**设计模式在Carlife中间件中的应用**

**作者：徐磊**

**摘 要**

在软件开发的过程中存在着这样一个普遍的事实，用户的需求会随着开发的进行不停的变更。随着时间的流逝，开发人员不断的修改代码，于是根据原先设计所得的系统，整体结构逐渐衰落。代码的质量慢慢沉沦，编码工作从严禁的工程堕落为胡吹乱劈的随性行为。于是在不破坏原有设计，不修改原有代码的基础上实现用户的新需求成为了一个迫在眉睫的问题。《设计模式》就是解决这一问题的利器。

古代有兵法《三十六计》，它是根据中国古代军事思想和丰富的斗争经验总结而成的兵书，以利后人学习兵法。上者领悟其神不拘一格可成兵法之家，中者融会贯通运用得当可成一军之将，下者知形式循规守据可自吹自擂。

软件的《设计模式》，则是对以前的编程高手开发经验的总结，以利后人学习编程。上者领悟其神不拘一格可成大牛，中者融会贯通运用得当可拿高薪，下者知形式循规守据可自吹自擂。

《设计模式》是重构的最终目标，给重构指明了方向。通过重构，你可以找出改变的平衡点。你会发现所谓的设计不再是一切动作的前提，而是在整个开发过程中逐渐浮现出来。在系统构筑过程中，你可以学习如何强化设计，其间带来的互动可以让一个程序在开发过程中持续保有良好的设计。

关键词 Singleton模式；Factory模式；

重构，设计模式

## Carlife简介

百度 CarLife 是一款手机车机互联方案，通过 CarLife 用户可以将移动设备（Mobile Device，简称 MD）上符合驾驶安全的应用通过多屏互动技术分享到车机（Head Unit，简称 HU）上显示，并通过车上的触摸屏、硬按键、旋钮操控和麦克风来控制 CarLife，此方案能充分利用手机和车机的各自优势，为驾驶员提供安全、丰富的信息娱乐体验。

## 互联技术

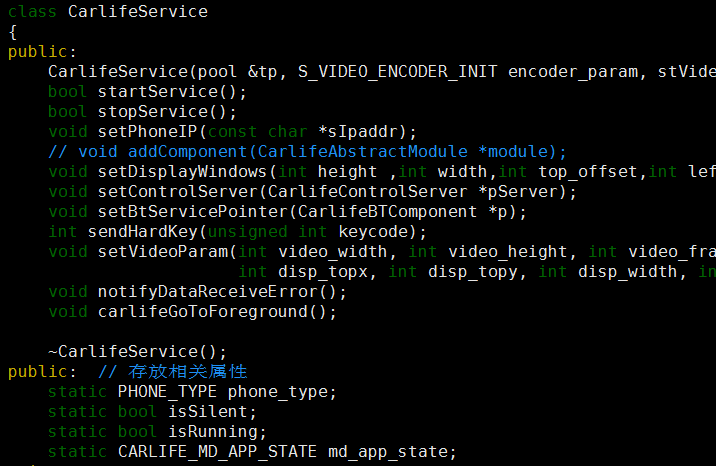
车载互联技术，实际上主要解决：连接方式，视频传输与显示，音频传输与播放，音频采集与传输，回控，通信协议等6个方面。

## 范畴

本文只描述连接方式中使用的Singleton模式、Factory模式。

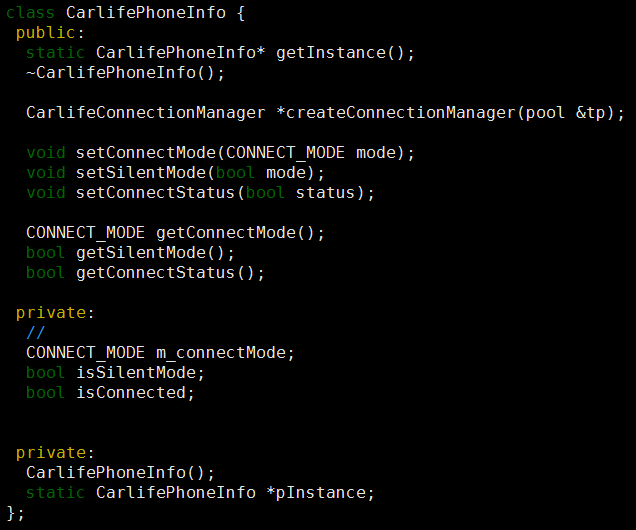
## Singleton模式

### 4.1 原有设计



CarlifeService类中充斥这大量静态成员变量，严重破坏了类的封装性。并且这些属性放在CarlifeService类中并不是很合理，因为其他功能也需要知道当前的连接方式，程序所处的状态。

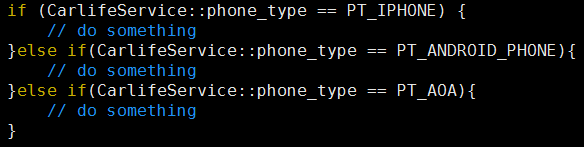
### 4.2 重构后Singleton模式设计



将连接方式、程序所处的状态等，使用Singleton模式，单独抽象出一个类，统一管理整个程序的连接信息。让程序更加清晰。

## Factory模式

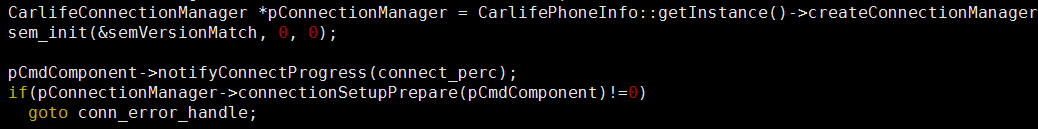
### 5.1 原有设计

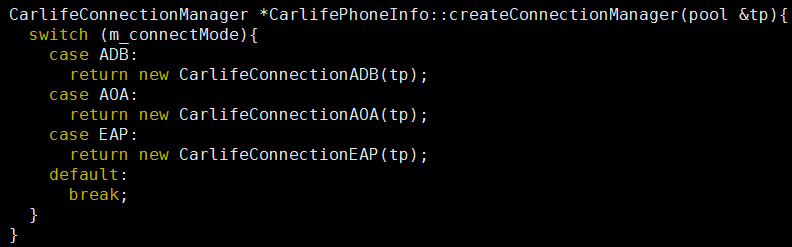


以前的设计是一种面向过程的设计思维，每种连接方式对应一种一个if语句，导致程序中有大段大段的if-else语句。刚开始只有adb和ncm连接方式，使用这种方式还能勉强应付。后来增加了EA的连接方式，再后来又增加了wifi的连接方式，最近在研究新的usbmuxd的连接方式，等研究完之后可能又需要增加一种连接方式。所以这端代码一改再改，而且if-else越来越长。

### 5.2 重构后Factory模式设计







采用工程模式之后如果要增加连接方式，只需要增加一个类，继承CarlifeConnectionManager类就，然后修改CarlifePhoneInfo类中创建连接对象的createConnectionManager函数就可以了。

## 参考文献

1.《设计模式 可复用面向对象软件的基础》

2.《重构 改善既有代码的设计》