Atitit 计算软件简史 艾提拉著

目录

[1.1. 第二代软件（1959~1965） 高级语言 第三代软件（1965~1971） os 1](#_Toc26809)

[1.2. 第四代软件（1971~1989）结构化的程序设计方法 与gui 1](#_Toc26048)

[1.3. 第五代软件（1990~今天） oo 和www 1](#_Toc31129)

[2. 艾提拉总结最佳的软件开发方法 1](#_Toc6451)

[2.1. 面向过程 1](#_Toc29989)

[2.2. 面向组件编程（COP） 1](#_Toc3199)

[2.3. 、面向方面编程（AOP） 2](#_Toc29329)

[2.4. 和面向服务编程（SOP） 2](#_Toc29416)

## 第二代软件（1959~1965） 高级语言 第三代软件（1965~1971） os

## 第四代软件（1971~1989）结构化的程序设计方法 与gui

      20世纪70年代出现了更好的程序设计技术--结构化的程序设计方法

## ****第五代软件（1990~今天） oo 和www****

      第五代软件中有三个著名事件。即在计算机软件业具有主导地位的Microsoft公司的崛起、面向对象的设计和编程方法以及万维网(World Wide Web)的普及。

# ****艾提拉总结最佳的软件开发方法****

## ****面向过程****

## **面向组件编程（COP）**

COP比OOP更进一步。通常OOP将数据对象组织到实体中。这种方法具有很多优点。但是，OOP有一个大的限制：对象之间的相互依赖关系。去掉这个限制的一个好的想法就是组件。组件和一般对象之间的关键区别是组件是可以替代的

## **、面向方面编程（AOP）**

将通用需求功能从不相关类之中分离出来；同时，能够使得很多类共享一个行为，一旦行为发生变化，不必修改很多类，只要修改这个行为就可以。   
AOP就是这种实现分散关注的编程方法，它将“关注”封装在“方面”中。

## **和面向服务编程（SOP）**

  SOP的思想明显不同于面向对象的编程，面向对象编程强烈的建议你应该将数据与其操作绑定。因此在面向对象编程风格中，每张CD 有它自己的CD播放机，他们之间不能被拆开。这听起来很奇怪，但是这就是我们建立许多已存软件系统的方式。

而SOP就不一样了，为了减少异构性、互操作性和不断改变的要求的问题，这样的体系结构应该提供平台来构建具有下列特征的应用程序服务：   
松散耦合、位置透明、协议独立   
    基于这样的面向服务的体系结构，服务使用者甚至不必关心与之通信的特定服务，因为底层基础设施或服务“总线”将代表使用者做出适当的选择。基础设施对请求者隐藏了尽可能多的技术。特别地，来自不同实现技术（如 J2EE 或 .NET）的技术规范不应该影响 SOP用户。如果已经存在一个服务实现，我们就还应该重新考虑用一个“更好”的服务实现来代替，新的服务实现必须具有更好的服务质量。

计算软件简史【转】 - 龙格泽月 - 博客园.html