<요구사항>

1. Rect 클래스 생성 및 width, height 멤버 변수 생성
2. Rect 메소드 생성
3. c의 가로, 세로, 면적을 나타내는 void 형의 show메소드를 생성

+operator(멤버 함수)

1. Rect 클래스 안에 operator+라는 연산자를 정의
2. 매개변수는 b로 하고 c의 가로와 세로를 구함.

-operator(외부 함수)

1. friend 형식으로 연산자 –를 Rect 클래스 안에 정의

2. 매개 변수로 Rect 클래스의 객체인 a와 b를 정의

main함수 설정

1. a와 b에 가로와 세로 값을 설정
2. a에서 b를 뺏을 때 c의 가로, 세로, 면적을 출력
3. a에서 b를 뺏을 때 c의 가로, 세로, 면적을 출력

<소스코드>

#include<iostream>

using namespace std;

class Rect;//오류를 방지 하기 위해 정의

class Rect {

private://가로와 세로를 private로 정의

int width;

int height;

public:

Rect(int width = 0, int height = 0) {this->width = width; this->height = height;}

//Rect메소드를 설정하고 이 메소드 안의 width와 height에 private으로 정의한

//변수를 대입

Rect operator +(Rect b);//맴버함수 형식의 연산자+를 설정

void show();//가로 세로 면적을 출력하는 메소드

friend Rect operator-(Rect a, Rect b);//외부함수 형식의 연산자 -를 설정, 그리고 friend로 설정

};

Rect Rect::operator+(Rect b) {

Rect c;

c.width = this->width + b.width;

c.height = this->height + b.height;

return c;

//클래스의 소속된 연산자를 재정의 하게되면 인자가 2개가 아니라 한개가 됨. 즉, 클래스

//자신이 이미 1개의 인자역활을 하기 때문에 그렇다. 위 예제의 + 연산자 오버로딩하는 함수를

//아래와 같이 수정해야 함

}

Rect operator-(Rect a, Rect b) {

Rect c;

c.width = a.width - b.width;

c.height = a.height - b.height;

return c;

//매개변수로 사용한 Rect 클래스의 객체인 a,b를 이용하여 c를 return

}

void Rect::show() {

cout << "가로는 " << width<< "," << "세로는" << height<< "," << "면적은" << height\*width << endl;

//가로 세로 면적을 출력하는 메소드

}

int main() {

Rect a(5, 8), b(1, 5), c;

//a와 b의 가로 세로를 설정

c = a - b;

c.show();

//a에서 b를 뺀 결과인 c의 show()값

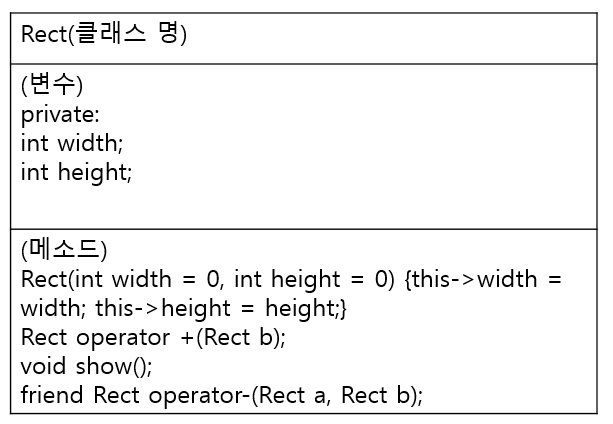
c = a + b;

c.show();

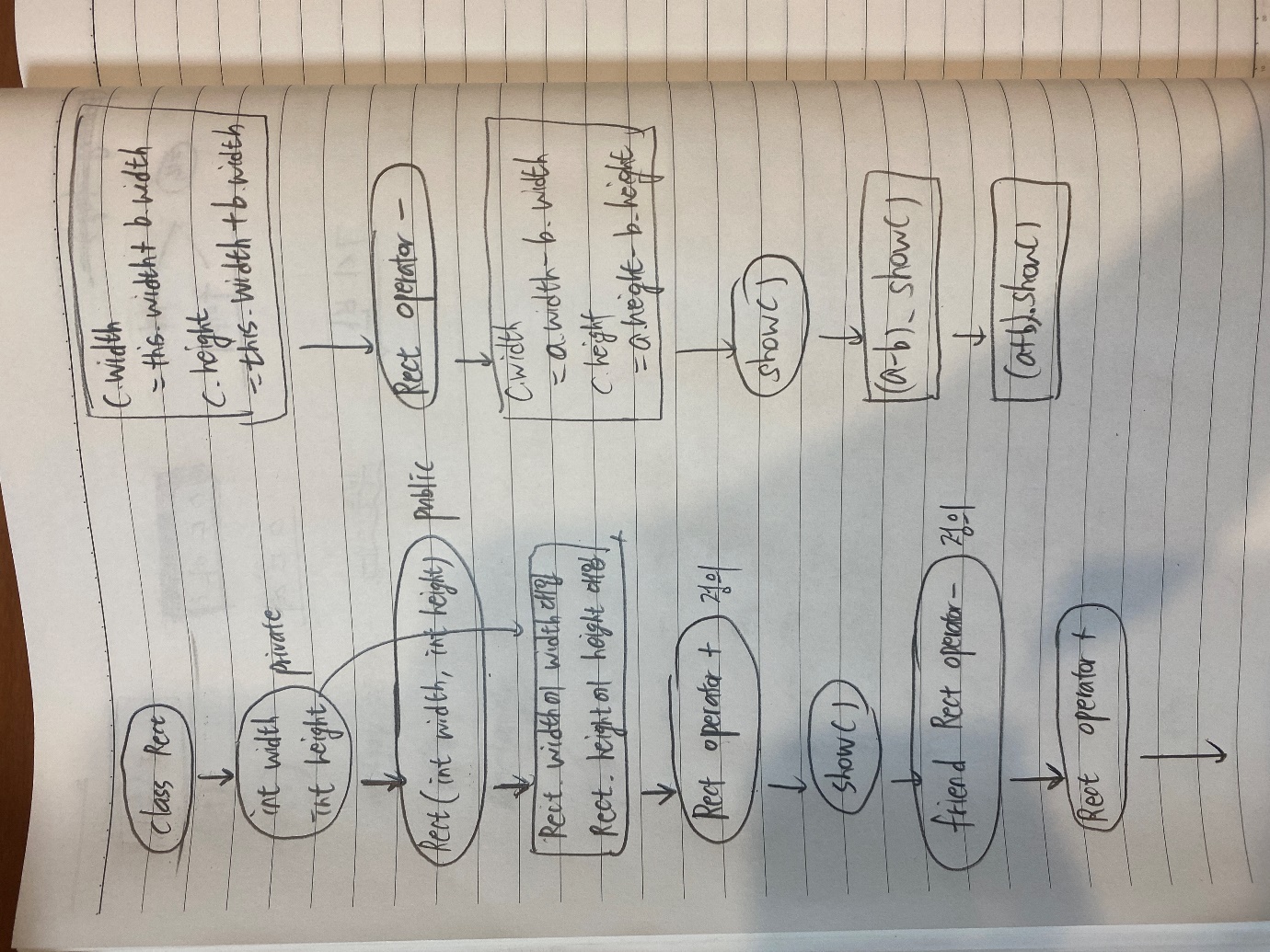
//a에서 b를 더한 결과인 c의 show()값

}

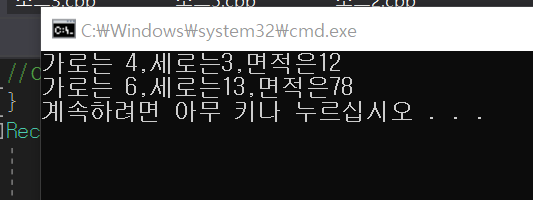
<클래스 설계도>



<설계도>



<실행결과>



<의견>

다소 무난한 것 같았지만 중간에 “이 연산자 함수에 매개변수가 너무 많습니다”라는 오류 메시지가 떴을 때는 많이 당황했다. 구글링해서 오류를 찾아본 결과 “클래스의 소속된 연산자를 재정의 하게되면 인자가 2개가 아니라 한개가 됨. 즉, 클래스 자신이 이미 1개의 인자역활을 하기 때문에 그렇다. 위 예제의 + 연산자 오버로딩하는 함수를 아래와 같이 수정해야 함”이라는 사실을 알게 되어서 신기하였다.