

## 2020/12/15\_COM\_第1课\_插件的设计

笔记本： COM

创建时间： 2020/12/15 星期二 15:00

作者： ileemi

---

- [COM组件 \(一套设计\)](#)
- [插件设计](#)
- [插件设计技术的好处](#)
- [CAD程序设计](#)
- [接口规范](#)
  - [响应多菜单事件](#)
- [MFC 哈希表](#)
- [添加新的接口](#)

讲解设计

## COM组件 (一套设计)

**组件对象模型**，微软设计规范。

用途：

## 插件设计

插件 (plugin)：是一种遵循一定规范的应用程序接口编写出来的程序。很多软件都有插件，插件有无数种，例如在IE中，安装相关的插件后，WEB浏览器能够直接调用插件程序，用于处理特定类型的文件。（这里的插件是针对 应用程序 的，并不是针对操作系统的）

现在软件开发注重处（性能、维护性（需求变化）等角度）：维护性

技术上解决这些问题：

- 动态加载更新后的代码（动态库）
- 将.dll按照功能进行拆分（便于管理，方便更新，dll名称需要固定）
- 程序接口的设计很重要
- 添加新的 dll 文件，在dll名称固定的前提下，需要更新主程序。不更新主程序可以添加一个配置文件，配置文件中填写需要加载的 dll 名称。

## 插件设计技术的好处

使用插件技术能够在分析、设计、开发、项目计划、协作生产和产品扩展等很多方面带来好处：

- 结构清晰、易于理解。由于借鉴了硬供总线的结构，而且各个插件之间是相互独立的，所以结构非常清晰也更容易理解。
- 易修改、可维护性强。由于插件与宿主程序之间通过接口联系，就像硬件插卡一样，可以被随时删除，插入和修改，所以结构很灵活，容易修改，方便软件的升级和维护。
- 可移植性强、重用力度大。
- 结构容易调整。系统功能的增加或减少，只需相应的增删插件，而不影响整个体系结构，因此能方便的实现结构调整。
- 插件之间的耦合度较低。由于插件通过与宿主程序通信来实现插件与插件，插件与宿主程序间的通信，所以插件之间的耦合度更低。
- 可以在软件开发的过程中修改应用程序。由于采用了插件的结构，可以在软件的开发过程中随时修改插件，也可以在应用程序发行之后，通过补丁包的形式增删插件通过这种形式达到修改应用程序的目的。
- 灵活多变的软件开发方式。

## CAD程序设计

插件适合团队开发，一个类、一个功能一个.dll。

插件是在程序运行的时候加载的。现在程序大多数程序都是用组件设计的。

工具 -- 调试 -- 符号 -- 关闭连接微软服务器下载 .pdb文件

## 接口规范

添加一个对应的接口头文件（将接口的声明放到该头文件中），接口名前添加特殊的前缀（防止名称冲突）。

## 响应多菜单事件

**范围响应消息：**

添加宏：**ON\_COMMAND\_RANGE**

原型：**ON\_COMMAND\_RANGE(id, idLast, memberFxn);**（参数三：函数指针）

使用示例：

**ON\_COMMAND\_RANGE(1000, 60000, &CView::OnPluginCmd);**

菜单ID在1000~60000之间的，在程序运行的时候都会可以被点击（不呈现灰色状态）。

使用 GetParent 获取主窗口

# MFC 哈希表

## CMap

使用时一般给四个参数（前两个参数给定模板的值，后两个参数是用来指定函数参数的类型（一般使用引用&）），使用示例如下：

## 添加新的接口

在程序中添加新的接口，主程序的代码需要进行对应的修改，对应的 .dll 也需要重新编写。程序的维护性不高。

添加新的功能，主程序不进行大的改动：

- 使用共用体，将对应的方法函数指针存储到其中
- 在动态库中导出一个函数指针数组（数组中存储对应的函数名）