

민 코 딩 훈 련 2 코 스

수업노트 LV-24



배우는 내용

1. Class 소개
2. String Class 사용하기
3. Parsing 메서드 (find, substr, stoi)
4. find 응용 : 단어 개수 찾기

Class 소개

c언어 struct와 c++ struct의 차이

- ▶ struct 안에 함수를 포함시킬 수 있음

```
struct Node {  
    int x;  
    int y;  
};  
  
struct Node t;  
  
t.x = 10;  
t.y = 20;
```

c언어 struct

```
struct Node {  
    int x;  
    int y;  
  
    void bts() {  
        cout << "호호";  
    }  
};  
  
Node t;  
  
t.x = 10;  
t.y = 20;  
t.bts();
```

c++ struct

public 제외하면 struct 와 class는 동일하다.

- ▶ struct에 함수를 추가할 수 있도록 만든 것이 class 이다.

- ▶ class, public 으로만
바꾸어 써주면 동작한다.

```
struct Node {  
    int x;  
    int y;  
  
    void bts() {  
        cout << "호호";  
    }  
};  
  
Node t;  
  
t.x = 10;  
t.y = 20;  
  
t.bts();
```

```
class Node {  
public:  
    int x;  
    int y;  
  
    void bts() {  
        cout << "호호";  
    }  
};  
  
Node t;  
  
t.x = 10;  
t.y = 20;  
  
t.bts();
```

용어를 암기하자.

▶ 필드

- ▶ class 또는 struct 안에 있는 변수

▶ 메서드

- ▶ class 또는 struct 안에 있는 함수

▶ 인스턴스

- ▶ class 또는 struct 타입의 변수
- ▶ “인스턴트 음식”의 인스턴트 용어가 아니다.

```
class Node {  
public:  
    int x;  
    int y;  
  
    void bts() {  
        cout << "호호";  
    }  
};  
  
Node t;  
  
t.x = 10;  
t.y = 20;  
  
t.bts();
```

class를 만드는 것과 사용하는 것의 차이

수업시간에 다루는 점

- ▶ Class를 만드는 것은 우리 수업에서 다루지 않음
- ▶ 이미 만들어진 Class에 대해서, 인스턴스를 만들어 사용하는 것에 포커스를 맞춤

Class Library

- ▶ Class Library에는 많은 Class들이 이미 만들어져있다.
- ▶ 필요한 Class의 인스턴스를 만들어 사용한다.

클래스 만들기

```
class Node {  
public:  
    int x;  
    int y;  
  
    void bts() {  
        cout << "호호";  
    }  
};
```

```
Node t;  
  
t.x = 10;  
t.y = 20;  
  
t.bts();
```

클래스 사용

재차 강조, 필수 암기 용어

▶ 필드

- ▶ class 또는 struct 안에 있는 변수

▶ 메서드

- ▶ class 또는 struct 안에 있는 함수

▶ 인스턴스

- ▶ class 또는 struct 타입의 변수
- ▶ “인스턴트 음식”의 인스턴트 용어가 아니다.

```
class Node {  
public:  
    int x;  
    int y;  
  
    void bts() {  
        cout << "호호";  
    }  
};  
  
Node t;  
  
t.x = 10;  
t.y = 20;  
  
t.bts();
```


string class 사용하기

string 헤더파일

`#include <string.h>`

- ▶ c언어에서 사용하던 문자열 함수 Library
- ▶ 종류 : strlen, strcmp, strcpy, strcat...

`#include <string>`

- ▶ string Class의 추가 기능들이 들어가 있음
- ▶ c언어의 ○

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string a = "ABCD";

    cout << a;

    return 0;
}
```

string 사용법

string 객체 생성

- ▶ string 인스턴스 생성 또는, **string 객체 생성** 이라고 부른다.
- ▶ int 타입 변수 사용 방법과 동일

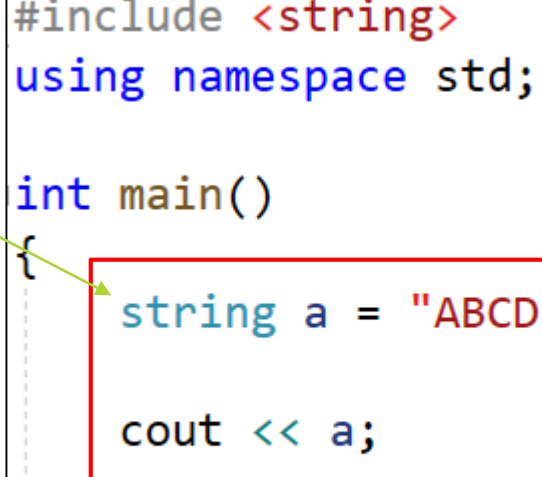
입/출력

- ▶ cin, cout 역시 변수 사용법과 동일

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string a = "ABCD";
    cout << a;

    return 0;
}
```



문자열 길이 출력하기

- ▶ 한 문자열 입력받고,
한 칸씩 띄어서 출력하는 프로그램 작성
- ▶ length 메서드

```
string str;  
cin >> str;  
  
for (int i = 0; i < str.length(); i++) {  
    cout << str[i] << " ";  
}
```

Microsoft Visual Studio 디버그 콘솔

```
KFCWORLD  
K F C W O R L D  
C:\Users\minco\source\repos\...  
습니다(코드: 0개).  
이 창을 닫으려면 아무 키나 ...
```

문자열 비교 / 추가

- ▶ int 변수와 사용방법이 동일하게 만들어져있다.

```
string str;  
cin >> str;  
  
if (str == "ABC") {  
    cout << "ABC가 입력 되었음";  
}
```

== 로 비교 가능 (strcmp 안써도 됨)

```
string str1 = "ABC";  
string str2 = "BTS";  
  
string str = str1 + str2;  
cout << str;
```

출력결과 : ABCBTS

[참고] printf로 string 출력하려면?

- ▶ string 객체를
cout 으로 출력하는 것은 쉽다.

```
string str1 = "ABC";  
cout << str1;
```

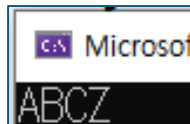
- ▶ string 객체를
printf 함수로 출력 → c_str() 메서드를 사용해야한다.

```
string str1 = "ABC";  
printf("%s", str1.c_str());
```

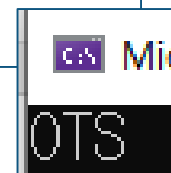
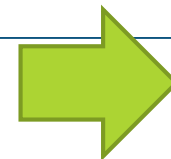
char [] 처럼 특정 index 접근 / 수정 가능

- ▶ index로 특정 문자 접근 / 수정 가능

```
string id = "ABC93Z";  
  
cout << id[0];  
cout << id[1];  
cout << id[2];  
  
cout << id[id.length() - 1];
```



```
string id = "ABC93Z";  
  
id = "BTS";  
id[0] = '0';  
  
cout << id;
```



Parsing 메서드

Parsing 파싱이란?

- ▶ 문자열 구조 파악을 뜻한다.
- ▶ 이곳에서 파싱은, 문자열의 구조를 파악하여, **필요한 정보만 가져오는 것**을 뜻한다.



파싱에 가장 많이 쓰이는 메서드 세 가지

find()

- ▶ 특정 문자열이 어디에 있는지 검색하여, index를 알려준다.

substr()

- ▶ 특정 index 부터 몇 글자수 문자열을 추출한다.

stoi()

- ▶ 문자열을 수로 변환하는 메서드

find 메서드

문자 or 문자열을 찾을 때 사용하는 메서드

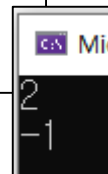
- ▶ 찾으면 : index값이 리턴된다.
- ▶ 못 찾으면 : (unsigned int) **-1**이 리턴된다.

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string str1 = "Z_KFC_ZZZ";

    cout << str1.find("KFC") << "\n";
    cout << (int)str1.find("BBQ") << "\n";

    return 0;
}
```



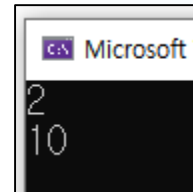
2
-1

[중요] find 메서드의 추가 옵션

- ▶ 몇 번 index부터 검색을 시작할지 명시할 수 있음
 - ▶ 파싱에 많이 사용되는 기능이기에 꼭 알아 둘 것.

```
string str1 = "Z_KFC_ZZZ_KFC_";  
cout << str1.find("KFC") << "\n";  
cout << str1.find("KFC", 4) << "\n";
```

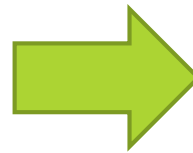
4번 index부터
검색을 시작한다.



substr 메서드

- ▶ 특정 문자열에서, 특정 개수 만큼 문자열을 추출한다.
- ▶ 2번 index 부터 세 글자를 추출하는 소스코드

```
string str1 = "Z_KFC_ZZZ_KFC_";  
string ret = str1.substr(2, 3);  
cout << ret;
```

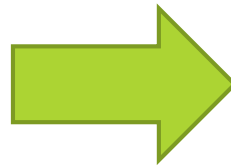


[유의사항]
글자 수를 기입해야한다.

stoi 함수

- ▶ 수 형태로 되어있는 문자열을
수로 변경하는 함수

```
string str = "1432";  
  
int ret = stoi(str);  
cout << ret + 100;
```

A screenshot of a Windows command prompt window. The title bar shows the Microsoft logo and the word "Microsoft". The command prompt displays the number "1532" on a black background, which is the result of the C++ code execution (1432 + 100).

Microsoft
1532

괄호 안에 있는 수 추출하여, + 10 값을 출력하기

1. find로 시작과 끝 검색하기
2. size 계산 후, substring으로 문자열 추출하기
3. 문자열을 수로 변경하여, + 10 값 출력

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string str = "ABC[13]IEB";

    int a = str.find('[');
    int b = str.find(']', a + 1);

    int size = b - a - 1;
    string word = str.substr(a + 1, size);

    cout << stoi(word) + 10;

    return 0;
}
```

find 응용 : 단어 개수 찾기

find 기본

for문을 쓰지 않고 구현하기

한 문자열에 "ABC" 가 2개 존재

두 ABC 문자열의 index 찾기

```
string str = "__ABC__ABC__";  
  
int a = str.find("ABC", 0);  
int b = str.find("ABC", a + 1);  
  
cout << a << b;
```

find 응용

for문 써서 구현하기

한 문자열에 "ABC" 가 2개 존재

두 ABC 문자열의 index 찾기

▶ for문으로 구현해보자.

```
string str = "__ABC__ABC__";

int a = -1;
for (int i = 0; i < 2; i++) {
    a = str.find("ABC", a + 1);
    cout << a;
}
```

단어 개수 구하기

긴 문자열에서, ABC가 총 몇개 존재할까?

```
string str = "__ABC__ABC__";

int a = -1;
int cnt = 0;
while(1) {
    a = str.find("ABC", a + 1);
    if (a == -1) break;
    cnt++;
}

cout << cnt;
```