|  |  |
| --- | --- |
| 1. 可变参数模板 2. Sizeof..运算符 3. 编写可变参数函数模板 | 可变参数模板  参数包  Args模板参数包  rest函数参数包  Template <typename T, typename…Args>  Void foo(const T&t,const Args& …rest);  Foo(I,s,42,d);  Foo(cons tint&,const string&,cons tint&,const double&)  Sizeof…运算符  Template<typename T>  Ostream &print(ostream &os, const T &t)  {  Return os<<t;  }  Template <typename T,typename.. Args>  Ostream &print(ostream &os,const T &t,const Args&…rest){  Os<<t<<”,”  Return print(os,rest…);  } |
| 1. 可变参数模板 2. 一个可变参数模板就是一个接受可变数目参数的模板函数或模板类。可变数目的参数被称为参数包。两种：模板参数包，函数参数包 3. 用一个省略号来指出一个模板参数或函数参数表示一个包。   模板用class…或typename…类型列表  用类型名后加一个省略号表示非类型参数的列表  函数列表中若一个参数的类型是一个模板参数包，则此参数也是一个函数参数包   1. 编译器还会推断包中参数的数目   Void foo(const string&,cons tint&,const char[3]&)都是const的引用   1. Sizeof..运算符 2. Sizeof…(Args) sizeof…(args) 3. 编写可变参数函数模板 4. 既不知道想要处理的实参的数目也不知道它们的类型，可变参数函数是很有用的 5. 用递归来打印每个函数参数包 6. 对于最后一次递归调用，两个print版本都是可行的。但是非可变参数模板比可变参数模板更特例化。所以选非可变参数版本 | |