

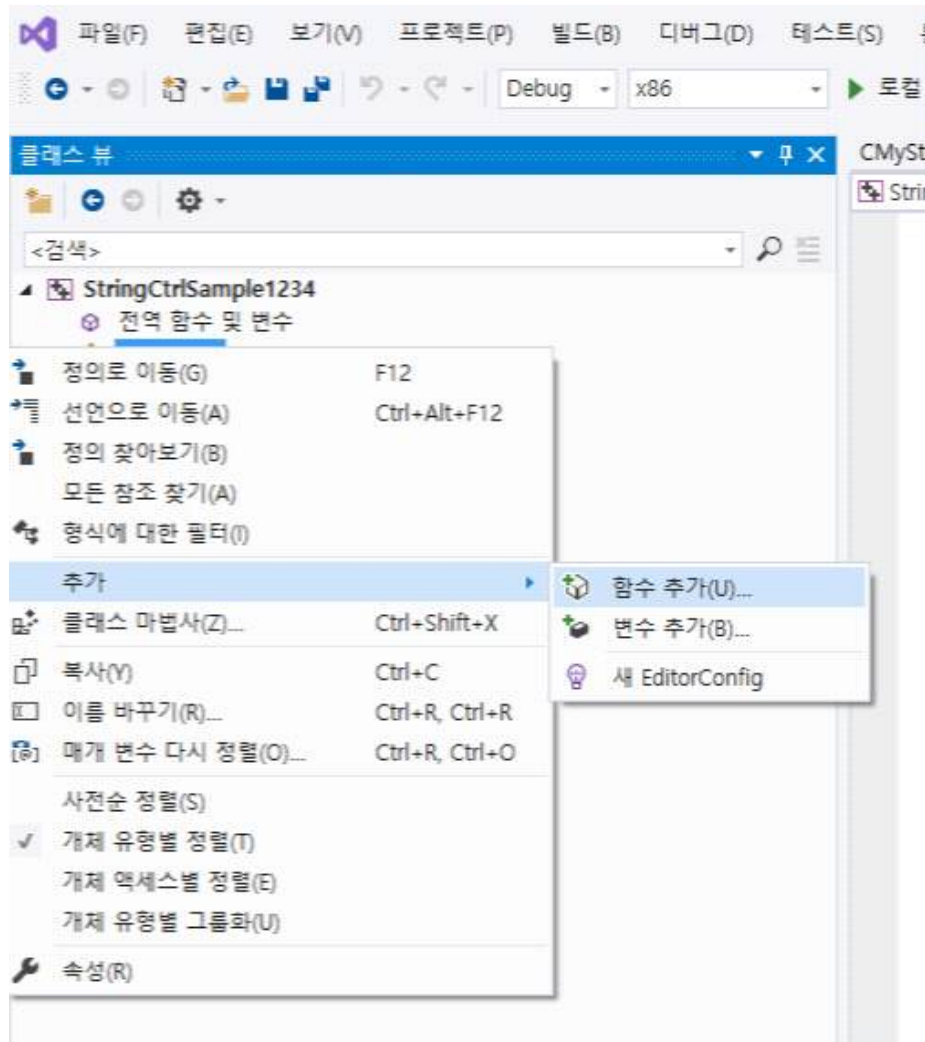
## 실습과제#12 : 문자열 필터의 또 다른 구현방법

### ■ 문자열 필터 다른 구현 방법

이번 실습 과제는 지금까지와는 달리 앞서 과제에 이어가는 것이 아니라 ‘실습과제 9’ 프로젝트에 이어서 진행합니다. 즉, SetString() 함수를 이용해 문자열 필터를 구현하기 직전으로 돌아가는 것입니다. 이번에도 단계별로 따라서 실습하기만 하면 됩니다.

```
// CMyStringEx 클래스가 아래와 같은 상태에서 진행!  
  
#pragma once  
#include "CMyString.h"  
class CMyStringEx : public CMyString  
{  
public:  
    CMyStringEx() {}  
    ~CMyStringEx() {}  
    CMyStringEx(const char* pszParam);  
  
    int Find(const char* pszParam);  
};
```

- CMyString 클래스에 함수를 추가합니다.(CMyStringEx 클래스가 아님!!!)
- CMyString 클래스에 CMyString::OnSetString() 함수 추가



## ■ CMyString 클래스에 함수 정보입력

함수 추가

함수 이름(U)

OnSetString

반환 형식(Y)

void

액세스(A)

public

.cpp 파일(F)

CMyString.cpp

매개 변수(P)

char \*pszData

int nLength

주석(M)

기타 옵션:

☐ 인라인(I)

☐ 정적(S)

☒ 시각적 개체(V)

☐ 순수(P)

확인

취소

- 함수이름 : OnSetString
- 반환형식 : void
- 액세스 : public
- cpp 파일 : CMyString.cpp
- 매개변수 : char\* pszData  
int nLength
- 시각적 개체(V) 선택-가상함수

## ■ CMyString 클래스 확인(함수 추가 확인)

// CMyString.h

```
#pragma warning(disable:4996)
#pragma once
class CMyString
{
public:
    CMyString();
    ~CMyString();
    // 변환생성자
    explicit CMyString(const char* szParam);

    // 복사생성자
    CMyString(const CMyString& rhs);

    // 이동생성자
    CMyString(CMyString&& rhs);

    int GetLength() const;

    int operator==(const CMyString& rhs);
    int operator!=(const CMyString& rhs);

    char& operator[](int nIndex);
    char operator[](int nIndex) const;

    CMyString operator+(const CMyString& rhs);
    CMyString& operator+=(const CMyString& rhs);
    CMyString& operator=(const CMyString& rhs);
    CMyString& operator=(const char* str);

    operator char* () const { return m_pszData; }

private:
    // 문자열을 저장하기 위해 동적 할당된 메모리를 가리키는 포
    char* m_pszData;
    // 저장된 문자열의 길이
    int m_nLength;
public:
    int SetString(const char* pszParam);
    const char* GetString() const;
    void Release();
    int Append(const char* pszParam);
    virtual void OnSetString(char* pszData, int nLength);
};
```

// CMyString.cpp

```
void CMyString::OnSetString(char* pszData, int nLength)
{
    // 아무런 코드도 추가하지 않는다.!
```

■ CMyString::SetString() 함수 수정

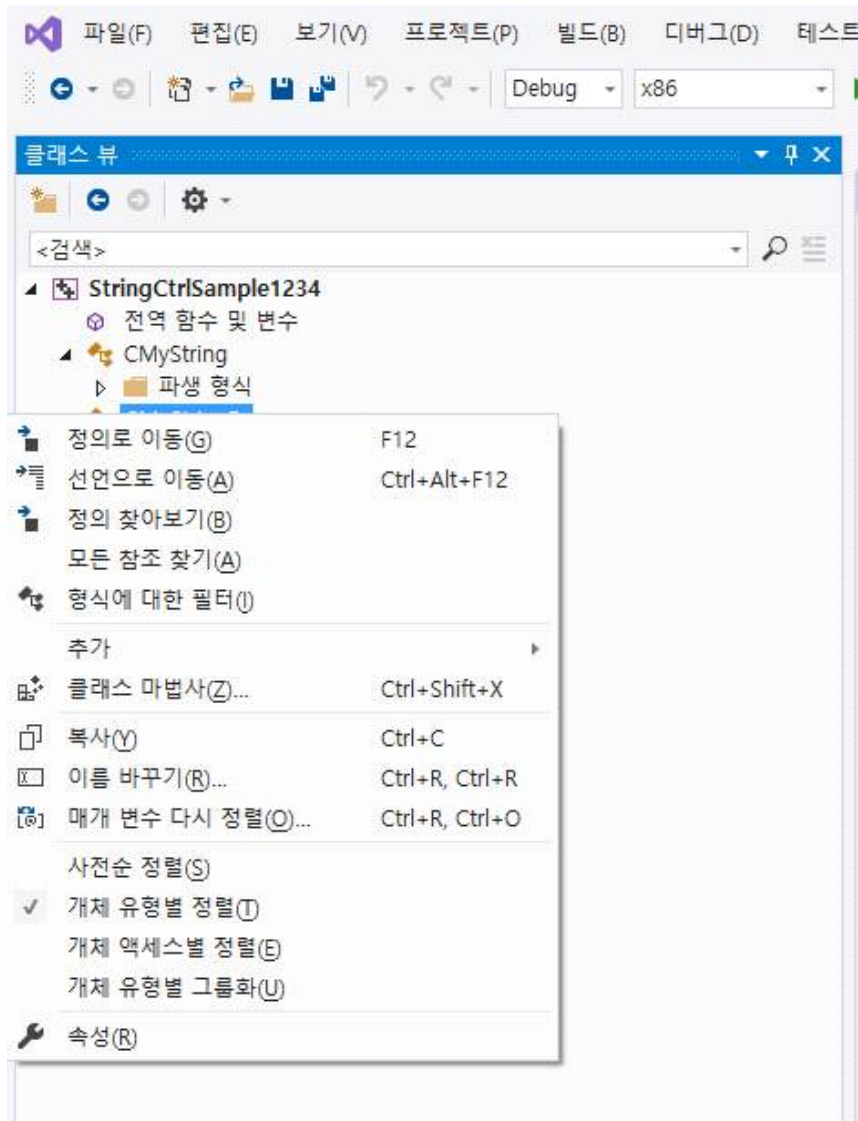
CMyString 클래스의 SetString() 함수가 결과를 반환하기 바로 전에 OnSetString() 가상함수를 호출하도록 CMyString.cpp 파일을 수정합니다.

```
int CMyString::SetString(const char* pszParam)
{
    // 미래를 호출한다.
    OnSetString(m_pszData, m_nLength);

    return m_nLength;
}
```

CMyString::SetString() 함수내부  
return 전에 문장 입력

- CMyStringEx 클래스에 CMyString::OnSetString() 함수를 재정의합니다.
- CMyStringEx 클래스에 CMyStringEx::OnSetString() 함수 추가



## ■ CMyStringEx 클래스에 함수 정보입력

함수 추가

함수 이름(U)

OnSetString

반환 형식(Y)

void

액세스(A)

public

.cpp 파일(F)

CMyStringEx.cpp

매개 변수(P)

char \*pszData

int nLength

주석(M)

기타 옵션:

☐ 인라인(I)

☐ 정적(S)

☒ 시각적 개체(V)

☐ 순수(P)

확인

취소

- 함수이름 : OnSetString
- 반환형식 : void
- 액세스 : public
- cpp 파일 : CMyStringEx.cpp
- 매개변수 : char\* pszData  
int nLength
- 시각적 개체(V) 선택-가상함수

■ CMyStringEx 클래스 확인(함수 추가 확인)

```
// CMyStringEx.h
#pragma once
#include "CMyString.h"
class CMyStringEx : public CMyString
{
public:
    CMyStringEx() {}
    ~CMyStringEx() {}
    CMyStringEx(const char* pszParam);

    int Find(const char* pszParam);

    virtual void OnSetString(char* pszData, int nLength);
};
```

```
// CMyStringEx.cpp
void CMyStringEx::OnSetString(char* pszData, int nLength)
{
    if (strcmp(pszData, "멍멍이아들") == 0)
    {
        strcpy(pszData, "*착한아들*");
    }
}
```



## StringCtrlSample.cpp 의 코드를 다음과 같이 수정

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


int main()
{
    CMyStringEx strTest;

    // 문자열이 필터링 되어 대체되는 경우
    strTest.SetString("멍멍이아들");
    cout << strTest << endl;

    // 필터링 되지 않는 경우
    strTest.SetString("학번 : 1234, 이름 : 홍길동"); // 학번과 이름은 본인 것으로 수정!!!
    cout << strTest << endl;

    return 0;
}
```

## 실행결과 확인!

 C:\Windows\system32\cmd.exe

\*착한아들\*

학번 : 1234, 이름 : 홍길동

계속하려면 아무 키나 누르십시오 . . .