

Programação Orientada a Objetos

Prof. Delano M. Beder

Roteiro 05

1. Crie um projeto (C++) denominado Namespace

2. Crie uma Nova classe C++ denominada Pessoa

Mude o nome dos arquivos gerados para Pessoa1.cpp e Pessoa1.h

2.1 Arquivo Pessoa1.h (inclua essa classe no namespace **ns1**)

```
#ifndef PESSOA1_H
#define PESSOA1_H

#include <string>
#include <iostream>
using namespace std;

namespace ns1 {

    class Pessoa {
    public:
        Pessoa(string nome, int idade);
        void imprime() const;
    private:
        string nome;
        int idade;
    };

}

#endif /* PESSOA1_H */
```

2.2 Arquivo denominado Pessoa1.cpp (inclua essa classe no namespace **ns1**)

```
#include "Pessoa1.h"

namespace ns1 {

    Pessoa::Pessoa(string nome, int idade) :
        nome(nome), idade(idade) {
    }

    void Pessoa::imprime() const {
        cout << "Nome: " << nome << endl;
        cout << "Idade: " << idade << endl;
    }

}
```

3. Crie uma Nova classe C++ denominada Pessoa

Mude o nome dos arquivos gerados para Pessoa2.cpp e Pessoa2.h

3.1 Arquivo Pessoa2.h (inclua essa classe no namespace **ns2**)

```
#ifndef PESSOA2_H
#define PESSOA2_H

#include <string>
#include <iostream>
using namespace std;

namespace ns2 {

    class Pessoa {
    public:
        Pessoa(string nome, string endereco);
        void imprime() const;
    private:
        string nome;
        string endereco;
    };

}

#endif /* PESSOA2_H */
```

3.2 Arquivo denominado Pessoa2.cpp (inclua essa classe no namespace **ns2**)

```
#include "Pessoa2.h"

namespace ns2 {

    Pessoa::Pessoa(string nome, string endereco) :
        nome(nome), endereco(endereco) {
    }

    void Pessoa::imprime() const {
        cout << "Nome: " << nome << endl;
        cout << "Endereço: " << endereco << endl;
    }

}
```

4. Arquivo main.cpp

```
#include "Pessoa1.h"
#include "Pessoa2.h"

using namespace ns1;

int main(int argc, char** argv) {

    Pessoa p1("Fulano", 18);
    ns2::Pessoa p2("Sincrano", "Endereco X");

    p1.imprime();

    cout << endl;

    p2.imprime();
}
```

5. Compile e execute (verifique a saída impressa)

6. Fim