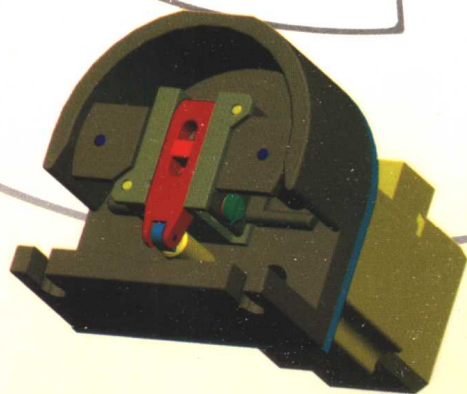
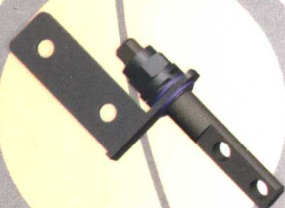




Pro/ENGINEER 工程师之路丛书

Pro/TOOLKIT 程序设计



李世国 编著



机械工业出版社
China Machine Press

Pro/TOOLKIT 程序设计

李世国 编著



机械工业出版社

本书详尽地介绍了在 Pro/ENGINEER 和 Pro/ENGINEER Wildfire 环境下运行的 Pro/TOOLKIT 应用程序开发技术, 主要包括 Pro/TOOLKIT 应用程序创建方法、特征创建技术、菜单设计、UI 对话框设计、基于 MFC 的用户界面设计、Pro/ENGINEER 模型的创建和管理、窗口和视图的显示与控制、设计参数创建与应用、尺寸约束的检索与应用和工程图的生成和控制等。

本书论述深入浅出、突出重点、层次清楚, 以简洁的程序代码说明了 Pro/TOOLKIT 应用程序开发中的关键技术和疑难问题。所有示例程序经过作者精心设计和上机验证。利用 Visual C++6.0 集成开发环境, 突破了 Pro/TOOLKIT 提供的用 MAKE 文件来管理、编译、连接和生成可执行程序的局限, 提出了一种既先进又便捷的应用程序创建方法。

所附光盘包含了本书全部实例的工程文件、源程序和资源文件, 非常便于学习、理解和使用。

本书的内容反映了基于三维 CAD/CAM 系统的应用和高级开发技术, 可作为产品数字化建模、三维模型库的生成以及三维参数化设计等 CAD 应用系统开发等方面的不可多得的学习教材和参考资料。

图书在版编目 (CIP) 数据

Pro/TOOLKIT 程序设计/李世国编著. -北京: 机械工业出版社, 2003.6

ISBN 7-111-12380-8

I. P… II. 李… III. 机械设计: 计算机辅助设计-应用软件,
Pro/TOOLKIT-程序设计 IV. TH122

中国版本图书馆 CIP 数据核字 (2003) 第 047128 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 吴宏伟 版式设计: 张丽花

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2003 年 7 月第 1 版第 1 次印刷

787mm×1092mm 1/16·29.5 印张·729 千字

0001-5000 册

定价: 47.00 元 (含 1CD)

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话: (010) 68993821、88379646

封面无防伪标均为盗版

前 言

产品的三维设计离不开 CAD/CAM 系统的支持, 具有技术成熟、功能强大的 Pro/ENGINEER 参数化造型系统在设计 and 制造业中引起了人们的重视, 并得到了日益广泛的应用。Pro/ENGINEER 作为通用的三维 CAD/CAM 系统在功能上基本能够满足产品三维设计要求, 但要高效地进行产品设计以满足更高层次的要求, 必须借助于高级开发工具包 Pro/TOOLKIT。

与使用 Pro/ENGINEER 相比, 掌握 Pro/TOOLKIT 程序设计技术要困难一些。主要问题是这方面的参考资料太少, 虽然 Pro/TOOLKIT 提供了开发应用程序的帮助文档 Pro/TOOLKIT APIWizard, 但要真正理解和灵活应用绝非易事。其原因是: 一方面帮助文档的内容非常之多, tkuse.pdf 文件有 1200 多页, 从中找出合适的切入点需要花费很大的精力; 另一方面, 帮助文档给出的实例主要针对 UNIX 系统的 C 程序代码, 程序代码长、可读性不尽人意、理解比较困难; 再加上实例程序还会涉及到与标准 C 语言无关、Pro/TOOLKIT 特有的菜单技术、信息文件等相关文件, 这样给用户增加了不小的难度。当然, 这并不是说使用 Pro/TOOLKIT 高不可攀, 而是入门不易。只要是跨过了这道门坎, 读者就会有“柳暗花明又一村”之感, 对这一点作者有很深切的感受。正是由于这个原因, 作者产生了写书的念头, 相信本书的出版会对读者有很大的帮助。

本书在编写过程中得到了江南大学设计学院领导的关怀和支持, 研究生何建军、李强、卜莹琼、梁琪等为本书的前期准备作了大量的信息收集和整理工作, 谨向他们表示衷心的感谢!

非常感谢机械工业出版社北京时代金科科技有限公司的信任和责任编辑为本书的出版所付出的辛勤劳动!

Pro/TOOLKIT 涉及的内容很多, 由于作者水平有限, 书中难免会有错译和疏漏之处, 欢迎读者批评指正。

编者

2003 年 4 月

目 录

前言

第 1 章 概述	1
1.1 Pro/TOOLKIT 开发环境介绍	1
1.1.1 Pro/TOOLKIT 简介	1
1.1.2 Pro/TOOLKIT 的安装	1
1.1.3 Pro/TOOLKIT 开发环境与 Pro/ENGINEER 版本的关系	2
1.2 有关 Pro/TOOLKIT 的基本知识	2
1.2.1 对象和动作	3
1.2.2 对象句柄	3
1.2.3 Pro/TOOLKIT 的函数原型和函数的返回值	4
1.3 帮助文档的使用方法	6
1.3.1 浏览器环境要求	6
1.3.2 Pro/TOOLKIT APIWizard 的启动	7
1.3.3 使用 Pro/TOOLKIT APIWizard	8
第 2 章 如何在 VC 环境开发 Pro/TOOLKIT 应用程序	10
2.1 创建 Pro/TOOLKIT 应用程序的基本方法	10
2.1.1 利用 Make 文件创建 Pro/TOOLKIT 应用程序	10
2.1.2 利用 VC 向导创建 Pro/TOOLKIT 应用程序	11
2.2 用 VC 创建 Pro/TOOLKIT 应用程序基本框架	11
2.3 Pro/TOOLKIT 应用程序设计	12
2.3.1 user_initialize()和 user_terminate()函数	13
2.3.2 Pro/TOOLKIT 应用程序主体部分的调用方法和相关函数	13
2.3.3 应用程序示例 (例 2-1)	16
2.4 编译和连接设置	19
2.4.1 设置包含文件路径	19
2.4.2 设置库文件路径	20
2.4.3 设置连接所需库文件	20
2.5 注册文件及 Pro/TOOLKIT 应用程序的运行	21
2.5.1 注册文件及实例	21
2.5.2 Pro/TOOLKIT 应用程序的运行	23
2.6 有关说明	25

第 3 章 菜单设计技术及应用	26
3.1 有关菜单的术语介绍	26
3.2 创建和操作菜单的基本方法和相关函数	26
3.2.1 向菜单栏中添加菜单条	26
3.2.2 向菜单条下添加复选按钮	28
3.2.3 向菜单条下添加单选按钮组	30
3.2.4 向菜单条下添加下级子菜单	32
3.2.5 对现有菜单命令的控制	32
3.3 创建和操作菜单的应用程序实例 (例 3-1)	33
3.3.1 实例概述及源程序	33
3.3.2 程序分析	40
3.3.3 程序运行	41
3.4 模式菜单设计	42
3.4.1 模式菜单的含义	42
3.4.2 模式菜单的菜单文件	42
3.4.3 模式菜单的创建方法	44
3.4.4 模式菜单的相关函数介绍	45
3.4.5 模式菜单设计实例	46
第 4 章 特征创建方法及程序设计	49
4.1 特征元素树	49
4.1.1 如何理解特征元素树	49
4.1.2 如何确定具体特征的特征元素树	49
4.2 特征元素值、特征元素路径和特征元素	51
4.2.1 特征元素值的设置和检索	51
4.2.2 特征元素路径	52
4.2.3 特征元素	53
4.3 创建特征的一般步骤及相关函数	54
4.3.1 基本步骤及函数用法	54
4.3.2 特征树的查询和分析	56
4.3.3 特征的重定义	56
4.4 创建基准平面	57
4.4.1 基准平面的特征元素树	57
4.4.2 创建默认基准面	60
4.4.3 创建偏距基准面	63
4.4.4 参照基准面的获取方法	66
4.5 创建默认坐标系	67
4.6 创建拉伸特征	69

4.6.1	拉伸特征元素树	69
4.6.2	拉伸特征的创建步骤	72
4.6.3	创建拉伸特征的程序设计及相关函数说明	73
4.6.4	创建第一个拉伸特征	82
4.7	创建旋转特征	82
4.7.1	旋转特征元素树	82
4.7.2	旋转特征的创建步骤及程序设计	84
4.8	创建倒角特征	91
4.8.1	倒角特征元素树	91
4.8.2	创建倒角特征的程序设计及边集的获取方法	92
4.9	创建孔特征及基准轴特征	95
4.9.1	直孔的特征元素树	96
4.9.2	创建直孔特征的程序设计	98
4.9.3	创建基准轴特征的程序设计	102
4.10	创建扫描特征	104
4.10.1	扫描特征的特征元素树	104
4.10.2	扫描特征分析及创建方法	105
4.10.3	创建扫描特征的程序设计及相关函数说明	106
4.11	创建特征的综合实例(例 4-1)	111
4.11.1	工程文件	111
4.11.2	应用程序的初始化和菜单定义	112
4.11.3	CreatFeature.cpp 和 CreateSweepFeat.cpp 文件	120
4.11.4	信息文件	122
第 5 章	UI 对话框	123
5.1	UI 对话框入门	123
5.1.1	资源文件及组成	123
5.1.2	资源文件的编写格式	127
5.1.3	UI 对话框控制程序设计与主要函数介绍	127
5.1.4	使用 UI 对话框的 Pro/TOOLKIT 应用程序设计方法(例 5-1)	129
5.2	UI 对话框元件及共有属性	134
5.2.1	UI 对话框的所有元件	134
5.2.2	显示 UI 对话框所有元件的程序(例 5-2)	142
5.2.3	UI 对话框元件的共有属性	144
5.3	UI 对话框属性及相关函数	145
5.4	UI 对话框元件的相关函数及编程	147
5.4.1	输入框的属性及相关函数	147
5.4.2	复选按钮的属性及相关函数	150

5.4.3	标签的属性及相关函数	151
5.4.4	按钮的属性及相关函数	152
5.4.5	单选按钮组的属性及相关函数	152
5.4.6	列表框的属性及相关函数	155
5.4.7	输入框、列表框等元件编程示例 (例 5-3)	161
5.4.8	选项菜单的属性及相关函数	174
5.4.9	滑块的属性及相关函数	176
5.4.10	指轮的属性及相关函数	178
5.4.11	指轮和滑块元件编程示例 (例 5-4)	179
5.4.12	微调按钮的属性及相关函数	185
5.4.13	文本输入区的属性及相关函数	190
5.4.14	选项卡的属性及相关函数	192
5.4.15	进度条的属性及相关函数	193
5.5	带有菜单条的 UI 对话框	195
5.5.1	UI 对话框菜单条的属性及相关函数	196
5.5.2	UI 对话框菜单区的属性及相关函数	198
5.5.3	UI 对话框菜单分级按钮的属性及相关函数	198
5.5.4	UI 对话框菜单编程示例 (例 5-5)	199
第 6 章	模型、实体及窗口控制与显示	207
6.1	模型的创建和检索	207
6.1.1	模型与模式的基本概念	207
6.1.2	模式的检索	208
6.1.3	模型对象及模型的创建	209
6.1.4	模型创建实例 (例 6-1)	213
6.1.5	模型的标识	214
6.1.6	Pro/ENGINEER 的当前进程、当前模型及相关函数	217
6.1.7	进程中的模型控制实例 (例 6-2)	219
6.1.8	模型文件的管理	226
6.2	模型项和相关函数	228
6.2.1	模型项的基本概念	228
6.2.2	模型项的相关函数	229
6.2.3	模型项名称检索和设置实例 (例 6-3)	230
6.3	实体和实体对象的访问	232
6.3.1	实体特征对象的访问	232
6.3.2	实体特征对象访问实例 (例 6-4)	235
6.3.3	实体其他对象的访问	241
6.4	窗口的显示与控制	242

6.4.1	窗口的显示.....	243
6.4.2	窗口的控制.....	244
6.4.3	窗口显示与控制实例 (例 6-5)	246
6.5	实体的定向.....	252
6.5.1	视图和窗口的变换矩阵.....	252
6.5.2	窗口和视图的控制.....	252
6.5.3	实体定向实例 (例 6-6)	255
第 7 章	参数和三维参数化程序设计.....	265
7.1	参数对象和参数的访问.....	265
7.1.1	参数对象和参数值.....	265
7.1.2	参数值的操作函数.....	267
7.1.3	参数的访问函数.....	268
7.2	参数的创建、检索和编辑.....	271
7.2.1	参数对象的创建方法.....	271
7.2.2	参数对象的检索方法.....	272
7.2.3	参数值的获取和修改.....	274
7.2.4	应用实例 (例 7-1)	275
7.3	基于三维模型的参数化程序设计.....	285
7.3.1	基本原理.....	285
7.3.2	三维模型样板的建立.....	286
7.3.3	应用实例 (例 7-2)	286
第 8 章	使用 MFC 的可视化对话框设计技术.....	300
8.1	如何使用 MFC 创建对话框.....	300
8.1.1	创建对话框的一般过程.....	300
8.1.2	创建一个简单的示例对话框 (例 8-1)	302
8.2	模式对话框设计及实例.....	307
8.2.1	模式对话框及例 8-2 简介.....	307
8.2.2	使用模式对话框的程序设计 (例 8-2)	308
8.3	无模式对话框设计及实例.....	313
8.3.1	无模式对话框的设计方法.....	313
8.3.2	无模式对话框实例 (例 8-3)	314
8.4	属性页对话框设计及实例.....	321
8.4.1	模式属性页对话框的设计方法.....	321
8.4.2	模式属性页对话框实例 (例 8-4)	322
8.4.3	无模式属性页对话框的设计方法.....	329
8.4.4	无模式属性页对话框实例 (例 8-5)	332

第 9 章 尺寸对象及关系式的操作和应用	343
9.1 尺寸对象.....	343
9.2 尺寸对象操作函数.....	343
9.2.1 尺寸对象的检索函数.....	343
9.2.2 尺寸信息检索函数.....	345
9.2.3 尺寸信息修改函数.....	347
9.2.4 尺寸显示控制函数.....	348
9.3 尺寸操作函数应用实例（例 9-1）	349
9.3.1 实例简介.....	349
9.3.2 界面设计.....	349
9.3.3 工程文件、主要程序文件及程序说明	350
9.4 尺寸文本与关系式.....	359
9.4.1 尺寸文本的检索与设置.....	359
9.4.2 关系式、关系式集对象的概念和关系式集指针的获得方法.....	361
9.4.3 关系式的操作函数.....	363
9.4.4 尺寸文本与关系式函数应用实例（例 9-2）	364
第 10 章 工程图的生成和控制.....	381
10.1 工程图环境设置.....	381
10.1.1 与工程图相关的配置文件和选项	381
10.1.2 工程图配置选项设置方法	385
10.1.3 工程图配置选项设置实例（例 10-1）	386
10.2 工程图的生成.....	400
10.2.1 工程图环境及工程图页面	400
10.2.2 工程图页面的创建方法.....	401
10.2.3 工程图页面的相关函数介绍	401
10.2.4 工程图页面操作实例（例 10-2）	404
10.3 视图的检索、创建和修改.....	410
10.3.1 视图的检索.....	410
10.3.2 视图的创建.....	413
10.3.3 视图的修改.....	414
10.3.4 工程图视图操作实例（例 10-3）	415
10.4 工程图尺寸及尺寸公差的显示和修改	428
10.4.1 尺寸显示和修改.....	428
10.4.2 尺寸的公差显示和修改.....	429
10.4.3 尺寸及尺寸公差显示和修改实例（例 10-4）	430
10.5 工程图表格的创建和编辑.....	441
10.5.1 表格的创建.....	442

10.5.2 表格的编辑.....	444
10.5.3 表格操作实例（例 10-5）	445
10.6 结束语.....	456
后记.....	457
参考文献	458

第 1 章 概 述

1.1 Pro/TOOLKIT 开发环境介绍

1.1.1 Pro/TOOLKIT 简介

Pro/TOOLKIT 是美国 Parametric Technology Corporation (PTC) 公司为 Pro/ENGINEER 软件提供的开发工具包，其主要目的是让用户或第三方通过 C 程序代码扩充 Pro/ENGINEER 系统的功能，开发基于 Pro/ENGINEER 系统的应用程序模块，从而满足用户的特殊要求。不仅如此，还可以利用 Pro/TOOLKIT 提供的 UI 对话框、菜单以及 VC 的可视化界面技术，设计出方便实用的人机交互界面，从而大大提高系统的使用效率。

Pro/TOOLKIT 工具包提供了开发 Pro/ENGINEER 所需的函数库文件和头文件，使用户编写的应用程序能够安全地控制和访问 Pro/ENGINEER，并可以实现应用程序模块与 Pro/ENGINEER 系统的无缝集成。

在 Pro/ENGINEER 17 版本及该版本以前，PTC 公司提供的开发工具是 Pro/DEVELOP，从 Pro/ENGINEER 18 版本起，Pro/ENGINEER 系统的开发环境采用 Pro/TOOLKIT。Pro/TOOLKIT 支持在 Windows 2000/NT/XP 操作系统中使用 C 和 C++ 语言设计程序，采用 Microsoft Visual C++ 6.0 作为编译器和连接器，并可以在 Microsoft Visual C++ 6.0 的集成环境下完成程序的设计、调试和编译。

1.1.2 Pro/TOOLKIT 的安装

Pro/TOOLKIT 工具包是随 Pro/ENGINEER 安装盘一起提供的，安装时在“元件”列表框中选“API Toolkits”复选框，并在“子元件”列表框中选“Pro/TOOLKIT”复选框，如图 1-1 所示，将在 Pro/ENGINEER 系统的默认目录下自动创建两个下级目录：protoolkit 和 prodevelop。

Protoolkit 目录中共有 4 个子目录：\i486_nt、\includes、\protk_appls 和 \protkdoc。头文件位于 protoolkit\includes 目录，库文件位于 protoolkit\i486_nt\obj 目录，这两种类型的文件是用 VC 开发 Pro/ENGINEER 应用程序必不可少的支持文件。其余两个子目录的内容分别为程序实例和帮助文档。

在 prodevelop 目录中包含了 \i486_nt、\includes 和 \protk_appls 三个子目录。分别保存了 Pro/DEVELOP 使用的函数库文件、头文件和程序实例。Pro/DEVELOP 中的函数大多数被 Protoolkit 中的函数所取代，但也可以在 Pro/TOOLKIT 中使用 Pro/DEVELOP 的函数。

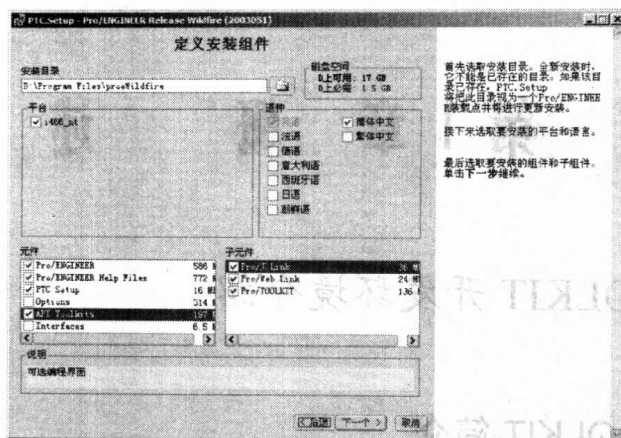


图 1-1 安装 Pro/TOOLKIT 的组件选择

1.1.3 Pro/TOOLKIT 开发环境与 Pro/ENGINEER 版本的关系

在不同 Pro/ENGINEER 版本提供的 Pro/TOOLKIT 开发环境中,其函数库以及头文件的定义会有一些变化,通常在编译时使用的头文件和在连接时使用的库文件最好应与 Pro/TOOLKIT 应用程序运行的 Pro/ENGINEER 环境的版本相一致。与 AutoCAD 的 ObjectARX 开发环境不同的是,使用低版本 Pro/TOOLKIT 的头文件和函数库生成的 Pro/TOOLKIT 应用程序可以在高版本的 Pro/ENGINEER 环境运行,反之则不行。例如,使用 Pro/ENGINEER 2001 版本的 Pro/TOOLKIT 的头文件和函数库生成的 Pro/TOOLKIT 应用程序可以在 Pro/ENGINEER Wildfire 环境顺利运行,但用 Pro/ENGINEER Wildfire 版本的 Pro/TOOLKIT 的头文件和函数库生成的 Pro/TOOLKIT 应用程序则不能在 Pro/ENGINEER 2001 环境运行。这样,在使用低版本 Pro/TOOLKIT 的头文件和函数库生成的 Pro/TOOLKIT 应用程序,如果不使用高版本 Pro/TOOLKIT 新增的函数,可以直接使用而不一定非要重新编译和连接。本书给出的实例主要针对 Pro/ENGINEER 2001 和 Pro/ENGINEER Wildfire,在这两个版本中,Pro/TOOLKIT 新增的函数主要是 UI 对话框的操作函数和特征元素树的变化,对其他函数没有影响。除了第 4 章的实例是用于 Pro/ENGINEER 2001 之外,其余各章的实例均可用于 Pro/ENGINEER Wildfire 或更高的版本。实际上,即使以后的高版本有了变化,根据帮助文档作适当修改也是很容易的。

1.2 有关 Pro/TOOLKIT 的基本知识

Pro/TOOLKIT 采用面向对象的程序设计方法 (Object-Oriented Programming)。在 Pro/ENGINEER 和应用程序之间主要是通过特定的数据结构来传递信息,对应用程序来说这种数据结构并不是直接访问的,而只能通过 Pro/TOOLKIT 提供的函数来访问。在设计 Pro/TOOLKIT 程序时,要涉及到大量的由 Pro/TOOLKIT 函数库提供的 C 函数,正确理解和使用这些函数是非常重要的。

1.2.1 对象和动作

对象 (Object) 和动作 (Action) 是 Pro/TOOLKIT 中最基本的概念。

1. 对象及对象名

Pro/TOOLKIT 的对象实质是一种类型为结构体的数据, 结构体中的成员描述了该对象的属性。例如, 名为 ProFeature 的特征对象的结构体定义为:

```
typedef struct pro_model_item
{
    ProType type;
    int id;
    ProMdl owner;
} ProFeature;
```

结构体中的 type、id 和 owner 成员分别描述了该对象的类型、标识号和上级对象。

为了便于区别, 在 Pro/TOOLKIT 中所有对象的命名约定为: Pro+<对象名>, 其中对象名用英文单词表示, 第一个字母大写。如 ProFeature (特征对象)、ProSurface (曲面对象) 和 ProSolid (实体对象) 等。

Pro/TOOLKIT 定义的对象分成两类: 第一类对象本身是 Pro/ENGINEER 数据库中的一个项, 如 ProFeature 和 ProSurface; 另一类对象是抽象或临时对象, 如调用有关选择操作时用来保存选择结果的数据对象。

2. 动作及 Pro/TOOLKIT 函数

对特定的 Pro/TOOLKIT 对象执行的某种操作称为动作, 动作的执行是通过调用 Pro/TOOLKIT 函数库提供的 C 函数来实现了。与动作相关的 Pro/TOOLKIT 函数命名约定为: Pro+<对象名>+<动作>, 表示<对象名>和<动作>的英文单词首字母均用大写表示, 如: ProSolidRegenerate() (实体再生)、ProFeatureDelete() (特征删除)。

在 Pro/TOOLKIT 中, 用来表示动作的常用动词及意义为:

- Get 直接从 Pro/ENGINEER 数据库中获得信息, 如 ProSolidOutlineGet() (获得实体轮廓)
- Eval 提供简单的计算结果, 如 ProEdgeLengthEval() (计算边长)。
- Compute 提供复杂的计算结果, 如 ProSolidOutlineCompute() (实体轮廓计算)。

在本书的所有实例中, 凡是以“Pro”为前缀的函数都是由 Pro/TOOLKIT 函数库提供的库函数, 除此之外的函数均是用户的自定义函数。注意到这一点是很重要的, 因为对于所有的 Pro/TOOLKIT 函数, 只管正确调用而无须考虑具体的定义。在编译时只要包含了相应的头文件, 在连接时附加了相应的库文件, 就会顺利完成 Pro/TOOLKIT 应用程序的创建。

1.2.2 对象句柄

在 Pro/TOOLKIT 中每一个对象对应于一个结构体, 定义该结构体类型的一个具体的结构体变量称为对象句柄, 如: ProPart 表示零件对象, 用下面的形式声明:

ProPart NewPart;

NewPart 就是一个对象句柄。由于 ProPart 是一个指向结构体的指针，其定义为：

```
typedef struct sld_part* ProPart;
```

因此，NewPart 也是一个指向结构体的指针。实际上 Pro/TOOLKIT 中的对象句柄也可以完全理解为对象指针。

按照对象句柄的定义和使用方式分成两种类型：OHandle（Opaque Handle，非透明句柄）和 DHandle（Database Handle，数据库句柄）。非透明句柄是使用 Pro/ENGINEER 数据库结构的内存地址引用 Pro/ENGINEER 对象，这是一种最简单的方式。使用非透明句柄（或称非透明指针）Pro/TOOLKIT 应用程序只能获得该对象的内存地址，不能访问该结构体的具体成员，而使用数据库句柄则可访问该对象结构体的具体成员。在 Pro/TOOLKIT 中只是给出了非透明句柄的声明，没有给出结构体的定义，如：

```
typedef struct geom* ProSurface;
typedef void* ProMdl;
typedef struct curve_header* ProEdge;
typedef struct sld_part* ProSolid;
typedef struct entity* ProPoint;
typedef struct entity* ProAxis;
typedef struct entity* ProCsys;
typedef struct entity* ProCurve;
```

其目的是防止 Pro/TOOLKIT 应用程序访问结构体中的成员。

对于数据库句柄，在 Pro/TOOLKIT 给出了结构体的定义，因此在 Pro/TOOLKIT 应用程序中可以访问该结构体的成员，如：

```
typedef struct pro_model_item
{
    ProType type;
    int id;
    ProMdl owner;
} ProModelitem, ProGeomitem;
```

1.2.3 Pro/TOOLKIT 的函数原型和函数的返回值

Pro/TOOLKIT 函数库提供的库函数均有相应的 ANSI 函数原型，并在相应的头文件中定义，在调用函数之前必须将头文件包含在 C 或 C++ 程序中。本书列出了 Pro/TOOLKIT 函数库中常用函数的原型及相应的头文件，并给出了参数的类型和用法。其他函数可参见 Pro/TOOLKIT 的帮助文档。

大多数 Pro/TOOLKIT 函数的返回值类型为 ProError，用来表示调用函数执行状态。ProError 为枚举类型的数据，定义为：

```
typedef enum ProErrors
```

```

{
PRO_TK_NO_ERROR                = 0,
PRO_TK_GENERAL_ERROR           = -1,
PRO_TK_BAD_INPUTS              = -2,
PRO_TK_USER_ABORT              = -3,
PRO_TK_E_NOT_FOUND             = -4,
PRO_TK_E_FOUND                 = -5,
PRO_TK_LINE_TOO_LONG          = -6,
PRO_TK_CONTINUE                = -7,
PRO_TK_BAD_CONTEXT             = -8,
PRO_TK_NOT_IMPLEMENTED         = -9,
PRO_TK_OUT_OF_MEMORY           = -10,
PRO_TK_COMM_ERROR              = -11, /* communication error */
PRO_TK_NO_CHANGE               = -12,
PRO_TK_SUPP_PARENTS            = -13,
PRO_TK_PICK_ABOVE              = -14,
PRO_TK_INVALID_DIR             = -15,
PRO_TK_INVALID_FILE            = -16,
PRO_TK_CANT_WRITE              = -17,
PRO_TK_INVALID_TYPE            = -18,
PRO_TK_INVALID_PTR             = -19,
PRO_TK_UNAV_SEC                = -20,
PRO_TK_INVALID_MATRIX          = -21,
PRO_TK_INVALID_NAME            = -22,
PRO_TK_NOT_EXIST               = -23,
PRO_TK_CANT_OPEN               = -24,
PRO_TK_ABORT                   = -25,
PRO_TK_NOT_VALID               = -26,
PRO_TK_INVALID_ITEM            = -27,
PRO_TK_MSG_NOT_FOUND           = -28,
PRO_TK_MSG_NO_TRANS            = -29,
PRO_TK_MSG_FMT_ERROR           = -30,
PRO_TK_MSG_USER_QUIT           = -31,
PRO_TK_MSG_TOO_LONG            = -32,
PRO_TK_CANT_ACCESS             = -33,
PRO_TK_OBSOLETE_FUNC           = -34,
PRO_TK_NO_COORD_SYSTEM         = -35,
PRO_TK_E_AMBIGUOUS             = -36,
PRO_TK_E_DEADLOCK              = -37,
PRO_TK_E_BUSY                  = -38,
PRO_TK_E_IN_USE                = -39,
PRO_TK_NO_LICENSE              = -40,
PRO_TK_BSPL_UNSUITABLE_DEGREE  = -41,

```

```

PRO_TK_BSPL_NON_STD_END_KNOTS    = -42,
PRO_TK_BSPL_MULTI_INNER_KNOTS    = -43,
PRO_TK_BAD_SRF_CRV                = -44,
PRO_TK_EMPTY                      = -45,
PRO_TK_BAD_DIM_ATTACH             = -46,
PRO_TK_NOT_DISPLAYED              = -47,
PRO_TK_CANT_MODIFY                = -48,
PRO_TK_CHECKOUT_CONFLICT          = -49,
PRO_TK_CRE_VIEW_BAD_SHEET         = -50,
PRO_TK_CRE_VIEW_BAD_MODEL         = -51,
PRO_TK_CRE_VIEW_BAD_PARENT        = -52,
PRO_TK_CRE_VIEW_BAD_TYPE          = -53,
PRO_TK_CRE_VIEW_BAD_EXPLODE       = -54,
PRO_TK_UNATTACHED_FEATS           = -55,
PRO_TK_REGEN_AGAIN                = -56,
PRO_TK_DWGCREATE_ERRORS           = -57,

PRO_TK_APP_XS_CALLBACKS           = -93,
PRO_TK_APP_STARTUP_FAIL           = -94,
PRO_TK_APP_INIT_FAIL              = -95,
PRO_TK_APP_VERSION_MISMATCH       = -96,
PRO_TK_APP_COMM_FAILURE           = -97,
PRO_TK_APP_NEW_VERSION            = -98,
PRO_TK_APP_UNLOCK                 = -99

```

```
} ProError, ProErr;
```

其中最常用的返回值是 `PRO_TK_NO_ERROR` 表示函数调用成功。在程序中，通过检查函数的返回值不仅便于在调试时找出错误，更重要是可以避免在执行时出现异常情况，提高了程序的可靠性。

1.3 帮助文档的使用方法

Pro/TOOLKIT 提供了两种形式的帮助文档：位于 `\protoolkit` 目录下的 `tkuse.pdf` 文件和位于 `\protoolkit\protkdoc` 目录下的 `Pro/TOOLKIT APIWizard` 文件。下面主要介绍在 Windows 2000/NT/XP 环境下 Pro/TOOLKIT APIWizard 帮助文档使用方法。

1.3.1 浏览器环境要求

Pro/TOOLKIT APIWizard 支持 Netscape Navigator V4 和 Internet Explorer V5 及更高版本的浏览器。

使用 Internet Explorer 5 最好安装 Java2 插件，使用 Netscape Navigator 最好安装 Java