



OBJECT ORIENTED PROCESS WITH TRACE

Final Presentation

OOPT 3rd Cycle

Team4:

202112348 조영탁

202211282 김종혁

202312373 정석현



목차

01 동작 화면



02 Brute Force Test 수정



03 Static Analysis 수정



04 개인적인 소감





CHAPTER 01 🔍

동작 화면

주문

```
어서오세요 .
시작하려면 's'를 입력하세요 .
(선결제 인증코드 입력을 원하시면 "인증코드"를 입력하세요 ...)
-----
s
-----
(1) 콜라 : 99개 1500원 (2) 사이다 : 99개 1500원 (3) 녹차 : 99개 1700원 (4) 홍차 : 99개 1700원 (5) 밀크티 : 99개 2000원
(6) 탄산수 : 99개 1400원 (7) 보리차 : 99개 1300원 (8) 캔커피 : 0개 1500원 (9) 물 : 0개 1000원 (10) 에너지 드링크 : 0개 2200원
(11) 유자차 : 0개 1800원 (12) 식혜 : 0개 1600원 (13) 아이스티 : 0개 1000원 (14) 딸기주스 : 0개 2000원 (15) 오렌지 주스 : 0개 2000원
(16) 포도 주스 : 0개 2000원 (17) 이온 음료 : 0개 1700원 (18) 아메리카노 : 0개 1900원 (19) 핫초코 : 0개 1800원 (20) 카페라떼 : 0개 2100원

음료 ID와 갯수를 입력해주세요 . (e.g. 3 2)
-----
10 10
-----
치 x: %d 결제를 원하시면 'y', 원하지 않으면 'n'를 입력해주세요
결제 여부 :
-----
y
-----
카드번호를 입력해주세요
-----
11111
-----
결제에 성공했습니다
-----
```



CHAPTER 01 🔍

동작 화면

인증 코드

```
-----
x좌표 : 10, y좌표 : 10, 인증번호 : CYAvj
-----
어서오세요 .
시작하려면 's'를 입력하세요 .
(선결제 인증코드 입력을 원하시면 "인증코드"를 입력하세요 ...)
-----

```

```
[100%] Built target gmock_main
❖ jeongseoghyeon@jeongseoghyeon-ui-MacBookAir build % ./dvm
-----
어서오세요 .
시작하려면 's'를 입력하세요 .
(선결제 인증코드 입력을 원하시면 "인증코드"를 입력하세요 ...)
-----
인증코드
-----
인증코드를 입력하세요
-----
CYAvj
-----
음료수명 : 에너지 드링크, 갯수 : 10
안녕히가십시오 .
-----
어서오세요 .
시작하려면 's'를 입력하세요 .
(선결제 인증코드 입력을 원하시면 "인증코드"를 입력하세요 ...)
-----

```



CHAPTER 01



동작 화면

사용자 주문 중 선결제 처리

(1) 콜라 : 0개 1500원 (2) 사이다 : 0개 1500원 (3) 녹차 : 0개 1700원 (4) 홍차 : 0개 1700원 (5) 밀크티 : 0개 2000원
(6) 탄산수 : 0개 1400원 (7) 보리차 : 0개 1300원 (8) 캔커피 : 0개 1500원 (9) 물 : 99개 1000원 (10) 에너지 드링크 : 99개 2200원
(11) 유자차 : 99개 1800원 (12) 식혜 : 99개 1600원 (13) 아이스티 : 99개 1000원 (14) 딸기주스 : 99개 2000원 (15) 오렌지 주스 : 99개 2000원
(16) 포도 주스 : 99개 2000원 (17) 이온 음료 : 99개 1700원 (18) 아메리카노 : 99개 1900원 (19) 핫초코 : 99개 1800원 (20) 카페라떼 : 79개 2100원

음료 ID와 갯수를 입력해주세요. (e.g. 3 2)

10 10

카드번호를 입력해주세요

11111

결제에 성공했습니다

음료수 명 : 에너지 드링크, 갯수: 10
안녕하십니까.

어서오세요.
시작하려면 's'를 입력하세요.
(선결제 인증코드 입력을 원하시면 "인증코드"를 입력하세요...)

인증코드

인증코드를 입력하세요

DkLHc

음료수 명 : 에너지 드링크, 갯수: 10
안녕하십니까.

어서오세요.
시작하려면 's'를 입력하세요.
(선결제 인증코드 입력을 원하시면 "인증코드"를 입력하세요...)

s

(1) 콜라 : 0개 1500원 (2) 사이다 : 0개 1500원 (3) 녹차 : 0개 1700원 (4) 홍차 : 0개 1700원 (5) 밀크티 : 0개 2000원
(6) 탄산수 : 0개 1400원 (7) 보리차 : 0개 1300원 (8) 캔커피 : 0개 1500원 (9) 물 : 99개 1000원 (10) 에너지 드링크 : 79개 2200원
(11) 유자차 : 99개 1800원 (12) 식혜 : 99개 1600원 (13) 아이스티 : 99개 1000원 (14) 딸기주스 : 99개 2000원 (15) 오렌지 주스 : 99개 2000원
(16) 포도 주스 : 99개 2000원 (17) 이온 음료 : 99개 1700원 (18) 아메리카노 : 99개 1900원 (19) 핫초코 : 99개 1800원 (20) 카페라떼 : 79개 2100원

음료 ID와 갯수를 입력해주세요. (e.g. 3 2)



CHAPTER 01 🔍

동작 화면

재고가 다른 자판기에도 없을 때

```
(1) 콜라 : 99개 1500원 (2) 사이다 : 99개 1500원 (3) 녹차 : 99개 1700원 (4) 홍차 : 99개 1700원 (5) 밀크티 : 99개 2000원
(6) 탄산수 : 99개 1400원 (7) 보리차 : 99개 1300원 (8) 캔커피 : 0개 1500원 (9) 물 : 0개 1000원 (10) 에너지드링크 : 0개 2200원
(11) 유자차 : 0개 1800원 (12) 식혜 : 0개 1600원 (13) 아이스티 : 0개 1000원 (14) 딸기주스 : 0개 2000원 (15) 오렌지주스 : 0개 2000원
(16) 포도주스 : 0개 2000원 (17) 이온음료 : 0개 1700원 (18) 아메리카노 : 0개 1900원 (19) 핫초코 : 0개 1800원 (20) 카페라떼 : 0개 2100원

음료 ID와 갯수를 입력해주세요. (e.g. 3 2)

-----
8 10
-----
선결제 가능한 자판기가 존재하지 않습니다.
-----
-----
어서오세요.
시작하려면 's'를 입력하세요.
(선결제 인증코드 입력을 원하시면 "인증코드"를 입력하세요...)
```



CHAPTER 01 🔍

동작 화면

인증코드 저장

```
/ certCode6Yxn5.json / ...  
{  
  "cert_code": "6Yxn5",  
  "item_code": 10,  
  "item_num": 10  
}
```

동작 화면

```
어서오세요 .
시작하려면 's'를 입력하세요 .
(선결제 인증코드 입력을 원하시면 "인증코드"를 입력하세요 ...)
```

```
s
```

```
(1) 콜라 : 0개 1500원 (2) 사이다 : 0개 1500원 (3) 녹차 : 0개 1700원 (4) 홍차 : 0개 1700원 (5) 밀크티 : 0개
(6) 탄산수 : 0개 1400원 (7) 보리차 : 0개 1300원 (8) 캔커피 : 0개 1500원 (9) 물 : 99개 1000원 (10) 에너지
(11) 유자차 : 99개 1800원 (12) 식혜 : 99개 1600원 (13) 아이스티 : 99개 1000원 (14) 딸기주스 : 99개 2000원
00원
(16) 포도 주스 : 99개 2000원 (17) 이온 음료 : 99개 1700원 (18) 아메리카노 : 99개 1900원 (19) 핫초코 : 99개
개 2100원
```

```
음료 ID와 갯수를 입력해주세요 . (e.g. 3 2)
```

```
14 10
```

```
카드번호를 입력해주세요
```

```
11111
```

```
결제에 성공했습니다
```

```
음료수 명 : 딸기주스 , 갯수 : 10
안녕히가십시오 .
```

```
build > {} card.json > # balance
```

```
1 {
2   "balance": 100000,
3   "card_num": 11111
4 }
```

```
build > {} card.json > ...
```

```
1 {
2   "balance": 80000,
3   "card_num": 11111
```

```
build > {} item14.json > ...
```

```
1 {
2   "item_code": 14,
3   "item_num": 99,
4   "item_price": 2000
5 }
```

```
uild > {} item14.json > ...
```

```
1 {
2   "item_code": 14,
   "item_num": 89,
   "item_price": 2000
```




Brute Force Test 수정

Test	Test #	Description	Expected output	Test output	P / F
선결제 음료 선택	1 - 1	타 자판기의 재고보다 작은 수량 입력	선결제 여부 확인	선결제 여부 확인	P
	1 - 2	타 자판기의 재고보다 큰 수량 입력	에러 메시지 출력 및 시작 화면으로 초기화	에러 메시지 출력 및 시작 화면으로 초기화	P
선결제 여부 확인	2 - 1	소문자 y 입력	카드 번호 입력 요구	문자 입력 요구	F
	2 - 2	소문자 n 입력	시작 화면으로 초기화	문자 입력 요구	F
	2 - 3	대문자 y 입력	카드 번호 입력 요구	카드 번호 입력 요구	P
	2 - 4	대문자 n 입력	시작 화면으로 초기화	시작 화면으로 초기화	P
	2 - 5	숫자 및 다른 문자 입력	y 또는 n 입력 요구	문자 입력 대기	F
	2 - 6	대문자 Y 또는 N 사이에 공백 입력	y 또는 n 입력 요구	문자 입력 대기	F
선결제 카드번호 입력	3 - 1	잘못된 숫자열 입력	결제 실패	결제 실패 및 시작화면으로 초기화	P
	3 - 2	문자를 포함해 입력	결제 실패	시작화면 무한 loop	F



Brute Force Test 수정

숫자 및 다른 입자 입력 및 y와 n 사이에 다른 문자열 입력

```
-----
(1) 콜라 : 99개 1500원 (2) 사이다 : 99개 1500원 (3) 녹차 : 99개 1700원 (4) 홍차 : 99개 1700원 (5) 밀크티 : 99개 2000원
(6) 탄산수 : 99개 1400원 (7) 보리차 : 99개 1300원 (8) 캔커피 : 0개 1500원 (9) 물 : 0개 1000원 (10) 에너지 드링크 : 0개 2200원
(11) 유자차 : 0개 1800원 (12) 식혜 : 0개 1600원 (13) 아이스티 : 0개 1000원 (14) 딸기주스 : 0개 2000원 (15) 오렌지주스 : 0개 2000원
(16) 포도주스 : 0개 2000원 (17) 이온음료 : 0개 1700원 (18) 아메리카노 : 0개 1900원 (19) 핫초코 : 0개 1800원 (20) 카페라떼 : 0개 2100원

음료 ID와 갯수를 입력해주세요. (e.g. 3 2)

-----
20 20
-----
치 x: %d 결제를 원하시면 'y', 원하지 않으시면 'n'를 입력해주세요
결제 여부:
-----
YON
올바르지 않은 입력입니다. y나 n을 입력하세요
y
-----
카드번호를 입력해주세요
-----
11111
-----
결제에 성공했습니다
-----
x좌표: 10, y좌표: 10, 인증번호: WvrdJ
-----
-----
어서오세요.
시작하려면 's'를 입력하세요.
(선결제 인증코드 입력을 원하시면 "인증코드"를 입력하세요...)
-----
-----
어서오세요.
시작하려면 's'를 입력하세요.
(선결제 인증코드 입력을 원하시면 "인증코드"를 입력하세요...)
-----
```

```
-----
(1) 콜라 : 99개 1500원 (2) 사이다 : 99개 1500원 (3) 녹차 : 99개 1700원 (4) 홍차 : 99개 1700원 (5) 밀크티 : 99개 2000원
(6) 탄산수 : 99개 1400원 (7) 보리차 : 99개 1300원 (8) 캔커피 : 0개 1500원 (9) 물 : 0개 1000원 (10) 에너지 드링크 : 0개 2200원
(11) 유자차 : 0개 1800원 (12) 식혜 : 0개 1600원 (13) 아이스티 : 0개 1000원 (14) 딸기주스 : 0개 2000원 (15) 오렌지주스 : 0개 2000원
(16) 포도주스 : 0개 2000원 (17) 이온음료 : 0개 1700원 (18) 아메리카노 : 0개 1900원 (19) 핫초코 : 0개 1800원 (20) 카페라떼 : 0개 2100원

음료 ID와 갯수를 입력해주세요. (e.g. 3 2)

-----
10 10
-----
치 x: %d 결제를 원하시면 'y', 원하지 않으시면 'n'를 입력해주세요
결제 여부:
-----
yon
올바르지 않은 입력입니다. y나 n을 입력하세요
y
-----
카드번호를 입력해주세요
-----
```



CHAPTER 02 🔍

Brute Force Test 수정

소문자 n 입력

```
-----
어 서 오 세 요 .
시 작 하 려 면 's'를 입 력 하 세 요 .
(선 결 제 인 증 코 드 입 력 을 원 하 시 면 "인 증 코 드 "를 입 력 하 세 요 ...)
-----
s
-----
(1) 콜 라 : 99개 1500원 (2) 사 이 다 : 99개 1500원 (3) 녹 차 : 99개 1700원 (4) 홍 차 : 99개 1700원 (5) 밀 크 티 : 99개 2000원
(6) 탄 산 수 : 99개 1400원 (7) 보 리 차 : 99개 1300원 (8) 캔 커 피 : 0개 1500원 (9) 물 : 0개 1000원 (10) 에 너 지 드 링 크 : 0개 2200원
(11) 유 자 차 : 0개 1800원 (12) 식 혜 : 0개 1600원 (13) 아 이 스 티 : 0개 1000원 (14) 딸 기 주 스 : 0개 2000원 (15) 오 렌 지 주 스 : 0개 2000원
(16) 포 도 주 스 : 0개 2000원 (17) 이 온 음 료 : 0개 1700원 (18) 아 메 리 카 노 : 0개 1900원 (19) 핫 초 코 : 0개 1800원 (20) 카 페 라 떼 : 0개 2100원

음 료 ID와 갯 수 를 입 력 해 주 세 요 . (e.g. 3 2)

-----
20 20
-----
치 x: %d 결 제 를 원 하 시 면 'y', 원 하 지 않 으 시 면 'n'를 입 력 해 주 세 요
결 제 여 부 :
-----
n
-----
어 서 오 세 요 .
시 작 하 려 면 's'를 입 력 하 세 요 .
(선 결 제 인 증 코 드 입 력 을 원 하 시 면 "인 증 코 드 "를 입 력 하 세 요 ...)
-----
```



CHAPTER 02 🔍

Brute Force Test 수정

카드 번호 입력 시 문자를 포함해 입력 오류

카드 번호를 입력해 주세요

bbcde

유효하지 않은 입력입니다. 숫자만 입력하세요.



Brute Force Test 수정

Test	Test #	Description	Expected output	Test output	P / F
선결제 카드번호 입력	3 - 3	잔액 충분한 상태에서 정확한 숫자열 입력	결제 성공, 타 자판기 위치 반환, 인증번호 출력	결제 성공, 타 자판기 위치 반환, 인증번호 출력	P
	3 - 4	잔액 부족한 상태에서 정확한 숫자열 입력	결제 실패	결제 성공	F
인증코드 입력	4 - 1	정확한 코드 입력	상품 반환	상품 반환	P
	4 - 2	이전에 사용한 코드 입력	에러 메시지 출력	에러 메시지 출력	P
	4 - 3	틀린 코드 입력	에러 메시지 출력	에러 메시지 출력 및 시작화면으로 초기화	P
Interrupt	5 - 1	client connected 메시지 출력 중 재고가 있는 음료 선택	카드번호 입력 요구	카드번호 입력 요구	P
	5 - 2	client connected 메시지 출력 중 재고가 부족한 음료 선택	선결제 여부 확인	선결제 여부 확인	P
	5 - 3	client connected 메시지 출력 중 '인증 코드' 입력	인증코드 입력 요구	인증코드 입력 요구	P



Brute Force Test 수정

카드번호를 입력하세요

11111

결제에 실패했습니다.
카드번호나 잔액을 확인하세요

어서오세요.
시작하려면 's'를 입력하세요.
(선결제 인증코드 입력을 원하시면 "인증코드"를 입력하세요...)

_



Static Analysis 수정

File	Line	Severity	Review
Bank.cpp	57	Medium	원인: 멤버 변수를 수정하지 않는 함수에 const 키워드가 선언되지 않았음. 대처: 기능이 아닌 Notation의 문제이므로 수정을 요구하지 않음.
CertCode.cpp	18	Low	원인: 클래스 인자를 불필요하게 명시. 대처: 가독성 문제이므로 여유 있을 때 수정 요구.
CertCodeManager.cpp	19	Medium	원인: 함수에 객체를 값으로 전달하여 오버헤드 발생. 대처: const를 사용하여 참조하는 방식으로 수정 요구.
	46	Medium	원인: 예외처리의 범위가 구체적으로 특정되지 않음. 대처: 주석으로라도 예외의 경우 명시 요구.
	46	Low	원인: Parameter의 미사용 지적. 대처: parameter 삭제가 가독성 저하를 일으킬 수 있으므로 수정을 요구하지 않음.
Input.cpp	8, 20, 38, 48, 55	Medium	원인: 멤버 변수를 수정하지 않는 함수에 const 키워드가 선언되지 않았음. 대처: 기능이 아닌 Notation의 문제이므로 수정을 요구하지 않음.
MSG.cpp	45	High	원인: 경로 변수를 const로 설정하지 않음. 대처: 경로를 변경하는 경우가 없으면 수정 요구.
	50	Medium	원인: 함수에 Parameter가 많음 대처: 수정을 요구하지 않음.



Static Analysis 수정

File	Line	Severity	Review
MSG.cpp	69, 79, 89, 99	Medium	원인: 예외처리의 범위가 구체적으로 특정되지 않음. 대처: 주석으로라도 예외의 경우 명시 요구.
	114	High	원인: Global variable을 const로 선언하지 않음. 대처: 수정 요구.
	141	Low	원인: Parameter의 미사용 지적. 대처: Parameter 삭제가 가독성 저하를 일으킬 수 있으므로 수정을 요구하지 않음.
	231, 299, 451	Medium	원인: long을 int로 바꾸며 정확성 저하. 대처: 꼭 int를 써야하는게 아니면 long으로 수정 요구.
	235	Blocker	원인: Index가 범위를 벗어날 위험이 높음. 대처: Index의 범위를 제한하는 코드 추가 요구.
	300	Low	원인: 조건문에 사용되는 변수를 조건문에서 선언하지 않고 외부에서 선언함. 대처: 외부에서 선언하는 것이 더 이해하기 쉬울 수 있어 수정을 요구하지 않음.
	374, 410	Low	원인: 변수 미사용 대처: 변수 삭제 요구
ShowScreen.cpp	16, 119	Medium	원인: 함수에 객체를 값으로 전달하여 오버헤드 발생. 대처: const를 사용하여 참조하는 방식으로 수정 요구.



CHAPTER 03 🔍

Static Analysis 수정

Severity ?	
Blocker	1
High	6
Medium	65
Low	22
Info	0

총 : 94개

Severity ?	
Blocker	0
High	3
Medium	50
Low	9
Info	0

총 : 62개



개인적인 소감 - 조영탁

객체지향이 무엇인지에 대해서 포괄적으로 배울 수 있었다.
아키텍처적, 디자인적, 코드적 객체지향 수준이 존재한다는 것을 알 수 있었고,
각각의 적용 방법에 대해서 학습 및 일부를 적용해볼 수 있었다.

또한 코드를 작성하기 전,
OOAD 과정을 거치면서 협업가능하고 체계적으로 프로젝트를 진행할 수 있게 되었다.

방학중에 이번에 다루지 않았던,
아키텍처 리스크 관리와 Dev/Ops 과정을 학습하여, 나만의 프로젝트를 만들어보고자 한다.



개인적인 소감 - 김종혁

평소 개발을 시작하기 전에 문서화를 해본 경험이 별로 없었는데, 잘 정리된 문서가 조직 내에서 개발을 할 때 어떤 식으로 도움이 되는지 직접 경험해볼 수 있는 좋은 시간이었다. 충분히 고민한 usecase의 경우 구현 단계가 빠르고 용이하게 완료되었지만, usecase를 엄밀하지 못하게 작성하거나 무시할 때 발생하는 파급효과를 경험해보면서 디자인 단계도 개발과 동등하게 중요하다는 것을 깨달았다.

다만 OOAD에 미숙할 시 완벽하지 않은 디자인으로 구현단계에 들어갈 확률이 높을 것 같다.

이 경우 디자인 단계를 다시 갈아엎어야 하는 리스크가 발생하게 되므로

OOAD를 적용하기 위해

조직 내 인원들이 해당 방법론에 대한 충분한 훈련이 필요할 것 같다는 요구사항이 존재하는 것 같다.

검증팀의 존재로 미처 생각하지 못한 문제나 개선사항을 지적받은 점도 매우 값진 경험이었다.

iteration을 더 많이 돌고, c++이 훈련이 된 상태로 프로젝트에 참가했으면 좋았을 걸 이라는 아쉬움이 남는다.

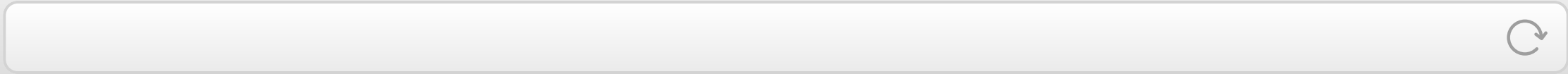


개인적인 소감 - 정석현

중간고사 이전에 한 Use-Case 다이어그램이나 Class 다이어그램과 같이 이전까지는 코드를 짜기 전까지는 이런 일련의 작업이 거의 하지 않았기에 이러한 작업이 코드를 작성하는 데 도움이 된다는 것을 느꼈다.

하지만 코드를 구현하면서 부터는 소켓 통신도 잘 모르고 C++도 사용해보지 않았고 할 수 있는 것이 코드 읽기 정도였고 Unit Code Test는 아직도 어떻게 하는지 모를 정도로 따라가기가 버거웠다 그래서 조금 더 많은 것을 알고 왔다면 조금 더 도움이 되지 않았을까라는 생각이 든다.

그래도 이러한 경험을 통해서 이번에 그린 Diagram을 바탕으로 코드를 짤 때와 검증팀이 테스트해주는 것을 바탕으로 디자인하고 코드를 작성하는 게 생각보다 신경써야 하는 게 많다는 것을 이번 기회에 느끼게 되었다.



감사합니다