



# Programação III

Jordana S. Salamon

[jssalamon@inf.ufes.br](mailto:jssalamon@inf.ufes.br)

[jordanasalamon@gmail.com](mailto:jordanasalamon@gmail.com)

DEPARTAMENTO DE INFORMÁTICA

CENTRO TECNOLÓGICO

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

# Namespaces

# Espaços de nome

- ▶ À medida que os códigos crescem em tamanho, aumenta a chance de colisão de nomes;
- ▶ Para evitá-lo, C++ oferece o mecanismo de *namespaces*;
  - ▶ Dois nomes iguais em espaços de nomes diferentes não colidem;
  - ▶ A API padrão do C++ é declarada no espaço de nome **std**, já visto anteriormente.
- ▶ Define-se um *namespace* como se define classes:

```
namespace mylib {  
    /* ... */  
}
```



# Espaços de nome

- ▶ Ao contrário de classes, a redefinição de um *namespace* (ex.: em outro arquivo) continua a definição anterior;
- ▶ É possível criar apelidos para *namespaces*:

```
namespace VitorSouzaLibrary {  
    void hello() { cout << "Hello, world!\n"; }  
}  
namespace vs1 = VitorSouzaLibrary;  
  
int main () {  
    VitorSouzaLibrary::hello();  
    vs1::hello();  
}
```

- ▶ Como já visto, podemos usar também a diretiva:  
**using namespace** <nome-do-namespace>;



# Modularização com *namespaces*

```
#include <string>
using namespace std;
#ifndef DEPARTAMENTO_H_
#define DEPARTAMENTO_H_ ←
#include "Funcionario.h"

namespace exercicio02 {
    class Departamento {
        string nome;
        Funcionario* funcionarios[100];
        int numFuncs;
    public:
        /* ... */
    };
}

#endif
```

Como em C, `#define` é usado para evitar declaração repetida.



# Modularização com *namespaces*

```
#include <string>
using namespace std;
#ifndef FUNCIONARIO_H_
#define FUNCIONARIO_H_
namespace exercicio02 {
    class Departamento;

    class Funcionario {
        string nome;
        double salario;
        string dataAdmissao;
        Departamento* departamento;
        friend class Departamento;

public:
        /* ... */
    };
}
#endif
```

Não podemos aqui incluir “Departamento.h” pois gera inclusão circular. Portanto, fazemos uma declaração antecipada (*forward declaration*).

# Modularização com *namespaces*

```
#include <iostream>
#include "Departamento.h"
namespace exercicio02 {
    Departamento::Departamento(const char* nome) {
        /* ... */
    }
    /* ... */
}
```

```
#include <iostream>
#include "Departamento.h" ← Departamento.h inclui
                             Funcionario.h.
```

```
namespace exercicio02 {
    Funcionario::Funcionario(const char* nome, double
                             salario, const char* dataAdmissao) {
        /* ... */
    }
    /* ... */
}
```

Artigo interessante sobre o assunto:  
<http://www.cplusplus.com/forum/articles/10627/>

That's all Folks!



nemo