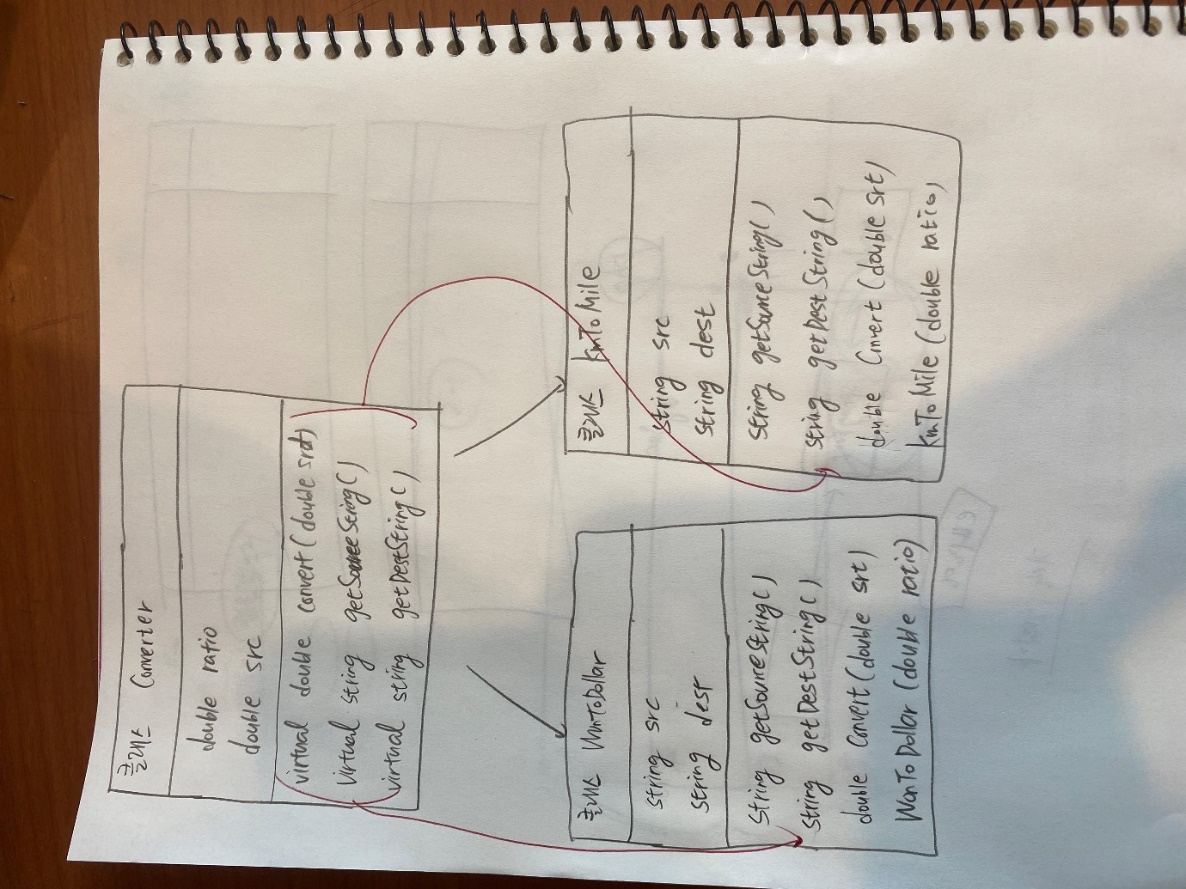
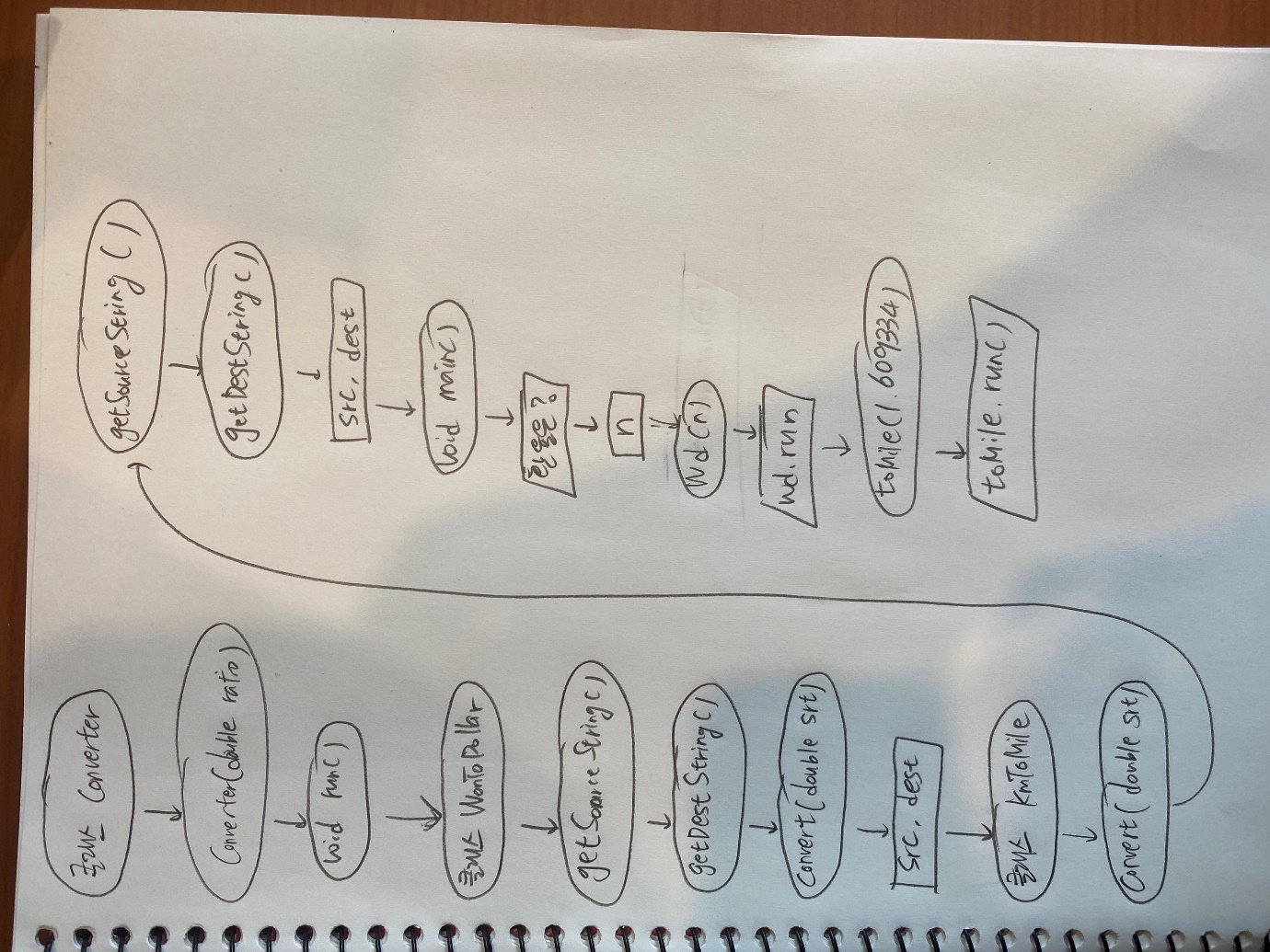
<요구사항>

1. 가상 함수를 가지는 부모 클래스를 생성
2. 단위를 나타내는 string형식의 문자 선언
3. 반환하려는 값 double형으로 선언
4. 원을 달러로 바꾸어주는 클래스인 WonToDollar클래스 선언(Converter 클래스 상속)
5. Km를 mile로 바꾸어주는 클래스인 KmtoMile클래스 선언(Converter클래스 상속)
6. 환율 입력
7. 환율을 바탕으로 원을 입력하면 이를 달러로 변환하여 출력
8. Km를 입력하면 이를 mile로 변환하여 출력

<클래스 설계도>



<프로그램 설계도>



<소스코드>

#include<iostream>

using namespace std;

class Converter {

protected:

double ratio;//비율, 여기서는 1달러당 ratio원, 1km당 ratio Mile

virtual double convert(double srt) = 0;//가상함수

virtual string getSourceString() = 0;//가상함수, src 단위 명칭

virtual string getDestString() = 0;//가상함수, dest 단위 명칭

public://어느 곳에서든 사용할 수 있음

Converter(double ratio) { this->ratio = ratio;}//생성자

void run() {//반환의 결과를 출력하는 메소드

double src;//바꾸려고 하는 값

cout << getSourceString() << "을" << getDestString() << "로 바꿉니다" << endl;//원->달러 혹은 km->Mile

cout << getSourceString() << "을 입력하세요" << endl;//반환하고 싶은 값의 단위를 입력

cin >> src;//반환하고 싶은 값을 입력

cout << "반환 결과" << convert(src) << getDestString() << endl;//바꾸려는 값+단위 꼴로 출력됨

//반환하고 싶은 값을 convert 메소드에 넣음.

}

};

class WonToDollar : public Converter {//원을 달러로 바꾸어 주는 클래스

string src, dest;//string 형식의 src와 dest 선언, (단위)

protected://선언된 클래스나 상속을 하는 클래스에서만 사용될 수 있음

string getSourceString() { return src; }//원

string getDestString() { return dest; }//달러

double convert(double srt) { return srt / Converter::ratio; }//Converter클래스의 ratio 사용 가능(protected이므로)

//convert(double srt)이 메소드를 사용하면 반환된 값을 구할 수 있음.

public://모든 곳에서 WonToDollar메소드르 사용할 수 있음.

WonToDollar(double ratio) :Converter(ratio) {

src = "원"; dest = "달러";

}

};

class KmToMile :public Converter {

string src, dest;

protected://선언된 클래스나 상속하는 클래스 내에서 사용가능

double convert(double srt) { return srt / Converter::ratio; }

string getSourceString() { return src; }//km

string getDestString() { return dest; }//Mile

public:

KmToMile(double ratio) :Converter(ratio) {

src = "km"; dest = "Mile";

}

};

void main() {

cout << "환율은 어떻게 되나요?" << endl;

int n;//환율

cin >> n;//환율을 입력(1달러는 n원이다)

WonToDollar wd(n);//입력한 n이 ratio가 되어서 반환하고 싶은 값인 src를 입력하면 이를 ratio로 나누어서 출력

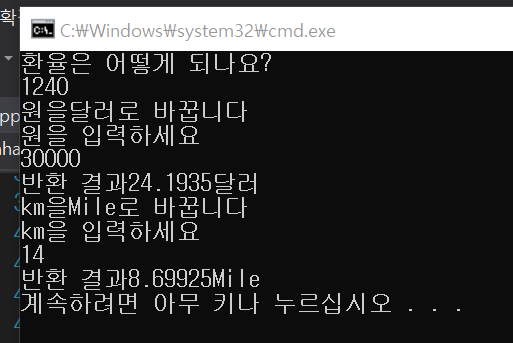
wd.run();//결과를 알려주는 메소드

KmToMile toMile(1.609334);//ratio는 1.609334가 되어 입력하는 km를 ratio로 나누어줌

toMile.run();//결과를 알려주는 메소드

}

<실행결과>



<의견>

처음에 코딩을 할 때 반환하려는 값과 단위를 헷갈려서 헤맸던 것 같다. 클래스에서는 단순히 어떤식으로 출력을 할지만 상속을 받을 것이기 때문에 똑같은 클래스를 상속받는 WonToDollar클래스와 KmToMile클래스에서 단위를 표시할 때 에매할 수 있다. 그래서 각 클래스마다 따로 단위를 저장해두고 getSourceString메소드의 src와 getDestString클래스의 dest만 상속받아 사용을 한다. 정말 괜찮은 아이디어인 것 같아 앞으로 자주 써야겠다는 생각을 했다.