

C++11可变参数模板新特性

可变参数模板(variadic template)使得编程者能够创建这样的模板函数和模板类，即可接受可变数量的参数。

可变参数模板和普通模板的语义是一样的，只是写法上稍有区别，声明可变参数模板时需要在typename或class后面带上省略号"..."

```
template<class ...Args>
void ShowList(Args... args)
{}
```

特别注意，语法并不支持使用args[i]的方式来获取参数包中的参数。因此要获取参数包中的各个参数，只能通过展开参数包的方式来获取，一般我们会通过递归或逗号表达式来展开参数包。

递归法

```
//递归终止函数
template<class T>
void ShowListArg(const T& t)
{
    cout << t << endl;
}
//展开函数
template<class T, class ...Args>
void ShowListArg(T value, Args... args)
{
    cout << value << " "; //打印传入的若干参数中的第一个参数
    ShowListArg(args...);   //将剩下参数继续向下传
}
//供外部调用的函数
template<class ...Args>
void ShowList(Args... args)
{
    ShowListArg(args...);
}
```

逗号表达式

```
//支持无参调用
void ShowList()
{
    cout << endl;
}
//处理函数
template<class T>
```

```
void PrintArg(const T& t)
{
    cout << t << " ";
}
//展开函数
template<class ...Args>
void ShowList(Args... args)
{
    int arr[] = { (PrintArg(args), 0)... }; //列表初始化+逗号表达式
    cout << endl;
}
```