# uniqe\_ptr用法详解

unique\_ptr是C++标准库自C++11起开始提供的类型。它是一种在异常发生时可帮助避免资源泄露的智能指针。一般而言，这个智能指针实现了独占式拥有概念，意味着它可确保一个对象和其相应资源同一时间只被一个指针拥有。一旦拥有者被销毁或变成空，或开始拥有另一个对象，先前拥有的那个对象就会被销毁，其任何相应资源也会被释放。

所以可以看如下程序

#include <memory>

int main()

{

unique\_ptr<int> ptr (new int(0));

return 0;

}

# **智能指针简单实现**

基于引用计数的智能指针可以简单实现如下（详细解释见程序中注释）：

#include <iostream>

using namespace std;

template <class T>

class SmartPtr

{

public :

SmartPtr(T \*p);

~SmartPtr();

SmartPtr(const SmartPtr<T> &rhs);

SmartPtr<T>& operator=(const SmartPtr<T> &rhs); // 浅拷贝

private:

T \*ptr;

// 将use\_count声明成指针是为了方便对其的递增或递减操作

int \*use\_count;

};

template<class T>

SmartPtr<T>::SmartPtr(T \*p):ptr(p)

{

try

{

use\_count = new int(1);

}

catch(...)

{

delete ptr;

ptr = nullptr;

use\_count = nullptr;

cout <<”allocate memory for use\_count fails”<<endl;

exit(1);

}

cout <<”Constructor is called!”<<endl;

https://www.codeprj.com/blog/46e2b31.html(参考博客）