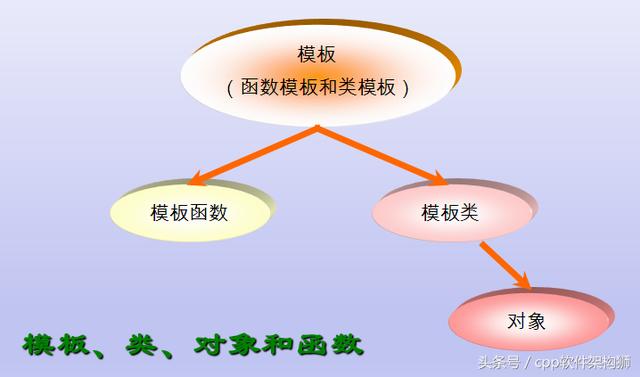
**C++进阶篇-函数模板**

cpp软件架构狮 2018-07-28 19:53:02

**从本期开始C++进阶篇的讲解，C++基础知识咱们就不在重复了，请自行学习！**

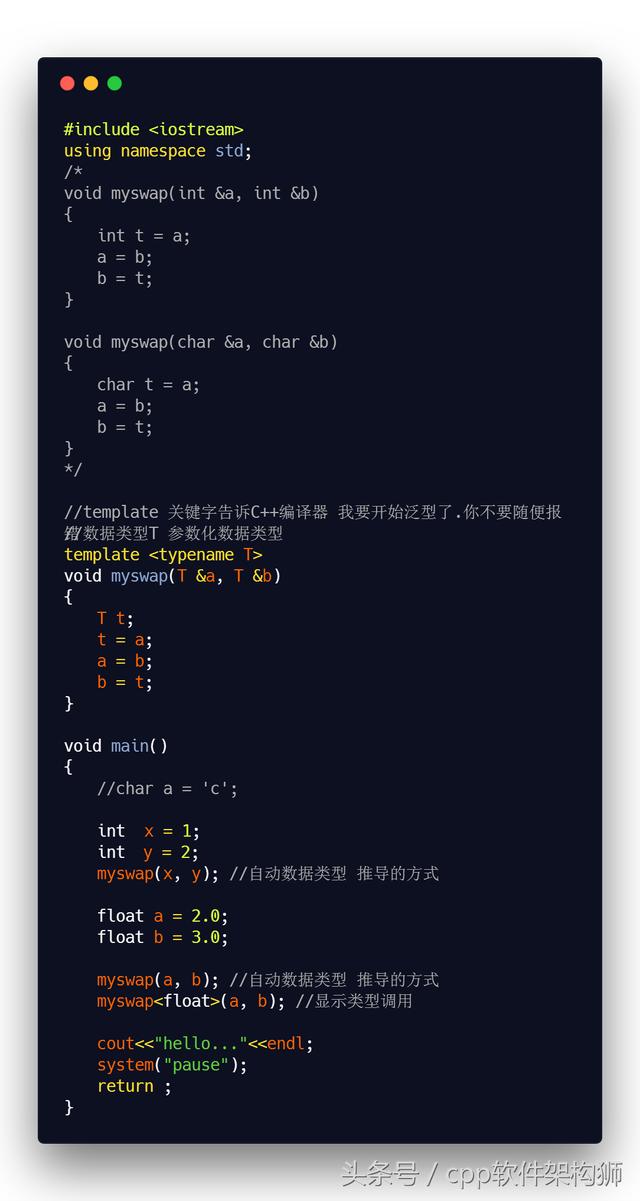
C++提供了函数模板(function template)。所谓函数模板，实际上是建立一个通用函数，其函数类型和形参类型不具体指定，用一个虚拟的类型来代表。这个通用函数就称为函数模板。凡是函数体相同的函数都可以用这个模板来代替，不必定义多个函数，只需在模板中定义一次即可。在调用函数时系统会根据实参的类型来取代模板中的虚拟类型，从而实现了不同函数的功能。



**C++提供两种模板机制：函数模板、类模板**

首先提个问题，为什么要有函数模板？

我们看个案例，**需求：写n个函数，交换char类型、int类型、double类型变量的值。**



如果按照原始方法，需要三个同样功能的函数，这样的代码对程序员简直就是噩梦，为了解决这一痛点，函数模板就诞生了。

**函数模板语法**



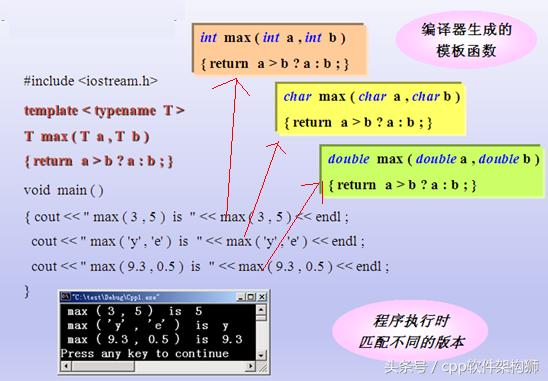
**函数模板调用**

myswap<float>(a, b); //显示类型调用

myswap(a, b); //自动数据类型推导

**函数模板和模板函数**

看下图基本就能看懂他们的关系了。



**函数模板做函数参数**

老规矩，看代码是学习最快的方式，我不说话。



**函数模板遇上函数重载**

**函数模板和普通函数区别结论：**

* 函数模板不允许自动类型转化
* 普通函数能够进行自动类型转换

**函数模板和普通函数在一起，调用规则：**

* 函数模板可以像普通函数一样被重载
* C++编译器优先考虑普通函数
* 如果函数模板可以产生一个更好的匹配，那么选择模板
* 可以通过空模板实参列表的语法限定编译器只通过模板匹配