

# SolidWorks

# 2008 二次开发基础 与实例(VC++)

王文波 涂海宁 熊君星 编著



清华大学出版社

# SolidWorks

## 2008 二次开发基础 与实例 (VC++)

王文波 涂海宁 熊君星 编著

清华大学出版社  
北京

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

## 图书在版编目(CIP)数据

SolidWorks 2008 二次开发基础与实例(VC++)/王文波,涂海宁,熊君星编著. —北京:清华大学出版社,2009.8

ISBN 978-7-302-20432-9

I. S… II. ①王… ②涂… ③熊… III. ①计算机辅助设计—应用软件,SolidWorks 2008 ②C语言—程序设计 IV. TP391.72 TP312

中国版本图书馆CIP数据核字(2009)第104741号

责任编辑:冯昕 赵从棉

责任校对:赵丽敏

责任印制:李红英

出版发行:清华大学出版社

地址:北京清华大学学研大厦A座

<http://www.tup.com.cn>

邮编:100084

社总机:010-62770175

邮购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印刷者:北京市人民文学印刷厂

装订者:三河市溧源装订厂

经销:全国新华书店

开本:175×245

印张:18.75 字数:386千字

附光盘1张

版次:2009年8月第1版

印次:2009年8月第1次印刷

印数:1~4000

定价:35.00元

---

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:033873-01

# 前 言

SolidWorks 是一款非常优秀的三维机械软件。SolidWorks 在企业、单位中的应用越来越广泛,学习和使用 SolidWorks 的人员越来越多。企业为了提高效率,提高市场竞争力,必然有快速开发新产品、形成自身产品特色的需求。这些需求又使得 SolidWorks 软件二次开发的需求也日益迫切。

目前,有关 SolidWorks 软件二次开发的书籍很少,而且以介绍使用 VB 进行 SolidWorks 软件二次开发为主。广大学习者只能依靠 SolidWorks API 帮助自己摸索。但是 SolidWorks API 的帮助全是英文的,这对于广大的 SolidWorks 软件使用人员和开发人员阅读理解有一定的困难,而且 SolidWorks API 帮助偏重于理论上的介绍,联系实际应用的很少,广大学习者在学习开发的过程中肯定会遇到各种困难。为了帮助他们更好、更快地掌握 SolidWorks 软件的二次开发,作者将自己的学习过程和所操作的实际项目的内容进行总结提炼,编写了本书。

由于 SolidWorks API 的内容非常多,SolidWorks 软件二次开发牵涉的面非常广,不可能在一本书中将其完全论述,所以,本书把 SolidWorks 软件二次开发的内容归为用户界面、文件操作、特征与草图、零件、工程图、装配体等几个主要方面。针对每种类型都介绍了一些实例,并对实例所能完成的功能、实例实现的步骤、实例中所用到的 SolidWorks API 函数和所有程序进行了详细的讲解。本书所举的实例有其特定的应用场合,但读者只要掌握了本书所论述的 SolidWorks 二次开发的知识和二次开发的方法,理解了二次开发的思路,就能根据实际的开发需求,查找 SolidWorks API 帮助中相关的函数,解决开发中的实际问题,实现满足企业实际开发需求的功能。

随书附带的光盘中,包括本书所有的 SolidWorks 二次开发源代码和模型文件。

由于编者水平有限,书中难免有疏漏和不足之处,希望广大读者批评指正,编者邮箱为 wbingd@tom.com。

编 者

2009.4

# 目 录

<b>第 1 章 SolidWorks 二次开发基础</b> .....	<b>1</b>
1.1 SolidWorks API 基础知识 .....	1
1.1.1 SolidWorks API 中的术语 .....	1
1.1.2 SolidWorks API 对象概述 .....	7
1.1.3 SolidWorks API 的语法 .....	13
1.2 用 VC++ 6.0 开发 SolidWorks 的步骤 .....	23
1.2.1 SolidWorks API SDK 的安装 .....	23
1.2.2 第一个插件程序 .....	26
1.2.3 注册表与 SolidWorks 二次开发 .....	40
<b>第 2 章 文件操作</b> .....	<b>43</b>
2.1 获得文件的历史版本信息 .....	43
2.2 遍历所有打开的文件 .....	46
2.3 查找文件所参考的其他文件 .....	49
2.4 新建零件文件 .....	52
2.5 自定义属性页 .....	54
<b>第 3 章 特征与草图</b> .....	<b>58</b>
3.1 获得草图及草图中的所有线段 .....	58
3.2 获得样条曲线的参数 .....	63
3.3 列出草图线段的约束 .....	68
3.4 生成草图样条曲线 .....	72
3.5 获得所选特征的父特征 .....	76
3.6 草图关联检查 .....	79
3.7 隐藏 FeatureManager 中所选的特征 .....	81

3.8	创建拉伸特征	86
<b>第 4 章</b>	<b>零件与几何体</b>	<b>91</b>
4.1	获取用户选择的对象	91
4.2	获得零件的配置	98
4.3	更改视图比例	103
4.4	动态旋转模型	106
4.5	获得并设置零件的单位	108
<b>第 5 章</b>	<b>工程图</b>	<b>111</b>
5.1	标题栏规范检查	111
5.2	工程图属性检查	119
5.3	提取材料明细表信息	123
5.4	移动所有的尺寸到指定的图层	128
5.5	将自定义属性和摘要中的内容写入标题栏	133
5.6	新建工程图文件	139
<b>第 6 章</b>	<b>装配体</b>	<b>142</b>
6.1	非模态对话框	142
6.2	遍历装配并获得各装配件的质量	149
6.3	获得装配件的约束状态	155
6.4	检查装配件中各零件是否处于隐藏状态	160
6.5	获得装配件中各零件的属性	166
6.6	遍历装配体中的同心配合装配关系及其配合参数	171
6.7	新建装配体文件	187
6.8	零件装配	188
<b>第 7 章</b>	<b>对话框与数据库</b>	<b>193</b>
7.1	模态对话框	193
7.2	在模态对话框中使用 SolidWorks API	197
7.3	非模态对话框	202
7.4	MFC 控件使用实例	205
7.5	数据库	225
<b>第 8 章</b>	<b>参数化设计</b>	<b>231</b>
8.1	使用系列零件设计表建立新配置	231



8.2	修改尺寸实现参数化设计 .....	236
8.3	Excel 在 SolidWorks 二次开发中的应用 .....	244
8.4	齿轮的参数化设计 .....	256
<b>第 9 章</b>	<b>使用 Visual C++ .NET 开发 SolidWorks .....</b>	<b>266</b>
9.1	使用 VC++ .NET 制作 SolidWorks 二次开发插件 .....	266
<b>第 10 章</b>	<b>异步模式 .....</b>	<b>274</b>
10.1	SolidWorks 二次开发的异步模式 .....	274
<b>附录</b>	<b>.....</b>	<b>283</b>
<b>参考文献</b>	<b>.....</b>	<b>292</b>

## SolidWorks 二次开发基础

本章分为两大部分,第一部分介绍 SolidWorks 二次开发的基本概念和基础知识,包括 SolidWorks API 对象和 SolidWorks API 的语法。第二部分通过一个实例来介绍 SolidWorks 2008 二次开发的基本过程。

当前主流的三维软件,如 SolidWorks、UG、Pro/E 等,都提供了二次开发接口(API),使用户能在软件本身功能的基础上开发出新的功能模块,满足特定的需求。二次开发的优势在于它可以通过三维软件提供的 API 调用三维软件本身的功能,直接在已有功能的基础上开发特定功能的软件模块。二次开发的不足之处是开发的自由度受限于开发平台接口的开放程度,SolidWorks 二次开发接口(SolidWorks API)有着良好的开发性和兼容性。SolidWorks 软件本身的功能大部分都可以通过调用 API 命令实现,同时在 SolidWorks 2008 API 中还加入了支持 OLE 对象操作的接口(比如图片和 Excel 文档)。

SolidWorks 2008 API 对二次开发人员有着良好的支持,在 SolidWorks 中有针对 Visual C++ 6.0 的开发向导和 Visual C++ .NET 的开发向导,这些开发向导为开发人员构造出一个基本的二次开发框架,降低了进行二次开发的难度,极大地提高了开发效率。SolidWorks 支持开发人员采用多种开发工具进行二次开发,在 SolidWorks 2008 中,开发人员可以采用的开发工具有 Visual C++ 6.0、Visual Basic、VBA、Visual C++ .NET、Visual Basic .NET 和其他支持 COM/OLE 的开发工具(比如 DELPHI)。

### 1.1 SolidWorks API 基础知识

#### 1.1.1 SolidWorks API 中的术语

##### 1. COM

COM(Component Object Model,组件对象模型)技术是 SolidWorks 2008 API



的基础。下面简单介绍几个与 COM 技术相关的概念。

- COM 对象

COM 对象就是以对象形式封装的一个实体,包含接口、属性和事件。每一个 COM 对象都有唯一的标志,称为 CLSID,它是一个 128 位的全局唯一标识符。下面是一个 CLSID 例子:

```
{224AB178-3EC3-4580-980C-242BB199539C}
```

注册组件后 CLSID 会被写入系统注册表中,在系统注册表的 CLSID 下,有该组件对象所在的文件路径。应用程序在加载组件时可根据 CLSID 找到组件对象,进而使用组件对象。在进行 SolidWorks 二次开发时,会经常接触到 COM 对象,如 SldWorks 就是一个 COM 对象。

COM 对象实际上是一个 C++ 类,而接口都是纯虚类,对象从接口派生而来。我们可以简单地用纯粹的 C++ 的语法形式来描述 COM:

```
class IObject
{
public:
    virtual Function1(...)
    virtual Function2(...)
    :
};
class MyObject : public IObject
{
public:
    virtual Function1(...) {...}
    virtual Function2(...) {...}
    :
};
```

IObject 就是我们常说的接口,MyObject 就是所谓的 COM 对象。接口都是纯虚类,它所包含的函数都是纯虚函数,而且没有成员变量。而 COM 对象就是从这些纯虚类继承下来的派生类,它实现了这些虚函数,仅此而已。从上面也可以看出,COM 组件是以 C++ 为基础的,特别重要的是虚函数和多态性的概念,COM 中所有函数都是虚函数,都必须通过虚函数表 VTable 来调用。

- COM 接口

COM 对象以接口方式提供服务,COM 接口是 COM 对象与使用 COM 对象的客户程序进行通信的唯一通道。

COM 对象有 3 个最基本的接口类,分别是 IUnknown、IClassFactory、IDispatch。

COM 规范规定任何对象、任何接口都必须从 IUnknown 继承。IUnknown 包含

3 个函数,分别是 QueryInterface、AddRef、Release。这 3 个函数是很重要的,而且它们的排列顺序也是不可改变的。QueryInterface 用于查询组件实现的其他接口,也就是看看这个组件的父类中还有哪些接口类;AddRef 用于增加引用计数;Release 用于减少引用计数。引用计数也是 COM 中的一个非常重要的概念。简单地可以这样理解,COM 组件是个 DLL,当客户程序要用它时,就将它装到内存中。另一方面,一个组件也不是只给一个客户用的,可能会有很多个程序同时都要用到它。但实际上 DLL 只装载了一次,即内存中只有一个 COM 组件,那 COM 组件由谁来释放?不可能由客户程序,因为如果你释放了组件,那别人怎么用,所以只能由 COM 组件自己来负责。因此出现了引用计数的概念,COM 维持一个计数,记录当前有多少客户在用它,每多一次调用计数就加一,少一个客户用它就减一,当最后一个客户释放它的时候,COM 知道已经没有人用它了,它的使用已经结束了,就把自己释放了。引用计数是 COM 编程中非常容易出错的一个地方,但所幸 VC 的各种各样的类库里已经基本上把 AddRef 的调用给隐含了,我们只需在适当的时候调用 Release。至少在两种情况下要记住调用 Release,第一种是调用了 QueryInterface 以后,第二种是调用了任何得到一个接口的指针的函数以后。

IClassFactory 的作用是创建 COM 组件。我们已经知道 COM 组件实际上就是一个类,一般是用 new 命令实例化一个类对象的,COM 组件也同样如此。但是谁来 new 它呢?不可能是客户程序,因为客户程序不可能知道组件的类名字。如果客户知道组件的类名字,那组件的可重用性就要打个大大的折扣了。事实上客户程序只不过知道一个代表着组件的 128 位的数字串而已。所以客户无法自己创建组件,而且如果组件是在远程的机器上,更不可能 new 出一个对象来。所以创建组件的责任就交给了一个单独的对象,这个对象就是类厂。每个组件都必须有一个与之相关的类厂,这个类厂知道怎样创建组件。当客户请求一个组件对象的实例时,实际上这个请求交给了类厂,由类厂创建组件实例,然后把实例指针交给客户程序。这个过程在跨进程及远程创建组件时特别有用,因为这时不是一个简单的 new 操作就可以的了,它必须经过调度,而这些复杂的操作都交给类厂对象去做了。IClassFactory 一个最重要的函数就是 CreateInstance,顾名思义就是创建组件实例。一般情况下不会直接调用它,API 函数都为我们封装好它了,只有某些特殊情况下才会由我们来调用,这也是使用 VC 编写 COM 组件的优点,使我们有了更多的控制机会。

IDispatch 叫做调度接口。它的作用何在呢?除了 C++ 外还有很多别的语言,比如 VB、VJ、VBScript、JavaScript 等。我们知道 COM 组件是 C++ 类,是靠虚函数表来调用函数的,对于 VC 来说毫无问题,这本来就是针对 C++ 而设计的;以前 VB 不能,但现在 VB 也可以用指针了,也可以通过 VTable 来调用函数了;VJ 也可以,但仍然有些语言不行,那就是脚本语言,典型的如 VBScript、JavaScript。不行的原因在于它们并不支持指针,也就无法用多态性和调用虚函数。现在网页上用的都是这些脚本语言,而分布式应用也是 COM 组件的一个主要市场,它不得被这些脚本语言

所调用。调度接口把每一个函数每一个属性都编上号,客户程序调用这些函数属性的时候只需把这些编号传给 IDispatch 接口即可, IDispatch 再根据这些编号调用相应的函数,当然实际的过程远比这复杂。IDispatch 接口的主要函数是 Invoke,客户程序都调用它,然后 Invoke 再调用相应的函数。

- IDL

COM 是与语言硬件无关的,所以需要一种通用的语言来定义接口,使接口既能被 C++ 使用,也能被 Java 使用。COM 接口是通过引入 IDL 语言解决这个问题的。

IDL (Interface Define Language, 接口定义语言) 是一种与平台硬件无关的接口描述语言,不同的编译平台通过 IDL 编译工具(如 Visual C++ 中的 MIDL)生成相应语言的接口(如 MIDL 就把 IDL 描述的接口生成 C++ 接口)。下面是一个 IDL 定义的 IMyAddinObj 接口,该接口中有 3 个方法:

```
interface IMyAddinObj: IDispatch
{
[id(1),helpstring("Start Notepad")] HRESULT StartNotepad;
[id(2),helpstring("Toolbar Update")] HRESULT ToolbarUpdate([out,retval] long *
status);
[id(3),helpstring("Display PropertyManagerPage")] HRESULT DisplayPMP();
};
```

COM 接口是组件间进行通信的基础,是软件组件的关键。在进行 SolidWorks 二次开发时,也是首先得到 SolidWorks API 对象的接口,然后再使用接口提供的方法。一个 COM 对象可以有一个或多个 COM 接口。

- 方法

方法是指一个接口中提供的供 COM 客户使用的函数,这是开发人员最终使用的函数,可以在 ISldWorks 接口中看到很多的方法。开发人员在使用 COM 组件时最终也是调用接口的方法。

- 事件

事件是指一个组件对象拥有的消息响应机制,开发人员可以通过在插件中建立响应机制来获取该事件,从而进行一定的处理。

- 组件模块

组件模块为组件对象提供活动空间,并且完成组件对象的注册,通常是以 DLL 的形式表现。COM 对象就封装在组件模块中,一个组件模块中可以有一个或多个组件对象。

- 引用计数

引用计数是用来控制 COM 对象生存的一个参数,当该参数为 0 时,组件从内存空间中释放。引用计数是一个很繁琐的参数,简单地说,每调用 AddRef 会增加一次,调用 Release 会减少一次。

## 2. OLE

OLE(Object Linking and Embedding,对象链接与嵌套)是 Microsoft 公司与计算机界合作并发展起来的产品规范,从开发人员的角度讲,OLE 是一种能使他们共享不同应用程序信息资源的技术。通过使用 OLE 控件,能够构造声、文、图、像、影于一体的复合文档,可以很容易地集成应用程序。ActiveX 控件就是在 OLE 控件技术上发展起来的。

SolidWorks 是围绕 OLE 设计的,它也是 SolidWorks API 的基础。而且在 SolidWorks API 中也在不断加强 OLE 开发的支持,最新的 SolidWorks 2008 API 明显的特征就是可以通过 SolidWorks API 创建和访问 OLE 对象。

OLE 自动化是 SolidWorks API 构造的基础,也是深入理解 SolidWorks API 的关键。OLE 自动化是一种为 Windows 设计的交互进程通信机制,它让一个应用程序驱动或自动完成另一个应用程序。OLE 自动化包括两部分应用程序:一是把 OLE 组件暴露出来使之可控制的应用程序,叫做 OLE 自动化服务器;二是利用 OLE 服务器提供的组件对该对象进行控制操作的应用程序,叫做 OLE 客户。SolidWorks 本身就是 OLE 自动化服务器,而在 SolidWorks 提供接口进行开发得到的插件就是 OLE 客户。

每个 OLE 对象都有属性(Property)和方法(Method),OLE 自动化服务器的编写者决定哪些属性和方法将通过 OLE 自动化暴露出来,而 OLE 自动化服务器的私有对象对于外部用户来说是不可见的。OLE 客户编程人员只需知道包含对象的应用程序的名称、对象的类的名称以及要使用的属性和方法的名称,就可以使用自动化服务器。

在 SolidWorks 2008 二次开发中,SolidWorks API 就是 SolidWorks 作为 OLE 自动化服务器提供的属性和方法,我们开发的插件就是使用这些接口的 OLE 客户。

## 3. ATL

在 SolidWorks 2008 提供的二次开发向导中,Visual C++ 6.0 平台上的向导是基于 ATL 技术构造的。ATL(ActiveX Template Library)是一套 C++ 模板库,与 MFC 的概念相似。使用 ATL 能够快速地开发出高效、简洁的代码,同时对 COM 组件的开发提供最大限度的代码自动生成以及可视化支持。为了方便使用,从 Microsoft Visual C++ 5.0 版本开始,Microsoft 把 ATL 集成到 Visual C++ 开发环境中。

使用 ATL 进行插件开发有以下几个优点:

(1) 为维护代价很高的数据类型(如接口指针、VARIANT、BSTR 和 HWND)提供了包装类。

(2) 提供了一些类,它们实现了诸如 IUnknown、IClassFactory、IDispatch、

IPersistXxx、IConnectionPointContainer 和 IEnumXxx 这些基本的 COM 接口。

(3) 管理 COM 服务器的类,它们用于暴露类对象、自注册和服务器生命周期管理。

(4) 节省手工录入的向导(Wizard)。

#### 4. 开发工具

SolidWorks API 是基于 COM 组件技术构造的,SolidWorks 通过 COM 技术为开发人员提供了强大的二次开发接口,因此凡是支持 COM 编程的开发工具,如 Visual C++、Visual Basic、Delphi 等均可用于 SolidWorks 的二次开发。在 SolidWorks 2008 提供的向导中,支持以下平台,即 Visual Basic 6.0、Visual C++ 6.0、Visual C++ .NET、Visual Basic .NET 和 Visual C# .NET。

最常用的是 Visual Basic、Visual C++ 和 Delphi,三者各有优势。

- Visual C++ 6.0

Visual C++ 6.0 是当今最流行的软件开发工具之一,是程序员的首选编程利器。它提供了功能强大的集成开发环境,用以方便、有效地管理、编写、编译、跟踪 C++ 程序,大大加速了程序员的工作,提高了程序代码的效率。

使用 Visual C++ 进行 SolidWorks 二次开发的优势如下:

(1) 可以使用 GDI 对象和设备环境类所提供的绘图函数,可以轻易地实现绘图功能而且无需考虑具体设备情况。

(2) Visual C++ 6.0 开发环境十分友好,其高度的可视化开发方式和强大的向导工具(AppWizards)能够帮助用户轻松地开发出多种类型的应用程序。大多数情况下,用户只需向自动生成的程序框架中填充定制代码即可,而且使用 ClassWizard 还能够大大简化这个过程。

(3) Visual C++ 6.0 有着强大的调试功能,能够帮助开发人员寻找错误和提高程序效率。

(4) Visual C++ 与 SolidWorks 有极好的连接性,能直接调用许多资源,方便地在 SolidWorks 上添加命令和各种控件。

因此,可以说它是 SolidWorks 的最佳开发工具,使用 Visual C++ 能最大程度地使用 SolidWorks API,并且开发的插件与其他两种开发方式相比有更好的效率,更适用于进行大型系统的开发。使用 Visual C++ 进行二次开发的不足之处是对开发人员的要求较高。

- Visual Basic

使用 Visual Basic 进行二次开发的优势在于容易使用,上手快。在 Visual Basic 中引入了控件的概念,各种各样的按钮、文本框、无线钮都是控件的种类。Visual Basic 把这些控件模式化,并且每个控件都有若干属性用来控制控件的外观和工作方法。Visual Basic 在原有 Basic 语言的基础上进一步发展,至今包含了数百条语句、

函数及关键词,其中很多和 Windows GUI 有直接关系。专业人员可以用 Visual Basic 实现其任何 Windows 编程语言的功能,而初学者只要掌握几个关键词就可以建立实用的应用程序。

与 Visual C++ 这样主流的语言相比,Visual Basic 具有不可取代的简单易用性和真正的“所见即所得”特性。Visual Basic 最大的特点就是简单。使用 Visual Basic 开发软件,开发周期短,代码效率高。

同 Visual C++ 一样,Visual Basic 也和 SolidWorks 具有良好的连接性,它的优势更主要体现在使用方便、简洁,适合初学者使用;不足之处在于开发的程序与 Visual C++ 开发相比,效率较低,而且不适合开发大型系统。

- VBA

VBA(Visual Basic for Applications)是 Microsoft 公司推出的一种可以被多种应用程序共享的、针对应用程序内部可编程、通用的可视化应用程序编程语言。它虽然具有十分完整的程序语言基本结构,但它提供的并不单单是程序,而是对 MS Office 各种软件功能的一种综合控制。因此,相对于 Visual Basic for Windows 而言,VBA 是一种面向用户(End User Oriented)的控制语言。除此之外,VBA 还具有应用程序生成器的特性。因为用户只要将想自动化的操作记录成宏,就可以直接产生宏程序代码(以后不必再费事编写程序,只需把操作过程记录下来,就会自动产生所需的程序代码)。

VBA 是一套完整的应用程序开发环境,它为用户和开发人员提供了一种应用程序间通用的应用程序语言,减少了学习时间和支持费用;而且,VBA 也为开发人员提供一种开发方法,用于开发集成多个应用程序的系统。在 Microsoft Office 中,所有的应用程序都可以使用 VBA。不想多花时间学习 Visual Basic 的用户,可以直接应用附加在应用软件中的 VBA,从而实现繁琐、机械的日常工作的自动化,提高用户的办公效率。在 AutoCAD 中就有大量的 VBA 应用,同样在 SolidWorks 中也有不少类似的应用,但由于它只有依托其他软件才能运行,而且缺乏通用性,不能编译,上述这些使它不能成为二次开发的得力工具,充其量是个简化操作、自动化控制的工具。

## 1.1.2 SolidWorks API 对象概述

### 1. SolidWorks 2008 API

SolidWorks 2008 API 也称为 SolidWorks 应用程序开发接口,是指 SolidWorks 2008 程序暴露给开发人员使用的接口。就如 Windows 2008 API 是 Windows 操作系统的接口,供开发人员通过 API 调用系统函数创建进程、线程一样,SolidWorks 2008 API 为开发人员提供 SolidWorks 2008 软件本身的功能,如生成拉伸特征、生成草图等,开发人员可以通过在这些接口的基础上实现需要的功能。

SolidWorks 2008 API 是由组成 SolidWorks 2008 软件的 COM 组件提供的, 每一个 SolidWorks 2008 对象都是采用 COM 技术构造的, 因此有着与一般 COM 对象同样的性质, 都是由接口、属性、方法与事件组成的。

以 PartDoc 对象为例子, 该对象是管理零件文档的对象, 它提供的 SolidWorks API 接口为 IPartDoc。PartDoc 对象有 3 个开发人员可访问的属性: MaterialIdName、MaterialPropertyValues 和 MaterialUserName, 分别对应 PartDoc 对象的材质 ID、材质值和材质名称。PartDoc 对象的方法有很多, 比如 FirstFeature()、GetMaterialPropertyName2(), 访问这些方法的唯一途径就是通过接口 IPartDoc。PartDoc 对象的事件有 filesaveNotify(文件保存事件)、DestroyNotify(文件退出事件)等。每一个 SolidWorks 2008 API 都是由这 4 个基本部分组成。

## 2. SolidWorks 2008 API 对象结构

SolidWorks 2008 API 通过面向对象思想组织所有的接口对象, 如图 1.1 所示。SolidWorks 2008 API 对象可以分为以下几大类。

- 应用程序对象

应用程序对象包括 SldWorks、ModelDoc2、PartDoc、AssemblyDoc 及 DrawingDoc 对象。PartDoc、AssemblyDoc、DrawingDoc 分别对应【新建】|【文件】中的 3 种文件类型即零件、装配体和工程图。其中每一种的下一级都有多个对象。ModelDoc2 对象提供了这 3 种文件类型共有的功能, 如打印、文件保存等。

ModelDoc2 与 PartDoc、AssemblyDoc 及 DrawingDoc 的关系是, 前者包含了后 3 者, 因此, 这几个接口之间可以通过 COM 的 QueryInterface 进行查询, 如图 1.2 所示。

PartDoc、AssemblyDoc 和 DrawingDoc 又是由多个其他的 SolidWorks 对象组成, 如 PartDoc 由 LightDialog 对象(描述灯光环境)和 Body2 对象及 Body2 下面多个对象(用于描述 PartDoc 对象的几何数据)组成, 如图 1.3 所示。

AssemblyDoc 与 DrawingDoc 一样, 也是由多个对象组成。AssemblyDoc 对象由 LightDialog 对象(描述灯光环境)、Component2 对象(描述装配体的组成部件)、Mate2 对象(描述装配关系)组成, DrawingDoc 对象由 LightDialog 对象(描述灯光环境)、Sheet 对象(描述 Bom 表)和 View 对象及其下面多个对象(描述工程图内容)组成, 如图 1.4 所示。

- 配置文件对象

配置文件对象管理零件中不同模块(零件文档模式)与装配体中不同零件(装配体文档模式)的状态。在零件文档模式下, 可以将其中的复杂特征设置成压缩模式; 在装配体文档模式下, 可以将其中的一个或多个零件设置成压缩模式。

- 事件对象

SolidWorks 2008 API 接口中提供了对事件的支持, 当前版本中支持的事件类

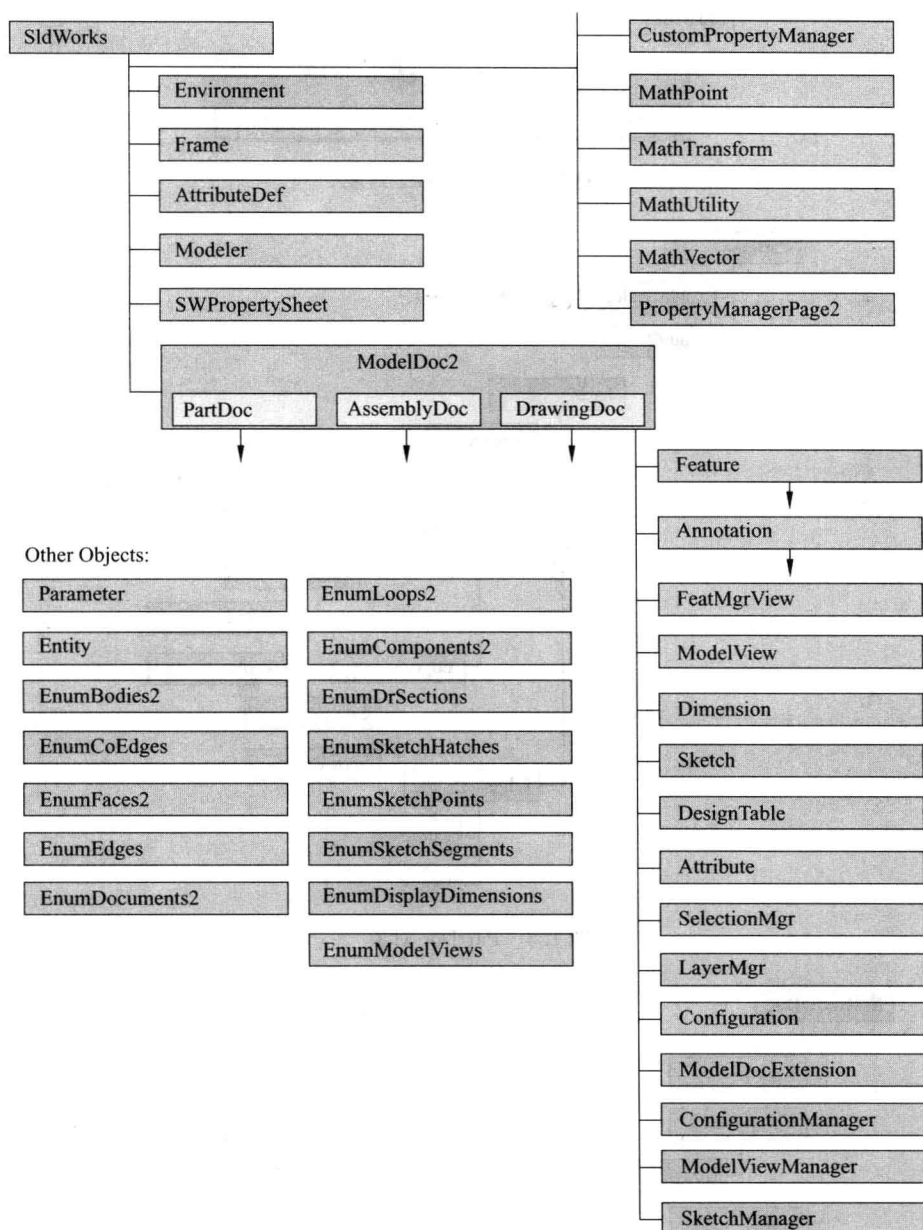


图 1.1 SolidWorks 2008 API 接口对象

型有 AssemblyDoc 事件、DrawingDoc 事件、FeatMgrView 事件、ModelView 事件、PartDoc 事件、SldWorks 事件及 SWPropertySheet 事件。开发人员可以截获应用程序中的事件并根据需要加入相应的功能。



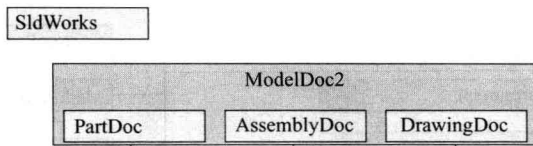


图 1.2 ModelDoc2 对象

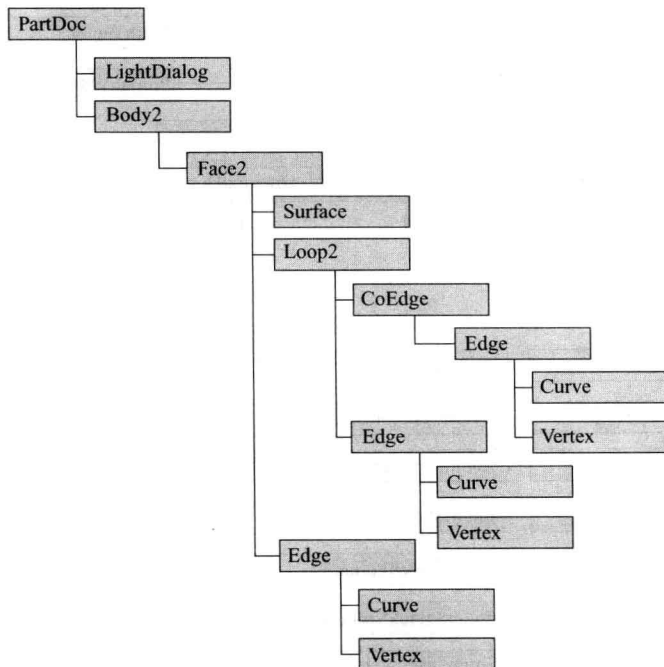


图 1.3 PartDoc 对象

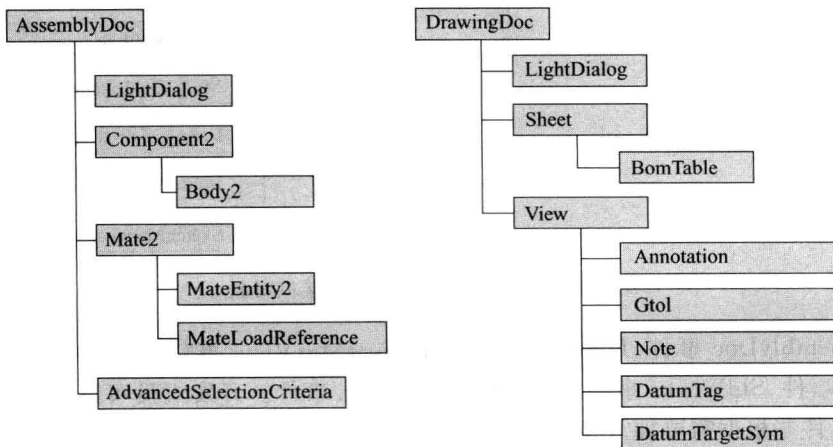


图 1.4 AssemblyDoc 与 DrawingDoc 对象