**关于未实例化模板类中引用其他类型在不同编译器下的测试**

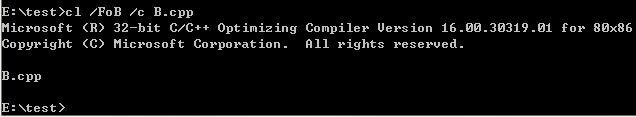
下面分别在 VS2010，gcc 3.3.4，gcc 4.1.2中对下面代码进行编译。

测试代码：

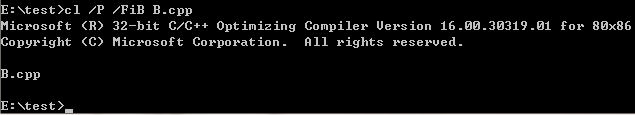


## 在 VS2010中测试

B.cpp可以成功编译生成B.obj目标文件。



查看B.cpp的预编译文件：





## 在gcc 3.4.4 中测试

生成目标文件失败，在模板类B前添加class A的声明后可以成功编译。

gcc -c B.cpp -o B.o



查看预编译文件：

gcc -E B.cpp –o B.i



## 在gcc 4.1.2中测试

生成目标文件失败，在模板类B前添加class A的声明后依然报错。

解决方法：模板类B在使用A时，必须事先知道A的完整定义。

gcc -c B.cpp -o B.o



查看预编译文件：

gcc -E B.cpp –o B.i



解决方法：注意B.h中#include “A.h”的位置。

此问题属于模板的语义检查范畴。按一般标准，模板类不具现，是不需要关心语义的，因为根本不会用到，但这里明显gcc做了更深入的检查。



**结论：（对于未实例化模板类B）**

* VS2010 模板类B中不需要事先知道类型A的定义。
* gcc 3.3.4 模版类B中必须事先知道类型A的声明。
* gcc 4.1.2 模板类B中必须事先知道类型A的定义。

## 补充：针对 circular inclusion 的解决方法

**情形1：两个普通类A和B相互依赖**

解决方法：将头文件中相互依赖的实现部分置于各自的CPP文件中。





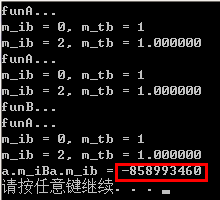


**情形2：其中一个为模板类的两个类相互依赖（对于需要实例化的情形）**

解决方法：将模板类B.h中的#include “A.h”置于模板类B的定义之后。







**参考：**

一个关于C++模板编译的问题

<http://topic.csdn.net/u/20090116/15/222aa93f-388e-4516-a527-11109daf4a75.html>