객체 프로그래밍 프로젝트 보고서

20190971\_김신영

## 타이틀 출력

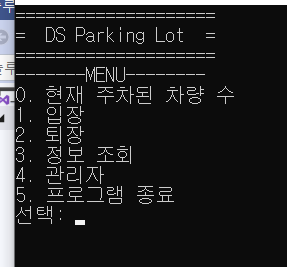
Main.cpp 11~13번째 줄

//타이틀 출력

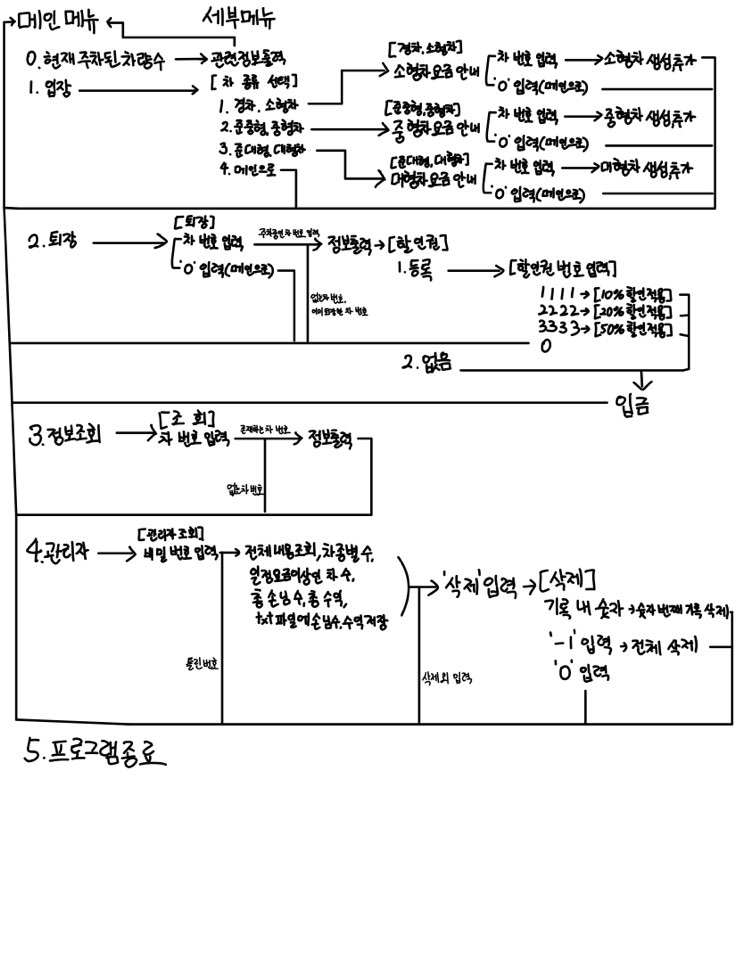
cout << "====================" << endl;

cout << "= DS Parking Lot =" << endl;

cout << "====================" << endl;



## 2. 콘솔 기반의 메뉴



0. 주차된 차량 수 조회

1. 생성, 추가

2. 주차장에서 퇴장, 할인권 입력, 입금

3. 해당 차 정보조회

4. 관리자 비밀번호, 전체 내용 조회, 관리자에 유용한 정보 조회, 파일에 내용 저장

단일 삭제, 전체 삭제

5. 프로그램 종료

=>기본메뉴(생성, 추가, 삭제, 전체 내용 조회)와 추가메뉴(주차된 차량 수 조회, 퇴장, 할인권, 입금, 단일 정보 조회, 관리자 비밀번호, 유용한 정보조회)를 메인 메뉴와 세부 메뉴에 담았습니다.

Main.cpp 15~51번째 줄

while (true) {

handler.showMenu();//메인메뉴

try {

cout << "선택: ";

cin >> choice;

if (cin.fail() == true)throw strException();//문자입력시

cout << endl;

switch (choice)

{

case COUNT:

handler.CarCount();

break;

case IN:

handler.InCar();

break;

case OUT:

handler.OutCar();

break;

case INFO:

handler.ShowInfo();

break;

case MENEGER:

handler.Meneger();

break;

case EXIT:

return 0;

default:

cout << "잘못된 숫자 선택" << endl<<endl;

}

}

catch (strException& expt) {

expt.ShowExceptionInfo();

cin.clear();

cin.ignore(256, '\n');

}

}

캡쳐 화면은 1. 타이틀 출력 의 캡쳐 화면 참조

## 3. 추상 클래스

Car.h

class Car{

private:

static int count;//클래스 변수-고유번호 부여

int carSize;//차 종(크기)

protected:

int num;//고유번호

string carName;//문자열 변수-차 번호

time\_t inTime;//차가 들어온 시간

time\_t outTime;//차가 나간 시간

int CouFee;//쿠폰적용비용

int fee;//비용

int t;//차가 주차장에 머문 시간

public:

Car(int carSize, string carName);

void SetCouFee(int f);

virtual void Out()=0;//순수 가상 함수-주차장에서 차 나감

int GetFee();

int GetCouFee();

time\_t GetOutTime();

string GetCarName() const;

virtual void ShowCarInfo()=0;//순수 가상 함수-정보출력(입장번호(고유번호),차번호,차종,입장시간,퇴장시간,요금,쿠폰적용요금)

};

## 4. 추상 클래스를 상속하는 3개 이상의 클래스

1)SmallCar.h

class SmallCar :public Car{

private:

const int BasicRate=500;//기본요금(10분(초)간)

const int Extra=200;//추가요금(10분(초)마다)

void timeCalc() {

this->outTime = time(NULL);

t = (int)(outTime - inTime);

t = t / 10;//실행 시간상 임의로 10분을 10초로 설정}

public:

SmallCar(int carSize, string carName) :Car(carSize, carName) {}

void Out() {

timeCalc();//시간 계산

fee = BasicRate + Extra \* (t);

ShowCarInfo();}

virtual void ShowCarInfo() {

//출력 시간 설정

struct tm pTimeInfo;

localtime\_s(&pTimeInfo, &inTime);

int month = pTimeInfo.tm\_mon + 1;

int day = pTimeInfo.tm\_mday;

int hour = pTimeInfo.tm\_hour;

int min = pTimeInfo.tm\_min;

int sec = pTimeInfo.tm\_sec;

cout << "입장번호: " << num << endl;

cout << "차 번호: " << carName << endl;

cout << "차 종: " << "경차, 소형차" << endl;//소형차 클래스이므로

cout << "입장시간: " << month << "월 " << day << "일 " << hour << "시 " << min << "분 " << sec << "초" << endl;

if (outTime != NULL) {//차가 주차장을 나갔을 경우 조회시 퇴장시간 보임

//출력시간을 퇴장시간으로

localtime\_s(&pTimeInfo, &outTime);

month = pTimeInfo.tm\_mon + 1;

day = pTimeInfo.tm\_mday;

hour = pTimeInfo.tm\_hour;

min = pTimeInfo.tm\_min;

sec = pTimeInfo.tm\_sec;

cout << "퇴장시간: " << month << "월 " << day << "일 " << hour << "시 " << min << "분 " << sec << "초" << endl;

}

else

cout << "퇴장시간: " << endl;

if (fee != NULL) {//차가 주차장을 나갔을 경우 조회시 요금 보임

cout << "요금: " << fee << endl;

}

else

cout << "요금: " << endl;

if (CouFee != NULL) {//쿠폰을 적용했을 경우 쿠폰적용요금 보임

cout << "쿠폰 적용 요금: " << CouFee << endl;

}

}

};

2)MidCar.h

class MidCar :public Car{

private:

const int BasicRate = 750;//기본요금(20분(초)간)

const int Extra = 350;//추가요금(10분(초)마다)

void timeCalc() {

this->outTime = time(NULL);

t = (int)(outTime - inTime);

t = t / 10;//실행 시간상 임의로 10분을 10초로 설정}

public:

MidCar(int carSize, string carName) :Car(carSize, carName) {}

virtual void Out(){

timeCalc();//시간 계산

if (t <= 1)

fee = BasicRate;

else

fee = BasicRate + Extra \* (t - 1);

ShowCarInfo();}

virtual void ShowCarInfo() {

//출력 시간 설정

struct tm pTimeInfo;

localtime\_s(&pTimeInfo, &inTime);

int month = pTimeInfo.tm\_mon + 1;

int day = pTimeInfo.tm\_mday;

int hour = pTimeInfo.tm\_hour;

int min = pTimeInfo.tm\_min;

int sec = pTimeInfo.tm\_sec;

cout << "입장번호: " << num << endl;

cout << "차 번호: " << carName << endl;

cout << "차 종: " << "준중형차, 중형차" << endl;//중형차 클래스이므로

cout << "입장시간: " << month << "월 " << day << "일 " << hour << "시 " << min << "분 " << sec << "초" << endl;

if (outTime != NULL) {//차가 주차장을 나갔을 경우 조회시 퇴장시간 보임

//출력시간을 퇴장시간으로

localtime\_s(&pTimeInfo, &outTime);

month = pTimeInfo.tm\_mon + 1;

day = pTimeInfo.tm\_mday;

hour = pTimeInfo.tm\_hour;

min = pTimeInfo.tm\_min;

sec = pTimeInfo.tm\_sec;

cout << "퇴장시간: " << month << "월 " << day << "일 " << hour << "시 " << min << "분 " << sec << "초" << endl;}

else

cout << "퇴장시간: " << endl;

if (fee != NULL) {//차가 주차장을 나갔을 경우 조회시 요금 보임

cout << "요금: " << fee << endl;}

else

cout << "요금: " << endl;

if (CouFee != NULL) {//쿠폰을 적용했을 경우 쿠폰적용요금 보임

cout << "쿠폰 적용 요금: " << CouFee << endl;

}

}

};

3)LargeCar.h

class LargeCar :public Car{

private:

const int BasicRate = 1000;//기본요금(30분(초)간)

const int Extra = 500;//추가요금(10분(초)마다)

void timeCalc() {

this->outTime = time(NULL);

t = (int)(outTime - inTime);

t = t / 10;//실행 시간상 임의로 10분을 10초로 설정}

public:

LargeCar(int carSize, string carName) :Car(carSize, carName) {}

virtual void Out(){

timeCalc();//시간 계산

if (t <= 2)

fee = BasicRate;

else

fee = BasicRate + Extra \* (t - 2);

ShowCarInfo();}

virtual void ShowCarInfo() {

//출력 시간 설정

struct tm pTimeInfo;

localtime\_s(&pTimeInfo, &inTime);

int month = pTimeInfo.tm\_mon + 1;

int day = pTimeInfo.tm\_mday;

int hour = pTimeInfo.tm\_hour;

int min = pTimeInfo.tm\_min;

int sec = pTimeInfo.tm\_sec;

cout << "입장번호: " << num << endl;

cout << "차 번호: " << carName << endl;

cout << "차 종: " << "준대형차, 대형차" << endl;//대형차 클래스이므로

cout << "입장시간: " << month << "월 " << day << "일 " << hour << "시 " << min << "분 " << sec << "초" << endl;

if (outTime != NULL) {//차가 주차장을 나갔을 경우 조회시 퇴장시간 보임

//출력시간을 퇴장시간으로

localtime\_s(&pTimeInfo, &outTime);

month = pTimeInfo.tm\_mon + 1;

day = pTimeInfo.tm\_mday;

hour = pTimeInfo.tm\_hour;

min = pTimeInfo.tm\_min;

sec = pTimeInfo.tm\_sec;

cout << "퇴장시간: " << month << "월 " << day << "일 " << hour << "시 " << min << "분 " << sec << "초" << endl;}

else

cout << "퇴장시간: " << endl;

if (fee != NULL) {//차가 주차장을 나갔을 경우 조회시 요금 보임

cout << "요금: " << fee << endl;}

else

cout << "요금: " << endl;

if (CouFee != NULL) {//쿠폰을 적용했을 경우 쿠폰적용요금 보임

cout << "쿠폰 적용 요금: " << CouFee << endl;

}

}

};

## 5. 관리 Handler 클래스 Handler.h

class Handler

{

private:

vector<shared\_ptr<Car>> CarArr;//차 담는 벡터-스마트 포인터 이용

vector<int> numVec;//iterator 구현 위한 차 수 많큼의 벡터

vector<int>::iterator it;

vector<int> sizeVec;//차 종 담은 벡터

vector<int>feeVec;//비용 담은 벡터

static int SelecCar;//현재 선택중인 차의 벡터 위치 저장

public:

void showMenu(void)const;//메뉴

void CarCount(void);//현재 주차된 차량 수

void InCar(void);//입장

void OutCar(void);//퇴장

void ShowInfo(void);//정보출력

void Meneger(void);//관리자 메뉴

protected:

void Small(void);//소형차 입장시

void Mid(void);//중형차 입장시

void Large(void);//대형차 입장시

void Coupon(void);//쿠폰 사용시

void NCoupon(void);//쿠폰 사용 안할시

void CouponA(void);//10%쿠폰

void CouponB(void);//30%쿠폰

void CouponC(void);//50%쿠폰

void Delete(void);//관리자 삭제 메뉴

};

## 6. 생성자, 복사생성자, 소멸자 : 이후의 기능으로 생략 가능

## 7. 연산자 오버로딩

+와 -를 오버로딩

Handler.h 48~53번째 줄, Handler.cpp 415~418번째 줄에서 사용

void operator+(int op) {

s.append(to\_string(op));

}

void operator-(string op2) {

s.append(op2);

}

## 8. String

Car.h 15번째 줄

string carName;//문자열 변수-차 번호

## 9. 예외처리

Exception.h

//비용보다 지불금액이 적을 때 호출

class InsuffException {

private:

int fee;

int money;

public:

InsuffException(int f, int mon):fee(f),money(mon) {}

void ShowExceptionInfo() const {

cout << endl<<"금액이 "<<fee-money<< "원 부족합니다." << endl<<endl;}

};

//숫자 입력 구간에 문자 입력시 호출

class strException {

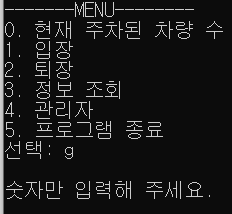
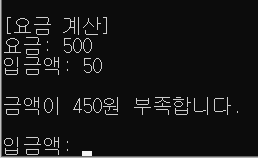
public:

void ShowExceptionInfo() const {

cout << endl<< "숫자만 입력해 주세요." << endl<<endl;

}

};

## 10. STL

컨테이너: vector 또는 map이용

Handler.h 15번째 줄

vector<shared\_ptr<Car>> CarArr;//차 담는 벡터-스마트 포인터 이용

이터레이터를 사용한 데이터 접근

Handler.h 16, 17번째 줄, Handler.cpp 26번째 줄

vector<int> numVec;//iterator 구현 위한 차 수 많큼의 벡터

vector<int>::iterator it;

for (it = numVec.begin(); it != numVec.end(); it++)

2개 이상의 알고리즘 사용

Handler.cpp 410~413번째 줄

count(sizeVec.begin(), sizeVec.end(), 1)

count(sizeVec.begin(), sizeVec.end(), 2)

count(sizeVec.begin(), sizeVec.end(), 3)

count\_if(feeVec.begin(), feeVec.end(), [](int e) {return e >= 3000; })

## 11. 스마트 포인터

Shared\_ptr 이용

Handler.h 15번째 줄

vector<shared\_ptr<Car>> CarArr;//차 담는 벡터-스마트 포인터 이용

## 12. 삽입 연산자 <<와 추출 연산자 >> 를 활용한 입출력

Main.cpp 18,19번째 줄

cout << "선택: ";

cin >> choice;



## 13. 파일 입출력

Handler.cpp 419~422번째 줄

ofstream writeFile;//파일 입출력

writeFile.open("test.txt");//없으면 생성

writeFile.write(string.GetS().c\_str(), string.GetS().size());//str의 string 내용 test.txt에 저장(총 손님 수와 총 수익 저장)

writeFile.close();

