



揭开 C++ 新标准的神秘面纱

——C++11 简介

C++11 的新特性

- 改善内存管理——智能指针和最小垃圾回收
- 减少冗余代码——类型推导
- 改变思考方式——lambda 表达式
- 并行编程支持——内存模型、线程、原子操作
- 挖掘优化潜能——右值引用

.....

改善内存管理

- 显式内存管理及弊端

- 智能指针

- 垃圾回收

auto_ptr

unique_ptr

shared_ptr

weak_ptr

```
#include <memory>
#include <iostream>
using namespace std;
int main()
{
    shared_ptr<int> sp1(new int(22));
    auto sp2 = sp1;
    cout << *sp1 << endl;
    *sp2 = 1;
    cout << *sp1 << endl;
    return 0;
}
```



类型推导

- auto 关键字
- decltype 关键字

```
#include <string>
#include <iostream>
using namespace std;
int main()
{
    string str1 {"Hello,world!"};
    decltype(str1) str2;
    for(auto i = str1.begin(); i != str1.end() ; ++i)
        str2.push_back(*i);
    cout << str2 << endl;
    return 0;
```



lambda 函数

```
#include <vector>
#include <iostream>
using namespace std;

int main()
{
    vector<int> vec{1,2,3,7,5,9,0};
    auto Print = [&]()
    {
        for(auto& e: vec)
            cout << e << " ";
        cout << endl;
    };
    Print();
    vec.push_back(2);
    Print();
}
```



右值引用

```
string s;
```

```
string&& ss = s;
```

