

Universidade Tecnológica Federal do Paraná UTFPR – Campus Curitiba

Orientação a Objetos - Programação em C++

Grupo de Slides 13:
A classe Predefinida *String*.

Prof. Jean Marcelo SIMÃO

Classe *String*

Uma classe Predefinida.

Exemplos adaptados do livro dos Deitels:

- Deitel H. M., Deitel, P. J. "C++ Como Programar". 3a Edição Bookman, 2001.

Atribuição e concatenação de *strings*

```
#include "stdafx.h"

#include <iostream>
using std::cout;
using std::endl;

#include <string>
using std::string;

int _tmain ( int argc, _TCHAR* argv[] )
{
    string s1 ( "Bom dia" ), s2, s3;
    s2 = s1; // Atribui s1 a s2 com =
    s3.assign( s1 );

    cout << "S1: " << s1 << endl;
    cout << "S2: " << s2 << endl;
    cout << "S3: " << s3 << endl;
    cout << endl;

    // modifica S2 e S3.
    s2[ 0 ] = s3[ 2 ] = 'r';

    cout << "S1: " << s1 << endl;
    cout << "S2: " << s2 << endl;
    cout << "S3: " << s3 << endl;
    cout << endl;

    // demonstrando o método at()
    int tam = s3.length();
    for (int x = 0; x < tam; ++x)
    {
        cout << s3.at( x );
    }
    cout << endl;
}
```

```
// concatenação
string s4 ( s1 + " gente"); // vira "Bom dia gente" em s4

// += sobregarregado
s3 += "cidadaos";
s1.append(" pessoal."); // cria "Bom dia pessoal" em s1

cout << "S1: " << s1 << endl;
cout << "S2: " << s2 << endl;
cout << "S3: " << s3 << endl;
cout << "S4: " << s4 << endl;
cout << endl;

// acrescenta ! as posições até o final de S1
string s5;
s5.append( s1, 4, s1.size() );

cout << "S1: " << s1 << endl;
cout << "S2: " << s2 << endl;
cout << "S3: " << s3 << endl;
cout << "S4: " << s4 << endl;
cout << "S5: " << s5 << endl;
cout << endl;

return 0;
}
```

```
C:\WINDOWS\system32\cmd.exe
S1: Bom dia
S2: Bom dia
S3: Bom dia

S1: Bom dia
S2: rom dia
S3: Bor dia

Bor dia
S1: Bom dia pessoal
S2: rom dia
S3: Bor dia cidadaos
S4: Bom dia gente

S1: Bom dia pessoal
S2: rom dia
S3: Bor dia cidadaos
S4: Bom dia gente
S5: dia pessoal

Press any key to continue . . . _
```

```

#include "stdafx.h"
#include <iostream> using std::cout; using std::endl;
#include <string> using std::string;
int _tmain(int argc, _TCHAR* argv[])
{
    string s1 ("Testando as funcoes de comparacao"),
           s2 ("Bom dia"), s3 ("Ops"), z1 ( s2 );
    cout << "s1: " << s1 << endl;
    cout << "s2: " << s2 << endl;
    cout << "s3: " << s3 << endl;
    cout << "z1: " << z1 << endl;
    cout << endl;

    // comparação de s1 e z1
    if ( s1 == z1 )
    {
        cout << "s1 == z1" << endl;
    }
    else
    {
        if ( s1 > z1 )
        {
            cout << "s1 > z1" << endl;
        }
        else
        {
            cout << "s1 < z1" << endl;
        }
    }
}

// comparacao de S1 com S2
int f = s1.compare( s2 );
if ( f == 0 )
{
    cout << "s1.compare( s2 ) == 0" << endl;
}
else
{
    if ( s1 > z1 )
    {
        cout << "s1.compare( s2 ) > 0" << endl;
    }
    else
    {
        cout << "s1.compare( s2 ) < 0" << endl;
    }
}
}

```

```

// comparacao de S1 (elementos 2 - 3) e s3 (elementos 0 - 5)
f = s1.compare( 2, 3, s3, 0, 5 );
if ( f == 0 )
{
    cout << "s1.compare( 2, 3, s3, 0, 5 ) == 0" << endl;
}
else
{
    if ( s1 > z1 )
    {
        cout << "s1.compare( 2, 3, s3, 0, 5 ) > 0" << endl;
    }
    else
    {
        cout << "s1.compare( 2, 3, s3, 0, 5 ) < 0" << endl;
    }
}

// comparacao de s2 e z1
f = z1.compare( 0, s2.size(), s2 );
if ( f == 0 )
{
    cout << "z1.compare( 0, s2.size, s2 ) == 0" << endl;
}
else
{
    if ( s1 > z1 )
    {
        cout << "z1.compare( 0, s2.size, s2 ) > 0" << endl;
    }
    else
    {
        cout << "z1.compare( 0, s2.size, s2 ) < 0" << endl;
    }
}

return 0;
}

```

Comparação de String

```
C:\WINDOWS\system32\cmd.exe
s1: Testando as funcoes de comparacao
s2: Bom dia
s3: Ops
z1: Bom dia

s1 > z1

s1.compare< s2 > > 0
s1.compare< 2, 3, s3, 0, 5 > > 0
z1.compare< 0, s2.size, s2 > == 0
Press any key to continue . . . _
```

Substrings

```
#include "stdafx.h"

#include <iostream>
using std::cout;
using std::endl;

#include <string>
using std::string;

int _tmain(int argc, _TCHAR* argv[])
{
    string s ( "O aeroplano desapareceu no horizonte." );

    cout << s.substr( 6, 5 ) << endl << endl;

    return 0;
}
```



A screenshot of a Windows command prompt window. The title bar reads "C:\WINDOWS\system32\cmd.exe". The window content shows the output of the program: "plano" followed by "Press any key to continue . . . _". The window has standard Windows window controls (minimize, maximize, close) in the top right corner and a scrollbar at the bottom.

Intercambiando strings

```
#include "stdafx.h"

#include <iostream>
using std::cout;
using std::endl;

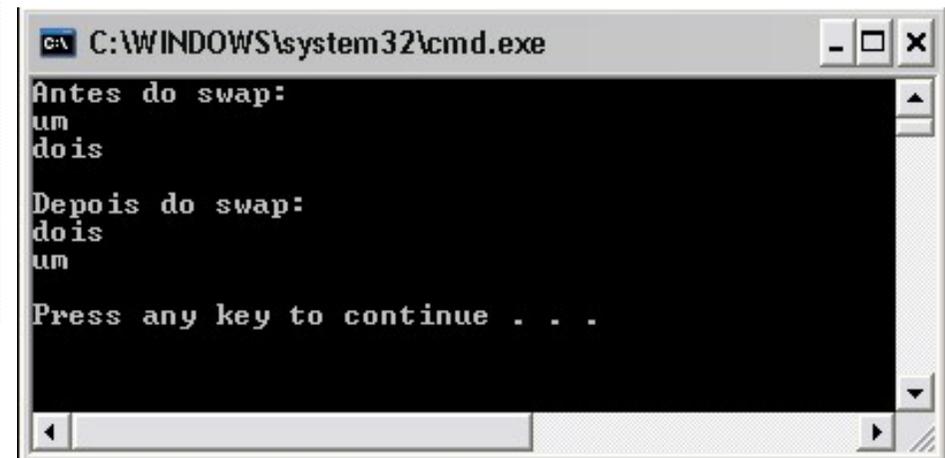
#include <string>
using std::string;

int _tmain(int argc, _TCHAR* argv[])
{
    string primeiro ("um"), segundo ("dois");

    cout << "Antes do swap: " << endl;
    cout << primeiro          << endl;
    cout << segundo           << endl;
    cout << endl;

    primeiro.swap( segundo );
    cout << "Depois do swap: " << endl;
    cout << primeiro          << endl;
    cout << segundo           << endl;
    cout << endl;

    return 0;
}
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The output of the program is displayed as follows:

```
Antes do swap:
um
dois

Depois do swap:
dois
um

Press any key to continue . . .
```

Características de strings

```
#include "stdafx.h"

#include <iostream>
using std::cout;
using std::cin;
using std::endl;

#include <string>
using std::string;

void imprimeSituacao( const string &str );

int _tmain(int argc, _TCHAR* argv[])
{
    string s;

    cout << "Características antes de leitura." << endl;
    imprimeSituacao( s );

    cout << "Digite uma string: " << endl;
    cin >> s;
    cout << endl;
    cout << "A string digitada foi : " << s << endl;

    cout << endl;
    cout << "Características depois da leitura: " << endl;
    imprimeSituacao( s );

    s.resize( s.length() + 10);

    cout << "Características depois de redimensionar: " << endl;
    imprimeSituacao ( s );

    return 0;
}
```

```
void imprimeSituacao( const string &str )
{
    cout << " Capacidade:      " << str.capacity()    << endl
         << " Tamanho máximo: " << str.max_size()    << endl
         << " Tamanho atual:    " << str.size()         << endl
         << " Comprimento atual:" << str.length()       << endl
         << " Vazio:           " << ( str.empty() ? "verdade" : "falso" )
         << endl << endl;
}
```

```
C:\WINDOWS\system32\cmd.exe
Características antes de leitura.
Capacidade:          15
Tamanho máximo:     4294967294
Tamanho atual:       0
Comprimento atual:   0
Vazio:              verdade

Digite uma string:
Conhecimento é fundamental.

A string digitada foi : Conhecimento

Características depois da leitura:
Capacidade:          15
Tamanho máximo:     4294967294
Tamanho atual:       12
Comprimento atual:  12
Vazio:              falso

Características depois de redimensionar:
Capacidade:          31
Tamanho máximo:     4294967294
Tamanho atual:       22
Comprimento atual:  22
Vazio:              falso

Press any key to continue . . . _
```

Encontrar caracteres em strings.

```
#include "stdafx.h"

#include <iostream> using std::cout; using std::cin; using std::endl;
#include <string> using std::string;

int _tmain ( int argc, _TCHAR* argv[] )
{
    // o compilador concatena todas as partes em um só string.
    string s ( " Oh ! Que saudades que tenho "
              "\n Da aurora da minha vida,"
              "\n Da minha infância querida"
              "\n Que os anos não trazem mais!"
              "\n Que amor, que sonhos, que flores,"
              "\n Nasquelas tardes fagueiras,"
              "\n À sombra das bananeiras, "
              "\n Debaixo dos laranjais! "
              "\n ... de Casimiro de Abreu. " );

    // encontra a primeira e última posições de "Que"
    cout << " String original: " << endl
          << " A substring 'Que' foi encontrada (via find) na posição: "
          << s.find ( "Que" ) << endl
          << " A substring 'Que' foi encontrada (via find) na posição: "
          << s.rfind ( "Que" ) << endl
          << " A substring 'Que' foi encontrada (via find) na posição: "
          << s.find ( "Que" ) << endl << endl;

    // encontra o primeiro e último caracter existente na string "Que"
    cout << " String original: " << endl
          << " Caracter de 'Que' encontrado por primeiro (via find_first_of) na posição: "
          << s.find_first_of ( "Que" ) << endl
          << " Caracter de 'Que' encontrado por último (via find_last_of) na posição: "
          << s.find_last_of ( "Que" ) << endl << endl;

    // encontra o primeiro e último caracter NÃO existente na string "Que"
    cout << " String original: " << endl
          << " Caracter NAO em 'Que' encontrado por primeiro (via find_first_of) na posição: "
          << s.find_first_not_of ( "Que" ) << endl
          << " Caracter NAO em 'Que' encontrado por último (via find_last_of) na posição: "
          << s.find_last_not_of ( "Que" ) << endl << endl;

    return 0;
}
```

C:\WINDOWS\system32\cmd.exe

String original:

```
A substring 'Que' foi encontrada (via find) na posição: 6  
A substring 'Que' foi encontrada (via find) na posição: 118  
A substring 'Que' foi encontrada (via find) na posição: 6
```

String original:

```
Caracter de 'Que' encontrado por primeiro (via find_first_of) na posição: 6  
Caracter de 'Que' encontrado por último (via find_last_of) na posição: 260
```

String original:

```
Caracter NAO em 'Que' encontrado por primeiro (via find_first_of) na posição: 0  
Caracter NAO em 'Que' encontrado por último (via find_last_of) na posição: 262
```

Press any key to continue . . . _

Substituir caracteres em strings.

```
#include "stdafx.h"
#include <iostream> using std::cout; using std::cin; using std::endl;
#include <string> using std::string;

int _tmain (int argc, _TCHAR* argv[])
{
    // o compilador concatena todas as partes em um só string.
    string s ( " Oh ! Que saudades que tenho"
              "\n Da aurora da minha vida,"
              "\n Da minha infância querida"
              "\n Que os anos não trazem mais!"
              "\n Que amor, que sonhos, que flores,"
              "\n Nasquelas tardes fagueiras,"
              "\n À sombra das bananeiras, "
              "\n Debaixo dos laranjais! "
              "\n ... de Casimiro de Abreu. " );

    // Remove todos os caracteres da posição 62 até o fim de s
    s.erase( 62 );

    // encontra a primeira e última posições de "Que"
    cout << "String depois do Erase: " << s << endl << endl;

    // substitui todos os espaços por um ponto.
    int x = s.find(" ");
    while ( x < string::npos )
    {
        s.replace( x, 1, "." );
        x = s.find( " ", x + 1 );
    }
    cout << "String depois da primeira substituição: " << s << endl << endl;

    x = s.find(".");
    while ( x < string::npos )
    {
        s.replace( x, 2, "xxxxx;yyy", 5, 2 );
        x = s.find( ".", x + 1 );
    }
    cout << "String depois da segunda substituição: " << s << endl << endl;

    return 0;
}
```

```
C:\WINDOWS\system32\cmd.exe
String depois do Erase: Oh ? Que saudades que tenho
Da aurora da minha vida,
Da

String depois da primeira substituição: .Oh.?.Que.saudades.que.tenho..
.Da.aurora.da.minha.vida,.
.Da.

String depois da segunda substituição: ;;h;;;ue;;audades;;ue;;enho;;
;;a;;urora;;a;;inha;;ida,;;;a;;

Press any key to continue . . . _
```

Inserir caracteres em strings.

```
#include "stdafx.h"
#include <iostream> using std::cout; using std::cin; using std::endl;
#include <string> using std::string;

int _tmain (int argc, _TCHAR* argv[])
{
    string s1 ( "início fim" ),
           s2 ( "meio" ),
           s3 ( "12345678" ),
           s4 ( "xx" );

    cout << "Strings iniciais: " << endl
         << "s1: " << s1 << endl
         << "s2: " << s2 << endl
         << "s3: " << s3 << endl
         << "s4: " << s4 << endl << endl;

    // Insere "meio" na posição 7 de s1
    s1.insert( 7, s2 );

    // Insere "xx" de S4 na posição 3 de s3
    s3.insert( 3, s4, 0, string::npos);

    cout << "Strings depois das inserções: " << endl
         << "s1: " << s1 << endl
         << "s2: " << s2 << endl
         << "s3: " << s3 << endl
         << "s4: " << s4 << endl << endl;

    return 0;
}
```

```
C:\WINDOWS\system32\cmd.exe
Strings iniciais:
s1: início fim
s2: meio
s3: 12345678
s4: xx

Strings depois das inserções:
s1: início meio fim
s2: meio
s3: 123xx45678
s4: xx

Press any key to continue . . . _
```

Conversão de objetos string para variáveis char*.

```
#include "stdafx.h"
#include <iostream>
using std::cout; using std::endl;
#include <string>
using std::string;

int _tmain ( int argc, _TCHAR* argv[] ) {
    string s ( "STRINGS" );

    const char *ptr1 = 0;
    int tam = s.length();
    char *ptr2 = new char[ tam + 1 ]; // incluindo o nulo

    // Copia caracteres do string para a memória alocada.
    s.copy( ptr2, tam, 0);
    ptr2[ tam ] = 0; // acrescenta terminador nulo

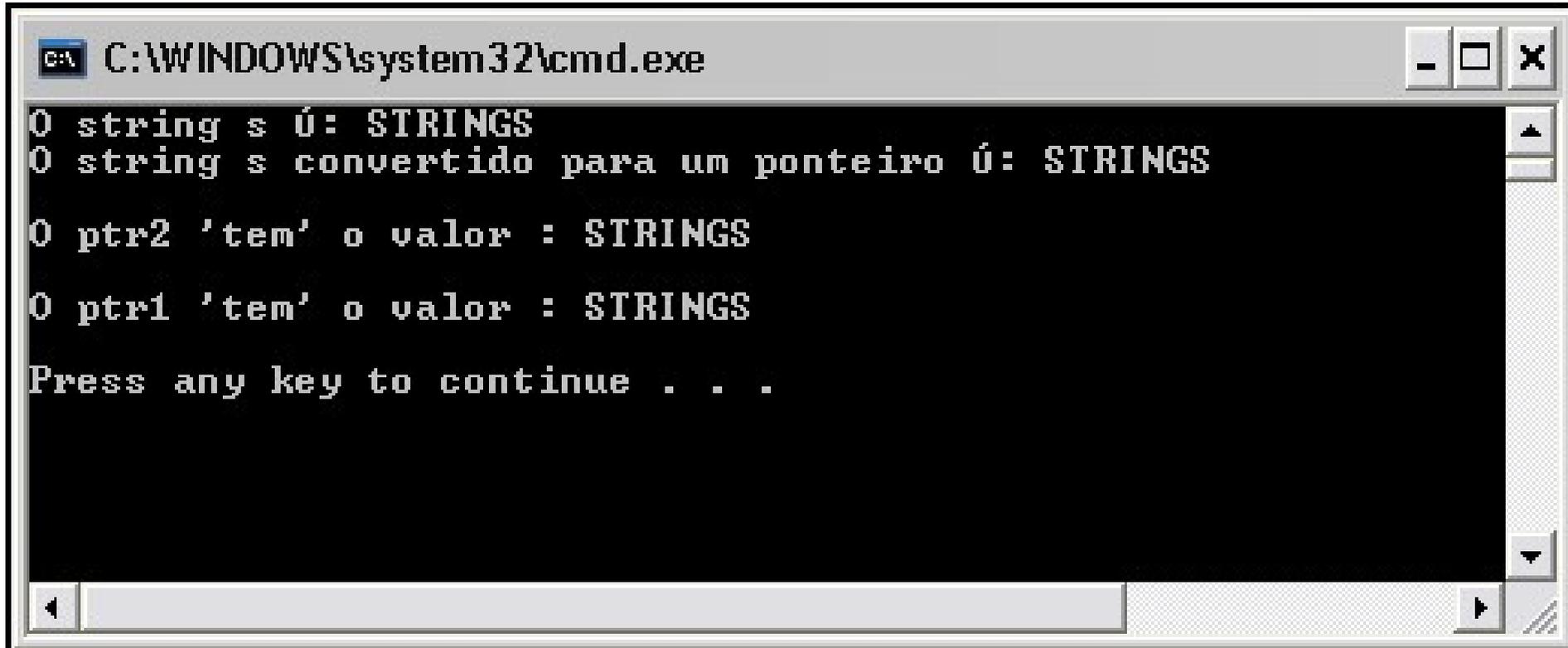
    // Saída
    cout << "O string s é: " << s << endl;
    cout << "O string s convertido para um ponteiro é: " << s.c_str() << endl << endl;

    cout << "O ptr2 'tem' o valor : " << ptr2 << endl << endl;

    // Atribui o const char* retornado pela função data() ao ponteiro ptr1
    // Nota: esta é uma atribuição perigosa, uma vez que se a string s for
    // modificada, o ponteiro ptr1 pode se tornar inválido.

    ptr1 = s.data();
    cout << "O ptr1 'tem' o valor : ";
    for ( int k = 0; k < tam; ++k ) {
        cout << *( ptr1 + k );
    }
    cout << endl << endl;

    delete [] ptr2;
    return 0;
}
```



A screenshot of a Windows command prompt window. The title bar reads "C:\WINDOWS\system32\cmd.exe". The window contains the following text:

```
0 string s ú: STRINGS
0 string s convertido para um ponteiro ú: STRINGS
0 ptr2 'tem' o valor : STRINGS
0 ptr1 'tem' o valor : STRINGS
Press any key to continue . . .
```

```
#include <iostream>
using std::cout;
using std::endl;

#include <string>
using std::string;

int _tmain (int argc, _TCHAR* argv[])
{
    string s ( "Apresentando iteradores." );

    const_iterator iterador1;

    iterador1 = s.begin();

    cout << "O conteúdo de s usando o iterador1: " << endl << endl;

    while ( iterador1 != s.end() )
    {
        cout << *iterador1;
        ++iterador1;
    }

    cout << endl;

    return 0;
}
```



```
C:\WINDOWS\system32\cmd.exe
O conte-do de s usando o iterador1:
Apresentando iteradores.
Press any key to continue . . .
```

Processamento de strings em streams

```
#include "stdafx.h"
#include <iostream>
using std::cout; using std::endl;
#include <string>
using std::string;
#include <sstream>
using std::ostringstream;
int _tmain ( int argc, _TCHAR* argv[] )
{
    ostringstream StringSaida;

    string s1 ( "Saída de diversos tipos de dados " ),
           s2 ( "usando um objeto ostringstream: " ),
           s3 ( "\n                double:  "),
           s4 ( "\n                int:      "),
           s5 ( "\n                endereço de int:  ");

    double d = 123.4567;
    int i = 22;

    StringSaida << s1 << s2 << s3 << d << s4 << i << s5 << &i;

    cout << "StringSaida contém: " << endl
         << StringSaida.str() << endl;

    cout << endl << endl;

    StringSaida << " Mais caracteres adicionados no stream.";

    cout << " Após as inserções adicionais no stream, " << endl
         << " StringSaida contém: " << endl
         << StringSaida.str() << endl;

    cout << endl << endl;

    return 0;
}
```

Exercícios

- No projeto que desenvolvemos antes em sala (o das universidades), utilize a classe predefinida *string* em vez de vetores de *chars* (*caracteres*).