名字空间（NameSpace）

朱晓旭

苏州大学计算机科学与技术学院

程序规模的扩大

- C语言
  - 用户自定义函数与库函数同名，则封库函数
- 随着程序规模的扩大
  - 函数名、类名同名问题难以避免
  - 引入名字空间

名字空间定义

- 名字空间是一种程序描述逻辑分组的机制
  - 名字由程序员指定
- 名字空间是一种作用域
- 作用域一般是{}括起来的一块区域，如函数、类、if-else控制语句等都有作用域
- 缺省情况下，我们在全局域中定义各种语言符号：函数、类、模板、全局对象等

优点

- 名字空间可以跨越编译单元；
- 名字空间中声明的标识符
  - 具有一定的封装屏蔽特征
  - 不同名字空间的同名变量不会发生冲突

定义名字空间

- 格式

namespace 名称
{
}

名字空间不一定连续，可以累积

```
// ----- MathLib.h -----
namespace  MathLib
{
      int Max(int, int) ;
}
```

```
// ----- MathLib.cpp ------
namespace  MathLib
{
      int Max(int a, int b)
      {    return  (a>b?a:b);    }
}
```

嵌套名字空间

- 名字空间可以包含嵌套的名字空间

```
namespace namespace_name1
{
        // …
         namespace namespace_name2
        {
                  // …
        }
}       // 外层名字空间
```

- 和函数内局部作用域一样{}内嵌套{}

名字空间成员

- 限定在名字空间{}范围内的成员，有函数、类、模板、常量对象等
- 名字空间成员隶属于名字空间作用域，其使用以作用域运算符::来限定

访问名字空间的内容

- 直接访问

```
#include<iostream>
void main ()
{  std::cout<<"hello"<<std::endl;}
```

- 使用using

```
#include<iostream>
using namespace std;
void main ()
{  cout<<"hello"<<endl;}
```

using指示符

- using指示符是用于使用名字空间的符号
- 形式：

  using namespace 已定义名字空间名；

  如：using namespace MathLib;

  使用：Max(x, y);

例子

- myspace1.h

```
namespace myspace
{
    class CTime
    {
        int hour,minute,second;
    public:
        CTime(int h=0,int m=0,int s=0);

    };
}
```

类函数实现

- myspace1.cpp

```
#include "myspace1.h"

myspace::CTime::CTime(int h,int m,int s)
{
    hour=h;
    minute=m;
    second=s;
}
```

访问自己的名字空间中的类

```
#include "myspace1.h"
void main()
{
   myspace::CTime time1;
}
```

无名名字空间

- 无名名字空间——C++中可以用未命名的名字空间声明一个局部于某一文件的符号
- 无名名字空间的定义不能跨越多个文本文件（内部所用，不与其它文件冲突）
- 例如

```
// ----- MathLib.cpp ------
namespace  {
    void swap( double *d1, double *d2 ) { /* ... */ }
}  //  函数swap()只在文件MathLib.cpp中可见
```

C++的标准名字空间

- 在新的C++库标准中都是用名字空间
- 标准C++库中的所有组成部分都是在一个被称为 std 的名字空间中声明和定义的
- 在标准头文件如<vector>或<iostream> 中声明的函数对象和类模板都被声明在名字空间std中
- VC的Include文件夹下有两类头文件：带有.h和不带.h扩展名的，要使用不带扩展名的头文件，必须：using namespace std;