

C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>



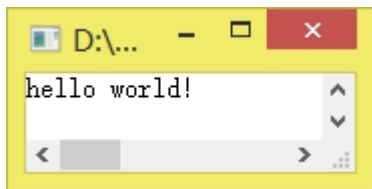


本节主题：

命名空间要解决的问题

命名空间namespace是个老相识

```
#include <iostream>
using namespace std;
int main( )
{
    cout<<"hello world!"<<endl;
    return 0;
}
```



```
#include <iostream>
```

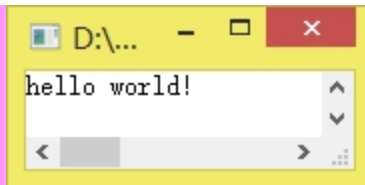
```
int main( )
```

```
{
```

```
    std::cout<<"hello world!"<<std::endl;
```

```
    return 0;
```

```
}
```



```
#include <iostream>
```

```
int main( )
```

```
{
```

```
    cout<<"hello world!"<<endl;
```

```
    return 0;
```

```
}
```

```
In function 'int main()':
5   error: 'cout' was not declared in this scope
5   note: suggested alternative:
62  note:   'std::cout'
5   error: 'endl' was not declared in this scope
5   note: suggested alternative:
562 note:   'std::endl'
```

实例：iostream 中的std(节选)

```
#include <bits/c++config.h>
#include <ostream>
#include <istream>

namespace std _GLIBCXX_VISIBILITY(default)
{
    _GLIBCXX_BEGIN_NAMESPACE_VERSION

    extern istream cin;      /// Linked to standard input
    extern ostream cout;    /// Linked to standard output
    extern ostream cerr;    /// Linked to standard error (unbuffered)
    extern ostream clog;    /// Linked to standard error (buffered)

    _GLIBCXX_END_NAMESPACE_VERSION
} // namespace
```

- 为了解决C++标准库中的标识符与程序中的全局标识符之间以及不同库中的标识符之间的同名冲突，标准C++库的所有标识符都定义在一个名为std的命名空间中。
- 在程序中用到C++标准库时，使用std作为限定

```
std::cout<<"OK."<<std::endl;
```
- 当不发生冲突时，可以在文件的开头加入以下声明

```
using namespace std;
using std::string;
using std::cout;
using std::cin;
```

大型项目中，命名是个大问题

- ❏ 惯用手法：头文件中定义类，在主文件用#include命令行包含头文件
- ❏ 在大型应用程序中，可能在不同的头文件中用了相同的名字来命名所定义的类或函数.....

```
//文件h1.h
```

```
class Table{
```

```
public:
```

```
    Table(int l, Shape s, int p){.....}
```

```
private:
```

```
    int legs;
```

```
    Shape surface;
```

```
    int price;
```

```
};
```

```
.....
```

```
//文件prog1.cpp
```

```
#include "h1.h"
```

```
int main( )
```

```
{
```

```
    Table table(4,"round",100);
```

```
    .....
```

```
}
```

```
//文件h2.h
```

```
class Table{
```

```
public:
```

```
    Table(int r, int c, int b){.....}
```

```
private:
```

```
    int rows;
```

```
    int cols;
```

```
    int border;
```

```
};
```

```
//文件prog2.cpp
```

```
#include "h1.h"
```

```
#include "h2.h"
```

```
int main( )
```

```
{
```

```
    Table table(4,"round",100);
```

```
    Table table(4,5,1);
```

```
    .....
```

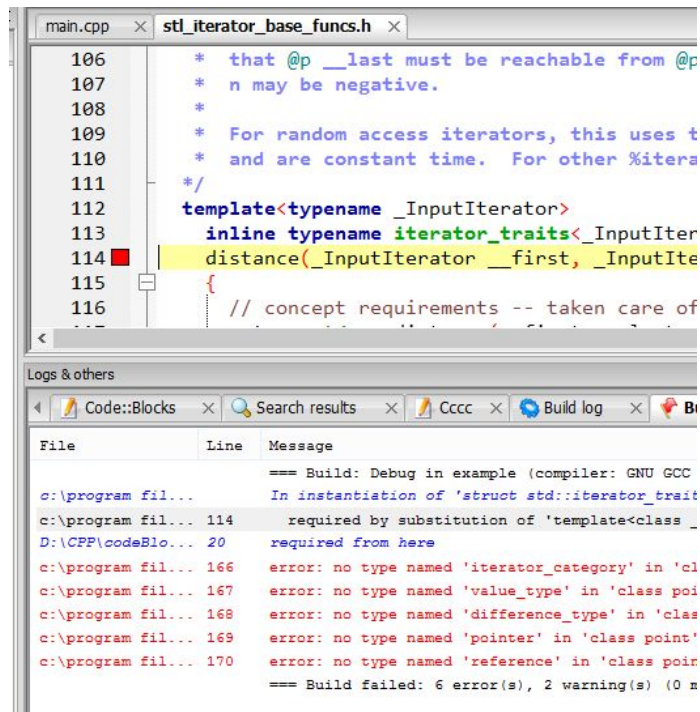
```
}
```

为什么需要命名空间

- 命名空间是ANSI C++引入的可以由用户命名的作用域，用来处理程序中常见的同名冲突。
- 在C++程序中，有大量的变量、函数和类名，如果它们都存在于同一个全局命名空间中，会导致很多冲突。
 - 自定义函数与标准库函数函数重名
 - 程序中使用的名字与第三方库中名称相同
 - 同一项目不同模块（可能由不同程序员负责）中名字的冲突
- 命名空间的机制，对于声明于其中的名称都进行了本地化
 - 相同的名称可以在不同的上下文中使用，而不会引起名称的冲突。
 - C++中的标准库被定义到std命名空间中，减少了与用户自定义名称的冲突。
 - 在自己的程序中创建自己的命名空间，避免本地化的名称与外界冲突。

吓人的错误!

```
#include <iostream>
#include <math.h>
using namespace std;
class point
{
public:
    double x;
    double y;
};
double distance(point p1,point p2)
{
    double d=sqrt((p1.x-p2.x)*(p1.x-p2.x)+(p1.y-p2.y)*(p1.y-p2.y));
    return d;
}
int main()
{
    point p1= {3,5},p2= {6,9};
    cout<<distance(p1,p2);
}
```



```
main.cpp x stl_iterator_base_funcs.h x
106 * that @p __last must be reachable from @p
107 * n may be negative.
108 *
109 * For random access iterators, this uses t
110 * and are constant time. For other %itera
111 */
112 template<typename _InputIterator>
113 inline typename iterator_traits<_InputIter
114 distance(_InputIterator __first, _InputIte
115 {
116 // concept requirements -- taken care of
// ...

Logs & others
Code::Blocks x Search results x Cccc x Build log x
File Line Message
=== Build: Debug in example (compiler: GNU GCC)
In instantiation of 'struct std::iterator_trait
c:\program fil... 114 required by substitution of 'template<class _
D:\C\CP\codeBlo... 20 required from here
c:\program fil... 166 error: no type named 'iterator_category' in 'cl
c:\program fil... 167 error: no type named 'value_type' in 'class poi
c:\program fil... 168 error: no type named 'difference_type' in 'clas
c:\program fil... 169 error: no type named 'pointer' in 'class point'
c:\program fil... 170 error: no type named 'reference' in 'class poi
=== Build failed: 6 error(s), 2 warning(s) (0 m
```

THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：www.csdn.net

企业服务：<http://ems.csdn.net/>

人才服务：<http://job.csdn.net/>

CTO俱乐部：<http://cto.csdn.net/>

高校俱乐部：<http://student.csdn.net/>

程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>

项目外包：<http://www.csto.com/>

CSDN博客：<http://blog.csdn.net/>

CSDN论坛：<http://bbs.csdn.net/>

CSDN下载：<http://download.csdn.net/>