

단국대학교 REPORT

제목 : c++상점 관리 프로그램 (중간고사 대체과제)

과목명 : 객체지향프로그래밍(sw)

교수명 : 송인식 교수님

이름 : 황예진

학번 : 32195044

소속학과/학년 : 소프트웨어학과 / 2학년

1. 프로그램 설명

‘상점 관리 프로그램’은 실제로 상점에서 사용할만한 기능을 모아놓은 프로그램이다. 해당 프로그램에서는 아래와 같은 기능을 제공한다. (기능별 자세한 계획은 https://github.com/Hwangyejin884/-_.git 의 README.md에 첨부한 제안서에서 확인할 수 있다.)

(1) 제품 입고 기능

- 사용법 : 메인 메뉴에서 1을 입력하면 제품명을 입력할 수 있다. 이때, 입력한 제품명이 이미 등록되어 있는 제품이라면 메인 메뉴로 돌아가고, 등록되어 있지 않은 제품이라면 가격과 수량을 입력한 뒤 등록이 진행된다.

(2) 제품명, 가격 확인 기능

- 사용법 : 메인 메뉴에서 2를 입력하면 입고된 제품의 이름과 가격을 확인할 수 있다. 이때 확인 범위를 전체 명단 / 개별 명단 중에서 선택할 수 있다. 개별 명단을 선택했다면 확인할 제품명을 입력한 뒤, 가격을 확인 할 수 있다.

(3) 재고 관리 기능

- 사용법 : 메인 메뉴에서 3을 입력하면 제품 당 남아 있는 재고를 확인할 수 있다. 만약 재고가 5개 이하라면 이름과 재고 수량 옆에 ‘품절임박’이라는 문구가 뜬다. 이때 확인 범위를 전체 명단 / 개별 명단 중에서 선택할 수 있다. 개별 명단을 선택했다면 확인할 제품명을 입력한 뒤 수량을 입력할 수 있다.

(4) 제품 판매 기능

- 사용법 : 메인 메뉴에서 4를 입력하면 판매 기능을 이용할 수 있다. 이때 구입한 제품의 종류 개수를 입력하면 해당 개수만큼 상품명 입력받기를 반복한다. 모두 입력이 완료되면 총 결제 금액과 함께 포인트 적립 기능이 실행된다. 이때, 매장의 회원이 경우, 포인트가 1000점 이상일 때 포인트를 사용할 수 있고, 결제 금액에서 사용한 포인트만큼 제외된다. 회원이 아닐 경우, 회원 가입을 원하면 이름/전화번호를 입력받고 회원 정보에 등록된다.

★ 포인트가 1000점 이상일 때 결제 금액에서 포인트를 사용할 수 있는 기능은 제안서에서 추가된 기능이다. 제안서를 기반으로 만들었지만 포인트를 적립만 하고 사용하는 기능을 계획하지 않아, 프로그램을 만들면서 추가하게 되었다.

(5) 누적 판매량 확인 기능

- 사용법 : 메인 메뉴에서 5를 입력하면 누적 판매량 확인 기능을 이용할 수 있다. 이때 확인 범위를 전체 명단 / 개별 명단 중에서 선택할 수 있다. 마찬가지로 개별 명단을 선택했다면 확인할 제품명을 입력한 뒤, 누적된 판매량을 확인할 수 있다.

(6) 고객 포인트 확인 기능

- 사용법 : 메인 메뉴에서 6을 입력하면 고객 포인트 확인 기능을 이용할 수 있다. 해당 기능에서는 무분별하게 고객 정보를 보게 하지 않기 위해 개별 확인 기능만 제공한다. 만약 고객이 원할 경우 조회할 고객의 이름을 입력받으면 포인트와 전화번호를 제공한다.

(7) 종료

- 사용법 : 메인 메뉴에서 7을 입력하면 상점 관리 프로그램을 종료한다.

전체적인 프로그램의 틀은 Store 클래스에서 각 기능에 대한 멤버 함수를 선언해 main()에서 클래스의 멤버 함수를 호출하는 방식이다. 이때, Store에서 중복되는 기능은 Base라는 부모 클래스의 멤버 함수로 미리 선언하고, Store 클래스가 Base 클래스를 상속하게 한다. 프로그램에서 요구하는 정보들은 크게 상품(Product)과 고객(Customer)로 나눌 수 있는데 이것을 구현하기 위해 구조체를 사용했다. 상품(Product) 구조체에는 제품명, 가격, 재고 수량, 누적 판매량 멤버가 있고, 고객(Customer) 구조체에는 고객명, 포인트, 전화번호 멤버가 있다.

이외의 코드에 대한 자세한 내용은 아래 코드 분석에서 설명한다.

2. 코드 분석

가장 먼저 설명할 것은 프로그램의 기본 틀이 되는 **Store.h** 부분이다. 본 프로그램에서 필요한 헤더 파일을 include 한다.

```
1 #include <iostream>
2 #include <string>
3 #include <limits>
4 using namespace std;
```

기본적인 <iostream>과 문자열 입력 및 검색을 위한 <string>, Base 클래스의 예외 처리 함수에서 사용할 INT_MAX를 위한 <limits>까지 include해준 뒤 편의성을 위해 이름 공간을 std로 선언한다.

고객과 상품의 정보를 저장할 구조체를 선언한다. 상품(Product) 구조체에는 이름(string), 가격(int), 재고 수량(int), 누적 판매량(int) 멤버가 있고, 고객(Customer) 구조체에는 이름(string), 포인트(int), 전화번호(string) 멤버가 있다. 이후 구조체 배열로 선언해 정보를 저장한다.

```

6  typedef struct { //상품 정보 구조체
7      string name; //제품명
8      int price; //가격
9      int amount; //재고 수량
10     int sales_rate; //누적 판매량
11
12 } Product;
13
14 typedef struct { //고객 포인트 구조체
15     string name; //고객명
16     int point; //포인트
17     int number; //전화번호
18 } Customer;
19

```

```

20 class Base {
21 public:
22     int return_int(int min, int max); //int형 값을 반환하는 함수
23 };
24

```

Base 클래스는 Store 클래스의 부모 클래스이다. Store클래스에서 int형 정수를 받을 때, int가 아닌 다른 형의 문자를 입력하면 생기는 오류를 제거하기 위한 return_int 멤버 함수를 가지고 있다. Store 클래스의 여러 멤버 함수에서 공통적으로 쓰이므로 클래스 상속을 이용해 하나로 묶었다. 자세한 내용은 함수 본체를 선언할 때 하겠다.

```

26 class Store : public Base { //상점 클래스 (재고, 판매 관리용)
27 public:
28     void Receiving(); //제품 입고 기능
29     void CheckProduct(); //제품명, 가격 확인 기능
30     void CheckAmount(); //재고 관리 기능
31     void Sales(); //제품 판매 기능
32     void Check_SaleProduct(); //누적 판매량 확인 기능
33     void CheckPoint(); //고객 포인트 확인 기능
34 };
35

```

본 Store 클래스는 Base 클래스를 상속받으며 프로그램 설명에서 언급한 기능을 담당할 멤버 함수들이 있다. 이 또한 자세한 내용은 각 함수의 본체를 선언할 때 한다.

```

1  #include "Store.h"
2  #define P_SIZE 30 //상품 갯수
3  #define C_SIZE 10 //고객 명수
4  #define PRICE 100000 //상품 1개의 최고 가격
5  #define AMOUNT 100 //상품 종류당 최고 수량
6
7  Product product[P_SIZE]; //30개의 상품 정보
8  Customer customer[C_SIZE];
9  int pro_index = 0; //초기 상품 정보 인덱스 0으로 초기화
10 int customer_index = 0; //초기 고객 정보 인덱스 0으로 초기화
11

```

다음은 클래스의 멤버 함수를 구체화하는 선언부 작성이다. Store.cpp에는 멤버 함수 본체 선언, 구조체 배열 선언 등의 내용이 있다.

다음 내용은 Store.h를 따른다. 먼저 조건문이나 연산에서 쓰일 변동 가능한 수를 #define을 통해 설정해준다. 본 프로그램에서는 등록할 수 있는 상품 개수, 가입할

수 있는 고객 명수, 상점의 상품 당 최고 가격, 종류 당 최고 수량을 일정 수로 제한한다. 다음으론 고객, 상품을 저장하기 위해 제한된 숫자를 이용해 구조체 배열을 선언한다. 이때, 초기 상품 정보 인덱스 수는 0으로 초기화한다.

```

13 int Base::return_int(int min, int max) { //int형 반환해주는 함수
14     int select_num;
15     while (1) { //MIN, MAX 사이의 수를 입력할때까지 반복
16         cout << "   입력: ";
17         cin >> select_num;
18         if (select_num <= max && select_num >= min) { //min과 max 사이의 수라면
19             return select_num; //입력받은 select_num 값 반환
20             break;
21         }
22         else {
23             cin.clear();
24             cin.ignore(INT_MAX, '\n');
25             cout << "※※ 잘못된 입력입니다. 다시 입력하세요. ※※ " << endl << endl;
26         }
27     }
28 }
29

```

Base 클래스의 멤버 함수 return_int는 int형 입력만 처리하고, 이외의 입력을 했을 시 코드가 무한반복되는 오류를 피하기 위한 것이다. return_int 함수를 호출할 때 min, max를 지정해주고 함수 내에서 범위 내의 입력이 됐을 경우 break:로 탈출하며, 이때까지 while문을 이용해 무한 반복한다. int형이 아닌 입력을 했을 시, cin.clear()함수와 cin.ignore()함수를 이용해 초기화한 뒤 다시 입력을 받는다. 이때, 정수로 제한하기 위해 INT_MAX를 ignore 함수의 인자로 준다.

```

31 void Store::Receiving() { //case 1의 제품 입고 관리함수
32     string name;
33     int i = 0;
34     cin.ignore(); //getline() buffer clear 문제 예방
35     cout << "● 제품 입고 기능을 선택하셨습니다. ● " << endl;
36     cout << "[ 입고할 제품명을 입력하세요. ]" << endl;
37     cout << "제품명 : ";
38     getline(cin, name);
39     for (i = 0; i < PSIZE; i++) {
40         if (name == product[i].name) { // i 인덱스에서 값을 찾았을 시
41             cout << "※※ 이미 등록돼 있는 제품입니다. ※※ " << endl;
42             cout << "└ 수량을 변경은 판매 시에만 가능합니다. " << endl
43                 << endl << endl;
44             return;
45         }
46     }
47     cout << "■ 등록 가능한 제품명입니다. ■" << endl;
48     product[pro_index].name = name;
49     cout << "가격을 입력하세요. ";
50     product[pro_index].price = Base::return_int(0, PRICE);
51     cout << "수량을 입력하세요. ";
52     product[pro_index].amount = Base::return_int(1, AMOUNT);
53
54     cout << product[pro_index].name << "가 가격 " << product[pro_index].price <<
55         "로 책정되었습니다. 수량은 " << product[pro_index].amount << "입니다" << endl;
56     cout << endl << endl;
57     pro_index++; //기능 수행 후 인덱스 위치 +1
58     return;
59 }

```

main함수의 swtich문에서 case 1에 해당하는 함수이다. 제품명은 공백이 있을 수

있으므로 getline을 이용해 입력을 받는다. 입력된 제품명을 for문을 통해 구조체 배열 확인을 PSIZE(상품 최대 개수)만큼 반복한다. if문을 통해 배열에 있는 상품을 입력했을 시 메인 메뉴로 돌아가고, 배열에 없는 상품을 입력했다면 가격과 수량을 입력한다. 이때, int형만 입력받아야 하므로 Base 클래스의 return_int 멤버 함수를 이용한다. 가격의 범위는 0원 ~ PRICE(최대 가격), 수량은 1개 ~ AMOUNT(최대 수량)으로 설정한다. 모두 입력이 끝났으면 입고된 제품의 정보를 띄운 후 상품 구조체 배열의 초기 인덱스를 +1 한 뒤 함수를 끝낸다.

```

62 void Store::CheckProduct() { //case 2의 제품명/가격 확인 기능
63     int select_num;
64     string select_name;
65     cout << "● 제품명 / 가격 확인 기능을 선택하셨습니다. ●" << endl;
66     cout << "[ 원하시는 기능의 번호를 입력해주세요. ]" << endl;
67     cout << "1. 전체 명단 확인 " << endl;
68     cout << "2. 개별 명단 확인 " << endl;
69     select_num = Base::return_int(1, 2);
70
71     switch (select_num) {
72     case 1: //전체 명단 확인
73         cout << " [ 이름          가격      ] " << endl;
74         for (int i = 0; i < PSIZE; i++)
75             cout << "      " << product[i].name << "      " << product[i].price << endl;
76         break;
77     case 2: //개별 명단 확인
78         cin.ignore();
79         cout << "[ 확인할 제품명을 입력하세요. ]" << endl;
80         cout << "name : ";
81         getline(cin, select_name);
82         cout << " [ 이름          가격      ] " << endl;
83         for (int i = 0; i < PSIZE; i++) {
84             if (select_name == product[i].name) {
85                 cout << "      " << product[i].name << "      "
86                     << product[i].price << endl << endl;
87                 break;
88             }
89         }
90         break;
91     default:
92         cout << "※※ number는 1, 2 만 입력하실 수 있습니다. ※※" << endl;
93     }
94 }

```

main함수의 switch문에서 case 2에 해당하는 함수이다. 전체 명단, 개별 명단 중 선택하기 위해 Base클래스의 return_int를 사용해주고 case문을 이용해 각 번호에 맞는 문을 실행한다. 전체 명단 확인은 for문을 사용해 상품 구조체 배열 전체를 출력하고, 개별 명단 확인은 getline으로 입력받은 string 변수와 배열을 대조해 맞는 가격을 함께 출력한다.


```

97 void Store::CheckAmount() { //case3의 재고 관리 기능
98     int select_num;
99     string select_name;
100     cout << "● 재고 관리 기능을 선택하셨습니다. ●" << endl;
101     cout << "[ 원하시는 기능의 번호를 입력해주세요. ]" << endl;
102     cout << "1. 전체 명단 확인 " << endl;
103     cout << "2. 개별 명단 확인 " << endl;
104     select_num = Base::return_int(1, 2);
105
106     switch (select_num) {
107     case 1: //전체 명단 확인
108         cout << " [ 이름          재고 ] " << endl;
109         for (int i = 0; i < PSize; i++) {
110             cout << " " << product[i].name << " " << product[i].amount;
111             if (product[i].amount <= 5 && product[i].amount != 0) //이부분도 수정
112                 cout << " ★품질임박★ " << endl;
113             else
114                 cout << endl;
115         }
116         break;

```

main함수의 switch문에서 case 3에 해당하는 함수이다. 마찬가지로 전체 명단, 개별 명단 선택을 위해 return_int 함수를 호출하고, 전체 명단은 for문을 이용해 PSize 만큼 반복해 전체 명단을 출력한다. 이때, for문 안에 if문으로 해당 i 인덱스의 amount가 5개 이하일 때, 재고 수량 옆에 ★품질임박★ 이라는 문구를 띄운다.

```

117     case 2: //개별 명단 확인
118         cin.ignore();
119         cout << "[ 확인할 제품명을 입력하세요. ]" << endl;
120         cout << "name : ";
121         getline(cin, select_name);
122         cout << " [ 이름          재고 ] " << endl;
123         for (int i = 0; i < PSize; i++) {
124             if (select_name == product[i].name) {
125                 cout << " " << product[i].name << " " << product[i].amount;
126                 if (product[i].amount <= 5)
127                     cout << " ★품질임박★ " << endl;
128                 else
129                     cout << endl;
130                 break;
131             }
132         }
133         break;
134     default:
135         cout << "※※ 1, 2 만 입력하실 수 있습니다. ※※" << endl;
136     }
137 }

```

개별 명단의 구조는 (1) 입고 기능과 동일하다. 추가된 것은 ★품질임박★문구를 위한 if문이다.

```

140 void Store::Sales() { //case4의 판매 기록 관리
141     int howMany, save_price = 0, how_amount, use_point;
142     string what_Pname, what_Cname, what_number;
143     char yes_no;
144
145     cout << "● 판매 기록 관리 기능을 선택하셨습니다. ●" << endl;
146     cout << "[ 현재 고객님의서 몇 종류의 상품을 구매하셨나요? ]";
147     cout << "종류: ";
148     howMany = Base::return_int(1, PSIZE);
149     for (int i = 0; i < howMany; i++) { //구매한 종류의 갯수만큼 반복
150         cout << "[ 구매하신 상품명을 입력하세요. ]" << endl;
151         cout << "Product Name : "; cin >> what_Pname;
152
153         while (1) { //제대로 된 상품명을 입력할 때까지 반복
154             for (int j = 0; j < PSIZE; j++) { //상품명 찾기
155                 if (what_Pname == product[j].name) { //상품명 찾기 성공
156                     cout << "[ " << product[j].name << "의 수량을 입력하세요. ]"
157                         << endl;
158                     how_amount = Base::return_int(1, AMOUNT);
159                     product[j].amount -= how_amount; //재고 수량 빼기
160                     product[j].sales_rate += how_amount; //판매 기록 저장
161                     save_price += product[j].price * how_amount; //가격*수량 저장
162                     break;
163                 }
164             }
165             break;
166         }
167     }

```

main함수의 switch문에서 case 4에 해당하는 함수이다. 구매한 종류의 수를 입력받고, 종류의 수만큼 for문을 반복한다. 그런 다음 while(1)문을 이용해 등록되어 있는 상품명을 입력할 때까지 반복, 맞는 상품명을 찾았다면 return_int()로 수량을 입력받고 그 수만큼 재고 수량에서 빼기, 판매 기록에 더하기, 미리 선언해둔 save_price변수에 가격*수량을 저장한다.

```

168     cout << "■ 총 금액은 " << save_price << "원입니다. ■" << endl;
169     cout << "■ 저희 매장은 총 구매 금액의 5%를 포인트로 적립합니다. ■" << endl;
170     cout << "■ 포인트가 1000점 이상이면 현금처럼 사용하실 수 있습니다. " << endl;
171     cout << "● 저희 매장의 회원이십니까? ( Y / N ) : "; cin >> yes_no;
172     if (yes_no == 'Y') { //매장 회원일 때
173         cout << "[ 이름을 입력해주세요 ]" << endl << " 이름: "; cin >> what_Cname;
174         for (int i = 0; i < CSIZE; i++) {
175             if (what_Cname == customer[i].name) {
176                 if (customer[i].point >= 1000) {
177                     cout << "● 포인트를 사용하시겠습니까? ( Y / N ) : " << endl;
178                     cin >> yes_no;
179                     if (yes_no == 'Y') {
180                         cout << customer[i].name << "님의 포인트는 " << customer[i].point
181                             << "점 입니다. 얼마나 사용하시겠습니까? : ";
182                         use_point = Base::return_int(1, customer[i].point);
183                         cout << "■ 결제하실 금액은 " << save_price - use_point
184                             << "원입니다. ■" << endl;
185                         customer[i].point -= use_point;
186                     }
187                     else if (yes_no == 'N')
188                         cout << "■ 결제하실 금액은 " << save_price << "원입니다. ■" << endl;
189                 }
190             }
191             else
192                 cout << "■ 결제하실 금액은 " << save_price << "원입니다. ■" << endl;
193             customer[i].point += (save_price * 0.05);
194             cout << save_price * 0.05 << " 적립되어 " << customer[i].name
195                 << "님의 포인트가 " << customer[i].point << "점이 되었습니다."
196                 << endl << "☆☆☆☆ 구매해주셔서 감사합니다. ☆☆☆☆" << endl << endl;
197             return;
198         }
199     }
200 }

```


가격 누적이 완료되면 총 금액을 뺀 뒤 포인트 적립을 위해 if문을 실행한다. 다중if문과 회원정보를 찾기 위한 for문은 앞서 설명한 포맷과 동일하고, 포인트를 사용할 때는 사용할 포인트(use_point)만큼 고객 구조체 배열에서 뺀다.(사용 내역) 마지막으로 최종 결제할 금액을 출력한 뒤 함수를 종료한다.

```

201 else if (yes_no == 'N') { //매장 회원이 아닐 때
202     cout << "● 회원가입을 원하십니까?( Y / N ) : "; cin >> yes_no;
203     if (yes_no == 'Y') {
204         cout << "[ 고객님의 이름과 전화번호를 입력하세요. ]" << endl;
205         cout << "Name : "; cin >> customer[customer_index].name;
206         cout << "Number : "; cin >> customer[customer_index].number;
207         customer[customer_index].point += save_price * 0.05;
208         cout << save_price * 0.05 << "포인트가 적립되어 "
209             << customer[customer_index].name
210             << "님의 포인트가 " << customer[customer_index].point
211             << "점이 되었습니다." << endl;
212         customer_index++;
213         cout << "■ 결제하실 금액은 " << save_price << "원입니다. ■" << endl;
214         cout << "☆☆☆☆ 구매해주셔서 감사합니다. ☆☆☆☆" << endl << endl;
215         return;
216     }
217     else {
218         cout << "■ 결제하실 금액은 " << save_price << "원입니다. ■" << endl;
219         cout << "☆☆☆☆ 구매해주셔서 감사합니다. ☆☆☆☆" << endl << endl;
220         return;
221     }
222 }
223 }
224

```

매장 회원이 아닐 때는 회원가입을 원하는지, 원하지 않는지에 따라 결과가 달라진다. 원하지 않으면 결제 금액을 출력한 뒤 그대로 종료하지만, 회원 가입을 원하면 이름과 전화번호를 입력받아 고객 구조체 배열의 customer_index 위치에 저장 후, 해당 위치의 point 멤버에 포인트를 적립한 뒤 종료한다. (당일 구매해서 얻은 포인트는 해당 결제에서 사용할 수 없다.)

```

226 void Store::Check_SaleProduct() { //case5의 누적 판매량 확인 기능
227     int select_num;
228     string select_name;
229     cout << "● 누적 판매량 확인 기능을 선택하셨습니다. ●" << endl;
230     cout << "[ 원하시는 기능의 번호를 입력해주세요. ]" << endl;
231     cout << "1. 전체 명단 확인 " << endl;
232     cout << "2. 개별 명단 확인 " << endl;
233     select_num = Base::return_int(1, 2);
234
235     switch (select_num) {
236
237     case 1: //전체 명단 확인
238         cout << " 【 이름          재고          】 " << endl;
239         for (int i = 0; i < PSize; i++)
240             cout << product[i].name << "          " << product[i].sales_rate << endl;
241         return;
242     }
243

```

main함수의 switch문에서 case 5에 해당하는 함수이다. 이 함수도 위에서 설명한 전체 명단 / 개별 명단으로 나누어지는 기능에 구조가 동일하므로 설명은 생략한다.

```

243 case 2: //개별 명단 확인
244     cin.ignore();
245     cout << "[ 확인할 제품명을 입력하세요. ]" << endl;
246     cout << "name : ";
247     getline(cin, select_name);
248     cout << " [ 이름 재고 ] " << endl;
249     for (int i = 0; i < PSIZE; i++) {
250         if (select_name == product[i].name) {
251             cout << product[i].name << " " << product[i].sales_rate
252                 << endl << endl;
253             return;
254         }
255     }
256     cout << "※※ 일치하는 제품명이 없습니다. 초기화면으로 돌아갑니다. ※※"
257         << endl << endl;
258 default:
259     cout << endl;
260 }
261 }

```

이때, 일치하는 제품명이 없으면 초기화면으로 돌아간다.

```

264 void Store::CheckPoint() { //고객 포인트 확인 기능
265     string what_name;
266     cout << "● 고객 포인트 확인 기능을 선택하셨습니다. ●" << endl;
267     cout << "※ 고객의 포인트는 개별 확인 기능만 지원합니다. ※" << endl;
268     cout << "[ 조회하실 고객의 이름을 입력해주세요. ]" << endl;
269     cout << "Name : "; cin >> what_name;
270     for (int i = 0; i < CSIZE; i++) {
271         if (what_name == customer[i].name)
272             cout << "Name : " << customer[i].name << " Number : " << customer[i].number
273                 << " Point : " << customer[i].point << endl;
274         return;
275     }
276     cout << "일치하는 고객이 없습니다. " << endl << endl;
277 }

```

main함수의 switch문에서 case 6에 해당하는 함수이다. 프로그램 설명에서 언급했듯이 고객 포인트와 정보 조회는 개별만 가능하므로, 조회를 원하는 고객명을 입력하면 for문으로 구조체 배열에서 find 후 찾기에 성공하면 결과를 출력한다.

마지막으로 **main.cpp**이다. 이곳에서는 본 프로그램을 실행하며, 7을 입력해 종료를 선택할 때까지 무한 반복된다.

```

4 int main() {
5     int select_num; //기능 선택용 번호
6
7     Store yejin_store; //yejin_store 객체 생성
8

```

역시 Store.h를 따른다. main함수에서는 기능 선택용 변수를 선언하고 Store 클래스의 객체 yejin_store를 생성한다. (yejin은 제작자의 이름)

```

9   while (true) {
10      cout << endl << "★★★★ 판매/재고 관리 ★★★★★" << endl;
11      cout << "          [ 상점용 ]          " << endl;
12      cout << "본 프로그램은 상점용 판매/재고 관리 프로그램입니다." << endl;
13      cout << "1. 제품 입고 관리" << endl;
14      cout << "2. 제품명/가격 확인" << endl;
15      cout << "3. 재고 수량 확인" << endl;
16      cout << "4. 제품 판매 및 포인트 적립" << endl;
17      cout << "5. 제품 당 누적된 판매량 확인" << endl;
18      cout << "6. 고객 포인트 관리" << endl;
19      cout << "7. 프로그램 종료 " << endl;
20      cout << "사용할 기능의 번호를 입력해주시오." << endl;
21      select_num = Base::return_int(1, 7);
22  }
23
24

```

프로그램의 메인 메뉴 인터페이스이다. 역시 return_int(1, 7) 함수를 이용해 번호를 입력받는다.

```

25  switch (select_num) {
26  case 1: //제품 입고 기능
27      yejin_store.Receiving();
28      break;
29
30  case 2: //제품명과 가격을 확인하는 기능
31      yejin_store.CheckProduct();
32      break;
33
34  case 3: //수량을 확인하는 재고 관리 기능
35      yejin_store.CheckAmount();
36      break;
37
38  case 4: //제품 판매 및 포인트 적립 기능
39      yejin_store.Sales();
40      break;
41
42  case 5: //제품 당 누적된 판매량을 보여주는 기능
43      yejin_store.Check_SaleProduct();
44      break;
45
46  case 6: //고객ID의 포인트를 확인하는 기능
47      yejin_store.CheckPoint();
48      break;
49
50  case 7: //프로그램 종료
51      cout << "    이용해주셔서 감사합니다." << endl;
52      cout << "    상점 관리 프로그램을 종료합니다." << endl;
53      return 0;
54
55  default:
56      cout << "없는 기능의 번호를 입력하셨습니다." << endl;
57      cout << "초기 화면으로 되돌아갑니다.";
58      break;
59  } //switch 괄호
60  } //while 괄호
61  return 0;
62  } //main 괄호

```

1~7 사이의 수를 case문을 통해 나누어 실행한다. 각 기능에 맞는 Store클래스의 멤버함수를 호출해 실행한다. 혹시나 오류를 방지해 default문을 작성한다.

3. 마무리

중간대체과제로 만들게 된 프로그램이지만 ‘상점 관리 프로그램’을 제작하면서 제안서의 중요성을 다시금 깨닫게 되었다. 코드를 작성하면서 지금까지 객체지향프로그래밍에서 배웠던 수많은 기능들을 제대로 다 활용하지 못한 것 같아 아쉬움이 남지만, 이 시도를 기회 삼아 다음 프로젝트에서는 다양한 기능을 색다른 방법으로 제작해보고 싶은 흥미가 생겼다. 완성 후 코드를 실행해보면서 사용자 인터페이스 또한 프로그램 제작 시 중요한 요소 중 하나인 것을 느꼈다. 작은 기호지만 구역을 나누었을 때, 사용자는 더 편하게 프로그램을 사용할 수 있었다.