

C 的函数和模板函数 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/459/2021_2022_C___E7_9A_84_E5_87_BD_E6_c97_459860.htm 在c 没有模板（template）机制的时候，我们使用的就是普通的函数，我们要实现一个加法函数，他能够对两个参数求和，并根据参数的类型返回具有适当类型的值，就必须手工书写所有的代码：
short sum(short a,short b) {return a b.} int sum(int a,int b){return a b.}
float sum(float a,float b){ return a b.} 非常麻烦，可以用c 的模板函数来表达“通用型的函数”
template T sum(T a,T b) { return a b. } 保存为sumtest.h 现在，c 编译器可以根据我们调用sum函数的参数类型“现场”生成一个适当的函数，然后调用它。例如：
#include #include “sumtest.h” using namespace std. int main(void) { float fa=1,fb=3,fs. fs=sum(fa,fb). cout} [root@localhost html]# g -o sumtest sumtest.cpp
[root@localhost html]# ./sumtest sum(float,float) 1 and 3=4 看上面的代码，c 编译器为我们生成一个“float版本”的sum函数并调用它。如果我们给出的参数类型不一样，则编译器会报错。例如
#include #include "sumtest.h" using namespace std. int main(void) { float fa=1,fs. int lb=3. fs=sum(fa,lb). cout return 0. }
[root@localhost html]# g -o sumtest sumtest.cpp sumtest.cpp: In function `int main ()' : sumtest.cpp:10: no matching function for call to `sum (float amp.)' 由于函数模板并不支持两个不同类型的参数求和，所以C 编译器会报告无法生成真正的函数，从而让程序员有机会知道调用参数除了问题。如果不是用模板函数而用普通函数，即使参数类型不完全一致也可能会通过

编译。例如 `#include using namespace std. float sum(float a,float b) { return a b. } int main(void) { float fa=1,fs. int lb=3. fs=sum(fa,lb). cout return 0. }` [root@localhost html]# `g -o sumtest sumtest.cpp` [root@localhost html]# `./sumtest1 sum(float,Int) 1 and 3=4` 因为c中，int类型可以自动转换成float类型，于是这种情况下不会报错。函数模板不是真正的函数，它只是c编译器生成具体函数的一个模子。所以不能把函数模板的声明和定义分开放在不同的文件中，而普通的函数可以这样做。C函数还有一个问题就是和c的函数的区别。最近在www.chinaunix.net C/C论坛上出现了几个关于c语言写的代码，c编译器可以通过编译，采用c编译器就不行了。就是这个问题。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com