

新标准C++程序设计

北京大学信息学院 郭 炜

GWPL@PKU.EDU.CN

智能指针shared_ptr

- shared_ptr模板和auto_ptr有类似之处，包括成员函数和auto_ptr也大多类似。它比auto_ptr的改进之处在于，auto_ptr对象独享对指针的托管权，而多个shared_ptr对象可以共同托管一个指针p，在最后一个托管p的shared_ptr对象消亡的时候，才会执行“delete p”。

智能指针shared_ptr

```
//program 20.6.1.1.cpp shared_ptr
#include <memory>
#include <iostream>
using namespace std;
class A
{
public:
    int n;
    A(int v):n(v){ }
    ~A() { cout << n << " destructor" << endl; }
};
int main()
{
    shared_ptr<A> sp1(new A(2));
    shared_ptr<A> sp2(sp1);
    cout << sp1->n << "," << sp2->n << endl;
}
```

输出结果:
2,2
2 destructor

无序容器(哈希表)

//program 20.6.2.cpp unordered_map示例

```
#include <iostream>
```

```
#include <string>
```

```
#include <unordered_map>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    unordered_map<string,int> turingWinner; //图灵奖获奖名单
```

```
        turingWinner.insert(make_pair("Dijkstra",1972));
```

```
        turingWinner.insert(make_pair("Scott",1976));
```

```
        turingWinner.insert(make_pair("Wilkes",1967));
```

```
        turingWinner.insert(make_pair("Hamming",1968));
```

```
turingWinner["Ritchie"] = 1983;
```

```
    string name;
```

```
    cin >> name; //输入姓名
```

```
        unordered_map<string,int>::iterator p =  
turingWinner.find(name);//据姓名查获奖时间  
        if( p != turingWinner.end())  
            cout << p->second;  
        else  
            cout << "Not Found" << endl;  
        return 0;  
    }
```

正则表达式

//program 20.6.3.1.cpp 正则表达式

```
#include <iostream>
#include <regex> //使用正则表达式须包含此文件
using namespace std;
int main()
{
    regex reg("b.?p.*k");
    cout << regex_match("bopggk",reg) <<endl; //输出 1 表示匹
    配成功
    cout << regex_match("boopgggk",reg) <<endl; //输出 0 表示
    匹配失败
    cout << regex_match("b pk",reg) <<endl; //输出 1 表示匹配
    成功
    regex reg2("\\d{3}([a-zA-Z]+).((\\d{2}|N/A)\\s\\1");
    string correct="123Hello N/A Hello";
    string incorrect="123Hello 12 hello";
```

```
cout << regex_match(correct,reg2) <<endl; //输出 1 表示匹
```

配成功

```
cout << regex_match(incorrect,reg2) << endl; //输出 0 表示匹
```

配失败

```
}
```