* **String 클래스**

문자열 생성

* String s = “welcome”; ⬄ string s(“Welcome”);
* Char s1[] = “welcome”;
* String s(s1);

문자열 추가 (기존의 string 뒤에 추가됨)

Append(추가; +의 개념)

- append(string)

- append(s,index, n) : s의 index위치에서 n개의 문자를 string에 추가

- append(s,n): s의 처음부터 n개의 문자를 string에 추가

- append(n,ch): n개의 문자 ch를 string에 추가

Assign(대입; 교체의 개념)

- assign(s[]:char): 문자 배열 또는 문자열 대입

- assign(s, index, n)

- assign(s, n)

- assign(n, ch)

검색,제거,삭제,검사

* at(index: int): char ; 인덱스 위치의 문자 반환
* clear(): void ; 전체 다 지움 & size 0 & capacity 그대로
* erase(index, n): string ; index위치에서 시작해 n개의 문자 제거
* empty(): bool

문자열 크기

-length()

-size()

-capacity()

-c\_str() = data() :char[] ; 문자열에 대한 C-문자열을 반환

문자열 비교

-compare

s1.compare(s2) : s1 문자열(기준) (>,<,=) s2 문자열

ex) s1 > s2 => 반환값: +

* Substr

substr(index, n) : 인덱스로부터 string의 n개 부분 문자열 반환

substr(index): 인덱스에서 시작하여 끝까지 반환

ex) string s1("Welcome");

cout << s1.substr(0, 1) << endl; // W 반환

cout << s1.substr(3, 3) << endl; // com 반환

* Find

find(ch)

find(ch, index)

fing(s)

find(s,index)

* Insert & replace

Insert(index,s) : string의 인덱스 자리에 삽입

Insert(index, n, ch)

replace(index, n, s): string의 위치 인덱스로부터 n개를 s로 교체

* **Keywords**

**-**const

매개 변수 앞 -> 매개 변수를 상수로

함수 앞-> 반환을 상수로(함수 반환 값 변경x 읽기만)

함수 뒤 -> 멤버를 상수로(멤버 변수 변경x, 읽기만)

-static

함수 내: 한 번만 초기화 ,함수 빠져나가도 소멸x, 다시 호출해도 초기화x

전역 공간: 선언된 파일 내에서 참조 허용

클래스 내: 클래스 당 하나씩 생성 & 생성자로 초기화 x, 객체 멤버 x

-inline

함수 내부의 코드 재사용 가능, 실행속도 빨라짐, 메모리 낭비o

함수 내용 호출 -> 실행 X => 호출하는 코드 자체가 함수 내용의 코드