|  |  |
| --- | --- |
| 1. 模板参数与作用域 2. 模板声明 3. 使用类的类型成员 4. 默认模板实参 5. 模板默认实参与类模板 | <typename Foo>  <typename T,typename T>  T::size\_type \*p;  Typename T::value\_type  Typename F = less<T>  F f = F()  Template <class T = int> class nu{} |
| 1. 模板参数与作用域 2. 模板参数的名字不止为T 3. 模板参数名的可用范围是在其声明之后，至模板声明或定义结束之前 4. 模板参数名会隐藏外层作用域声明相同的名字 5. 模板内不能重用模板参数名，但是可以在c+14 6. 模板声明 7. 声明中的模板参数的名字不必与定义中相同，有点编译器检测不到定义 8. 但是一个给定模板的每个声明和定义必须有相同数量和种类，即（类型或非类型）参数   Note：特定文件所需要的所有模板的声明通常一起放在在文件开始位置，出现于任何使用这些模板之前P617解释   1. 使用类的类型成员 2. T::size\_type \*p,编译器不知道size\_type是static数据成员，还是类型 3. C+假定用::访问的是类型，用模板类型参数的类型成员时，必须显示告诉编译器这是一个类型，用typename，不能用class，若没有typename则被看做是值了。？？ 4. 但是若带有()，是函数，不用typename。主要是mem这种成员，需要显示Typename为类型，不然为值了 5. 默认模板实参 6. 模板参数提供了默认实参，对应的函数参数也可以提供默认实参。 7. 当传给默认实参的实参时会重新推断类型。 8. 只有右边的参数都有模板实参才可以有默认实参。 9. 模板默认实参与类模板 10. 若给类模板其所有模板参数都提供了默认实参，且希望使用这些默认实参，必须在模板名之后跟一个空尖括号对。表示使用默认类型。 | |