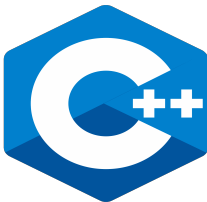


C++ : espace de noms (namespace)

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en programmation par contrainte (IA)
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`



Namespace ?

- Espace de stockage abstrait (n'existe pas physiquement)
- Répertoire virtuel
- Conçu pour développer des structures fonctionnelles sans avoir de conflit de nom

Considérons les quatre fichiers suivants

Fichier `salutationFr.h`

```
#ifndef SALUTATIONFR_H
#define SALUTATIONFR_H

void direBonjour();

#endif
```

Fichier `salutationEn.h`

```
#ifndef SALUTATIONEN_H
#define SALUTATIONEN_H

void direBonjour();

#endif
```

Fichier `salutationFr.cpp`

```
#include <iostream>
#include "salutationFr.h"

using namespace std;

void direBonjour()
{
    cout << "Bonjour" << endl;
}
```

Fichier `salutationEn.cpp`

```
#include <iostream>
#include "salutationEn.h"

using namespace std;

void direBonjour()
{
    cout << "Good morning" << endl;
}
```

C++

Le contenu de la fonction principale `main`

```
#include <iostream>

#include "salutationFr.h"
#include "salutationEn.h"

using namespace std;

int main()
{
    direBonjour();
}
```

C++

Le contenu de la fonction principale `main`

```
#include <iostream>

#include "salutationFr.h"
#include "salutationEn.h"

using namespace std;

int main()
{
    direBonjour();
}
```

En exécutant, le résultat est

multiple definition of 'direBonjour()'

Première solution

Faire `include` d'un seul fichier

© Achref EL MOUELHI ©

Première solution

Faire `include` d'un seul fichier

Mais

dans certains cas on a besoin d'utiliser les deux

Première solution

Faire `include` d'un seul fichier

Mais

dans certains cas on a besoin d'utiliser les deux

Deuxième solution

Définir chacune de ses fonctions dans un `namespace` différent.

Définissons des namespaces dans les quatre fichiers précédents

Fichier salutationFr.h

```
#ifndef SALUTATIONFR_H
#define SALUTATIONFR_H

namespace fr
{
    void direBonjour();
}

#endif
```

Fichier salutationEn.h

```
#ifndef SALUTATIONEN_H
#define SALUTATIONEN_H

namespace en
{
    void direBonjour();
}

#endif
```

Fichier salutationFr.cpp

```
#include <iostream>
#include "salutationFr.h"

using namespace std;

namespace fr
{
    void direBonjour()
    {
        cout << "Bonjour" << endl;
    }
}
```

Fichier salutationEn.cpp

```
#include <iostream>
#include "salutationEn.h"

using namespace std;

namespace en
{
    void direBonjour()
    {
        cout << "Good morning" << endl;
    }
}
```

Il reste qu'à préciser le `namespace` au moment de l'appel

```
#include <iostream>

#include "salutationFr.h"
#include "salutationEn.h"

using namespace std;

int main()
{
    fr::direBonjour();
    en::direBonjour();
}
```

Il reste qu'à préciser le `namespace` au moment de l'appel

```
#include <iostream>

#include "salutationFr.h"
#include "salutationEn.h"

using namespace std;

int main()
{
    fr::direBonjour();
    en::direBonjour();
}
```

En exécutant, le résultat est

```
Bonjour
Good morning
```

C++

On peut aussi utiliser l'opérateur `using` comme pour `std`

```
#include <iostream>

#include "salutationFr.h"
#include "salutationEn.h"

using namespace std;
using namespace fr;

int main()
{
    direBonjour();
    en::direBonjour();
}
```

C++

On peut aussi utiliser l'opérateur `using` comme pour `std`

```
#include <iostream>

#include "salutationFr.h"
#include "salutationEn.h"

using namespace std;
using namespace fr;

int main()
{
    direBonjour();
    en::direBonjour();
}
```

Même résultat

```
Bonjour
Good morning
```

On peut aussi imbriquer les namespaces (contenu de `salutationEn.h`)

```
#ifndef SALUTATIONEN_H
#define SALUTATIONEN_H

namespace en
{
    void direBonjour();
    namespace us
    {
        void direBonjour();
    }
}

#endif
```

Nouveau contenu de `salutationEn.cpp`

```
#include <iostream>
#include "salutationEn.h"

using namespace std;

namespace en
{
    void direBonjour()
    {
        cout << "Good morning" << endl;
    }
    namespace us
    {
        void direBonjour()
        {
            cout << "Good mornin" << endl;
        }
    }
}
```

C++

Pour tester

```
#include <iostream>

#include "salutationFr.h"
#include "salutationEn.h"

using namespace std;
using namespace fr;

int main()
{
    direBonjour();
    en::direBonjour();
    en::us::direBonjour();
}
```

C++

Pour tester

```
#include <iostream>

#include "salutationFr.h"
#include "salutationEn.h"

using namespace std;
using namespace fr;

int main()
{
    direBonjour();
    en::direBonjour();
    en::us::direBonjour();
}
```

En exécutant, le résultat est

```
Bonjour
Good morning
```