

C语言中的面向对象(1) - 类模拟和多态,继承 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/135/2021_2022_C_E8_AF_AD_E8_A8_80_E4_B8_AD_c97_135026.htm 在面向对象的语言里面，出现了类的概念。这是编程思想的一种进化。所谓类：是对特定数据的特定操作的集合体。所以说类包含了两个范畴：数据和操作。而C语言中的struct仅仅是数据的集合。

(liyuming1978@163.com) 1 . 实例：下面先从小例子看起

```
#ifndef C_Class #define C_Class struct #endif C_Class A { C_Class A *A_this. void (*Foo)(C_Class A *A_this). int a. int b. }. C_Class B{ //B继承了A C_Class B *B_this. //顺序很重要 void (*Foo)(C_Class B *Bthis). //虚函数 int a. int b. int c. }. void B_F2(C_Class B *Bthis) { printf("It is B_Fun\n"). } void A_Foo(C_Class A *Athis) { printf("It is A.a=%d\n",Athis->a). //或者这里 // exit(1). // printf("纯虚 不允许执行\n"). //或者这里 } void B_Foo(C_Class B *Bthis) { printf("It is B.c=%d\n",Bthis->c). } void A_Creat(struct A* p) { p->Foo=A_Foo. p->a=1. p->b=2. p->A_this=p. } void B_Creat(struct B* p) { p->Foo=B_Foo. p->a=11. p->b=12. p->c=13. p->B_this=p. } int main(int argc, char* argv[]) { C_Class A *ma,a. C_Class B *mb,b. A_Creat(amp.b). mb=amp.a. ma=(C_Class A*)mb. //引入多态指针 printf("%d\n",ma->a). //可惜的就是 函数变量没有private ma->Foo(ma). //多态 a.Foo(amp.b). //成员函数，因为效率问题不使用函数指针 return 0. } 输出结果： 11 It is B.c=13 It is A.a=1 It is B_Fun
```

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