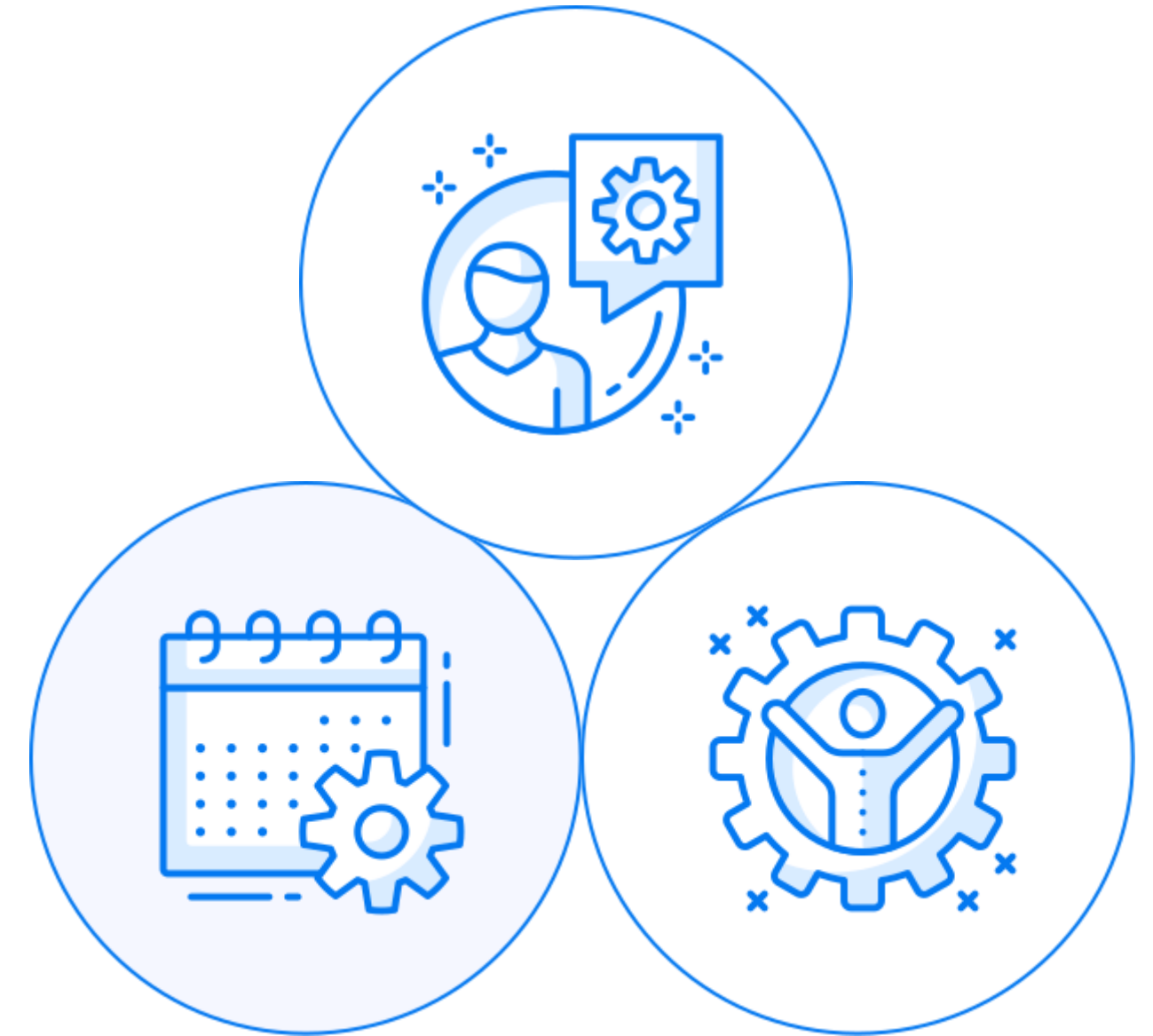


 객체지향개발방법론

OOPT 2050

3rd Cycle



팀원: 201912315 김성유 | 201411282 양대경 | 202011396 황정승

https://github.com/brian770/OOPT_development

목차

1. 구현 마무리 + 동작 화면
 2. 변동 사항
 3. CI/CD
 4. 개인 소감
-

구현 마무리 + 동작 화면

생성자를 다르게 하여 2개의 dvm 생성 (DVM dvm1("T1", 10, 20, 5001); DVM dvm2("T2", 4, 5, 5002);)

```
PS C:\OOPT_development> ./dvm1
[P2PServer] Listening on port 5001...
음료수 구매(1) | 인증코드 입력(2) | 구매 종료(3): 1
ID: 1, Name: 콜라, Price: 1500, Stock: 0
ID: 2, Name: 사이다, Price: 1400, Stock: 0
ID: 3, Name: 녹차, Price: 1300, Stock: 0
ID: 4, Name: 홍차, Price: 1350, Stock: 56
ID: 5, Name: 밀크티, Price: 1800, Stock: 47
ID: 6, Name: 탄산수, Price: 1200, Stock: 0
ID: 7, Name: 보리차, Price: 1100, Stock: 0
ID: 8, Name: 캔커피, Price: 1600, Stock: 0
ID: 9, Name: 물, Price: 1000, Stock: 35
ID: 10, Name: 에너지드링크, Price: 2000, Stock: 0
ID: 11, Name: 유자차, Price: 1500, Stock: 0
ID: 12, Name: 식혜, Price: 1400, Stock: 0
ID: 13, Name: 아이스티, Price: 1350, Stock: 45
ID: 14, Name: 딸기주스, Price: 1700, Stock: 44
ID: 15, Name: 오렌지주스, Price: 1700, Stock: 0
ID: 16, Name: 포도주스, Price: 1700, Stock: 0
ID: 17, Name: 이온음료, Price: 1300, Stock: 0
ID: 18, Name: 아메리카노, Price: 1900, Stock: 18
ID: 19, Name: 핫초코, Price: 1800, Stock: 57
ID: 20, Name: 카페라떼, Price: 2000, Stock: 0
음료수 번호를 입력해주세요: █
```

```
PS C:\OOPT_development> ./dvm2
[P2PServer] Listening on port 5002...
음료수 구매(1) | 인증코드 입력(2) | 구매 종료(3): 1
ID: 1, Name: 콜라, Price: 1500, Stock: 0
ID: 2, Name: 사이다, Price: 1400, Stock: 68
ID: 3, Name: 녹차, Price: 1300, Stock: 0
ID: 4, Name: 홍차, Price: 1350, Stock: 16
ID: 5, Name: 밀크티, Price: 1800, Stock: 0
ID: 6, Name: 탄산수, Price: 1200, Stock: 0
ID: 7, Name: 보리차, Price: 1100, Stock: 70
ID: 8, Name: 캔커피, Price: 1600, Stock: 0
ID: 9, Name: 물, Price: 1000, Stock: 0
ID: 10, Name: 에너지드링크, Price: 2000, Stock: 0
ID: 11, Name: 유자차, Price: 1500, Stock: 0
ID: 12, Name: 식혜, Price: 1400, Stock: 0
ID: 13, Name: 아이스티, Price: 1350, Stock: 0
ID: 14, Name: 딸기주스, Price: 1700, Stock: 51
ID: 15, Name: 오렌지주스, Price: 1700, Stock: 57
ID: 16, Name: 포도주스, Price: 1700, Stock: 0
ID: 17, Name: 이온음료, Price: 1300, Stock: 0
ID: 18, Name: 아메리카노, Price: 1900, Stock: 64
ID: 19, Name: 핫초코, Price: 1800, Stock: 0
ID: 20, Name: 카페라떼, Price: 2000, Stock: 14
음료수 번호를 입력해주세요: █
```

음료수 재고는 랜덤 생성하여 배정(각 무작위 7종류 1~99개의 무작위 개수 할당)

구현 마무리 + 동작 화면

T1 자판기에 재고가 부족한 2번 음료수 주문 -> T1~T7의 자판기에게 broadcast(본인 T1 제외) -&> T2 자판기에게 응답 수신

통신(발신)
디버그용
출력문

```
음료수 번호를 입력해주세요: 2
수량을 입력해주세요: 2
[MsgManager] 브로드캐스트 전송

[MsgManager] 수신 메시지: {"dst_id":"T1","msg_content":{"coor_x":4,"coor_y":5,"item_code":2,"item_num":68},"msg_type":"resp_stock","src_id":"T2"}
[P2PClient] connection failed to 127.0.0.1:5003 error code: 0
[P2PClient] connection failed to 127.0.0.1:5003 error code: 0
[P2PClient] connection failed to 127.0.0.1:5004 error code: 10061
[P2PClient] connection failed to 127.0.0.1:5004 error code: 0
[P2PClient] connection failed to 127.0.0.1:5005 error code: 10061
[P2PClient] connection failed to 127.0.0.1:5005 error code: 0
[P2PClient] connection failed to 127.0.0.1:5006 error code: 10061
[P2PClient] connection failed to 127.0.0.1:5006 error code: 0
[P2PClient] connection failed to 127.0.0.1:5007 error code: 10061
[P2PClient] connection failed to 127.0.0.1:5007 error code: 0
선결제 하시겠습니까? Yes(Y|y) No(N|n): y
```

통신(수신)
디버그용
출력문

```
[MsgManager] 수신 메시지: {"dst_id":"T1","msg_content":{"availability":"T","item_code":2,"item_num":2},"msg_type":"resp_prepay","src_id":"T2"}

[MsgManager] 수신 메시지: {"dst_id":"T2","msg_content":{"cert_code":"8rM8S","item_code":2,"item_num":2},"msg_type":"req_prepay","src_id":"T1"}
```

```
ID: 5, Name: 밀크티, Price: 1800, Stock: 0
ID: 6, Name: 탄산수, Price: 1200, Stock: 0
ID: 7, Name: 보리차, Price: 1100, Stock: 70
ID: 8, Name: 캔커피, Price: 1600, Stock: 0
ID: 9, Name: 물, Price: 1000, Stock: 0
ID: 10, Name: 에너지드링크, Price: 2000, Stock: 0
ID: 11, Name: 유자차, Price: 1500, Stock: 0
ID: 12, Name: 식혜, Price: 1400, Stock: 0
ID: 13, Name: 아이스티, Price: 1350, Stock: 0
ID: 14, Name: 딸기주스, Price: 1700, Stock: 51
ID: 15, Name: 오렌지주스, Price: 1700, Stock: 57
ID: 16, Name: 포도주스, Price: 1700, Stock: 0
ID: 17, Name: 이온음료, Price: 1300, Stock: 0
ID: 18, Name: 아메리카노, Price: 1900, Stock: 64
ID: 19, Name: 핫초코, Price: 1800, Stock: 0
ID: 20, Name: 카페라떼, Price: 2000, Stock: 14
음료수 번호를 입력해주세요:
```

```
[MsgManager] 수신 메시지: {"dst_id":0,"msg_content":{"item_code":2,"item_num":2},"msg_type":"req_stock","src_id":"T1"}

[MsgManager] 수신 메시지: {"dst_id":"T2","msg_content":{"cert_code":"8rM8S","item_code":2,"item_num":2},"msg_type":"req_prepay","src_id":"T1"}
[]
```

통신(수신)
디버그용
출력문

선결제 선택 -> 대안 자판기로 선정된 T2 자판기에게 메시지 전송 -> availability가 T인 응답 메시지 수신 + T2 자판기에 인증코드 저장

구현 마무리 + 동작 화면

T1 자판기에서 선결제 진행 -> 올바른 카드 번호 입력 -> 결제 성공 여부와 인증코드, 대안 자판기 좌표 출력 -> 다음 구매 진행

```
{
  "_id": "T1"
}
카드 번호를 입력해주세요(5자리): 12345
결제 성공하였습니다.
인증코드는 8rM8S입니다.
대안 자판기의 위치는 4, 5 입니다.
음료수 구매(1) | 인증코드 입력(2) | 구매 종료(3):
```

T2 자판기에서 4번 음료수 1개 결제 진행 -> 올바른 카드 번호 입력 -> 결제 성공 여부와 구매 항목 출력 -> 다음 구매 진행

T2 자판기에 2번, 4번 음료수의 재고가 차감됨

```
ID: 1, Name: 콜라, Price: 1500, Stock: 0
ID: 2, Name: 사이다, Price: 1400, Stock: 68
ID: 3, Name: 녹차, Price: 1300, Stock: 0
ID: 4, Name: 홍차, Price: 1350, Stock: 16
```

선결제 진행

일반 결제



```
ID: 1, Name: 콜라, Price: 1500, Stock: 0
ID: 2, Name: 사이다, Price: 1400, Stock: 66
ID: 3, Name: 녹차, Price: 1300, Stock: 0
ID: 4, Name: 홍차, Price: 1350, Stock: 15
```

음료수 번호 및 수량 입력 시, 유효하지 않은 입력 예외처리

```
음료수 번호를 입력해주세요: 0
음료수 번호는 1 ~ 20 사이의 값입니다.
음료수 번호를 입력해주세요: a
음료수 번호는 1 ~ 20 사이의 값입니다.
음료수 번호를 입력해주세요: 4
수량을 입력해주세요: 0
음료수 수량은 1~99 까지만 선택 가능합니다.
수량을 입력해주세요: b
음료수 수량은 1~99 까지만 선택 가능합니다.
수량을 입력해주세요: 2
카드 번호를 입력해주세요(5자리):
```

올바르지 않은 카드 번호 입력 시

```
카드 번호를 입력해주세요(5자리): 11111
카드 정보가 불일치합니다.
음료수 구매(1) | 인증코드 입력(2) | 구매 종료(3):
```

경고문 출력 -> 다음 구매 진행

구현 마무리 + 동작 화면

T1 자판기에게 발급 받은 인증 코드 -> T2 자판기에게 입력 -> 구매 항목 출력 -> 다음 구매 진행

결제 성공하였습니다.

인증코드는 **8rM8S**입니다.

대안 자판기의 위치는 **4, 5** 입니다.

음료수 구매(1) | 인증코드 입력(2) | 구매 종료(3):

음료수 구매(1) | 인증코드 입력(2) | 구매 종료(3): 2

인증코드를 입력해주세요(5자리): **8rM8S**

사이다 2개 구매되었습니다.

음료수 구매(1) | 인증코드 입력(2) | 구매 종료(3):

이미 사용된 인증 코드는 삭제됨(재사용 불가)

음료수 구매(1) | 인증코드 입력(2) | 구매 종료(3): 2

인증코드를 입력해주세요(5자리): **8rM8S**

사이다 2개 구매되었습니다.

음료수 구매(1) | 인증코드 입력(2) | 구매 종료(3): 2

인증코드를 입력해주세요(5자리): **8rM8S**

해당 인증코드가 존재하지 않습니다.

음료수 구매(1) | 인증코드 입력(2) | 구매 종료(3):

모든 자판기에 부족한 재고(1번 음료수) 구매 시도 시

음료수 구매(1) | 인증코드 입력(2) | 구매 종료(3): 1

ID: 1, Name: 콜라, Price: 1500, Stock: 0

ID: 2, Name: 사이다, Price: 1400, Stock: 0

ID: 3, Name: 녹차, Price: 1300, Stock: 0

해당 인증코드가 존재하지 않습니다.

음료수 구매(1) | 인증코드 입력(2) | 구매 종료(3): 1

ID: 1, Name: 콜라, Price: 1500, Stock: 0

ID: 2, Name: 사이다, Price: 1400, Stock: 66



"대안 자판기 없습니다." 메시지 출력 -> 다음 구매 진행

1

수량을 입력해주세요: 1

[MsgManager] 브로드캐스트 전송

[MsgManager] 수신 메시지: {"dst_id":"T2","msg_content":{"coor_x":10,"coor_y":20,"item_code":1,"item_num":0},"msg_type":"resp_stock","src_id":"T1"}

[P2PClient] connection failed to 127.0.0.1:5003 error code: 0

[P2PClient] connection failed to 127.0.0.1:5003 error code: 0

[P2PClient] connection failed to 127.0.0.1:5004 error code: 10061

[P2PClient] connection failed to 127.0.0.1:5004 error code: 0

[P2PClient] connection failed to 127.0.0.1:5005 error code: 10061

[P2PClient] connection failed to 127.0.0.1:5005 error code: 0

[P2PClient] connection failed to 127.0.0.1:5006 error code: 10061

[P2PClient] connection failed to 127.0.0.1:5006 error code: 0

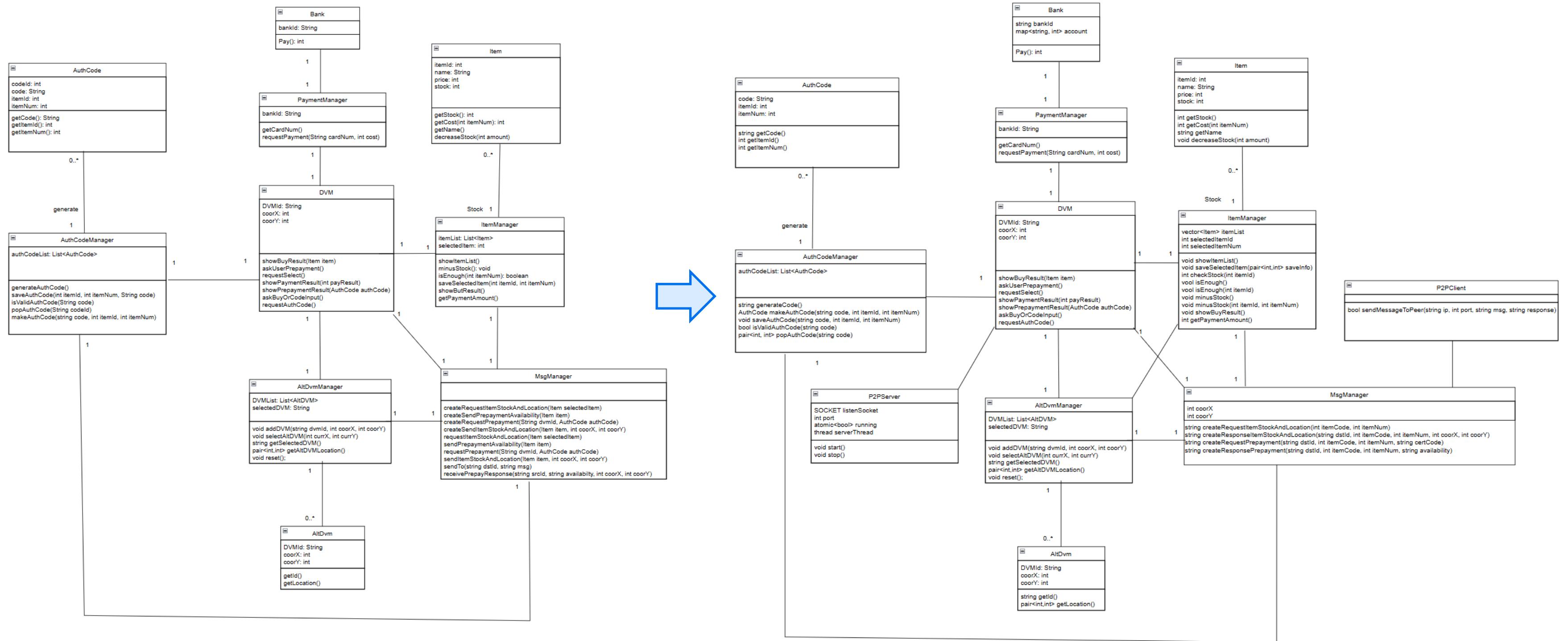
[P2PClient] connection failed to 127.0.0.1:5007 error code: 10061

[P2PClient] connection failed to 127.0.0.1:5007 error code: 0

대안 자판기가 없습니다.

음료수 구매(1) | 인증코드 입력(2) | 구매 종료(3):

변동 사항



CI/CD

```
Test Case 1      <error> (follows [if])
| DrinkIDInput (User) : else

Test Case 2      <error> (follows [if])
| DrinkNumInput (User) : else

Test Case 3      <error> (follows [if])
| SelectPrepay (User) : else

Test Case 4      <error> (follows [if])
| StockID (AltDVM) : else

Test Case 5      <error> (follows [if])
| AltDVMStockNum (AltDVM) : else

Test Case 6      <error> (follows [if])
| AltDVMLocation (AltDVM) : else

Test Case 7      <error> (follows [if])
| AltDVMPrepayAvail (AltDVM) : else

Test Case 8      <error>
| ReqDrinkID (AltDVM) : else

Test Case 9      <error>
| ReqDrinkNum (AltDVM) : else
```

1. Fail (무한루프 발생)
2. Fail (무한루프 발생)
3. Pass
4. Fail (타 자판기와 연결 구현 안됨)
5. Fail (타 자판기와 연결 구현 안됨)
6. Fail (타 자판기와 연결 구현 안됨)
7. Fail (타 자판기와 연결 구현 안됨)
8. Fail (타 자판기와 연결 구현 안됨)
9. Fail (타 자판기와 연결 구현 안됨)

```
pair<int, int> DVM::requestSelect() {
    int itemId, quantity;
    while (true) {
        cout << "음료수 번호를 입력해주세요: ";
        cin >> itemId;

        if (cin.fail() || itemId < 1 || itemId > 20) {
            cin.clear(); // 입력 스트림 오류 플래그 초기화
            cin.ignore(numeric_limits<streamsize>::max(), '\n'); // 잘못된 입력 무시
            cout << "음료수 번호는 1 ~ 20 사이의 값입니다." << endl;
            continue;
        }
        break;
    }

    while (true) {
        cout << "수량을 입력해주세요: ";
        cin >> quantity;
        if (cin.fail() || quantity < 1 || quantity > 99) {
            cin.clear(); // 입력 스트림 오류 플래그 초기화
            cin.ignore(numeric_limits<streamsize>::max(), '\n'); // 잘못된 입력 무시
            cout << "음료수 수량은 1~99 까지만 선택 가능합니다." << endl;
            continue;
        }
        break;
    }

    return {itemId, quantity};
}
```

기존에는 입력 받은 값이 정수가 아닌 경우, 무한루프가 발생
--> 잘못된 입력이 들어왔을 때 clear와 ignore 함수를 통해서 에러 플래그를 초기화하고 입력 버퍼를 비워 무한루프 문제 해결

다른 나머지 fail들은 타 자판기와 연결이 구현이 안되어 test 진행이 어려웠음

현재 구현은 완료했으나 최종 구현이 늦게 되어서 추가 피드백을 받지 못함

개인 소감

김성유

C++언어로 코딩이 익숙치 않은데다, 이 정도로 큰 규모의 객체지향 프로그램을 개발해본 경험이 부족하여 개발에 많은 어려움이 있었지만, 하나의 큰 시스템을 역할별로 객체단위로 명확히 나누어 파악하고 개발하는데 많은 경험을 얻게 되었다. 아쉬운 점으로는 메시지 관리 담당 및 통신 담당 객체를 나눌 생각을 초기 OOPT 단계에서 하지 못해, 개발하는 동안 많은 어려움을 겪었다. 하지만 이렇게 이전 단계로 돌아가 분석한 구조를 수정하고 개선해 나가는 것 또한 OOPT의 의의라 생각하기에 의미 있었다 생각한다.

양대경

분산형 자판기 개발을 통해 객체지향의 핵심 개념인 캡슐화, 책임 분리, 유연한 구조 설계의 중요성을 체감했다. UML 기반 설계는 팀 협업과 시스템 이해를 돕는 데 큰 역할을 했고, 유지보수성과 확장성을 고려한 개발의 가치를 배웠다. 단순 구현을 넘어 체계적인 소프트웨어 설계 경험이 되어 의미 있었다.

황정승

학기 처음에 수업을 들으면서 설계 하는 단계에서는 객체지향적으로 설계 하는 것이 왜 디버깅과 확장에 용이한 지 크게 와닿지 않았었다. 하지만 개발 단계에 들어가면서 꼼꼼하게 설계하지 못한 점과 객체 지향적으로 설계하지 못한 부분 때문에 구현하는데 큰 어려움을 겪었다. 또한, 잘 설계 되어진 팀의 구현 과정을 보면서 블록 단위로 끼워 넣는 방식으로 비교적 쉽게 구현 할 수 있음을 알게 되었다. OOPT2050 단계 사이클을 완벽하게 수행하진 못했지만 객체지향 사고방식을 체득하고 중요성을 깨닫게 되는 매우 의미 있는 수업이었다.

 객체지향개발방법론

**THANK YOU FOR
WATCHING**
