Atitit 面向对象 封装的实现原理

[1.1. 动态对象的模拟使用map+函数接口可以实现 1](#_Toc6109)

[1.2. 在用结构体 + 函数指针 模拟 对象 1](#_Toc3915)

[1.3. This指针。。 1](#_Toc29728)

[1.4. “继承”方式2 1](#_Toc12275)

[1.5. 函数的重载，应该类似js 函数有个默认函数实现 1](#_Toc6770)

数据封装

封装的定义：

封装是一个面向对象编程的概念，它将数据和操作数据的函数绑定在一起，避免外部对数据的干扰或滥用。数据封装是面向对象概念中的数据隐藏概念的核心

在我看来，类和对象过于细粒度，而真正应该关心的隔离、API 等应该是在模块、组件、库之间的边界上。据我的经验，OOP（Java / Scalar）的代码通常没有任何模块或组件的概念。开发者专注于在每个类上设置便捷，而不去考虑哪组类应该放在一起组成一个独立、可重用、一致的逻辑单元。

## 动态对象的模拟使用map+函数接口可以实现

## 在用结构体 + 函数指针 模拟 对象

封装

## This指针。。

不要通过对象方法传入进去，可以在methInvokr预先注入this

## “继承”方式2

这种方式主要通过内嵌结构体指针来达到“继承”。这样可以减少重复的代码，而且可以减少漏写的问题。（这种方法当修改父类时，并不影响子类结构，只要修改相应函数功能即可。但是调用父类函数时，显得很臃肿）

## 函数的重载，应该类似js 函数有个默认函数实现

其实在C语言中，并不可以函数同名，因为重载在C语言中也成为不可能的事件了（可能只是我不会实现而已）。

 重载，我们常说的重载大概有三种含义：

* 其一，函数重载，指函数名字一样，参数个数、类型不一样的函数声明和实现。由于 C 编译器的缘故，不支持。不过这个影响不大。
* 其二，重定义或者说覆盖，指派生类中定义与基类签名一样（名字、返回值、参数完全一样）的非虚函数，这样派生类的中的函数会覆盖基类的同签名函数，通过成员操作符访问时无法访问基类的同签名函数。
* 其三，虚函数重写，指在派生类中实现基类定义的虚函数或纯虚函数。虚函数是实现多态的关键，可以在结构体中使用函数指针来表达，但要完全实现，也很麻烦。

  我们平常在交流时通常不明确区分上面三种类型的重载，这里出于习惯，也不作区分

C语言中的面向对象技术---模拟对象(百度移动端笔试题之一) - - 博客频道 - CSDN.NET.html

作者:: 绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ）

汉字名：艾提拉（艾龙），   EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

Atiend