

실습과제#5 : 이동생성자 호출

■ 이동생성자

그동안 우리가 실습과제로 만들어 온 StringCtrlSample 프로젝트의 CMyString 클래스도 이동생성자가 꼭 필요한 경우라 할 수 있습니다. 아래 코드를 따라서 작성하고 결과를 확인해주세요!!

■ 이동생성자 선언

CMyString.h 파일에 이동생성자 선언 추가

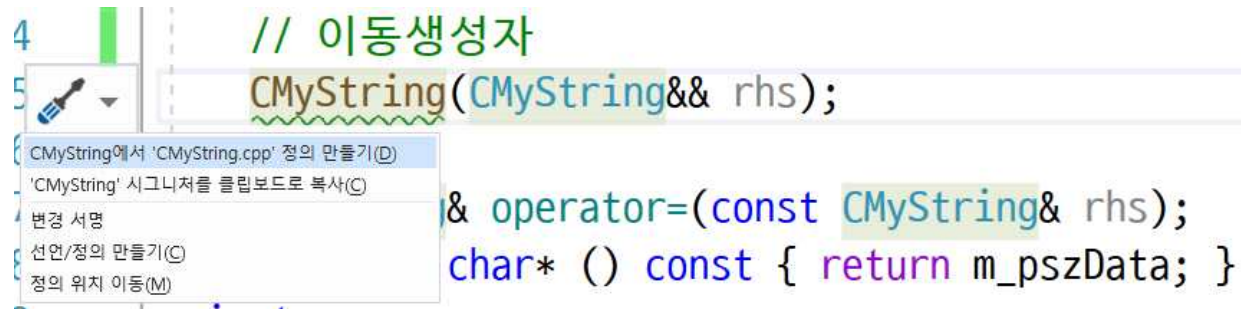
```
class CMyString
{
public:
    CMyString();
    ~CMyString();
    // 변환생성자
    explicit CMyString(const char* szParam);

    // 복사생성자
    CMyString(const CMyString& rhs);

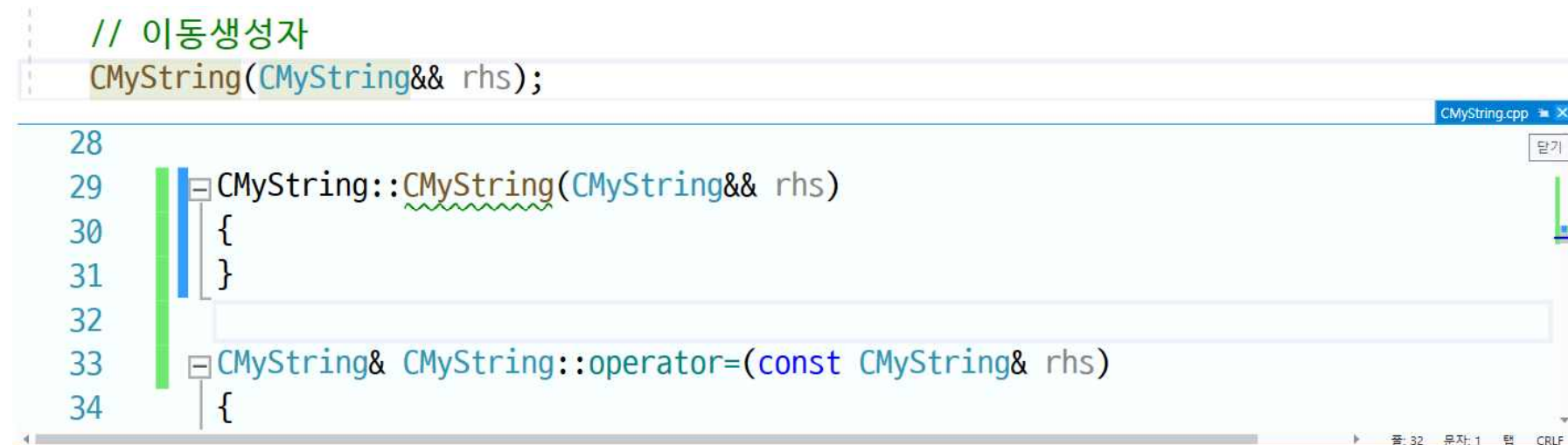
    // 이동생성자
    CMyString(CMyString&& rhs);
```

■ 이동 생성자 정의 추가

생성자 `CMyString(CMyString&& rhs);`를 클릭하고 `Alt+Enter` 눌러서 `CMyString` 에서 '`CMyString.cpp`'정의 파일 만들기 선택



아래와 같이 정의파일에 추가된 거 확인하고 저장



CMyString.cpp 파일에 이동생성자 함수 작성!

```
CMyString::CMyString(CMyString&& rhs)
: m_pszData(nullptr)
, m_nLength(0)
{
    cout << "CMyString 이동생성자 호출" << endl;

    // 얇은 복사를 수행해도 상관없다. 어차피 원본이 곧 소멸되기 때문이다.
    m_pszData = rhs.m_pszData;
    m_nLength = rhs.m_nLength;

    // 원본 임시 객체의 멤버들은 초기화 한다. 절대로 해제하면 안된다.
    rhs.m_pszData = nullptr;
    rhs.m_nLength = 0;
}
```

StringCtrlSample.cpp 의 코드를 다음과 같이 수정

```
#include <iostream>
#include "CMyString.h"
using namespace std;

CMyString TestFunc()
{
    CMyString strTest("TestFunc() return");
    cout << strTest << endl;

    return strTest;
}

int main()
{
    // 이름없는 임시 객체가 만들어진다.
    TestFunc();

    return 0;
}
```

실행결과 확인!

TestFunc() return
CMyString 이동생성자 호출
계속하려면 아무 키나 누르십시오 . . .

TestFunc()함수를 호출한 main()함수에서 이름없는 임시객체가 ‘이동생성자’에 의해 생성됨을 확인!!