



c++ CRTP（奇异的递归模板模式)介绍



creekee
与苟且安然相处，方能寻得诗和远方。

+ 关注他

38 人赞同了该文章

什么是CRTP?

简单来说，CRTP有两大特性：

- 1. 继承自模板类。
- 2. 派生类将自身作为参数传给模板类。

翻译成代码，长这样：

```
// 我们先定义一个模板类作为基类
template <typename T>
class Base
{
    ...
};
// 定义一个派生类，这个类继承以自身作为参数的基类
class Derived : public Base<Derived>
{
    ...
};
```

问题来了，为什么要这样做呢？

这样做的目的其实很明确，从基类对象的角度来看，派生类对象其实就是本身，这样的话只需要用一个static_cast就可以把基类转化成派生类，从而实现基类对象对派生对象的访问。

我们还是来看看代码：

```
template <typename T>
class Base
{
public:
    void doSomething()
    {
        T& derived = static_cast<T&>(*this);
    }
};

class Derived : public Base<Derived>
{
public:
    void doSomething()
    {
        std::cout << " Derived class " << std::endl;
    }
};
```

很多小伙伴可能发现了，这里将基类转换成派生类用的是static_cast静态绑定，而普通基类转派生类用的是dynamic_cast动态绑定。动态绑定的目的是为了确​​保你所转化的派生类是正确的，而对于CRTP来说，基类是继承于模板类的参数，也就是派生类本身。这也正是CRTP这种设计的目的。

CRTP的优点是什么呢？

多态是个很好的特性，但是动态绑定比较慢，因为要查虚函数表。而使用 CRTP，完全消除了动态绑定，降低了继承带来的虚函数表查询开销。

好了，这篇只是对CRTP的一个介绍，下次写写我们如何有效的使用CRTP。

编辑于 2020-12-31 07:48

C++ 应用 C / C++



写下你的评论...

4 条评论

默认 时间

Micro

绝赞好文！
2021-04-22

回复 赞

Wind

llvm 里 visitor 就是这么设计的
03-03

回复 赞

7ED55

”从基类对象的角度来看，派生类对象其实就是本身“。
这个”本身“是指基类对象还是派生类对象？
2020-11-26

回复 赞

冯Jungle

我的理解：在内存布局上，派生类的subobject就是基类本身的内存布局，所以，以基类的角度看派生类的内存布局，跟基类本身的内存布局是一样的，因为派生类的subobject的内存布局其实就是基类，所以通过static_cast一定可以将基类安全地转换为派生类。以派生类的角度看派生类的内存布局，其subobject就是基类，然后才是派生类自身定义的属性。所以这里的“本身”就是基类视角下的派生类内存布局。
2021-08-09

回复 赞

文章被以下专栏收录



知言
致知言，以养浩然之气。



c++：从入门到放弃