

share_ptr实现

share_ptr是C++11提供了一种智能指针，例程1的sharePtr是share_ptr的模拟实现(实现尚不完整)，仔细阅读代码，体味其代码实现。

例程1

```
第1行 #include<iostream>
第2行 using namespace std;
第3行 template<typename T>
第4行 class sharePtr{
第5行 public:
第6行     struct Box{
第7行         unsigned int count = 0;
第8行         T* pData = nullptr;
第9行         void LocateMem(unsigned int N){
第10行             cout << "Here!" << endl;
第11行             this->pData = new T[N];
第12行             this->count++;
第13行         }
第14行     };
第15行 public:
第16行     sharePtr(){};
第17行     explicit sharePtr(unsigned int N) {
第18行         this->pAddr = new Box;
第19行         this->pAddr->LocateMem(N);
第20行     }
第21行     sharePtr(const sharePtr<T> & me){
第22行         this->pAddr = me.pAddr;
第23行         this->pAddr->count++;
第24行     }
第25行     ~sharePtr() {
第26行         unsigned int usedCount = this->use_count();
第27行         if (usedCount == 1){
第28行             delete[] this->pAddr->pData;
第29行             delete this->pAddr;
第30行             cout << "~sharePtr() delete" << endl;
第31行         }
第32行         else{
第33行             this->pAddr->count--;
第34行             cout << "~sharePtr() --" << endl;
第35行         }
第36行     }
第37行     sharePtr<T>& operator=(const sharePtr<T> &me){
第38行         unsigned int usedCount = this->use_count();
```

```

第39行         switch (usedCount) {
第40行             case 0://没有分配
第41行                 this->pAddr = me.pAddr;
第42行                 this->pAddr->count++;
第43行                 break;
第44行             case 1://仅分配一次
第45行                 delete[] this->pAddr->pData;
第46行                 delete this->pAddr;
第47行                 this->pAddr = me.pAddr;
第48行                 this->pAddr->count++;
第49行                 break;
第50行             default://超过1次
第51行                 this->pAddr->count--;
第52行                 this->pAddr = me.pAddr;
第53行                 this->pAddr->count++;
第54行                 break;
第55行         }
第56行         return *this;
第57行     }
第58行     T* get() const {
第59行         return this->pAddr->pData;
第60行     }
第61行     unsigned int use_count() {
第62行         return this->pAddr->count;
第63行     }
第64行 private:
第65行     Box * pAddr=nullptr;
第66行 };
第67行 template<typename T>
第68行 void Test(const sharePtr<T> & shareP) {
第69行     sharePtr<T> tmpSharePtr = shareP;
第70行     cout << tmpSharePtr.use_count() << endl;
第71行 }
第72行 int main() {
第73行     sharePtr<int> mySharePtr(10), tmpSharePtr(20);
第74行     for (int i = 0; i < 10; ++i)*(mySharePtr.get() + i) = 10 + i;
第75行     for (int i = 0; i < 20; ++i)*(tmpSharePtr.get() + i) = 20 + i;
第76行     sharePtr<int> hisSharePtr = mySharePtr;
第77行     cout << mySharePtr.use_count() << endl;
第78行     cout << hisSharePtr.use_count() << endl;
第79行     Test(hisSharePtr);
第80行     cout << hisSharePtr.use_count() << endl;
第81行     for (int i = 0; i < 10; ++i)cout << *(mySharePtr.get() + i) << " "; cout << endl;
第82行     for (int i = 0; i < 10; ++i)cout << *(hisSharePtr.get() + i) << " "; cout << endl;

```

第83行	for (int i = 0; i < 20; ++i)cout << *(tmpSharePtr.get() + i) << " "; cout << endl;
第84行	
第85行	hisSharePtr = tmpSharePtr;
第86行	for (int i = 0; i < 10; ++i)cout << *(mySharePtr.get() + i) << " "; cout << endl;
第87行	for (int i = 0; i < 20; ++i)cout << *(hisSharePtr.get() + i) << " "; cout << endl;
第88行	for (int i = 0; i < 20; ++i)cout << *(tmpSharePtr.get() + i) << " "; cout << endl;
第89行	
第90行	system("pause");
第91行	return 0;
第92行	}