سلسلة تعلم البرمجة بلغة C++ الحديثة

Learn Modern C++ Programming Course

إعداد المهندس أحمد الديب

#29: Namespaces p2

Namespaces Are Open

```
namespace A {
int f();  // now A has member f()
}
namespace A {
int g();  // now A has two members, f() and g()
}
```

That way, the members of a namespace need not be placed contiguously in a single file. This can be important when converting older programs to use namespaces.

Source: The C++ Programming Language (4th Edition), Bjarne Stroustrup

Namespace Aliases

```
namespace this_is_very_long_namespace_name {
struct point {
  int x;
  int y;
};
}  // namespace this_is_very_long_namespace_name


this_is_very_long_namespace_name::point pt1;


namespace short_name = this_is_very_long_namespace_name;  // Alias
short_name::point pt2;
```

This can immensely simplify the task of replacing one version of a library with another.

Source: The C++ Programming Language (4th Edition), Bjarne Stroustrup

Namespace Composition

```
namespace lib_1 {
class Line {};
}  // namespace lib_1


namespace lib_2 {
class Text {};
}  // namespace lib_2


namespace my_lib {
using namespace lib_1;
using namespace lib_2;
}  // namespace my_ib


my_lib::Line L1;
my_lib::Text T1;
```

Source: The C++ Programming Language (4th Edition), Bjarne Stroustrup

Namespaces and Overloading

```
// old A.h:
void f(int);


// old B.h:
void f(char);


// old user.c:
#include "A.h"
#include "B.h"
void g() {
  f('a');  // f() from B.h
}
```

```
// new A.h:
namespace A {
void f(int);
}
// new B.h:
namespace B {
void f(char);
}
// new user.c:
#include "A.h"
#include "B.h"

using namespace A;
using namespace B;
void g() {
  f('a');  // f() from B.h
}
```

Function overloading works across namespaces. This is essential to allow us to migrate existing libraries to use namespaces with minimal source code changes.

Source: The C++ Programming Language (4th Edition), Bjarne Stroustrup

inline namespace

```
namespace Interface {
inline namespace V3 {  // default
double f(double);
int f(int);
template <class T>
class C {};
}  // namespace V3

namespace V2 {
// ...
}

namespace V1 {
double f(double);
template <class T>
class C {};
}  // namespace V1
}  // namespace Interface
```

Source: The C++ Programming Language (4th Edition), Bjarne Stroustrup

inline namespace Example V1

```cpp
// interface-v1.h
namespace Interface_V1 {
  void print(void) {
    std::cout << "print v1" << "\n";
  }
}  // namespace Interface_V1


// interface.h
#include <iostream>
namespace Library {
  inline
  #include "interface-v1.h"
}  // namespace Library
```

```cpp
#include "interface.h"


int main() {
  //
  Library::print();                  // default
  Library::Interface_V1::print();  // V1
}
```

```
print v1
print v1
```

Source: The C++ Programming Language (4th Edition), Bjarne Stroustrup

inline namespace Example V2

```
// interface-v2.h
namespace Interface_V2 {
  void print_new(void) {
    std::cout << "print new v2" << "\n";
  }
  void print(void) {
    std::cout << "print v2" << "\n";
  }
}  // namespace Interface_V2
```

```
// interface.h
#include <iostream>
namespace Library {
  inline
  #include "interface-v2.h"
  #include "interface-v1.h"
}  // namespace Library
```

```
#include "interface.h"


int main() {
  //
  Library::print();      // default from V2
  Library::print_new();  // default from V2
}
```

print v2
print new v2

Source: The C++ Programming Language (4th Edition), Bjarne Stroustrup

inline namespace Example V3

```
// interface-v3.h
namespace Interface_V3 {
  void print_new(void) {
    std::cout << "print new v3“ << "\n";
  }
  #include "interface-v2.h"
  using Interface_V2::print;
}  // namespace Interface_V3
```

```
// interface.h
#include <iostream>
namespace Library {
  inline
  #include "interface-v3.h"
  #include "interface-v2.h"
  #include "interface-v1.h"
}  // namespace Library
```

```
#include "interface.h"


int main() {
  //
  Library::print();      // default from V2
  Library::print_new();  // default from V3
}
```

print v2
print new v3

Source: The C++ Programming Language (4th Edition), Bjarne Stroustrup

Thank you