

```

////三：模板显式实例化，模板声明
//////为了防止在多个.cpp文件中都实例化相同的类模板，所以c++11提出了一个解决方法，我们称为“显式实例化”；
// //通过“显式实例化”来比米娜这种生成多个相同类模板实例的开销。
// <int> tmpobj(6, 7);

//模板的实例化定义,只有1个，实例化声明可以有多个。
//总结：
// (1) vs2017不太推荐这个特色 I
// (2)
// template A<float>:
// extern template A<float>;

```

“显式实例化”手段中的“实例化定义”，这种实例化定义只需要在一个.cpp文件写就可以。

```

template A<float>; //编译器遇到这段代码就直接实例化出来一个 A<float>

```

“显式实例化”手段中的“实例化声明”

```

extern template A<float>; //其他所有.cpp文件都这么写；
// extern作用：不会在本文中生成一个extern后边所表示的模板的实例化版本代码。
// extern目的：告诉编译器，在其他的源文件(.cpp文件)中已经有了一个该模板的实例化版本了。

```

不推荐是因为，这种实例化，会直接把所有成员都实例化，而不是在调用的时候实例化。同时，这个extern有时候不能避免在当前文件再次生产相关代码，而不是仅仅去其他.cpp/h中寻找定义，所以有可能不仅没减轻编译负担，而是加重了