

Programmation orientée objet

Namespaces

Namespace

Problématique:

- Un programme comporte de nombreux identificateurs provenant de différentes librairies.
- Des conflits peuvent survenir quand une variable a le même identificateur qu'une autre variable.

Namespace std

- Un namespace bien connu par tous est std :
`using namespace std;`
- Toutes les fonctionnalités standards C++ en font partie :
 - `<iostream>`
 - `<fstream>`
 - `<string>`
- Les fonctions de la librairie C n'appartiennent à aucun namespace
 - `<cmath>`
 - `<ctime>`
 - `<cstdlib>`

Namespace std (suite)

- Qu'arrive-t-il si on ne déclare pas “using namespace std;” au début du programme :

```
#include <iostream>
#include <cmath>
```

```
int main()
{
    double angle = 3.1416 / 6.0;
    double sinus = sin( angle ); // Aucun namespace pour cmath

    // namespace std
    std::cout << "Angle : " << angle << std::endl;
    // Opérateur de résolution de portée ::
    std::cout << "Sinus : " << sinus << std::endl;
    return 0;
}
```

Exemple namespace

```
namespace nom1
{  int unEntier = 2;
    float floatToto;
};
namespace nom2
{  int unEntier ;
}
namespace nom3
{  int unEntier;
}
```

```
using namespace nom3;
int main()
{  nom1::unEntier = 45;
    nom2::unEntier = 32;
    // namespace nom3
    unEntier = 10;
    // namespace nom1
    nom1::floatToto = 2;
    return 0;
}
```