



객체 포인터와 객체 배열, 객체의 동적 생성

string 클래스를 이용한 문자열

2

□ C++ 문자열

- ▣ C-스트링
- ▣ C++ string 클래스의 객체

□ string 클래스

- ▣ C++ 표준 라이브러리, <string> 헤더 파일에 선언

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

▣ 가변 크기의 문자열

```
string str = "I love "; // str은 'I', ' ', 'I', 'o', 'v', 'e', ' '의 7개 문자로 구성
str.append("C++."); // str은 "I love C++."이 된다. 11개의 문자
```

- ▣ 다양한 문자열 연산을 실행하는 연산자와 멤버 함수 포함
 - 문자열 복사, 문자열 비교, 문자열 길이 등
- ▣ 문자열, 스트링, 문자열 객체, string 객체 등으로 혼용

string 객체 생성 및 입출력

3

문자열 생성

```
string str; // 빈 문자열을 가진 스트링 객체
string address("서울시 성북구 삼선동 389"); // 문자열 리터럴로 초기화
string copyAddress(address); // address를 복사한 copyAddress 생성

// C-스트링(char [] 배열)으로부터 스트링 객체 생성
char text[] = {'L', 'o', 'v', 'e', ' ', 'C', '+', '+', '\0'};
string title(text); // "Love C++" 문자열을 가진 title 생성
```

문자열 출력

- cout과 << 연산자

```
cout << address << endl; // "서울시 성북구 삼선동 389" 출력
cout << title << endl; // "Love C++" 출력
```

문자열 입력

- cin과 >> 연산자

```
string name;
cin >> name; // 공백이 입력되면 하나의 문자열로 입력
```

문자열 숫자 변환

- stoi() 함수 이용
 - 2011 C++ 표준부터

```
string s="123";
int n = stoi(s); // n은 정수 123. 비주얼 C++ 2010 이상 버전
```

```
string s="123";
int n = atoi(s.c_str()); // n은 정수 123. 비주얼 C++ 2008 이하
```

string 클래스의 주요 생성자

생성자	설명
<code>string()</code>	빈 문자열을 가진 스트링 객체 생성
<code>string(const string& str)</code>	<code>str</code> 을 복사한 새로운 스트링 객체 생성
<code>string(const char* s)</code>	C-스트링 <code>s</code> 의 문자열을 복사하여 스트링 객체 생성
<code>string(const char* s, int n)</code>	문자 배열 <code>s</code> 에서 <code>n</code> 개의 문자를 복사하여 스트링 객체 생성

string 클래스의 주요 멤버 함수

- 표에서 pos는 문자열 내의 문자 위치로서, 0부터 시작함

멤버 함수	설명
string& append(const string& str)	문자열 뒤에 str 추가
string& append(const string& str, int pos, int n)	str 문자열 내 pos 위치에서 n개의 문자를 현재 문자열 뒤에 덧붙임
string& insert(int pos, int n, const string& str)	문자열 pos 위치에 str 삽입
string& replace(int pos, int n, const string& str)	문자열의 pos 위치부터 n개 문자를 str 문자열로 대체
int size()	문자열의 길이 리턴. 문자열의 길이는 바이트 수(영어 문자 개수)
int length()	문자열의 길이 리턴. size()와 동일
int capacity()	할당된 메모리 크기 리턴
string& erase(int pos, int n)	pos 위치부터 n개 문자 삭제
void clear()	문자열 모두 삭제. 크기를 0으로 만듦
bool empty()	문자열 크기가 0이면 true, 아니면 false 리턴

string 클래스의 주요 멤버 함수

- 표에서 pos는 문자열 내의 문자 위치로서, 0부터 시작함

멤버 함수	설명
<code>char& at(int pos)</code>	pos 위치의 문자 리턴
<code>int find(const string& str)</code>	문자열의 처음부터 str을 검색하여 발견한 처음 인덱스 리턴, 없으면 -1 리턴
<code>int find(const string& str, int pos)</code>	문자열의 pos 위치부터 str을 검색하여 발견한 처음 인덱스 리턴, 없으면 -1 리턴
<code>int rfind(const string& str, int pos)</code>	문자열의 pos 위치부터 str을 검색하여 마지막에 발견한 인덱스 리턴, 없으면 -1 리턴
<code>int compare(const string& str)</code>	문자열과 str을 비교하여 같으면 0을, 사전 순으로 현재 문자열이 앞에 오면 음수, 뒤에 오면 양수 리턴
<code>string substr(int pos, int n)</code>	pos 위치부터 n개의 문자를 새로운 서브스트링으로 생성, 리턴
<code>void swap(string& str)</code>	현재 객체의 문자열과 str의 문자열을 서로 바꿔치기 함
<code>char* c_str()</code>	C-스트링 문자열 리턴

string 클래스의 연산자

▣ `string s = "C++"; string s1 = "C"; string s2 = "Java";` 일 때

연산자	설명	사용 예	결과
<code>s1 = s2</code>	s2를 s1에 치환	<code>s1 = s2</code>	s1 = "Java"
<code>s[]</code>	s의 [] 인덱스에 있는 문자	<code>char c = s[1]</code>	c = '+'
<code>s1 + s2</code>	s1과 s2를 연결한 새로운 문자열	<code>s1 + s2</code>	"CJava"
<code>s1 += s2</code>	s1에 s2 문자열 연결	<code>s1 += s2</code>	s1 = "CJava"
<code>stream << s</code>	s를 stream 스트림에 출력	<code>cout << s;</code>	"C++" 출력
<code>stream >> s</code>	stream에서 문자열을 s에 입력	<code>cin >> s;</code>	문자열 입력
<code>s1 == s2</code>	s1과 s2가 같은 문자열이면 true	<code>s1 == s2</code>	false
<code>s1 != s2</code>	s1과 s2가 다른 문자열이면 true	<code>s1 != s2</code>	true
<code>s1 < s2</code>	s1이 사전 순으로 s2보다 앞에 오면 true	<code>s1 < s2</code>	true
<code>s1 > s2</code>	s1이 사전 순으로 s2보다 뒤에 오면 true	<code>s1 > s2</code>	false
<code>s1 <= s2</code>	s1이 s2와 같거나 앞에 오면 true	<code>s1 <= s2</code>	true
<code>s1 >= s2</code>	s1이 s2와 같거나 뒤에 오면 true	<code>s1 >= s2</code>	false

예제 4-11 string 클래스를 이용한 문자열 생성 및 출력

8

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

int main() {
    // 스트링 생성
    string str; // 빈 문자열을 가진 스트링 객체 생성
    string address("서울시 성북구 삼선동 389");
    string copyAddress(address); // address의 문자열을 복사한 스트링 객체 생성

    char text[] = {'L', 'o', 'v', 'e', ' ', 'C', '+', '+', '\0'}; // C-스트링
    string title(text); // "Love C++" 문자열을 가진 스트링 객체 생성

    // 스트링 출력
    cout << str << endl; // 빈 스트링. 아무 값도 출력되지 않음
    cout << address << endl;
    cout << copyAddress << endl;
    cout << title << endl;
}
```

string 클래스
를 사용하기 위
해 반드시 필요

빈 문자열을 가
진 스트링 출력

서울시 성북구 삼선동 389
서울시 성북구 삼선동 389
Love C++

예제 4-12 string 배열 선언과 문자열 키 입력 응용

9

5 개의 string 배열을 선언하고 getline()을 이용하여 문자열을 입력 받아 사전 순으로 가장 뒤에 나오는 문자열을 출력하라. 문자열 비교는 <, > 연산자를 간단히 이용하면 된다.

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

int main() {
    string names[5]; // 문자열 배열 선언

    for(int i=0; i<5; i++) {
        cout << "이름 >> ";
        getline(cin, names[i], '\n');
    }

    string latter = names[0];
    for(int i=1; i<5; i++) {
        if(latter < names[i]) { // 사전 순으로 latter 문자열이 앞에 온다면
            latter = names[i]; // latter 문자열 변경
        }
    }
    cout << "사전에서 가장 뒤에 나오는 문자열은 " << latter << endl;
}
```

```
이름 >> Kim Nam Yun
이름 >> Chang Jae Young
이름 >> Lee Jae Moon
이름 >> Han Won Sun
이름 >> Hwang Su hee
사전에서 가장 뒤에 나오는 문자열은 Lee Jae Moon
```

예제 4-13 문자열을 입력 받고 회전시키기

10

빈칸을 포함하는 문자열을 입력 받고, 한 문자씩 왼쪽으로 회전하도록 문자열을 변경하고 출력하라.

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

int main() {
    string s;

    cout << "문자열을 입력하세요(한글 안됨) " << endl;
    getline(cin, s, '\n'); // 문자열 입력
    int len = s.length(); // 문자열의 길이

    for(int i=0; i<len; i++) {
        string first = s.substr(0,1); // 맨 앞의 문자 1개를 문자열로 분리
        string sub = s.substr(1, len-1); // 나머지 문자들을 문자열로 분리
        s = sub + first; // 두 문자열을 연결하여 새로운 문자열로 만듦
        cout << s << endl;
    }
}
```

문자열을 입력하세요 (한글 안됨)

I love you

love youl

love youl

ove youl I

ve youl lo

e youl lov

youl love

youl love

oul love y

ul love yo

I love you

예제 4-14 문자열 처리 응용 - 덧셈 문자열을 입력 받아 덧셈 실행

4+125+4+77+102 등으로 표현된 덧셈식을 문자열로 입력받아 계산하는 프로그램 작성하라.

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

int main() {
    string s;
    cout << "7+23+5+100+25와 같이 덧셈 문자열을 입력하세요." << endl;
    getline(cin, s, '\n'); // 문자열 입력
    int sum = 0;
    int startIndex = 0; // 문자열 내에 검색할 시작 인덱스
    while(true) {
        int flIndex = s.find('+', startIndex);
        if(flIndex == -1) { // '+' 문자 발견할 수 없음
            string part = s.substr(startIndex);
            if(part == "") break; // "2+3+", 즉 +로 끝나는 경우
            cout << part << endl;
            sum += stoi(part); // 문자열을 수로 변환하여 더하기
            break;
        }
        int count = flIndex - startIndex; // 서브스트링으로 자를 문자 개수
        string part = s.substr(startIndex, count); // startIndex부터 count 개의 문자로 서브스트링 만들기
        cout << part << endl;
        sum += stoi(part); // 문자열을 수로 변환하여 더하기
        startIndex = flIndex+1; // 검색을 시작할 인덱스 전진시킴
    }
    cout << "숫자들의 합은 " << sum;
}
```

7+23+5+100+25와 같이 덧셈 문자열을 입력하세요.
66+2+8+55+100
66
2
8
55
100
숫자들의 합은 231

예제 4-15 문자열 find 및 replace

&가 입력될 때까지 여러 줄의 영문 문자열을 입력 받고, 찾는 문자열과 대치할 문자열을 각각 입력 받아 문자열을 변경하라.

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

int main() {
    string s;
    cout << "여러 줄의 문자열을 입력하세요. 입력의 끝은 &문자입니다." << endl;
    getline(cin, s, '&'); // 문자열 입력
    cin.ignore();
    string f, r;
    cout << endl << "find: ";
    getline(cin, f, '\n'); // 검색할 문자열 입력
    cout << "replace: ";
    getline(cin, r, '\n'); // 대치할 문자열 입력

    int startIndex = 0;
    while(true) {
        int fIndex = s.find(f, startIndex); // startIndex부터 문자열 f 검색
        if(fIndex == -1)
            break; // 문자열 s의 끝까지 변경하였음
        s.replace(fIndex, f.length(), r); // fIndex부터 문자열 f의 길이만큼 문자열 r로 변경
        startIndex = fIndex + r.length();
    }
    cout << s << endl;
}
```

& 뒤에 따라 오는 <Enter> 키를 제거하기 위한 코드!!!

예제 4-15 실행 결과

여러 줄의 문자열을 입력하세요. 입력의 끝은 &문자입니다.

It's now or never, come hold me tight. Kiss me my darling, be mine tonight
Tomorrow will be too late. It's now or never, my love won't wait&

find: now

검색할 단어

replace: Right Now

대치할 단어

It's Right Now or never, come hold me tight. Kiss me my darling, be mine tonight
Tomorrow will be too late. It's Right Now or never, my love won't wait

& 뒤에 <Enter>
키 입력