

C++语言基础

迂者 - 贺利坚

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

<http://edu.csdn.net>





本节主题：

命名空间的使用

自定义命名空间

- ❏ 由程序设计者自己定义命名空间
- ❏ 程序设计者把一些全局实体分别放在各个命名空间中，从而与其他全局实体分隔开来

```
namespace ns1 //指定命名空间ns1
{
    int a;
    double b;
}
```

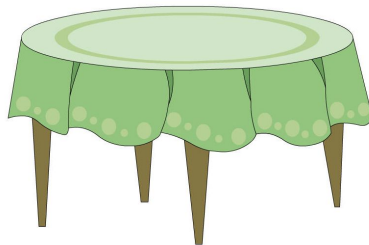
- ❏ 命名空间的作用是建立一些互相分隔的作用域，把一些全局实体分隔开来，以免产生名字冲突。

桌子归厨房，表格在办公室

```
//文件h1.h
namespace kichen
{
    class Table
    {
    public:
        Table(int l, Shape s, int p){.....}
    private:
        int legs;
        Shape surface;
        int price;
    };
    .....
}
```

```
//文件h2.h
namespace office
{
    class Table
    {
    public:
        Table(int r, int c, int b){.....}
    private:
        int rows;
        int cols;
        int border;
    };
    ....
}
```

```
//文件prog.cpp
#include "h1.h"
#include "h2.h"
int main( )
{
    office::Table t1(4,5,1);
    kichen::Table t2(4,"round",100);
    .....
}
```



声明一个命名空间的形式

```
namespace 命名空间名
{
    声明;
}
```

命名空间的声明部分可以包括以下类型:

- 变量(可以带有初始化)及常量;
- 函数(可以是定义或声明);
- 结构体;
- 类;
- 模板;
- 命名空间(在一个命名空间中又定义一个命名空间, 即嵌套的命名空间)。

//定义命名空间

```
namespace ns1
{
    const int RATE=0.08;
    double pay;
    double tax( )
    {
        return a*RATE;
    }
    namespace ns2
    {
        int age;
    }
}
```

//引用命名空间

```
cout<<ns1::RATE<<endl;
cout<<ns1::pay<<endl;
cout<<ns1::tax()<<endl;
cout<<ns1::ns2::age<<endl;
```

使用命名空间成员的方法

```
//定义命名空间
namespace ns1
{
    const int RATE=0.08;
    double pay;
    double tax( )
    {
        return a*RATE;
    }
    namespace ns2
    {
        int age;
    }
}
```

☞ 一般方法：引用命名空间成员时，用命名空间名对命名空间成员进行限定，以区别同名标识符

☞ 命名空间名::命名空间成员名

☞ 简化使用命名空间成员的机制

(1) 使用命名空间别名

namespace TV = Television; //TV等同Television

(2) 使用格式：using 命名空间成员名

using ns1::Student; //Student等同ns1::Student

(3) 使用：using namespace 命名空间名

```
std::cout<<"NS"<<std::endl;
```

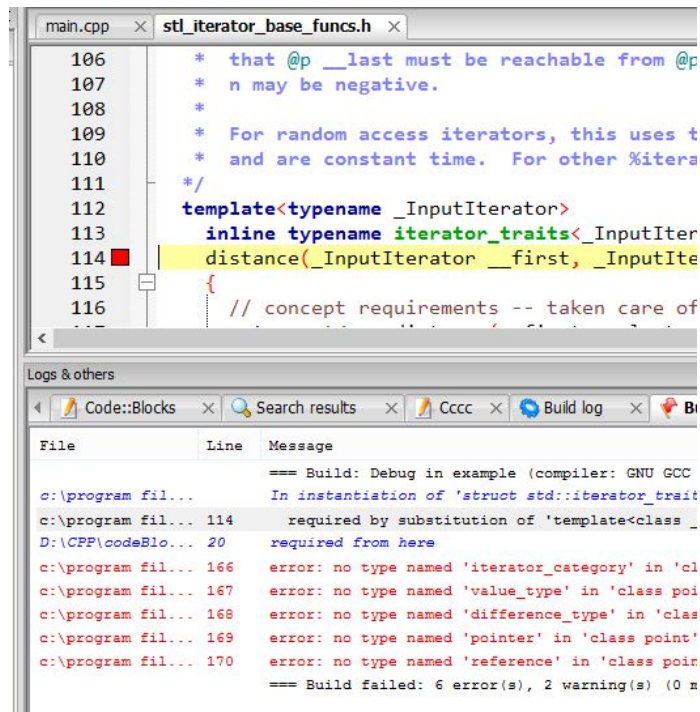
```
namespace s = std;
...
s::cout<<"NS"<<s::endl;
```

```
using std::cout;
...
cout<<"NS"<<std::endl;
```

```
using namespace std;
...
cout<<"NS"<<endl;
```

重名问题的解决

```
#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);
}
```



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/>

