

CI208: Programação de Computadores

Profa. Carmem Hara

Departamento de Informática/UFPR

4 de junho de 2024

Resumo

O Tipo String

Objetivos da aula

- Mostrar como manipular strings em C++

Declaração de Vetor de Caracteres

- **Strings** são vetores de caracteres (tipo char)
- O último caracter é o `'\0'`
- Exemplo:
 - `char nomeVetor[10] = {'a', 'n', 'a'};`
Isto é um vetor de char e NÃO um string
 - `char nomeString[10] = {'a', 'n', 'a', '\0'};`
Isto é um string
 - `char nome[10] = "ana";`
Isto é um string

- `cin >> variavel;`
 - ignora separadores iniciais (branco, TAB, ENTER)
 - para a leitura quando encontra um separador (branco, TAB, ENTER)
- `cin.getline( variavel, tam_maximo )`
faz a leitura até encontrar o caracter de final de linha (`'\n'`)

Exemplo com cin

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 #define TAM 50
6
7 int main(){
8     char nome[TAM] = "nao sabe";
9     char outroNome[TAM];
10
11     cout << "Entre com o seu nome: " ;
12     cin >> nome;
13
14     cout << "Seu nome: " << nome << " tem " << strlen(nome) << " letras\n";
15     cout << "Inicial: " << nome[0] << endl;
16     cout << "Tamanho na memoria: " << sizeof( nome ) << endl;
17
18     cout << "Entre com outro nome: ";
19     cin >> outroNome;
20     cout << "Voce digitou: " << outroNome << endl;
21     return 0;
22 }
```

Exemplo com cin.getline

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4
5  #define TAM 50
6
7  int main(){
8      char nome[TAM] = "nao sabe";
9      char outroNome[TAM];
10
11     cout << "Entre com o seu nome: " ;
12     cin.getline( nome, TAM );
13
14     cout << "Seu nome: " << nome << " tem " << strlen(nome) << " letras\n";
15     cout << "Tamanho na memoria: " << sizeof( nome ) << endl;
16
17     cout << "Entre com outro nome: ";
18     cin.getline( outroNome, TAM );
19     cout << "Voce digitou: " << outroNome << endl;
20     return 0;
21 }
```

- Atribuição: `strcpy( stringDestino, stringOrigem );`
- Comparação: `strcmp( string1, string2 );`
 - retorna zero se `string1 == string2`
 - retorna valor negativo se `string1 < string2`
 - retorna valor positivo se `string1 > string2`
- Concatenação: `strcat( string1, string2);`
concatena `string2` no `string1`

Exemplo

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4
5  #define TAM 50
6
7  int main(){
8      char nome[TAM] = "Maria";
9      char sobreNome[TAM] = "Silva";
10     char completo[TAM];
11
12     strcpy( completo, nome );           // NAO: completo= nome;
13     strcat( completo, " " );
14     strcat( completo, sobreNome );
15     cout << "Nome: " << completo << endl; // "Maria Silva"
16
17     if( strcmp( nome, sobreNome ) == 0 )
18         cout << "Nome e sobrenome sao iguais\n";
19     else if( strcmp( nome, sobreNome ) < 0 )
20         cout << "Nome eh menor que sobrenome na ordem lexicografica\n";
21     else
22         cout << "Nome eh maior que sobrenome na ordem lexicografica\n";
23     return 0;
24 }
```


O Tipo string em C++

- Não é preciso se preocupar com o tamanho do string
- Redimensionamento é feito automaticamente
- `string nome = "Maria";`

Exemplo

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  #define TAM 15
6
7  int main(){
8      string nome, sobrenome, completo;
9
10     cout << "Entre com o seu nome: " ;
11     cin >> nome;
12
13     cout << "Seu nome: " << nome << " tem " << nome.size() << " letras\n";
14     cout << "Inicial: " << nome[0] << endl;
15
16     cout << "Entre com o seu sobrenome: ";
17     cin >> sobrenome;
18
19     completo= nome;
20     completo= completo + " " + sobrenome;
21     cout << "Nome completo: " << completo << endl;
22     cout << "Nome completo: " << completo << " tem " << completo.size() << " letras\n";
23     return 0;
24 }
```