

c++中typename关键字总结

声明一个类型

typename的第一个作用在于声明一个类型。为什么类型还需要声明呢？因为编译器并不是总会知道那个名称是个类型。下面的代码会编译错：

```
1 template<typename C>
2 void print2nd(const C& container){
3     if(container.size() >= 2){
4         C::const_iterator it(container.begin());
5         ++it;
6         int value = *it;
7         cout<<value;
8     }
9 }
```

发生编译错误是因为编译器不知道C::const_iterator是个类型。万一它是个变量呢？C::const_iterator的解析有着逻辑上的矛盾：直到确定了C是什么东西，编译器才会知道C::const_iterator是不是一个类型；然而当模板被解析时，C还是不确定的。这时我们声明它为一个类型才能通过编译：

```
typename C::const_iterator it(container.begin());
```

嵌套从属名称

事实上类型C::const_iterator依赖于模板参数C，模板中依赖于模板参数的名称称为从属名称(dependent name)，当一个从属名称嵌套在一个类里面时，称为嵌套从属名称(nested dependent name)。其实C::const_iterator还是一个嵌套从属类型名称(nested dependent type name)。

嵌套从属名称是需要用typename声明的，其他的名称是不可以用typename声明的。比如下面是一个合法的声明：

```
1 template<typename C>
2 void f(const C& container, typename C::iterator iter);
```

如果你把const C&也声明了typename也是要编译错的哦：

```
1 template<typename C>
2 void f(typename const C& container, typename C::iterator iter);
```

错误输出：

```
error: expected a qualified name after 'typename'
```

一个例外

模板中的嵌套从属名称是需要typename声明的，然而有一个例外情况：在派生子类的基类列表中，以及构造函数基类初始化列表中，不允许typename声明。例如Derived继承自Base::D::Nested：

```
1 template<typename T>
2 class Derived: public Base<T>::Nested{ // 继承基类列表中不允许声明 'typename'
3 public:
4     explicit Derived(int x): Base<T>::Nested(x){ // 基类初始化列表不允许声明 'typename'
5         typename Base<T>::Nested tmp; // 这里是要声明的 6 |    }
7 };
```

traits

C++提供了一系列的traits模板，用来提供类型信息。比如：

```
1 template<typename IterT>
2 void workWithIterator(IterT it){
3     typename std::iterator_traits<IterT>::value_type tmp(*it);
4 }
```

其实上述模板方法也可以不同raits来实现，比如：

```
1 template<typename container>
2 void workWithIterator(typename container::iterator it){
3     typename container::value_type tmp(*it);
4 }
```

但traits提供了更加一致的使用方式以及容器实现的灵活性，模板代码也简洁了不少。尽管如此，程序员还是懒惰的。我们倾向于用typedef来给这些嵌套从属名称起一些别名：

```
1 template<typename IterT>
2 void workWithIterator(IterT it){
3     typedef typename std::iterator_traits<IterT>::value_type value_type;
4     value_type tmp(*it);
5 }
```

了解 C++ 之 typename 他知此事要躬行 2782
typename与class都可以用作模板形参定义的关键字，两者无异~~~ 可是，typename的用途并非仅限于此，如下面...

参与评论

请发表有价值的评论，博客评论不欢迎灌水，良好的社区氛围需大家一起维护。

👍

👎

评论

- C++ typename关键字_weixin_340995266的博客

template <typename T> 这里typename和class关键字的功能一样,是为了取代class关键字而生,原因是class用来代...

218
- C++中的typename关键字_u010585135的专栏_c++ typename

C++中的typename关键字 在C++中typename除了用于在定义模板函数和模板类中来替代关键字class,例如Templat...

121
- C++中typename关键字的使用方法和注意事项_热门推荐

1. 什么地方使用? 用在模板定义里, 标明其后的模板参数是类或参数。例如 template T foo(const T& t, const Y& y)...

3万+
- Error(63, 24) 错误: 枚举 switch case 标签必须为枚举常量的非限定名称

错误描述: Error(63, 24) 错误: 枚举 switch case 标签必须为枚举常量的非限定名称。 项目当中用到了枚举, 所...

4112
- 详解C++的模板中typename关键字的用法_c++typename,typename关键...

C++模板的使用可以提升代码的通用性、可移植性, 方便通用库的开发。与模板有关的关键字有两个: template 定...

211
- C++ typename关键字的作用_梦想博客

C++ typename关键字的作用 参考http://blog.163.com/cp7618@yeah/blog/static/702347772011122111034365/...

124
- C++ typename用法

先看这篇文章。https://blog.csdn.net/vanturman/article/details/80269081 https://blog.csdn.net/pizzq/article...

1288
- Item 42: typename的两种用法

Item 42: Understand the two meanings of 'typename' 时至今日还有人在论坛里问模板参数前的typename和class...

414
- C/C++编程: 关键字typename_最新发布

typename 在C++标准化工程中, 引入关键字typename是为了说明: 模板内部的标识符可以是一个类型。比如下面...

76
- C++typename的由来和用法

前言 在C++模板函数的使用过程中, 我们经常可以看到一个typename的使用, 例如这样的操作但是除此之外, 我...

1万+
- C++ (八) --typename用法

1. typename来源 1.1 原因1: 从表面上看, 下面模板的参数只支持用户自定义类型, 但实际上语言内置类型或者...

2132
- 连接oracle数据库经常遇到错误汇总

ORA-00001: 违反唯一约束条件 () ORA-00017: 请求会话以设置跟踪事件 ORA-00018: 超出最大会话数 ORA-0001...

8627
- c++---typename关键字

介绍typename之前, 应先了解几个概念: 1. 限定名与非限定名 限定名, 顾名思义, 是限定了命名空间的名称。#...

71
- [转载] 知无涯之C++ typename的起源与用法

原文http://feih.me/blog/2014/the-origin-and-usage-of-typename/ 如果想深入了解typename关键字的含义, 推荐...

600
- C++中typename用法

typename用法: 1、在c++模板中, 用于类型参数化的关键字 2、内联、依赖类型名 1. 在C++中typename一般用...

1617
- typedef和typename的区别

typename标示一个类型名, 而非定义一个类型, 以下声明了一个SeqIterator类型的变量it, 其中Seq是一个模板...

4658
- C++中typename的用法

typename主要作用是告诉compiler

2866

