

始终

不忘初心



# C++ 中的 typename 及 class 关键字的区别

📅 2018 年 03 月 16 日 | 📅 2020 年 02 月 04 日 | 📁 [Algorithm and Computer Science](#)

| 👁 2965

📄 1.9k | ⌚ 3 分钟

其实这是一个老掉牙的问题，网络上已有很多辨析。但看下来都有遗漏其中一个区别，所以有了这篇文章。

## 共同点

如果两个事物没有任何共同点，那么也就没必要讨论他们的区别——因为他们所有的特性都是区别。所以这里我们首先来看一下 `typename` 和 `class` 关键字的共同点。

在定义类模板或者函数模板时，`typename` 和 `class` 关键字都可以用于指定模板参数中的类型。也就是说，以下两种用法是完全等价的。这在大多数文章中都有提到。

```
1  template<typename T> /* class or function declaration */;  
2  template<class T>    /* class or function declaration */;
```

关于这个问题，Stan Lippman 曾在其[博客](#)中表示，最早 Stroustrup 使用 `class` 来声明模板参数列表中的类型是为了避免增加不必要的关键字；后来委员会认为这样混用可能造成概念上的混淆才加上了 `typename` 关键字。

## typename 独有的功能

除此之外，`typename` 还有其独有的功能。



由于 C++ 允许在类内定义类型别名，且其使用方法与通过类型名访问类成员的方法相同。故而，在类定义不可知的时候，编译器无法知晓类似 `Type::foo` 的写法具体指的是一个类型还是类内成员。

例如在以下代码中，类模板 `Bar` 的原意是使用类 `Foo` 实例化，而后引用其中的 `bar_type` 定义名为 `bar` 的类内成员。然而，就 `T::bar_type` 而言，编译器在编译期无法确定它究竟是不是一个类型。此时就需要 `typename` 关键字来辅助编译器的判断。

```
1  class Foo {
2      public:
3          typedef int bar_type;
4  };
5
6  template<typename T>
7  class Bar {
8      /* typename */
9      T::bar_type bar;
10 };
```

这在大多数文章中也都有提到。

值得一提的是，在编译期能够判断的情形，例如在上例中直接使用 `Foo::bar_type` 时，使用冗余的 `typename` 会报错。

只有少数文章提到了这一点。

## class 独有的功能

`class` 关键字最众所周知的功能是声明或定义一个类。这当然是其相对 `typename` 的一个独有功能。为了完整性，这里也列出。



除此之外，在模板的使用中，`class` 关键字也有其特有的功能。而这是绝大多数文章不会提及的。

C++ 的标准模板库中有名为 `std::stack` 的容器适配器，它能适配许多容器作为底层，实现栈的功能。其声明为

```
1 template <typename T, typename Containter = std::deque<T> >
2 class stack;
```

因此，在使用中，我们可以使用 `std::stack<int>` 来声明一个以 `std::deque<int>` 保存整型变量的栈；也可以使用 `std::stack<int, std::vector<int> >` 来声明一个以 `std::vector<int>` 保存整型变量的栈。

现在的问题是，是否有可能以类似 `Stack<int, std::vector>` 的形式，来达到同样的目的？

为此，我们需要有类似这样的声明

```
1 template <typename T,
2     template <typename E, typename = std::allocator<E> > class
3 class Stack;
```

由于 `Container` 必须是一个容器类模板，所以，如果不适用具体的模板参数实例化，就必须将其声明为一个类模板。故此，`Container` 之前需要保留标准库中容器类模板的模板参数。注意此处使用了标准库提供的内存分配器。

此处 `class` 特有的功能，体现在 `class Container` 之处。此处虽然是在声明 `Stack` 这个类模板，但是此处的 `class` 不能替换为 `typename`；否则编译器会报错。



不过，在 C++17 标准中，此处也允许使用 `typename` 了。参见[此处](#)。



俗话说，投资效率是最好的投资。如果您感觉我的文章质量不错，读后收获很大，预计能为您提高 10% 的工作效率，不妨小额捐助我一下，让我有动力继续写出更多好文章。

打赏

本文作者：Liam Huang

本文链接：<https://liam.page/2018/03/16/keywords-typename-and-class-in-Cxx/>

版权声明：本博客所有文章除特别声明外，均采用 [CC BY-NC-SA](#) 许可协议。转载请注明出处！

[# Cpp](#) [# Keywords](#)

◀ 三门问题的数学原理及模拟实验

乐理漫谈：音高、音程、泛音 ▶

0 条评论

未登录用户 ▼

