2023/02/19

---

1. #include <tchar.h>

: TCHAR라는 타입을 정의하기 위해 사용
_T("string")과 같은 _T매크로를 사용하여 다국어 지원 등 다양한 프로그램에서 문자열을 처리할 때 안정적 사용하기 위함

TCHAR : 유니코드, 멀티바이트 문자열을 모두 지원하기 위한 데이터 타입
-> 'TCHAR'로 선언된 변수는 프로젝트 설정에 따라 유니코드 문자열 또는 멀티바이트 문자열로 컴파일 됨

---

2. 멀티바이트 문자열 (Multi-byte character string)

: 문자열을 표현하기 위해 여러 개의 바이트를 사용하는 문자열을 의미

- 유니코드 문자열이 보편화되면서 사용이 줄어들고 있지만, 일부 멀티바이트 문자열을 사용하는 프로글매에서는 여전히 사용됨
- _TCHAR와 같은 TCHAR 매크로를 사용하면, 멀티바이트 문자열과 유니코드 문자열을 동시에 지원하는 코드 작성 가능

- 유니코드 문자열과 차이점 : 유니코드 문자열은 문자열 표현을 위해 2바이트(or 4바이트) 고정 길이 코드 유닛 사용

---

3. CString, char

- CString : 문자열을 처리하기 위한 클래스. 입력값이나 출력값을 문자열 형태로 처리할 때 유용
  C-style 문자열(char*)과 같은 문자열 처리를 쉽게 할 수 있는 유용한 기능들을 제공
- char : 문자 하나를 처리하기 위한 자료형 (연산자 처리)

---

4. '\0' (널문자)

- 문자열의 끝을 나타내기 위해 사용
- 사용자가 조건을 만족하는 문자(ex 사칙연산)을 입력하지 않았을 경우를 대비

---

5. string으로 변수를 선언하고 getline으로 문자를 입력 받는 이유

: string만을 사용하여 문자열을 입력받는 것 보다 **안전하고 유연한 문자열 처리를 위함**

1. 사용자가 입력한 문자열의 길이를 알 수 있음 (C스타일 문자열은 크기를 알 수 없어 문자열 처리에 제한O)
2. string 클래스는 문자열을 동적 할당하므로 메모리를 더 효율적으로 관리 가능
3. 입력받은 문자열에 대한 유효성 검사가 더 쉬움

- string : C++ STL문자열 클래스이며, 문자열을 처리하는데 유용한 여러가지 기능들 제공

2023/02/19

- getline : string 클래스에 정의된 멤버 함수로 입력 스트림에서 한 줄을 읽어 ‘string’ 객체에 저장

---

6. c_str()

: std:string 객체에 저장된 문자열을 C 스타일 문자열(const char* 형태)로 변환하여 반환
-> str_input에 ‘strA’를 할당하기 위해 사용
-> 변환해야 CSring 클래스에서 제공하는 다양한 문자열 연산을 사용 가능

- CString에 값을 저장하기 위함
- CString은 MFC 라이브러리에서 사용하는 문자열 클래스이며, 문자열 데이터를 포인터 형태로 저장한다. 따라서 CString에 값을 저장하기 위해서는 const char* or const wchar_t*와 같이 포인터 형태로 문자열을 전달해야 한다.
- CString 객체를 숫자형 변수로 변환 가능
- CString 객체를 double형 변수로 변환 가능 (_tstof 함수 사용)

---

7. GetAt 함수

GetAt 함수는 MFC CString 클래스에서 제공하는 함수 중 하나
CString 객체의 특정 위치에 있는 문자를 반환

예를 들어,
- CString str("hello")와 같이 문자열 "hello"를 가지는 CString 객체가 있다면, str.GetAt(0)은 'h'를 반환하고, str.GetAt(4)는 'o'를 반환

- 인자로 전달된 인덱스가 유효한지 확인하고, 유효하지 않은 경우에는 빈 문자('\0')를 반환

---

8. Replace() 함수로 공백 제거 하는 이유

- 공백이 포함된 수식을 계산하면 오류가 발생할 수 있기에 안정적인 계산을 위함