

# Universidade Tecnológica Federal do Paraná

## UTFPR – Campus Curitiba

---

# Orientação a Objetos

-

# Programação em C++

---

Grupo de Slides 12:  
*Const, Static, Namespace e Nested*

**Prof. Jean Marcelo SIMÃO**

# Métodos *Const*

```

#ifndef _MINHASTRING_H_
#define _MINHASTRING_H_

class MinhaString
{
public:

    MinhaString ( const char* s = "" );
    ~MinhaString ( );

    const char* getString ( ) const;

    // Sobre carga de operadores
    void operator = ( const char* s );
    void operator = ( MinhaString& s );

    bool operator == ( MinhaString& s );

private:
    void setString ( const char* s );

private:
    int tamanho;
    char* pStr; // Ponteiro para o início do string.

    const int TAM_MAX;
};

ostream &operator<<(ostream &saida, MinhaString& s);

#endif

```

```

MinhaString::MinhaString ( const char* s ) :
    TAM_MAX ( 5000 ),
    tamanho (strlen (s) )
{
    ...
}

...
void MinhaString::setString ( const char* s )
{
    pStr = new char [ tamanho + 1 ];
    strcpy ( pStr, s );
}

const char* MinhaString::getString() const
{
    return pStr;
}

...

```

# Atributos e Métodos *Static*

```

#ifndef _MINHASTRING_H_
#define _MINHASTRING_H_

class MinhaString
{
public:
    MinhaString (const char* s = "");
    ~MinhaString ( );
    const char* getString ( ) const;

    // Sobre carga de operadores
    void operator = (const char* s);
    void operator = (MinhaString& s);

    bool operator == (MinhaString& s);

    static const char* getNomeClasse();

private:
    void setString ( const char* s );

private:
    int tamanho;
    char* pStr; // Ponteiro para o início do string.

    static const char nomeclasse [12];
};

ostream &operator<<(ostream &saida, MinhaString& s);

#endif

```

```

#include "stdafx.h"
#include "MinhaString.h"

const char MinhaString::nomeclasse[12] = "MinhaString";

const char* MinhaString::getNomeClasse ( )
{
    return nomeclasse;
}

MinhaString::MinhaString ( const char* s ):
    tamanho( strlen( s ) )
{
    setString ( s );
}

MinhaString::~MinhaString ( )
{
    delete [] pStr;
}

void MinhaString::setString ( const char* s )
{
    pStr = new char[ tamanho + 1 ];
    strcpy ( pStr, s );
}

const char* MinhaString::getString ( ) const
{
    return pStr;
}

...

```

```

#ifndef _MINHASTRING_H_
#define _MINHASTRING_H_

class MinhaString
{
public:
    MinhaString (const char* s = "");
    ~MinhaString ( );
    const char* getString ( ) const;

    // Sobre carga de operadores
    void operator = (const char* s);
    void operator = (MinhaString& s);

    bool operator == (MinhaString& s);

    static const char* getNomeClasse();

    static int getCont() { return cont; }

private:
    void setString ( const char* s );

private:
    int tamanho;
    char* pStr; // Ponteiro para o início do string.

    static const char nomeclasse [12];

    static int cont;
};

ostream &operator<<(ostream &saida, MinhaString& s);

#endif

```

Atenção: a palavra *static* só aparece no .h.

```

#include "stdafx.h"
#include "MinhaString.h"

const char MinhaString::nomeclasse[12] = "MinhaString";

int MinhaString::cont = 0;

const char* MinhaString::getNomeClasse ( )
{
    return nomeclasse;
}

MinhaString::MinhaString ( const char* s ):
tamanho( strlen( s ) )
{
    cont++;
    setString ( s );
}

MinhaString::~~MinhaString ( )
{
    cont--;
    delete [] pStr;
}

void MinhaString::setString ( const char* s )
{
    pStr = new char[ tamanho + 1 ];
    strcpy ( pStr, s );
}

const char* MinhaString::getString ( ) const
{
    return sPtr;
}

...

```

```

#include "stdafx.h"
#include "MinhaString.h"

int _tmain ( int argc, _TCHAR* argv[] )
{
    MinhaString S1 ( "Minha primeira string soh minha." ),
                S2 ( "Minha segunda string soh minha." );

    cout << "Programa para testar o uso de static. " << endl << endl;

    cout << "Neste programa há objetos da classe: "
         << MinhaString::getNomeClasse() << endl << endl;

    cout << "O objeto S1 é da classe: " << S1.getNomeClasse() << endl;
    cout << "O objeto S2 é da classe: " << S2.getNomeClasse() << endl;

    cout << endl << endl;
    return 0;
}

```

```

C:\WINDOWS\system32\cmd.exe
Programa para testar o uso de static.

Neste programa há objetos da classe: MinhaString

O objeto S1 é da classe: MinhaString
O objeto S2 é da classe: MinhaString

Press any key to continue . . .

```

```

#include "stdafx.h"
#include "MinhaString.h"

int _tmain ( int argc, _TCHAR* argv[] )
{
    MinhaString S1 ( "Minha primeira string soh minha." ),
                S2 ( "Minha segunda string soh minha." ),
                S3 ( "wHATAEVER." );

    cout << S1.getCont() << S2.getCont() << S3.getCont() << endl << endl;

    cout << MinhaString::getCont() << endl << endl;

    cout << "Programa para testar o uso de static. " << endl << endl;

    cout << "Neste programa há objetos da classe: "
         << MinhaString::getNomeClasse() << endl << endl;

    cout << "O objeto S1 é da classe: " << S1.getNomeClasse() << endl;
    cout << "O objeto S2 é da classe: " << S2.getNomeClasse() << endl;

    cout << endl << endl;
    return 0;
}

```

```

#include "stdafx.h"
#include "MinhaString.h"

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

    cout << MinhaString::getCont() << endl << endl;

    MinhaString  S1 ( "Minha primeira string soh minha." ),
                 S2 ( "Minha segunda string soh minha." ),
                 S3 ( "wHATAEVER." );

    cout << S1.getCont() << S2.getCont() << S3.getCont() << endl << endl;

    cout << "Programa para testar o uso de static. " << endl << endl;

    cout << "Neste programa há objetos da classe: "
         << MinhaString::getNomeClasse() << endl << endl;

    cout << "O objeto S1 é da classe: " << S1.getNomeClasse() << endl;
    cout << "O objeto S2 é da classe: " << S2.getNomeClasse() << endl;

    cout << endl << endl;
    return 0;

}

```

```

#include "stdafx.h"
#include "MinhaString.h"

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

    cout << MinhaString::getCont() << endl << endl;

    MinhaString S1 ( "Minha primeira string soh minha." );

    cout << S1.getCont() << endl << endl;
    cout << MinhaString::getCont() << endl << endl;

    MinhaString S2 ( "Minha segunda string soh minha." ),
                S3 ( "wHATAEVER." );

    cout << S1.getCont() << S2.getCont() << S3.getCont() << endl << endl;

    cout << "Programa para testar o uso de static. " << endl << endl;

    cout << "Neste programa há objetos da classe: "
         << MinhaString::getNomeClasse() << endl << endl;

    cout << "O objeto S1 é da classe: " << S1.getNomeClasse() << endl;
    cout << "O objeto S2 é da classe: " << S2.getNomeClasse() << endl;

    cout << endl << endl;
    return 0;

}

```

# Exercício

- Fazer com que a classe *Pessoa*, do nosso “sistema de universidades”, tenha um atributo estático que permita informar quantos objetos de *Pessoa* (ou derivadas desta) existem no sistema.

# NameSpace

```

#ifndef _MINHASTRING_H_
#define _MINHASTRING_H_

namespace MinhasCoisasDeString
{

class MinhaString
{
public:
    MinhaString (const char* s = "");
    ~MinhaString();
    const char* getString() const;

    // Sobre carga de operadores
    void operator = (const char* s);
    void operator = (MinhaString& s);

    bool operator == (MinhaString& s);
    const static char* getNomeClasse();

private:
    void setString ( const char* s );

private:
    int tamanho;
    char* pStr; // Ponteiro para o início do string.
    static const char nomeclasse [12];
};

ostream &operator<<(ostream &saida, MinhaString& s);

}
#endif

```

## MinhaString.h

## MinhaString.cpp

```

#include "stdafx.h"

#include "MinhaString.h"
using namespace MinhasCoisasDeString;

const char MinhaString::nomeclasse [12] = "MinhaString";

const char* MinhaString::getNomeClasse()
{
    return nomeclasse;
}

MinhaString::MinhaString (const char* s):
tamanho( strlen( s ) )
{
    setString ( s );
}

MinhaString::~~MinhaString()
{
    delete [] pStr;
}

...

```

```
#include "stdafx.h"

#include "MinhaString.h"
using namespace MinhasCoisasDeString;

int _tmain(int argc, _TCHAR* argv[])
{
    MinhaString S1 ("Minha primeira string soh minha."),
                S2 ("Minha segunda string soh minha."),
                S3 ("Minha terceira string soh minha.");

    cout << "Programa para testar o uso de static. " << endl << endl;

    cout << "Neste programa há objetos da classe: "
         << MinhaString::getNomeClasse() << endl << endl;

    cout << "O objeto S1 é da classe: " << S1.getNomeClasse() << endl;
    cout << "O objeto S2 é da classe: " << S2.getNomeClasse() << endl;
    cout << "O objeto S3 é da classe: " << S3.getNomeClasse() << endl;

    cout << endl << endl;

    return 0;
}
```

```

#include "stdafx.h"

#include "MinhaString.h"

int _tmain(int argc, _TCHAR* argv[])
{
    MinhasCoisasDeString::MinhaString S1 ("Minha primeira string soh minha."),
        S2 ("Minha segunda string soh minha."),
        S3 ("Minha terceira string soh minha.");

    cout << "Programa para testar o uso de static. " << endl << endl;

    cout << "Neste programa há objetos da classe: "
        << MinhasCoisasDeString::MinhaString::getNomeClasse() << endl << endl;

    cout << "O objeto S1 é da classe: " << S1.getNomeClasse() << endl;
    cout << "O objeto S2 é da classe: " << S2.getNomeClasse() << endl;
    cout << "O objeto S3 é da classe: " << S3.getNomeClasse() << endl;

    cout << endl << endl;

    return 0;
}

```

# Classes Aninhadas (*Nested*)

## ConjuntoStrings.h

```
#ifndef _CONJUNTOSTRINGS_H_
#define _CONJUNTOSTRINGS_H_
namespace MinhasClasses
{
    class ConjuntoStrings
    {
        ...
        public:
            class MinhaString
            {
                public:
                    MinhaString (const char* s = "");
                    ~MinhaString ();
                    const char* getString () const;
                    // Sobre carga de operadores
                    void operator = (const char* s);
                    void operator = (MinhaString& s);
                    bool operator == (MinhaString& s);
                    const static char* getNomeClasse();
                    ...
                private:
                    void setString( const char* s );

                private:
                    int tamanho;
                    char* pStr;
                    ...
                    static const char nomeclasse[12];
            };

            void imprimeVet() const;
        private:
            MinhaString Vet [1000];
    };
}
#endif
```

```
#include "stdafx.h"
#include "ConjuntoStrings.h"
using namespace MinhasClasses;

const char ConjuntoStrings ::MinhaString::nomeclasse[12] = "MinhaString";

const char* ConjuntoStrings ::MinhaString::getNomeClasse()
{
    return nomeclasse;
}

ConjuntoStrings ::MinhaString::MinhaString (const char* s):
tamanho( strlen( s ) )
{
    setString( s );
}

ConjuntoStrings ::MinhaString::~MinhaString() { delete [] sPtr; }

void ConjuntoStrings ::MinhaString::setString (const char* s)
{
    sPtr = new char[ tamanho + 1 ];
    strcpy ( sPtr, s );
}

const char* ConjuntoStrings ::MinhaString::getString() const { return sPtr; }

void ConjuntoStrings::imprimeVet ( ) const
{
    // ...
}

...
```

```

#include "stdafx.h"
#include "ConjuntoStrings.h"
using namespace MinhasClasses;

int _tmain (int argc, _TCHAR* argv[])
{
    ConjuntoStrings::MinhaString  S1 ("Minha primeira string soh minha."),
                                   S2 ("Minha segunda string soh minha."),
                                   S3 ("Minha terceira string soh minha.");

    cout << "Programa para testar o uso de static. " << endl << endl;

    cout << "Neste programa há objetos da classe: "
         << ConjuntoStrings::MinhaString::getNomeClasse() << endl << endl;

    cout << "O objeto S1 é da classe: " << S1.getNomeClasse() << endl;
    cout << "O objeto S2 é da classe: " << S2.getNomeClasse() << endl;
    cout << "O objeto S3 é da classe: " << S3.getNomeClasse() << endl;

    cout << endl << endl;

    return 0;
}

```

```

#include "stdafx.h"
#include "ConjuntoStrings.h"
using namespace MinhasClasses;

int _tmain (int argc, _TCHAR* argv[])
{
    ConjuntoStrings::MinhaString  S1 ("Minha primeira string soh minha."),
                                   S2 ("Minha segunda string soh minha."),
                                   S3 ("Minha terceira string soh minha.");

    ConjuntoStrings CSs;

    cout << "Programa para testar o uso de static. " << endl << endl;

    cout << "Neste programa há objetos da classe: "
         << ConjuntoStrings::MinhaString::getNomeClasse() << endl << endl;

    cout << "O objeto S1 é da classe: " << S1.getNomeClasse() << endl;
    cout << "O objeto S2 é da classe: " << S2.getNomeClasse() << endl;
    cout << "O objeto S3 é da classe: " << S3.getNomeClasse() << endl;

    cout << endl << endl;

    return 0;
}

```

# Exercício 1

- Fazer com que *ElementoAluno* seja uma classe aninhada (privada) de *ListaAlunos*.

# Exercício 2

- Fazer com que *Elemento<Tipo>* seja uma classe aninhada (pública) de *Lista<Tipo>*.