

Standard Template Library String

Laboratório de Programação Competitiva I

Informações obtidas dos sites:

<https://web.eecs.umich.edu/~sugih/courses/eecs281/string.html>

Desenvolvido por: Dr. Mark J. Sebern e

<http://www.cplusplus.com/reference/string/string/>

Adaptação e apresentação em Impress por:

Rene Pegoraro

Pedro Henrique Paiola

Wilson M Yonezawa

ANSI String Class

- Classe que facilita a utilização de cadeias de caracteres em C++
 - `#include <string>`
- Não esqueça
 - `using namespace std;`

Iniciação

- Iniciando uma string vazia:
`string str1;`
- Com uma expressão:
`string str2 = str1;`
`string str3 = str1 + str2;`
`string str4(str2); // alternativa`
- Com uma cadeia literal:
`string str4 = "Hello there";`
`string str5("Goodbye");`
- Não é permitido:
`string str6 = 'A'; // incorreto`
`string str7('A'); // incorreto`
- Com caracter constante:
`string str7(10,'A'); // preenchida`
`// com 10 caracteres 'A'.`
- Com substring
`string str8 = "ABCDEFGHijkl";`
`string str9(str8,2,5); // inicia str9`
`// com "CDEFG"`

Comprimento

- Funções que retornam o comprimento de uma string
 - `size_type length() const;`
 - `size_type size() const;`
- Exemplo:

```
string str = "Hello";  
string::size_type len; // int também funciona  
len = str.length(); // len == 5  
len = str.size(); // len == 5
```

Convertendo para C-style string

- Retorna o ponteiro para o primeiro caracter do vetor do tipo char da cadeia de caracteres terminada por nulo
 - `const char* c_str() const;`
- Exemplo:

```
string str_cpp="abcd";  
char str_c[20];  
...  
strcpy(str_c, str_cpp.c_str());
```

Inserir uma String em Outra

- Inserir uma string em outra em uma posição determinada
 - `string& insert(size_type pos, const string& str);`
- Exemplo:

```
string str11 = "abcdefghi";  
string str12 = "0123";  
str11.insert (3, str12); // "abc0123defghi"
```

Apaga um Pedaco de uma String

- Apaga um Pedaco de uma String
 - `string& erase(size_type pos, size_type n);`
- Exemplo:

```
string str13 = "abcdefghi";  
str12.erase(5, 3); // "abcdei"
```

Substitui uma Substring por Outra

- Apaga uma substring e insere outra no lugar
 - `string& replace(size_type pos, size_type n, const string& str);`
- Exemplo:

```
string str14 = "abcdefghi";  
string str15 = "XYZ";  
str14.replace(4, 2, str15); // "abcdXYZghi"
```


Procura a Ocorrência de uma String em Outra

- Procura a ocorrência de uma String em outra String iniciando na posição pos
 - Retorna a primeira ocorrência
 - `size_type find(const string& str, size_type pos);`
 - Retorna a última ocorrência
 - `size_type rfind(const string& str, size_type pos);`
 - Se a string for encontrada, retorna o índice do início da ocorrência; senão, retorna o valor `string::npos`.
- Exemplo:

```
string str16 = "abcdefghi";  
string str17 = "def";  
string::size_type pos = str16.find(str17, 0); // retorna 3
```

Procura a Ocorrência de um dos Caracteres em uma String

- Procura a ocorrência de um dos caracteres na string iniciando na posição pos
 - Retorna a posição da primeira ocorrência
 - `size_t find_first_of(const string& str, size_t pos = 0);`
 - Retorna a última ocorrência
 - `size_t find_last_of(const string& str, size_t pos = npos);`
 - Se nenhum caractere for encontrado, retorna o índice da ocorrência; senão, retorna o valor `string::npos`.
- Exemplo:

```
std::string str("Um texto vai aqui.");  
std::size_t found = str.find_first_of("aeiou"); // retorna 4
```

Retorna uma Substring

- Retorna uma Substring com n caracteres a partir da posição pos
 - `string substr(size_type pos, size_type n);`
- Exemplo:
 `string str18 = "abcdefghi"`
 `string str19 = str18.substr(6, 2); // "gh"`

Operador = Atribuição

- Atribui uma string a outra

```
string string_one = "Hello";  
string string_two;  
string_two = string_one;
```
- Com cadeias literais

```
string string_three;  
string_three = "Goodbye";
```

- Com um único caracter

```
string string_four;  
char ch = 'A';  
string_four = ch;  
string_four = 'Z';
```

Operador + Concatenação

- Concatena duas Strings

```
string str1 = "Hello ";  
string str2 = "there";  
string str3 = str1 + str2; // "Hello there"
```
- Uma string com uma constante

```
string str1 = "Hello ";  
string str4 = str1 + "there";
```
- Uma string com um caracter único

```
string str5 = "The End";  
string str6 = str5 + '!';
```

Operador +=

Concatena e Atribui

- Concatena duas Strings

```
string str1 = "Hello ";  
string str2 = "there";  
str1 += str2; // "Hello there"
```
- Uma string com uma constante

```
string str1 = "Hello ";  
str1 += "there"; // "Hello there"
```
- Uma string com um caractere único

```
string str5 = "The End";  
str5 += '!'; // "The End!"
```

Operadores de Comparação

`==, !=, <, >, <=, >=`

- Compara duas strings

```
string str1 = "Joao";  
string str2 = "Maria";  
if (str1 > str2) {
```

...

- Uma string e uma cadeia literal

```
string str1 = "Joao";  
if (str1 != "Pedro") {
```

...

Operador []

Caractere Indicado por Índice

- Faz acesso ao caracter indicado pelo índice

```
string str10 = "abcdefghi";
```

```
char ch = str10[3]; // 'd'
```

```
str10[5] = 'X'; // "abcdeXghi"
```


Operador << e >>

Entrada e Saída no Modo Texto

- Escreve a string para o output stream
 string str1 = "Hello there";
 cout << str1 << endl;
- Le uma string a partir do input stream
 string str1;
 cin >> str1;