

Array and String

프로그래밍 입문(2)

Topics

- 배열 (Array)
 - 고정 길이 배열 vs. 동적 할당 배열
- 문자열 (String)
 - `char[]` 배열
 - `string` 클래스

Strings

- 문자열(string) = 문자들을 늘어놓은 것 또는 문자들의 ‘배열’.
- 문자형의 배열로 문자열을 나타낼 수 있다.
- 우리가 다루는 많은 글(text)은 모두 문자열의 집합입니다.
- 텍스트를 대상으로 한 빅데이터 분석, 기계학습의 시작은 모두 문자열을 처리하는 것.
- 따라서 문자열을 처리하는 방법을 아는 것은 매우 중요합니다.

문자열을 표현하는 방법

- char형의 배열: `char str[10] = "First Name";`
- string 클래스를 사용: `string str = "Last Name";`
- 아직 클래스에 대해서는 배우지 않았지만, 우선 string 클래스를 사용하는 방법만 익히도록 합니다.

char []

- char형의 배열의 각 원소에 한 글자씩을 저장하는 방식.
- 문자열의 끝을 표시하기 위해 마지막은 반드시 널 문자(Null character) '\0'가 들어야 합니다.
 - 배열의 크기가 문자열 길이+1 이상, 널문자가 없으면 일반적인 char형 배열.
- 일반적인 배열 초기화 방식 외에 " "를 사용한 초기화도 가능.
 - `char str[8] = { 'p', 'r', 'o', 'g', 'r', 'a', 'm', '\0' };`
 - `char str[8] = "program";`
 - `char str[] = "program";`

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----|-----|-----|-----|-----|-----|-----|------|
| 'p' | 'r' | 'o' | 'g' | 'r' | 'a' | 'm' | '\0' |

char []

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|------|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 'p' | 'r' | 'o' | 'g' | 'r' | 'a' | 'm' | '\0' |

- 각각의 문자에 배열처럼 인덱스를 사용하여 접근 가능.

• `str[3] = 'x';`

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|------|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 'p' | 'r' | 'o' | 'x' | 'r' | 'a' | 'm' | '\0' |

- 널문자를 대입하여 문자열 자르기를 할 수 있음.

• `str[3] = '\0'; → cout << str; → "pro";`

| | | | | | | | |
|-----|-----|-----|------|-----|-----|-----|------|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 'p' | 'r' | 'o' | '\0' | 'r' | 'a' | 'm' | '\0' |

string 클래스

- 사용을 위해 `#include <string>`이 필요.
- `char[]`를 사용하는 것에 비해 일반적으로 우리가 데이터를 다루는데 익숙한 방식으로 문자열을 다룰 수 있음.
 - `str1 = "string "; str2 = "class";`
 - `str = str1 + str2; → cout << str;`
`"string class"`
 - `str1 = str2; → cout << str1;`
`"class"`

다양한 문자열 처리 지원

- 문자열을 처리하기 위해 필요한 다양한 함수들을 라이브러리로 제공.
 - 문자열 길이
 - 특정 위치 문자 얻기
 - 부분 문자열
 - 문자열 검색
 - 문자열 바꾸기

문자열 길이

- `str.size()`함수를 사용할 수 있습니다.
 - `string str = "abcd"; int len = str.size();`
 - `len = 4;`
- `char[]`를 사용했을 경우, `#include <cstring>`으로 `strlen()`함수를 사용할 수 있습니다.
 - `char str[10] = "abcd"; int len = strlen(str);`
 - `len = 4;`

특정 위치 문자얻기

- `str.at(index);`
 - `string str = "abcde";`
 - `str.at(3) = 'd'`
- 인덱스는 0부터 시작한다는 점에 주의해야 함.
- e.g.) 4번째 문자는 인덱스 3, 7번째문자는 인덱스 6.

부분 문자열

- `str.substr(start_index, length)`
- 부분 문자열의 시작 인덱스와 얻고 싶은 부분 문자열의 길이를 입력.
- 처음부터 6개의 문자를 얻으려면: `str.substr(0, 6);`
- 8번째 문자부터 끝까지: `str.substr(7);`
- 4번째부터 6번째: `str.substr(4, 3);`

문자열 검색

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|
| p | r | o | g | r | a | m | | i | n | t | r | o | \0 |

- `str.find(query, start_index)`
 - query 문자열을 start_index부터 검색, 인덱스를 반환
- `string str = "program intro";`
 - `str.find("ro");` → 처음부터 "ro"를 검색
 - 찾은 "ro"의 첫번째 인덱스 1을 반환.
 - `str.find("ro", 2);` → 인덱스2부터 검색
 - 찾은 "ro"의 첫번째 인덱스 11을 반환.

문자열 바꾸기

- `str.replace(start_index, length, new_string);`
 - `start_index`부터 `length`개의 문자를 `new_string`으로 바꿈.
- `string str = "I like dog";`
 - `str.replace(7, 3, "dragon!");`
 - `str = "I like dragon!";`
- `find()`와 결합
 - `string q = "dog";`
 - `str.replace(str.find(q), q.size(), "dragon!");`

문자열 입력 받기

- 일반적인 cin을 사용한 입력시 발생하는 문제.
- cin은 사용자의 입력과 연결된 통로와도 같은 것 - iostream
- 첫번째 공백(whitespace)까지만을 하나의 문자열로 인식, 처리하고 그 뒤의 내용은 다음 문자열로 인식.
- `string input; → cin >> input;`
 - `string class`를 입력하고 `enter`를 누름.

문자열 입력 받기

- string class ←

- cin의 내부에는 다음과 같이 저장이 됨 (\n이 엔터에 해당).

cin s t r i n g c l a s s \n

- `cin >> input;`이 실행되면 왼쪽부터 처리하다 첫 공백이 나오는 시점에
 끊고 문자열로 판별.

cin **s** **t** **r** **i** **n** **g** **✓** **c** **l** **a** **s** **s** **\n**

- 이후에는 이미 처리된 문자열은 사라지고 나머지부분이 남음.

| | | | | | | | | | | | | |
|-----|---|---|---|---|---|----|--|--|--|--|--|--|
| cin | c | l | a | s | s | \n | | | | | | |
|-----|---|---|---|---|---|----|--|--|--|--|--|--|

- 다음으로 `cin >> input;`이 실행되어도 이미 들어온 값이 남아있으므로 다시 입력을 받지 않음.

| | | | | | | | | | | | | | |
|-----|---|---|---|---|---|----|----|--|--|--|--|--|--|
| cin | c | l | a | s | s | \n | 15 | | | | | | |
|-----|---|---|---|---|---|----|----|--|--|--|--|--|--|

문자열 입력 받기

- 온전히 한 줄을 다 입력받으려면 다음의 방법을 사용.
 - `cin.getline(배열, 길이)`
 - `getline(cin, string변수명)`
- `char input[20]; cin.getline(input, 20);`
- `string str; getline(cin, str);`

Summary

- 문자열을 표현하기 위한 두 가지 방식: char배열, string클래스
- 문자열을 처리하기 위한 다양한 라이브러리