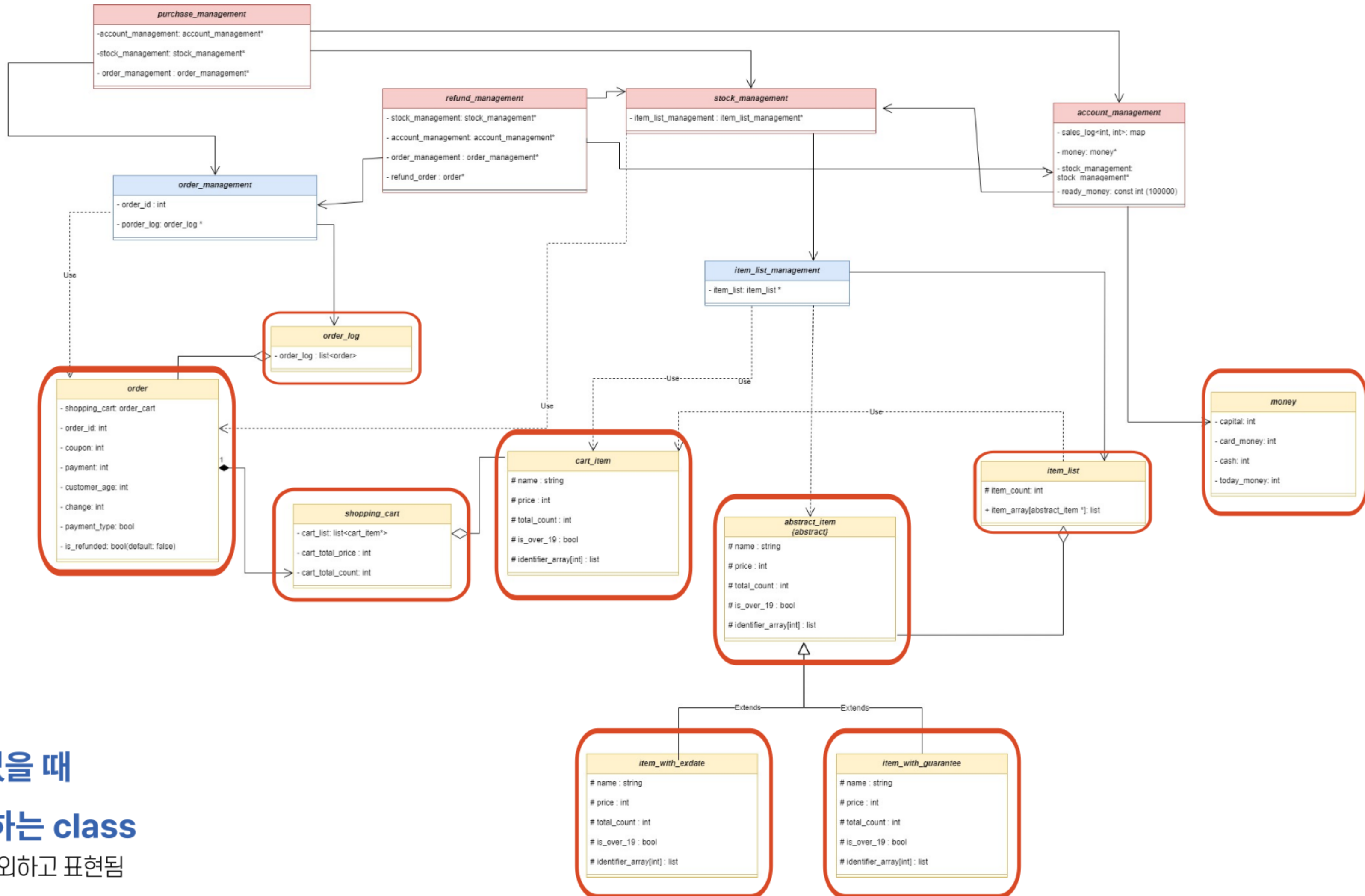


개수 증가
가격의 증액

How to "상품 관리"

02 본론

OOP의 적용



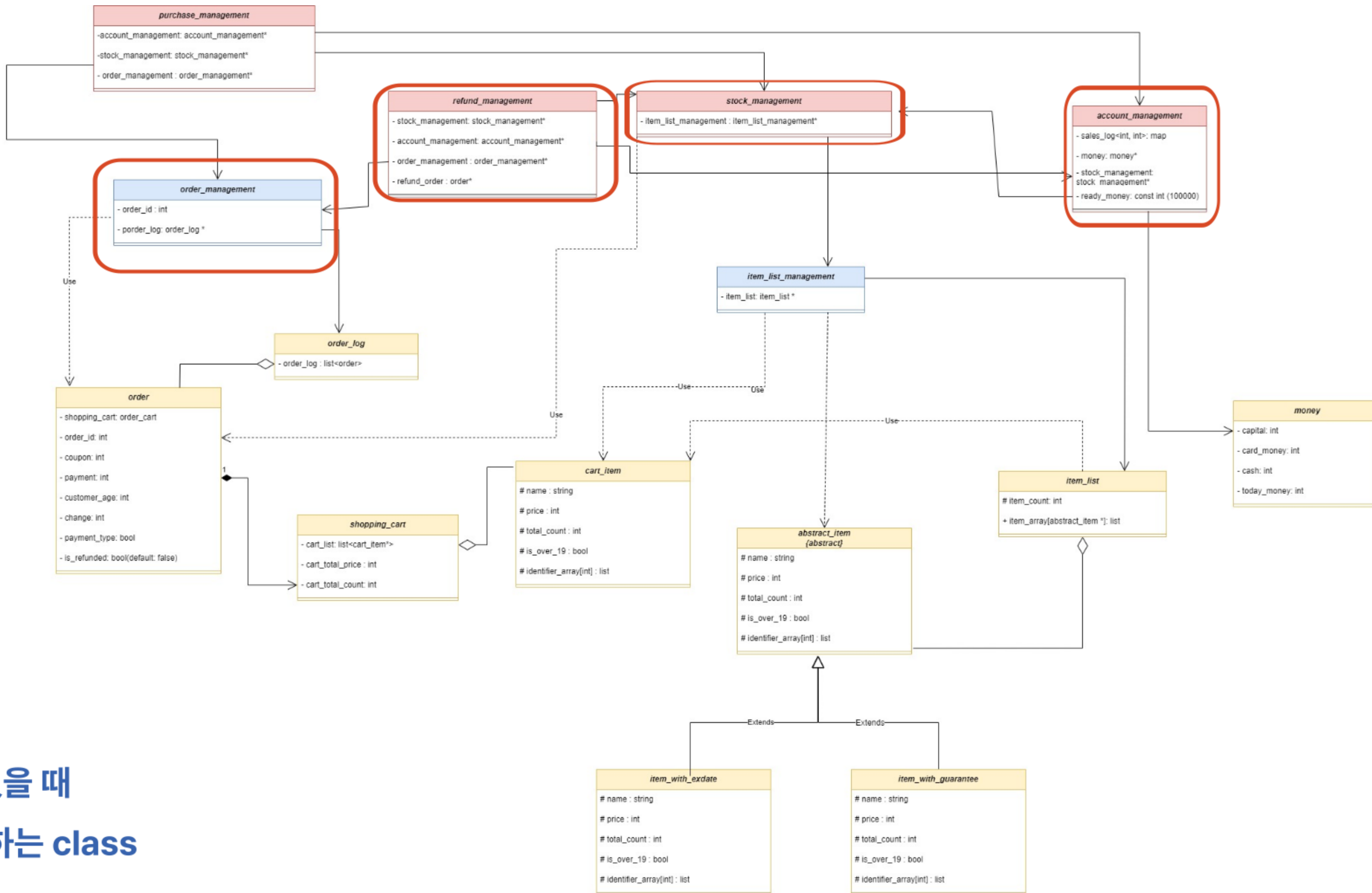
management 없을 때

환불 시 접근 해야하는 class

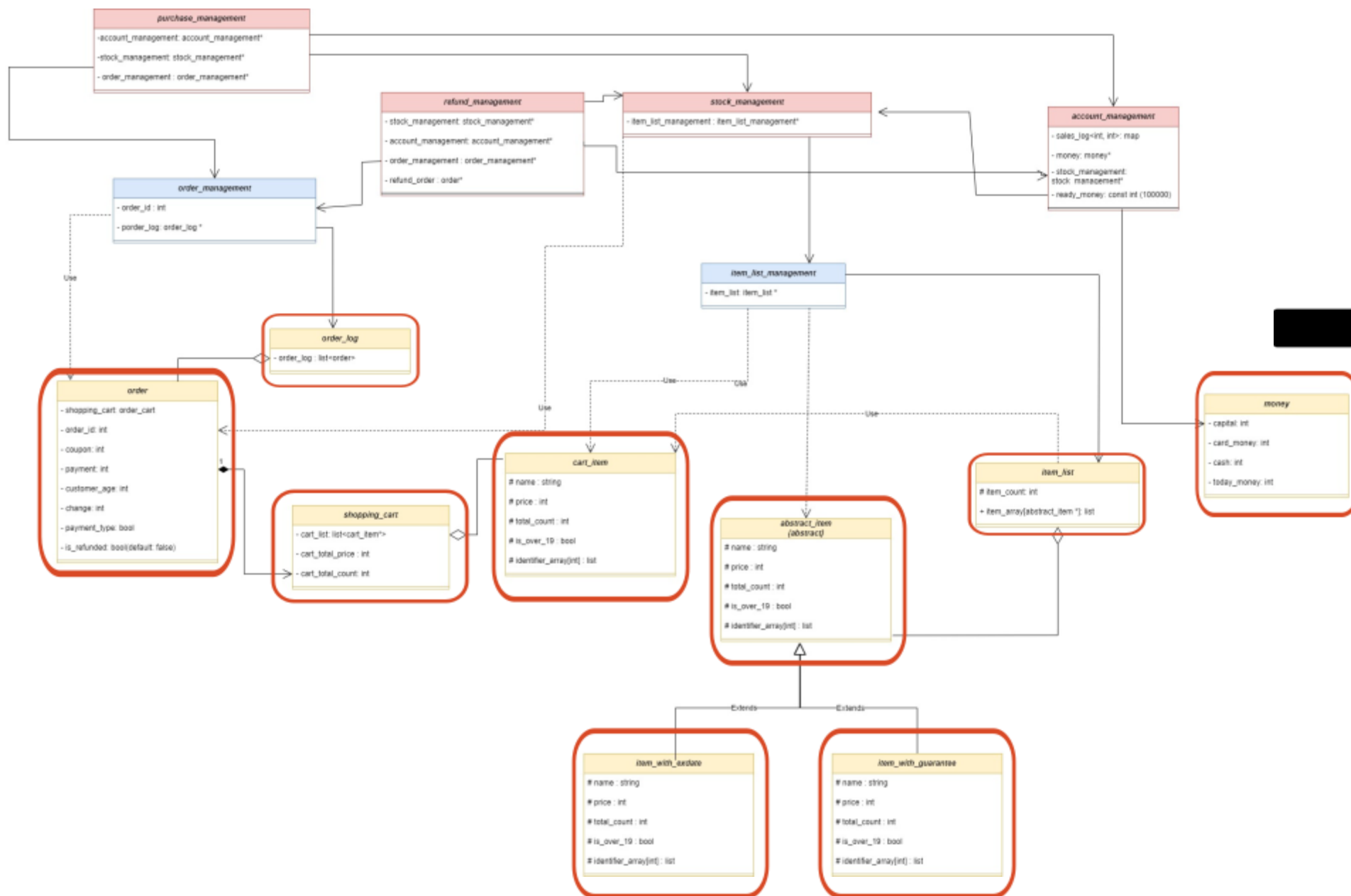
* 다음 UML은 메소드를 제외하고 표현됨

02 본론

OOP의 적용

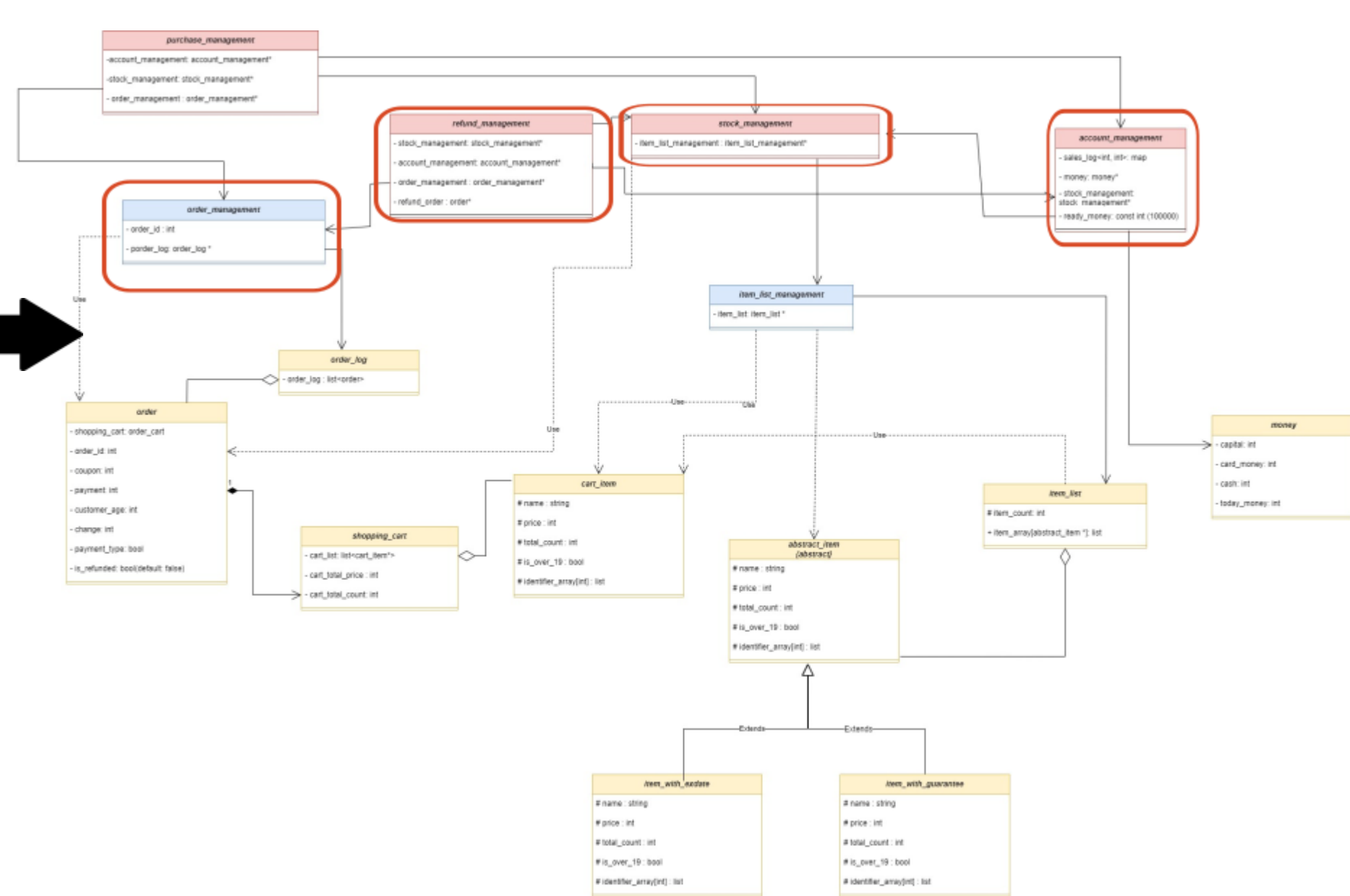


management 있을 때
환불 시 접근 해야하는 class



management 없을 때 환불 시 접근 해야하는 class

"Encapsulation"



management 있을 때 환불 시 접근 해야하는 class

"Single Responsibility"

2) POS기의 사용

02 본론

실제 POS기를 사용하는 직원



상황1



손님

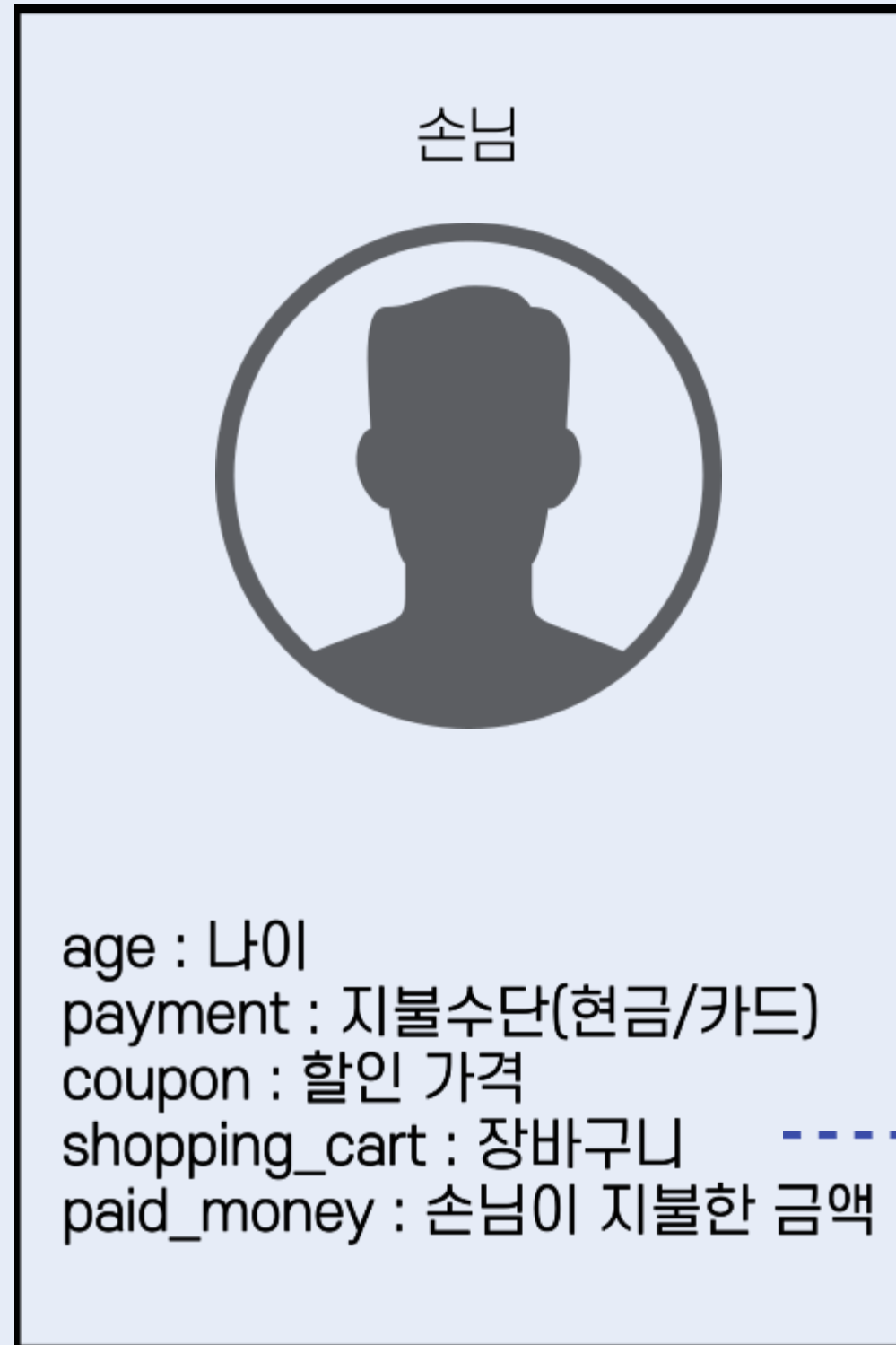


상황2



02 본론

상황 1. 손님과 직원(POS)의 상호작용 - 손님의 상품 구매

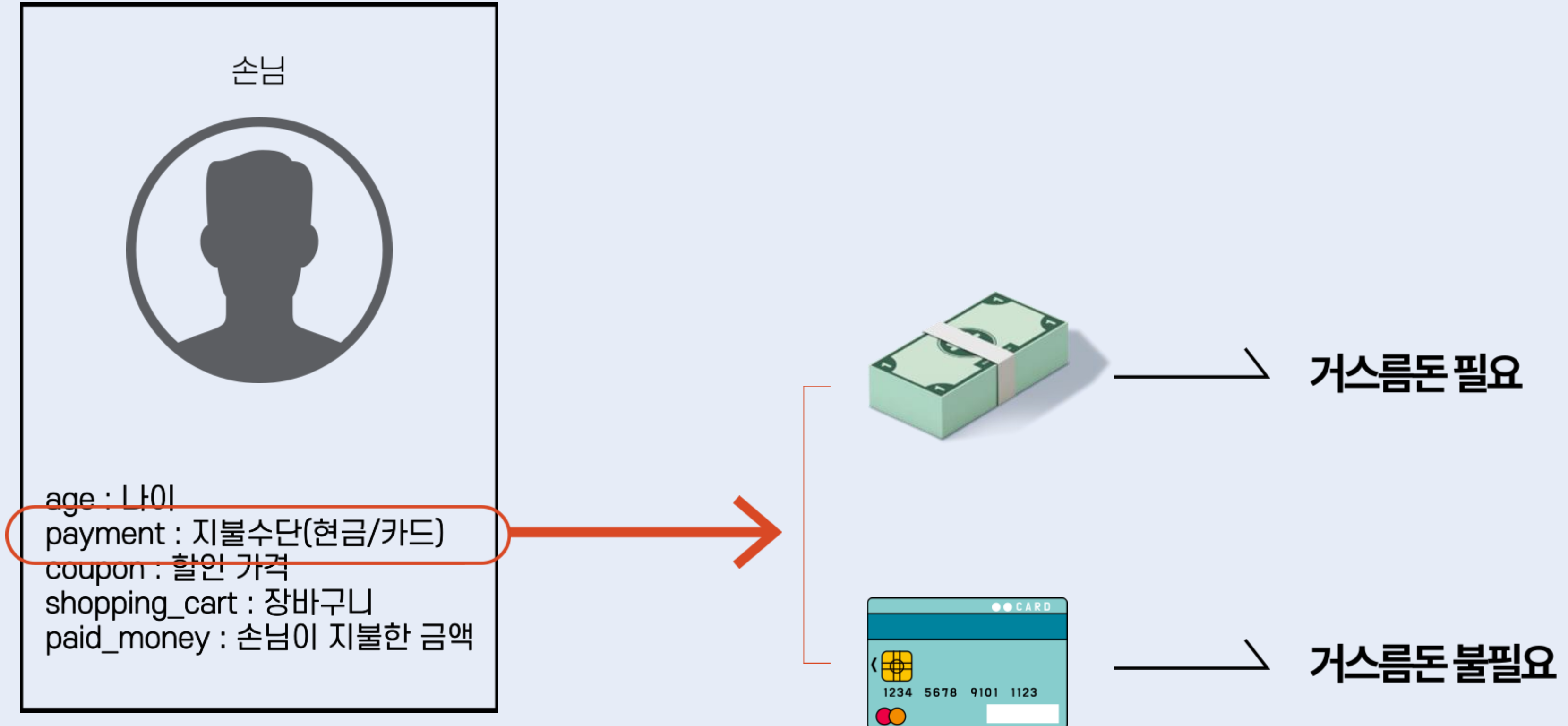


● 기능 제한

: shopping_cart에 담긴 물건은 수정될 수 없다.

02 본론

상황 1. 손님과 직원(POS)의 상호작용 - 손님의 상품 구매




02 본론

상황 1. 손님과 직원(POS)의 상호작용 - 손님의 상품 구매

(1) 구매기록의 추가

손님



age : 나이
payment : 지불수단(현금/카드)
coupon : 할인 가격
shopping_cart : 장바구니
paid_money : 손님이 지불한 금액



PAYMENT RECEIPT



영수증 출력, 환불에 이용

02 본론

상황 1. 손님과 직원(POS)의 상호작용 - 손님의 상품 구매

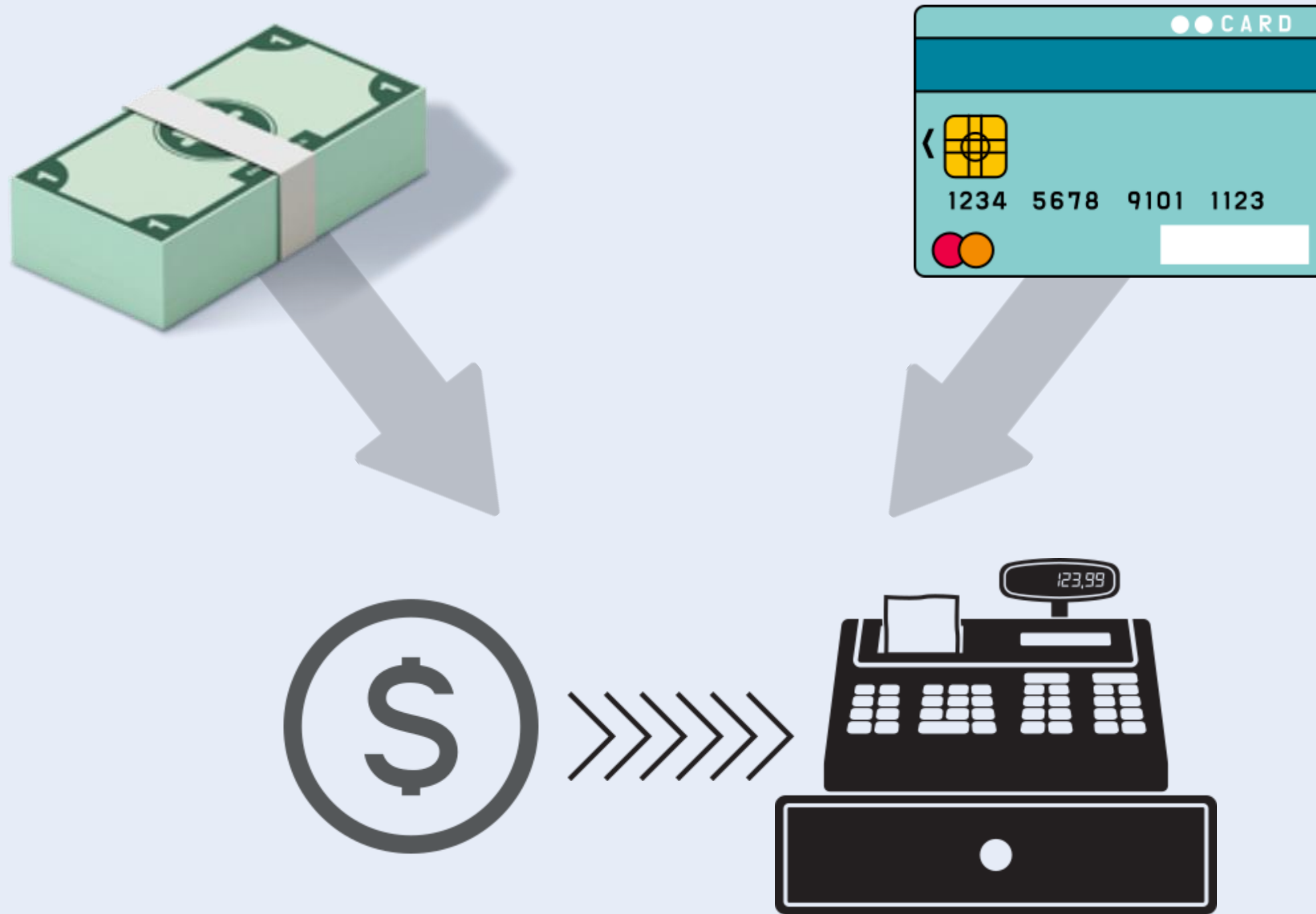
(2) 재고의 변경



02 본론

상황 1. 손님과 직원(POS)의 상호작용 - 손님의 상품 구매

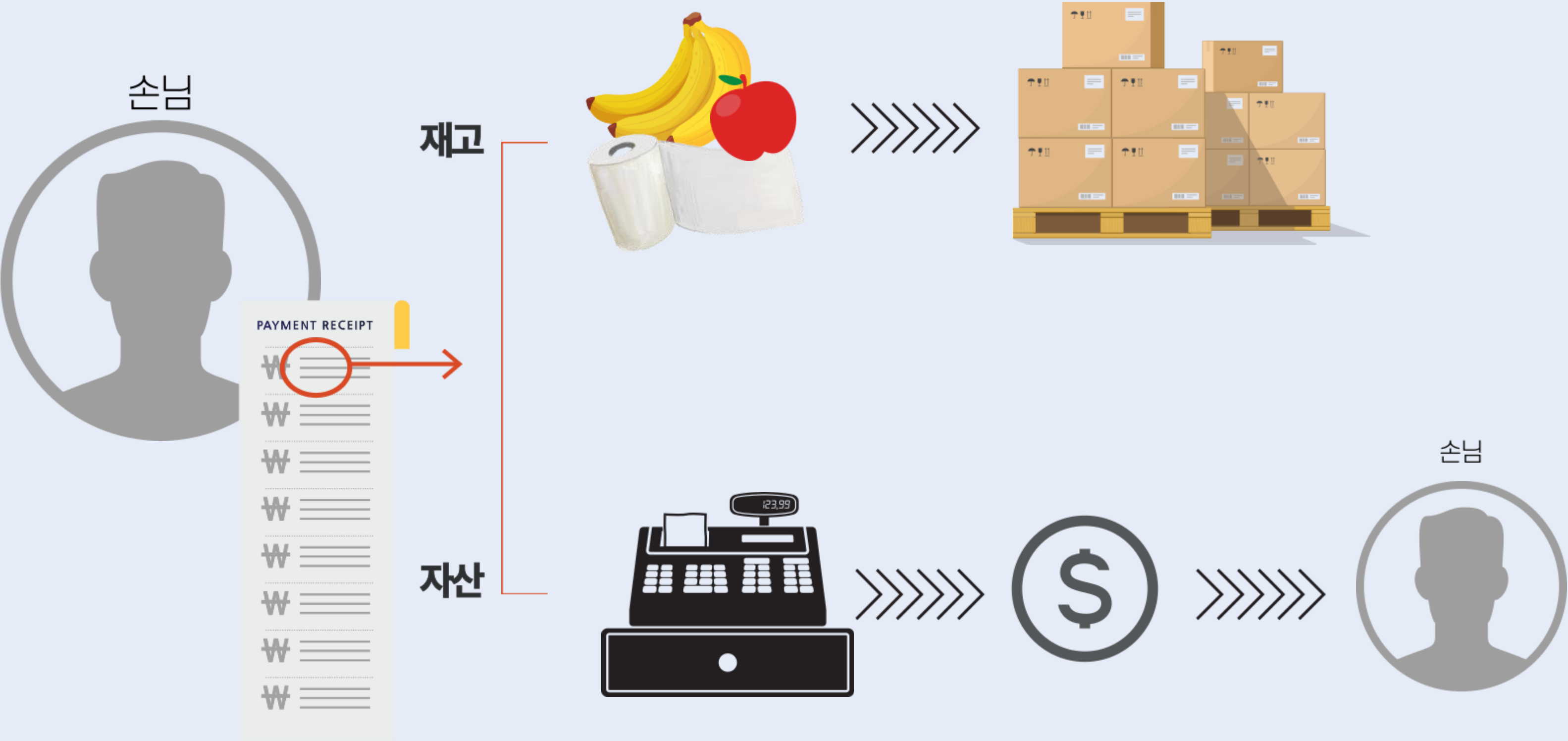
(3) 포스기 회계상의 변경



자산의 증가

02 본론

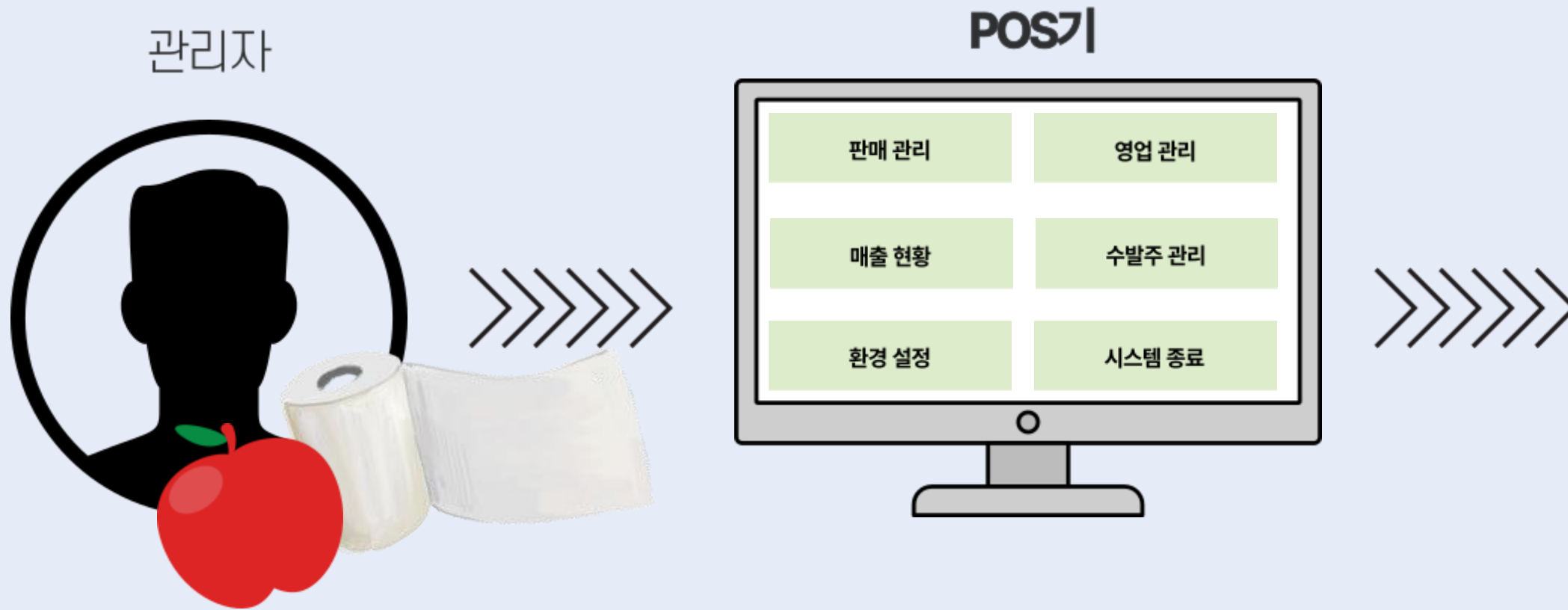
상황 1. 손님과 직원(POS)의 상호작용 - 손님의 상품 환불



02 본론

상황 2. 편의점과 직원(POS)의 상호작용 - 재고를 추가

휴지와 사과의 재고를 추가한다 가정



① 상품 목록에 이미 존재하는 상품
ex) 휴지

② 상품 목록에 존재하지 않는 상품
ex) 사과

02 본론

상황 2. 편의점과 직원(POS)의 상호작용 - 재고를 추가

① 상품 목록에 이미 존재하는 상품



+

유통기한 / 보증 기간

➤➤➤➤

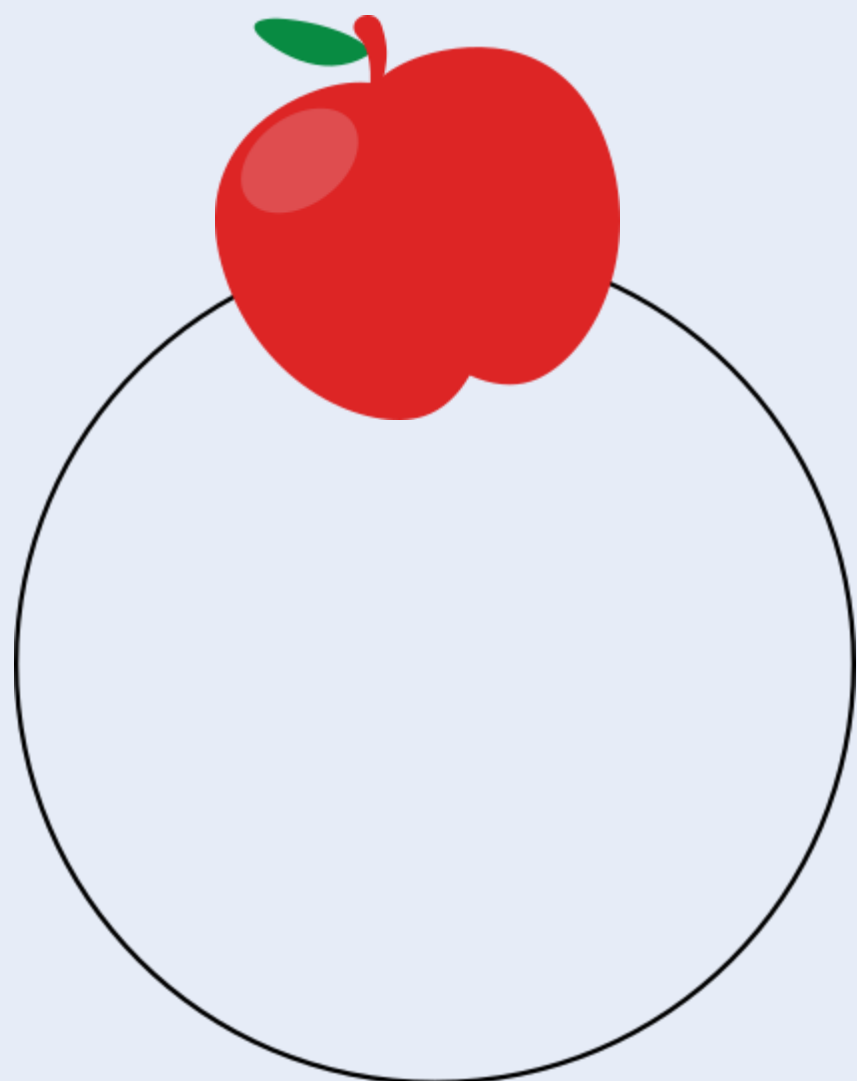


<입력 정보>

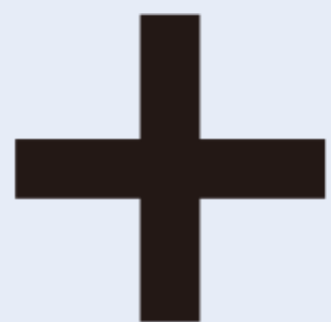
02 본론

상황 2. 편의점과 직원(POS)의 상호작용 - 재고를 추가

② 상품 목록에 존재하지 않는 상품



<기존 정보>



이름
가격

19세 미만 판매 가능?
유통기한 / 보증기한

<입력 정보>



02 본론

상황 2. 편의점과 직원(POS)의 상호작용 - 마감 정산 시행

(1) 회계상의 변화

<편의점 오픈>

현금 매출 = 10만원
카드 매출 = 0원
영업 준비금 = 10만원

<편의점 운영중>



<편의점 마감>

현금 매출 = 10만원
카드 매출 = 0원
영업 준비금 = 10만원

하루 매출액
= 현금 매출 + 카드 매출

02 본론

상황 2. 편의점과 직원(POS)의 상호작용 - 마감 정산 시행

(2) 매출기록의 추가



<매출 기록>



02 본론

상황 2. 편의점과 직원(POS)의 상호작용 - 마감 정산 시행

(3) 유통기한 / 보증기한이 지난 상품 폐기

<유통기한이 있는 상품>

유통 기한 > 오늘 날짜

<유통기한이 없는 상품>

보증 기한 > 오늘 날짜



폐기

03 결론

결론

03 결론

main()

```
POS 시스템을 시작합니다.
```

```
오늘의 날짜는 20221207 입니다.  
무엇을 하시겠습니까?
```

```
===== POS 목록 =====  
      1. 재고 관리          2. 구매  
      3. 환불              4. 회계  
      5. POS 시스템 종료(모든 정보 초기화)
```

```
=====
```

```
입력 (입력 예시: 1) :
```

03 결론

기능별 시행 - 재고 관리(1)

```
오늘의 날짜는 20221207 입니다.  
무엇을 하시겠습니까?
```

```
===== POS 목록 =====  
1. 재고 관리      2. 구매  
3. 환불          4. 회계  
5. POS 시스템 종료(모든 정보 초기화)
```

```
=====
```

```
입력 (입력 예시: 1) : 1
```

```
재고를 추가하시겠습니까? (Y / N) y
```

```
===== 재고 추가 시작 =====  
추가할 상품의 이름을 입력해주세요. (입력 예시: banana, 더이상 추가할 상품이 없을 경우 0입력) 우유  
우유은(는) 19세 미만 판매금지 상품입니까? (Y / N) n  
우유의 가격을 입력하세요. (입력 예시: 2000) 1000  
우유의 개수를 입력하세요. (입력 예시: 3) 5  
우유의 고유 정보는 무엇입니까? (1. 유통 기한 2. 보증 기한) (입력 예시: 1) 1  
우유 5개의 유통 기한을 입력하세요. (유통 기한이 같을 경우에도 각각 입력)  
(입력 예시: 20221219 20221219 20221224)(입력 후 Enter)  
20221213 20221213 20221213 20221213 20221213  
상품이 추가되었습니다.  
===== 재고 추가 완료 =====
```

03 결론

기능별 시행 - 재고 관리(2)

```
===== 재고 추가 시작 =====
추가할 상품의 이름을 입력해주세요. (입력 예시: banana, 더이상 추가할 상품이 없을 경우 0입력) 껌
껌은(는) 19세 미만 판매금지 상품입니까? (Y / N) N
껌의 가격을 입력하세요. (입력 예시: 2000) 500
껌의 개수를 입력하세요. (입력 예시: 3) 3
껌의 고유 정보는 무엇입니까? (1. 유통 기한 2. 보증 기한) (입력 예시: 1) 1
껌 3개의 유통 기한을 입력하세요. (유통 기한이 같을 경우에도 각각 입력)
(입력 예시: 20221219 20221219 20221224)(입력 후 Enter)
20230430 20230430 20230430
상품이 추가되었습니다.
===== 재고 추가 완료 =====

===== 재고 추가 시작 =====
추가할 상품의 이름을 입력해주세요. (입력 예시: banana, 더이상 추가할 상품이 없을 경우 0입력) 0
재고들을 추가하시겠습니까? (Y / N) n
재고들 추가하지 않습니다.
===== 재고 추가 취소 =====
```

03 결론

기능별 시행 - 구매(1)

오늘의 날짜는 20221207 입니다.
무엇을 하시겠습니까?

```
===== POS 목록 =====
1. 재고 관리          2. 구매
3. 환불             4. 회계
5. POS 시스템 종료(모든 정보 초기화)
```

입력 (입력 예시: 1) : 2

```
=====
1. 우유 가격: 1000    상품 목록
2. 껌   가격: 500     남은 수량: 5
                          남은 수량: 3
=====
```

```
===== 구매 시작 =====
손님의 나이를 입력해주세요: (입력 예시 : 20) : 20
```

위의 상품 목록을 참고하여 구매할 상품을 하나씩 입력해주세요.
(입력 예시: banana, 없다면 0입력)(한 번에 한 종류의 상품만 입력하세요.) : 우유
구매할 상품의 개수를 입력하세요. (입력 예시: 1) : 1
쇼핑카트에 상품이 담겼습니다.

위의 상품 목록을 참고하여 구매할 상품을 하나씩 입력해주세요.
(입력 예시: banana, 없다면 0입력)(한 번에 한 종류의 상품만 입력하세요.) : 껌
구매할 상품의 개수를 입력하세요. (입력 예시: 1) : 1
쇼핑카트에 상품이 담겼습니다.

03 결론

기능별 시행 - 구매(2)

```
위의 상품 목록을 참고하여 구매할 상품을 하나씩 입력해주세요.  
(입력 예시: banana, 없다면 0입력)(한 번에 한 종류의 상품만 입력하세요.) : 껌  
구매할 상품의 개수를 입력하세요. (입력 예시: 1) : 1  
쇼핑카트에 상품이 담겼습니다.
```

```
위의 상품 목록을 참고하여 구매할 상품을 하나씩 입력해주세요.  
(입력 예시: banana, 없다면 0입력)(한 번에 한 종류의 상품만 입력하세요.) : 0  
장바구니에 구매할 상품이 모두 추가되었습니다.  
결제를 시작합니다.  
쿠폰이 있다면 할인가를 입력하세요 (입력 예시 : 500, 없다면 0입력) : 0  
총 1500원 입니다.  
결제 방식을 선택해주세요 (0: 카드 1: 현금) : 0  
1500원 결제 완료.  
===== 구매 완료 =====
```

03 결론

기능별 시행 - 회계

오늘의 날짜는 20221207 입니다.
무엇을 하시겠습니까?

```
===== POS 목록 =====  
1. 재고 관리          2. 구매  
3. 환불              4. 회계  
5. POS 시스템 종료(모든 정보 초기화)
```

입력 (입력 예시: 1) : 4

원하시는 메뉴를 선택해 주세요.

```
=====
```

1. 마감 정산	2 : 매출 조회
----------	-----------

```
=====
```

입력 (입력 예시: 1) : 1

```
===== 마감 정산 =====
```

마감 정산을 시작하시겠습니까? (Y / N)

y

매출 로그 업데이트 완료

마감 정산이 완료되었습니다. 오늘 날짜가 변경됩니다.

```
===== 마감 정산 완료 =====
```

03 결론

기능별 시행 - 환불

```
오늘의 날짜는 20221208 입니다.  
무엇을 하시겠습니까?
```

```
===== POS 목록 =====  
1. 재고 관리          2. 구매  
3. 환불              4. 회계  
5. POS 시스템 종료(모든 정보 초기화)
```

```
=====
```

```
입력 (입력 예시: 1) : 3
```

```
환불을 진행하시겠습니까? (부분 환불은 가능하지 않습니다.) (Y / N)y
```

```
===== 환불 시작 =====  
환불 영수증 번호를 입력해 주세요. (입력 예시: 0)0  
내역 조회에 성공했습니다. 환불을 진행합니다.  
환불을 성공했습니다.  
===== 환불 종료 =====
```

03 결론

기능별 시행 - 시스템 종료

```
오늘의 날짜는 20221208 입니다.  
무엇을 하시겠습니까?
```

```
===== POS 목록 =====  
1. 재고 관리                2. 구매  
3. 환불                  4. 회계  
5. POS 시스템 종료(모든 정보 초기화)
```

```
=====
```

```
입력 (입력 예시: 1) : 5
```

```
POS 시스템을 종료합니다.
```

감사합니다