1. 해결과제

매장 포스 단말기, 즉 물품의 구매, 관리를 할 수 있는 프로그램을 구현을 해결 과제로 한다. 이를 클래스, 유도클래스, 파일입출력, 템플릿 등을 이용해 구현하도록 한다.

2. 프로그램 설명

우선 프로그램 구현 시점이 아닌 사용자의 시점에서 기능 설명하겠다.

프로그램은 크게 고객 모드/ 관리 모드로 볼 수 있다.

프로그램 시작 시에는 파일을 읽어오고 종료시에는 수정된 사항들을 파일에 저장하고 종료한다.

고객 모드

1) 상품 선택
   고객 모드에서는 콘솔 창에 메뉴를 나열하고 고객이 구매를 원하는 제품을 선택하도록 한다. 이 때, 메뉴는 유도클래스를 사용하여 같은 품목의 상품을 나열 해놓는다. 품목을 정리해 놓고, 품목 안에 있는 상품들을 선택하여 장바구니에 선택한 상품 개수만큼의 가격을 장바구니에 담아둔다.

2) 계산
   구매를 할 때에는 상품 선택 할 때에 장바구니에 담아둔 총 금액을 출력한다. 그리고 구매를 종료한다.

3) 재고, 판매량 수정
   구매를 종료하면 상품의 재고를 구매한 제품 수만큼 감소하고, 판매량 변수를 증가시킨다. 다시 main 함수로 돌아간다.

4) 검색

   상품의 이름으로 검색을 한다.

관리자 모드

1) 메뉴 선택
   콘솔창에 관리자가 관리할 메뉴를 출력해 놓는다. 메뉴에는 품목 추가/삭제, 상품 추가/삭제, 통계와 같은 메뉴가 존재한다.

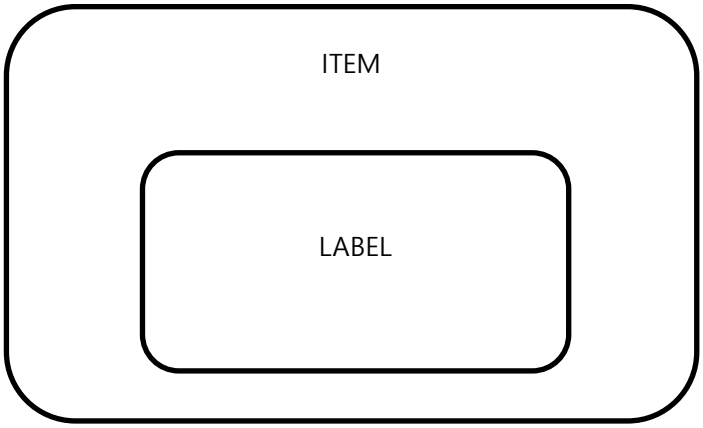
품목 추가/삭제 : 고객 모드에서 설명했던 대로 유도클래스를 통해 정의하고 생성함수를 통해 생성과 삭제를 한다.
상품 추가/삭제 : 상품도 클래스로 구현하여 이름과 가격, 원가 등의 정보를 담아둘 수 있도록 한다.

상품 수정 : 상품의 가격, 재고 등을 수정한다.
통계 : 통계에서는 제품의 수량과 수익, 순수익 등을 계산할 수 있는 창을 출력한다.

3. 자료구조

ITEM클래스는 LABEL클래스를 상속, 즉 LABEL클래스를 기반 클래스로 두고 ITEM클래스는 유도 클래스가 된다.

이를 도식화 하면



이와 같은 그림이 된다.

1) Label.h

class LABEL : 품목 클래스

string L_name : 품목 이름

void label_setname(string name) : 품목 이름 초기화

void print_label() : 품목 이름 출력

string get_label() : 품목 이름 반환

2) Item.h

class ITEM : 상품 클래스, 품목 클래스를 상속받음

string name : 상품의 이름

string code : 상품의 코드

int price : 상품의 가격

int c_price : 상품의 원가

int stock : 상품의 재고

int sales_volume : 상품의 판매량

void item_setname() : 이름 초기화 함수

void item_setcode() : 코드 초기화 함수

void item_setprice() : 가격 초기화 함수

void item_setc_price() : 원가 초기화 함수

void item_setstock() : 재고 초기화 함수

void item_sales_volume() : 판매량 초기화 함수 (판매량 = 0으로 초기화)

int profit() : 상품의 판매 수익

int net_profit() : 상품의 순이익

void print_item() : 상품의 정보 출력

int buy() : 상품 구매

int get_st() : 상품의 재고 출력

int get_sv() : 상품의 판매량 출력

void read(ifstream& in) : 파일 읽기

void write(ofstream& out) : 파일 저장

void edit() : 수정할 데이터 입력

string get_name() : 상품 이름 반환

3) Main.cpp

vector<string> n_label : c++에서 제공하는 std(Standard Template Library)을 사용하는 것으로 링크드 리스트와 유사한 방식으로 작동한다. 품목의 이름들을 저장하는 벡터.

vector<Item*> l_items : Item 클래스 객체를 저장하는 벡터.(동적할당을 위해 포인터형식으로 받는다)

void add_label(int a) : 품목 추가 함수

void show_label(int a) : 품목 출력 함수

int search_label(string name) : 품목 검색 함수, 해당 품목의 인덱스를 반환한다.

int search_item(string name) : 물품 검색 함수, 해당 물품의 인덱스를 반환한다.

void stat() : 통계 함수, 품목 -> 물품 -> 해당 물품의 통계 순의 출력

int serch_item_label(string name, int l) : 품목 삭제시 해당 품목에 해당하는 상품을 검색

void delete_label(int a) : 품목 삭제 함수

void add_item() : 상품 추가 함수

void delete_item() : 물품 삭제 함수

void edit_item() : 물품 정보 수정 함수

void print_all() : 품목, 물품을 전부 출력

4. 실행화면

5. 조원 역할

이문열 :

진태영 :

김현화 :

6. 결론 및 느낀 점

c에 익숙한 저희에게 클래스, 템플릿, 상속, 객체 등은 낯선 개념이었습니다. 본과제를 하며 목표로 정한 프로그램의 구현을 위해 그 개념들을 공부를 하며, 프로그램을 작성해야 했습니다. 클래

스 객체가 어떻게 생성이 되고 상속을 받는지를 코딩을 하며 에러가 왜 뜨는지 왜 작동이 안되는지를 생각해보고 또 찾아보며 많은 공부가 되었습니다. stl이라는 c++에서 제공해는 템플릿을 사용하여 템플릿에 대해서도 익숙해질 수가 있었습니다. 그리고 이를 팀프로젝트로 협업하여 해 보니 실제 실무에서 대규모의 프로젝트를 할 때 각자 어떤 파트를 어떻게 구현해야 서로 합칠 때 편하고 효율이 좋은지도 이번 기회를 통해 알게 되었습니다. 그리고 상속이라는 개념을 다시 한 번 다지고 구현을 해보니 어떤 상황에서 상속을 써야 적절한지를 알게 되었다.

7. 프로그램 코드

Label.h

```
#include <iostream>
#include <vector>
#include <fstream>
#include <string>

using namespace std;

class Label {
protected:
        string L_name;
public:
        void label_setname(string name);
        void print_label();
        string get_label();
};
```

Label.cpp

```
#include "Label.h"

void Label::print_label() {
        cout << "품목 명 : " << L_name;
}
void Label::label_setname(string name)
{
        L_name = name;
}
```

```
string Label::get_label() {
        return L_name;
}
```

Item.h

```
#include "Label.h"


class Item : public Label {
        string name;    //상품의 이름
        string code;    //상품의 코드
        int price;     //상품의 가격
        int c_price;   //상품의 원가
        int stock;     //상품의 재고
        int sales_volume;   //상품의 판매량
public:
        void item_setname();           //이름 초기화 함수
        void item_setcode();           //코드 초기화 함수
        void item_setprice();          //가격 초기화 함수
        void item_setc_price();        //원가 초기화 함수
        void item_setstock();          //재고 초기화 함수
        void item_sales_volume();      //판매량 초기화 함수 (판매량 = 0으로 초기화)
        int profit();   //상품의 판매 수익
        int net_profit();   //상품의 순이익
        void print_item();  //상품의 정보 출력
        int buy();         //상품 구매
        int get_st();       //상품의 재고 출력
        int get_sv();       //상품의 판매량 출력
        void read(ifstream& in);//파일 읽기
        void write(ofstream& out);//파일 저장
        void edit();             //데이터 입력
        string get_name();       //상품 이름 반환
};
```

Item.cpp

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include "ITEM.h"
using namespace std;

int Item::profit()
{
        int result = price * sales_volume;
        return result;
}
```

```
int Item::net_profit()
{
	int result = c_price * sales_volume;
	return result;
}

void Item::edit() {
	item_setname();
	item_setcode();
	item_setprice();
	item_setc_price();
	item_setstock();
	item_sales_volume();
}

void Item::print_item()
{
	cout << "상품명 : " << name << endl;
	cout << "가격 : " << price << endl;
	cout << "재고 : " << stock << endl;
	return;
}

int Item::buy()
{
	int temp;
	while (1)
	{
		cout << "해당 상품을 몇개 구매하시겠습니까?   ";
		cin >> temp;
		if (temp <= stock)
		{
			cout << "총 " << temp * price << "원 입니다."<<endl;
			stock = stock - temp;
			sales_volume = sales_volume + temp;
			break;
		}
		else
			cout << "재고가 부족합니다."<<endl;
	}
	return temp * price;
}
int Item::get_st()
{
	return stock;

}
int Item::get_sv()
{
	return sales_volume;
}

void Item::read(ifstream& in) {
	char buf[256];
	in.getline(buf, 256, '₩n');
```

```
        name = strtok(buf, "₩t");
        code = strtok(NULL, "₩t");
        Label::L_name = strtok(NULL, "₩t");
        price = stoi(strtok(NULL, "₩t"));
        stock = stoi(strtok(NULL, "₩n"));
}

void Item::write(ofstream& out) {
        out <<"₩n"<< name << "₩t" << code << "₩t" << Label::L_name << "₩t" << price << "₩t" <<
stock;
}

string Item::get_name() {
        return name;
}

void Item::item_setname()
{
        cout << "상품의 이름을 입력하세요. :";
        cin >> name;
}

void Item::item_setcode()
{
        cout << "상품의 코드를 입력하세요. : ";
                cin >> code;
}

void Item::item_setprice()
{
        cout << "상품의 판매 가격을 입력하세요. : ";
        cin >> price;
}

void Item::item_setc_price()
{
        cout << "상품의 원가를 입력하세요. : ";
        cin >> c_price;
}

void Item::item_setstock()
{
        cout << "상품의 재고를 입력하세요. : ";
        cin >> stock;
}

void Item::item_sales_volume()
{
        sales_volume = 0;
}
```

Main.cpp

```
#include "Item.h"
	vector<string> n_label;
	vector<Item*> l_items;


//품목 추가 함수
void add_label(int a)
{
	if (a >= 9)
	{
		cout << "더 이상 품목을 추가할 수 없습니다." << endl;
		return;
	}
	string temp;
	cout << "추가할 품목의 이름은 ? :";
	cin >> temp;
	n_label.push_back(temp);
}
//품목 출력 함수
void show_label(int a)
{
	cout << "**********" << endl;
	for (int i = 0; i < a; i++)
		cout <<i+1<<'.'<< n_label.at(i) << endl;
	cout << "**********" << endl;

}
//품목 검색 함수, 해당 품목의 인덱스를 반환
int search_label(string name)
{
	int i;
	for (i = 0; i <n_label.size();i++)
		if (n_label.at(i) == name)
			return i;
	cout << "해당 품목은 존재하지 않습니다."<<endl;
	return -1;
}
//물품 검색 함수, 해당 물품의 인덱스를 반환
int search_item(string name)
{
	for (int i=0; i <l_items.size();i++)
		if(l_items.at(i)->get_name()==name)
			return i;
	cout << "해당 상품은 존재하지 않습니다."<<endl;
	return -1;
}
//통계 함수, 품목 -> 물품 -> 해당 물품의 통계 순의 출력
void stat() {
	for (int i = 0; i < n_label.size(); i++) {
		cout << n_label.at(i) << endl;
		for (int j = 0; j < l_items.size(); j++) {
			if (n_label.at(i) == l_items.at(j)->get_label())
			{
```

```
                                cout << "상품명 : "<<l_items.at(j)->get_name() << endl;
                                cout << "재고 :   " << l_items.at(j)->get_st() << endl;
                                cout << "판매량 : " << l_items.at(j)->get_sv() << endl;
                                cout << "수익 : " << l_items.at(j)->profit() << endl;
                                cout << "순수익 : " << l_items.at(j)->net_profit() << endl;

                        }
                }
        }
}
//품목 삭제시 해당 품목에 해당하는 상품을 검색
int serch_item_label(string name, int i)
{
        for (i; i < l_items.size(); i++)
                if (l_items.at(i)->get_label() == name)
                        return i;
        return -1;
}
//품목 삭제 함수
void delete_label(int a)
{
        vector<string>::iterator iter;
        string tmp;
        int t = 0;
        int count = 0;
        int l = 0;
        if (a <= 0)
        {
                cout << "더 이상 품목을 삭제할 수 없습니다." << endl;
                return;
        }
        show_label(a);
        cout << "삭제할 품목 이름을 입력해주세요.";
        cin >> tmp;
        t = search_label(tmp);
        if (t == -1)
                return;
        n_label.erase(n_label.begin() + t);
        while (1){                                                              //지운
품목의 물품들을 삭제
                l = serch_item_label(tmp, count);
                if (l == -1)
                        break;
                delete(l_items.at(l));                                  //객체 해지
                l_items.erase(l_items.begin() + l);//벡터에서 지운 객체의 자리를 지움
                count = count + 1;
        }
}
//상품 추가 함수
void add_item()
{
        int l;
        Item *temp = new Item();
        cout << "추가하려는 상품의 정보를 입력하시오." << endl;
        temp->edit();
```

```
        cout << "상품의 품목을 고르세요" << endl;
        show_label(n_label.size());
        cin >> l;
        temp->label_setname(n_label.at(l-1));
        l_items.push_back(temp);
}
//물품 삭제 함수
 void delete_item()
{
        string tmp;
        int t;
        cout << "삭제하려는 상품을 입력하시오." << endl;
        cin >> tmp;
        t = search_item(tmp);
        if (t == -1)
                return;
        delete(l_items.at(t));          //객체 해지
        l_items.erase(l_items.begin() +t);
}
//물품 정보 수정 함수
 void edit_item() {
         string tmp;
         int t;
         cout << "수정하려는 상품을 입력하세요." << endl;
         cin >> tmp;
         t = search_item(tmp);
         if (t == -1)
                 return;
         cout << "새로운 정보를 입력하세요." << endl;
         l_items.at(t)->edit();
         cout << "수정되었습니다. " << endl;
 }
 //품목, 물품을 전부 출력
 void print_all() {
         for (int i = 0; i < n_label.size();i++) {
                 cout <<endl << n_label.at(i) << endl;
                 for (int j = 0; j < l_items.size();j++) {
                         if (n_label.at(i) == l_items.at(j)->get_label())
                                 l_items.at(j)->print_item();
                 }
         }
 }

int main() {

        n_label.reserve(10);   //품목은 최대 10개
        int total_price = 0;   //장바구니의 가격
        int command, command2;
        char yn;
        string buy_item;
        int t,k=0;
        int count = 0,cc=0;
        //파일 읽기
        ifstream in;
        ofstream out;
```

```
	in.open("itemlist.txt", ios::in);
	Item* tmp;
	char buf[256];
	in.getline(buf,256, '₩n');
	while (!in.eof()) {
		tmp = new Item();
		tmp->read(in);
		I_items.push_back(tmp);
		cc++;
	}
	in.close();

	//n_label에 품목이름을 넣어줌
	for (int i = 0; i<I_items.size(); i++) {
		count = 0;
		for (int j = 0; j < n_label.size(); j++)
			if (I_items.at(i)->get_label() == n_label.at(j))
				break;
			else
				count++;
		if (count == n_label.size())
			n_label.push_back(I_items.at(i)->get_label());
	}


	//구현부
	while (1) {
		cout << "(1) 고객모드  (2) 관리모드  (3) 종료" << endl;
		cin >> command;
		if (command < 1 && command>3)
			continue;

		else if (command == 1) {		//고객 모드
			while (1) {
				cout << "1. 상품 선택  2. 계산  3. 상품 검색" << endl;
				cin >> command2;
				if (command2 == 1) {
	//상품 선택
					print_all();
	//상품 출력
					cout << "어떤것을 구매하시겠습니까?";
					cin >> buy_item;
					t = search_item(buy_item);
					if (t != -1)
						total_price += I_items.at(t)->buy();
				}
				else if (command2 == 2) {			//계산
					cout << "총 " << total_price << "원 나왔습니다." <<
endl;
					total_price = 0;
					cout << "쇼핑을 그만하시겠습니까?[y/n]" << endl;
					cin >> yn;
					if (yn == 'y')
						break;
				}
```

```
                                else if (command2 == 3) {                       //상품
검색
                                        cout << "찾으시는 물건 이름을 입력하세요." << endl;
                                        cin >> buy_item;
                                        t = search_item(buy_item);
                                        if (t != -1)
                                                l_items.at(t)->print_item();    //해당
상품 출력
                                }

                        }
                }

                else if (command == 2) {                                //관리자모드
                        while (1) {
                                cout << "1.품목추가  2.품목삭제  3.상품추가  4.상품삭제
5.상품수정  6.통계보기  7.모드" << endl;
                                cin >> command2;
                                switch (command2) {
                                case 1:
                                        add_label(n_label.size());      //품목 추가
                                        show_label(n_label.size());         //추가한 후
현재의 품목 나열
                                        break;
                                case 2:
                                        delete_label(n_label.size());   //품목 삭제
                                        break;
                                case 3:
                                        add_item();                     //상품 추가
                                        break;
                                case 4:
                                        delete_item();                          //상품삭제
                                        break;
                                case 5:
                                        edit_item();
        //상품정보수정
                                        break;
                                case 6:
                                        stat();
        //통계보기
                                        break;
                                case 7:
                                        k = 1;
        //탈출조건
                                        break;
                                }
                                if (k == 1) break;                      //루프 탈출
                        }
                }
                else if (command == 3)                          //프로그램 종료(루프 탈출)
                        break;
        }
        //파일 저장
        out.open("itemlist.txt");
        for(int i =0; i<l_items.size();i++){
```

```
                l_items.at(i)->write(out);
        }
        out.close();
        //동적할당된 객체 해지
        for (int i = 0; i < l_items.size();i++) {
                delete(l_items.at(i));
        }
        l_items.clear();
        n_label.clear();
}
```