|  |  |
| --- | --- |
| 1. 包扩展 2. 理解包扩展 | 扩展  模式  Template <typename… Args>  Ostream &errorMsg(ostream &os, const Args&…rest)  {  Return print(os, debug\_rep(rest)…);  } |
| 1. 包扩展 2. 对一个包扩展。需要提供用于每个扩展元素的模式。扩展一个包将它分解成构成的元素。通过在模式右边放一个省略号…触发扩展操作 3. 每个元素应用模式，获得扩展后的列表 4. 模式：定义了扩展后参数包中每个元素的形式 5. 模式应该就是应该怎么样扩展，以函数形参的类型扩展（在函数形参中，扩展模板类型参数），可以自定义模式(在函数体中,扩展函数参数)， 6. Print(count,I,s,42)   Print(Ostream&, cons tint&,const string&,cons tint&)  Print(os,s,42);   1. 理解包扩展 2. 除了像上面将函数参数包被扩展为其构成元素 3. 可以将模式传入一个调用对象，则每个元素都会调用这个对象 4. Return print(os,debug\_rep(rest)…)   Print(cerr,debug\_rep(fcnName),debug\_rep(code.num()),…   1. 写法错误的：debug\_rep(rest…) 2. Note:扩展中的模式会独立地应用于包中的每个元素   // 扩展是生成以逗号分开的列表，不是执行语句！ | |