|  |  |
| --- | --- |
| 1. 控制实例化 2. 实例化定义会实例化所有成员 | 显示实例化  Extern template declaration  Template declaration  Extern template class Blob<string>;//声明  Template int compare(cons tint&,cons tint&)//定义  Extern template class Blob<string>;  Blob<string> sa1; |
| 1. 控制实例化 2. 相同的实例可能出现在多个对象文件中。当两个或多个独立编译的源文件使用了相同的模板，并提供了相同的模板参数时，每个文件中就都会有该模板的一个实例。   当模板被使用时会进行实例化。   1. 在多个文件中实例化相同模板的额外开销可能非常严重。可以通过显式实例化来避免这种开销 2. template class Blob<String>;代表实例化类模板的所有成员！ 3. 当编译器遇到extern模板声明时，不会再本文件中实例化代码。表示在其它位置有该实例的一个非extern声明（定义）。对于一个给定的实例化版本，可能有多个extern声明，但必须只有一个定义 4. 由于使用模板时自动对其实例化，extern声明必须出现在任何使用此实例化版本的代码之前。 5. 当在一个文件上写了一个实例化版本的extern声明，再下面写这个版本的定义则不在本文件中进行实例化 6. 需要将两个分离的声明与定义文件链接在一起。 7. 就是不知道若将一个实例化版本声明为extern的话再同一个文件使用这个实例化版本不知道会怎么样。 8. 实例化定义会实例化所有成员 9. 有点不懂，好像是当一个类模板的实例化定义会实例化该模板的所有成员，包括内联的成员函数。即使没有用到成员也会被实例化，用来显式实例化一个类模板的类型，必须用于模板的所有成员。类型也会被实例化调用构造函数 10. 这样显示实例化就只有一个   总：  就是当一个文件Blob<string>，另一个文件Blob<string>,则这两个文件都有string模板的Blob实例，都会实例化类模板的所有成员，则可在其中一个文件中  extern template class Blob<string>;声明代表其它类中有定义，不在被类中定义并实例，并可以使用Blob<string>  // 在另一个文件内些写下这个，代表在这个类实例化string版本Blob  template class Blob<string>  实例化定义：  Extern template declaration  Template declaration; | |