

MATLAB 实用指南系列

MATLAB 高级编程

苏金明 刘宏 刘波 编著

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



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

MATLAB 实用指南系列

MATLAB 高级编程

苏金明 刘 宏 刘 波 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书结合 MATLAB 最新版本 7.0 介绍面向对象编程技术、提高 MATLAB 编程效率的技巧、编译器 4.0, 以及 MATLAB 与外部程序 (包括 MATLAB 与 C/C++、Visual Basic、Visual C++、Excel、SPSS 以及硬件等) 的接口技术。还介绍 COM 生成器、Excel 生成器等接口工具。给出了一些实用性较强的实例。最后介绍与 MATLAB 界面设计有关的实用技巧。

本书内容丰富、新颖、实用, 是 MATLAB 高级编程方面不可多得的参考书。适合 MATLAB 中高级读者阅读。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目 (CIP) 数据

MATLAB 高级编程 / 苏金明, 刘宏, 刘波编著. —北京: 电子工业出版社, 2005.8

(MATLAB 实用指南系列)

ISBN 7-121-01376-2

I. M… II. ①苏…②刘…③刘… III. 计算机辅助计算—软件包, MATLAB 7.0—程序设计 IV. TP391.75

中国版本图书馆 CIP 数据核字 (2005) 第 093725 号

责任编辑: 王昌铭

印 刷: 北京东光印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 22 字数: 560 千字

印 次: 2005 年 8 月第 1 次印刷

印 数: 4000 册 定价: 35.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077。质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

近年来，国内流行多种科学计算软件。科学计算软件的使用，可以极大地提高科研人员的工作效率，可以更快更准确地完成计算方案的设计，可以在必要的时候用图形图像表示计算结果和描述运行机制。在数十种科学计算软件中，MATLAB 凭借其数值计算与图形可视化功能的完美结合，以及开放的设计理念，取得了很大的成功。目前，很多学校已经开设了这方面的课程，很多学生已经开始使用该软件完成论文设计。

我们从 1998 年开始接触 MATLAB，在深入学习和应用的基础上，于 2002 年、2004 年推出过 MATLAB 方面的图书。这些图书普遍受到读者的好评，其中《MATLAB 6.1 实用指南》被多所大学用作教材和教参。基于对 MATLAB 本身的理解和相关内容的积累，以及 MATLAB 目前在国内的发展和现状，经过长时间的酝酿，我们结合最新版本 7.0 编写了这套《MATLAB 实用指南系列》。

《MATLAB 实用指南系列》由《MATLAB 实用教程》、《MATLAB 工程数学》、《MATLAB 信号处理》、《MATLAB 图形图像》和《MATLAB 高级编程》等 5 本图书组成，其结构体系如图 0-1 所示。MATLAB 的主要内容可以概括为数值计算、图形图像和程序设计等 3 个方面。

《MATLAB 实用教程》结合大学课程介绍了这几个方面的基础和应用知识。其他图书在某个方面进行更深入的探讨。这样，整个系列具有清晰的层次感，并成为有机结合的整体。

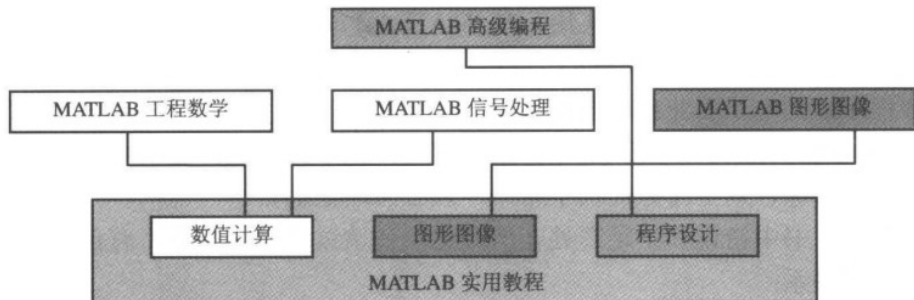


图 0-1 系列图书的结构体系

《MATLAB 实用教程》结合大学数学课程介绍高等数学、线性代数、计算方法、概率论与数理统计、最优化方法和偏微分方程数值解等的 MATLAB 实现方法；介绍 MATLAB 强大的图形功能，包括二维、三维图形的绘制，以及科学计算可视化、计算几何等；介绍 M 文件设计、界面设计、文件操作、编译和接口等程序设计方面的内容。每章安排了习题，便于巩固所学。

《MATLAB 工程数学》介绍几个与工程数学有关的工具箱，包括统计工具箱、最优化工具箱、偏微分方程数值解工具箱、样条工具箱和曲线拟合工具箱等。

《MATLAB 信号处理》介绍与信号处理有关的数字信号处理工具箱、阵列信号处理工具箱、时频分析工具箱和高阶谱分析工具箱。

《MATLAB 图形图像》介绍 MATLAB 图形图像的基础——句柄图形对象，以及图形着色、光照、材质、纹理映射、透明性和交互操作等高级绘图技术，然后介绍图像处理、虚拟

现实和地图制作等几个图形图像方面的工具箱。

《MATLAB 高级编程》由高级编程技术、程序编译、MATLAB 与外部程序接口和高级界面主题等几方面的内容组成，介绍一些非常实用的技术。其中 MATLAB 与外部程序接口包括 MATLAB 与 C/C++、Visual Basic、Visual C++、Excel、SPSS 以及硬件等接口，介绍了 COM 生成器、Excel 生成器等接口工具。该书还介绍了 MATLAB 7.0 全新的编译器。

总之，在编写过程中，我们希望在内容上尽可能地全面，既要满足大多数读者入门、应用的需要，又要满足中高级读者进阶的要求；既要有体系地介绍比较基础的知识，又要介绍图形可视化、工具箱内高级算法实现和高级接口技术等这样一些非常有价值、能反映 MATLAB 真实品质的内容。在我们力所能及的范围内，我们希望大家看到一个真实的 MATLAB。

关于本书

本书介绍 MATLAB 高级编程技术、编译、与外部程序接口和界面设计技巧等内容。

第 1~2 章介绍高级编程技术，包括面向对象编程和提高 MATLAB 编程效率的技巧。

第 3 章介绍 MATLAB 7.0 提供的最新编译器 4.0，该款编译器有比较大的改进。以前版本中运行时服务器的功能基本上包含到编译器中。

第 4~14 章全面地介绍 MATLAB 与外部程序接口技术，包括 MATLAB 与 C、Visual Basic、Visual C++、Excel、SPSS 以及硬件等接口；还介绍 COM 生成器、Excel 生成器等接口工具；给出了一些实用性较强的实例。

第 15 章介绍与 MATLAB 界面设计有关的实用技巧。

本书的特点是内容丰富、新颖、实用，是 MATLAB 高级编程方面不可多得的一本参考书。适合于 MATLAB 中高级读者阅读。

书中部分程序请到电子工业出版社网站下载。

本书第 7、9 章由刘宏编写，第 15 章由刘波编写，其余内容由苏金明编写。黄国明、周建斌、张莲花、向喜琼、刘玉珊等提供了帮助，衷心感谢他们！

由于水平有限，书中错误和不足之处在所难免，谨请读者批评指正！有任何问题，请通过电子邮件与我们联系：

苏金明 s_jm@263.net.cn

刘宏 liuh126@126.com

刘波 sclibo@mail.sc.cninfo.net

编著者

2005.6.16

目 录

第 1 章 面向对象编程	1
1.1 对象和类	1
1.1.1 面向对象编程的特点	1
1.1.2 MATLAB 的数据类层次	1
1.1.3 创建对象	2
1.1.4 调用对象的方法	2
1.2 在 MATLAB 中创建自己的类	2
1.2.1 MATLAB 类的方法集合	2
1.2.2 类目录	3
1.2.3 构造函数	3
1.2.4 设置和访问对象数据	4
1.2.5 类方法	4
1.2.6 引用和赋值	5
1.2.7 对象索引	6
1.2.8 识别对象	7
1.2.9 转换器方法	8
1.3 重载	8
1.3.1 运算符重载	8
1.3.2 函数重载	9
1.3.3 示例——polynom 类	9
1.4 继承	14
1.4.1 简单继承	14
1.4.2 多继承	15
1.4.3 示例——asset 类及其子类	15
1.5 组合	25
1.6 保存和装载对象	28
1.6.1 保存或载入时修改对象	28
1.6.2 示例——为 portfolio 类定义 saveobj 和 loadobj 方法	28
1.7 对象优先级	31
1.7.1 指定自定义类的优先级	31
1.7.2 在优先层次中定位	31
第 2 章 改善 MATLAB 的运行效率	32
2.1 改善运行的技巧	32
2.1.1 分析程序的运行状况	32
2.1.2 循环矢量化	33

2.1.3	数组的内存预分配	34
2.1.4	加速运行的其他方法	35
2.2	程序运行情况监测——Profiler	36
2.2.1	Profiler 的运行环境	36
2.2.2	使用 Profiler	36
2.2.3	监测图形用户界面的运行情况	37
2.2.4	从命令窗口监测语句	37
2.2.5	监测综述报表	37
2.2.6	监测详细报表	38
2.2.7	利用 Profiler 报表中的信息	40
2.2.8	改变 Profiler 的字体	40
2.3	使用 profile 函数	40
2.3.1	profile 函数的语法和使用步骤	40
2.3.2	profile 函数使用演示	41
2.3.3	对结果进行访问	42
2.4	有效使用内存	43
2.4.1	内存管理函数	43
2.4.2	节约内存的方法	43
第 3 章	编译器	45
3.1	概述	45
3.1.1	MATLAB 编译器 4.0 和以前版本的区别	45
3.1.2	MATLAB 编译器的基本功能	48
3.1.3	使用 MATLAB 编译器的基本步骤	49
3.1.4	MATLAB 编译器的局限性	50
3.1.5	关于运行时服务器(Runtime Server)	51
3.2	安装和注册	51
3.2.1	系统需求	52
3.2.2	安装	52
3.2.3	注册	53
3.2.4	几个问题	54
3.3	编译处理	55
3.3.1	MATLAB 编译器术语简介	55
3.3.2	输入和输出文件	57
3.3.3	应用程序的部署	58
3.3.4	使用 MCR	60
3.4	使用 mcc	61
3.4.1	命令概况	61
3.4.2	使用宏简化编译	62
3.4.3	使用路径名	62
3.4.4	使用束文件	63

3.4.5	使用打包器文件	63
3.4.6	使用注记	65
3.4.7	脚本文件	65
3.5	独立应用程序	65
3.5.1	C 独立应用程序	66
3.5.2	源代码只包括 M 文件	67
3.5.3	源代码包含 M 文件和 C/C++ 文件	68
3.6	库	68
3.6.1	C 共享库	68
3.6.2	C++ 共享库	72
3.6.3	MATLAB 编译器生成的接口函数	75
3.7	COM 和 Excel 组件	78
3.7.1	生成 COM 组件	78
3.7.2	生成 Excel 插件	79
第 4 章	MATLAB 调用动态链接库	80
4.1	库的载入和卸载	80
4.1.1	载入库	80
4.1.2	卸载库	80
4.2	获取库的信息	80
4.3	调用库函数	82
4.4	传递参数	82
4.5	数据转换	83
4.5.1	简单类型	84
4.5.2	枚举类型	85
4.5.3	结构	86
4.5.4	创建引用	89
4.5.5	引用指针	91
第 5 章	DDE (动态数据交换) 编程	92
5.1	DDE 概念和技巧	92
5.2	MATLAB 作为服务器	92
5.2.1	DDE 命名层次	93
5.2.2	MATLAB 主题和项目	93
5.3	MATLAB 作为客户	95
5.3.1	相关函数	95
5.3.2	DDE 提示链接	98
第 6 章	COM 编程	99
6.1	MATLAB COM 集成简介	99
6.1.1	概念和术语	99
6.1.2	支持的客户/服务器设置	100
6.1.3	注册控件和服务	102

6.2	MATLAB COM 客户支持	103
6.2.1	创建服务器进程	103
6.2.2	创建 ActiveX 控件	104
6.2.3	创建 DLL 组件的实例	107
6.2.4	创建 EXE 组件的实例	107
6.2.5	访问对象的接口	107
6.2.6	调用 COM 对象的命令	109
6.2.7	识别对象和接口	111
6.2.8	调用方法	112
6.2.9	对象属性	115
6.2.10	控件和服务器事件	121
6.2.11	编写事件处理程序	126
6.2.12	保存工作	129
6.2.13	释放 COM 接口和对象	130
6.2.14	识别对象	130
6.2.15	MATLAB 作为自动化客户示例	131
6.3	其他 COM 客户信息	132
6.3.1	使用 COM 集合	132
6.3.2	转换数据	132
6.3.3	将 MATLAB 用作 DCOM 客户程序	133
6.3.4	MATLAB COM 支持的局限性	133
6.4	MATLAB 自动化服务器支持	133
6.4.1	创建自动化服务器	134
6.4.2	连接已经存在的服务器	134
6.4.3	自动化服务器函数	135
6.4.4	MATLAB 自动化属性	137
6.5	其他自动化服务器信息	137
6.5.1	手工创建服务器	137
6.5.2	指定共享或独占服务器	137
6.5.3	将 MATLAB 用作 DCOM 服务器	138
第 7 章	MATLAB 与 C 接口	139
7.1	MATLAB 与 C 接口概述	139
7.2	C 语言的 MEX 文件	140
7.2.1	MEX 文件模式	140
7.2.2	第一个 MEX 文件	141
7.2.3	不同数据类型的传递	143
7.2.4	MEX 文件内存管理	149
7.2.5	MEX 文件调试	152
7.2.6	MEX 应用程序开发实例	153
7.3	C 引擎应用程序模式	157

7.3.1	MATLAB 引擎库函数介绍	157
7.3.2	MATLAB 引擎应用程序示例	159
7.3.3	在 Visual C++6.0 中编译、调试引擎应用程序	161
7.3.4	MATLAB 引擎应用程序实例开发	165
7.4	MAT 文件模式	168
7.4.1	MAT 文件格式介绍	168
7.4.2	MAT 文件示例	171
第 8 章	MATLAB 与 Visual Basic 接口	173
8.1	基于 OLE 的接口实现	173
8.1.1	实现 OLE 自动化	173
8.1.2	传递矩阵数据	177
8.1.3	传递字符串	182
8.1.4	处理工作空间的数据	183
8.1.5	传递和处理 MATLAB 函数	184
8.1.6	其他操作	186
8.2	基于 ActiveX 的接口实现	188
8.2.1	使用 ActiveX 控件	188
8.2.2	使用 ActiveX DLL	192
8.2.3	使用 ActiveX EXE	201
8.3	基于 COM 组件的接口实现	201
8.3.1	使用 COM 生成器	202
8.3.2	关于 MatrixVB	202
第 9 章	MATLAB 与 Visual C++ 接口	203
9.1	MATLAB 与 VC 混合编程接口	203
9.1.1	VC 与 MEX 文件示例一	203
9.1.2	VC 与 MEX 文件示例二	204
9.1.3	VC 与引擎应用程序	207
9.1.4	VC 与 MAT 文件	209
9.2	MCC	215
9.2.1	准备工作	215
9.2.2	建立独立应用程序示例	216
9.3	MATcom 与 Add-in	221
9.3.1	MATcom 安装与生成 Visual MATcom 工具条	221
9.3.2	m 文件转换示例——Test1	222
9.3.3	m 文件转换示例——Test2	225
9.3.4	matlib 数学库与 Mm 数据类型	228
第 10 章	MATLAB 与 Excel 接口	234
10.1	自动化链接	234
10.1.1	MATLAB 作为自动化客户端	234
10.1.2	MATLAB 作为自动化服务器端	236

10.2	Excel Link 插件	236
10.2.1	概述	236
10.2.2	安装和操作 Excel Link 插件	237
10.2.3	Excel Link 的函数	239
10.2.4	技巧和提示	240
10.2.5	Excel Link 使用实例	242
第 11 章	MATLAB 与 SPSS 接口	245
11.1	SPSS 软件	245
11.2	SPSS 中的对象	245
11.3	MATLAB 调用 SPSS	247
11.4	SPSS 调用 MATLAB	249
第 12 章	COM 生成器 (COM Builder)	252
12.1	创建 COM 生成器组件	252
12.1.1	创建工程	252
12.1.2	管理 M 文件和 MEX 文件	253
12.1.3	生成组件	254
12.2	利用 COM 生成器组件编程	254
12.2.1	给 COM 生成器组件对象添加方法和属性	254
12.2.2	给 COM 生成器组件对象添加事件	255
12.2.3	创建类实例	257
12.2.4	调用类实例的方法	259
12.2.5	处理 varargin 和 varargout 变量	259
12.2.6	在调用方法的过程中控制错误	260
12.2.7	修改标记	260
12.3	应用举例	261
12.3.1	创建 M 文件	261
12.3.2	创建 COM 生成器组件	262
12.3.3	在 Visual Basic 中使用 COM 组件	263
12.4	COM 组件的部署	268
12.4.1	组件打包	268
12.4.2	MCR	269
12.4.3	常见问题	269
12.5	深入 COM 生成器组件	270
12.5.1	COM 组件的兼容性	270
12.5.2	组件生成的内部过程	270
12.5.3	调用约定	271
12.5.4	组件注册	272
12.5.5	版本控制	275
12.5.6	数据转换	275
12.6	工具库	280

12.6.1	MWUtil 类	280
12.6.2	MWFlags 类	283
12.6.3	MWStruct 类	286
12.6.4	MWField 类	288
12.6.5	MWComplex 类	288
12.6.6	MWSparse 类	289
12.6.7	MWArg 类	290
12.6.8	3 个枚举类型	291
第 13 章	Excel 生成器 (Excel Builder)	292
13.1	创建 Excel 生成器插件	292
13.1.1	创建工程	292
13.1.2	管理 M 文件和 MEX 文件	293
13.1.3	生成组件	293
13.1.4	测试 VBA 模块	294
13.1.5	打包和发布组件	295
13.2	用 Excel 生成器组件编程	295
13.2.1	用 Excel 初始化生成器库	295
13.2.2	创建类的实例	296
13.2.3	调用类实例的方法	297
13.2.4	处理 varargin 和 varargout 参数	298
13.2.5	在调用方法的过程中控制错误	299
13.2.6	修改标记	299
13.3	魔方示例	302
13.3.1	一个输入的情况	302
13.3.2	使用多个文件和变量	304
13.4	谱分析示例	308
13.4.1	创建组件	308
13.4.2	将组件集成到 VBA 中	309
13.4.3	创建图形用户界面	310
13.4.4	保存和测试插件	315
13.4.5	打包组件	316
第 14 章	MATLAB 与硬件接口	317
14.1	MATLAB 串行通信接口简介	317
14.1.1	什么是 MATLAB 串行通信接口	317
14.1.2	支持的串行通信接口标准及平台	317
14.2	进一步了解串行接口	317
14.2.1	什么是串行通信	317
14.2.2	串行接口标准	318
14.2.3	串行接口信号及管脚分配	318
14.2.4	用串行电缆连接通信设备	318

14.2.5	查找所使用平台的串行接口信息	319
14.3	用串行接口进行通信	320
14.3.1	一个简单的例子	320
14.3.2	通信步骤及相关函数介绍	321
14.4	应用实例	325
14.4.1	实例 1——与示波器通信	325
14.4.2	实例 2——拆分输入数据	327
14.4.3	实例 3——计算机与计算机通信	328
14.5	串口 I/O 相关函数表	329
第 15 章	界面设计技巧	331
15.1	使用外部控件	331
15.2	控件的选择、移动、缩放和复制	335
15.3	控件标题文本的换行	336
15.4	将 MATLAB 绘制的图形显示到 VB 界面上	337

第 1 章 面向对象编程

本章介绍如何在 MATLAB 中实现面向对象编程。面向对象编程强调使用类和对象来编写程序。使用类和对象，可以向 MATLAB 添加新的数据类型和新的运算符。类描述对象的结构，并定义运算符的类型和可以操作对象的函数。对象是某个类的实例。

1.1 对象和类

可以把类看作是具有一定行为的新的数据类型。例如，Polynomial 类重新定义相加运算符 (+) 以后，可以对多项式完成正确的相加运算。运算符是类的一种方法。也可以把类看作是可作为单一实体的新项目。例如 arrow (箭头) 对象，它可以显示在图形上，并像句柄图形对象那样具有属性。可以通过简单地实例化 arrow 对象来创建箭头，可以为对象指定用于数据存储的 MATLAB 结构，并创建 M 文件的类目录来将类添加到 MATLAB 环境中。这些 M 文件基于对象进行操作并提供类的方法。类目录中还可以定义应用于对象的不同 MATLAB 运算符，包括算术运算、索引引用等。可以在类中重新定义内部运算符，即进行运算符重载。

1.1.1 面向对象编程的特点

使用设计良好的类时，面向对象编程可以显著提高代码的重用性，并使程序更易于维护和扩展。用类和对象编程与一般的结构化编程有下面一些主要区别。

- (1) 能实现函数和运算符重载。
- (2) 可以覆盖已经存在的 MATLAB 函数表示的方法。
- (3) 把用户定义的对象作为参数调用函数时，MATLAB 首先进行检查，看是否有定义对象类的方法。如果有，MATLAB 调用它。
- (4) 可以实现数据的封装。
- (5) 对象属性在命令行中不可见，只能用类方法访问它们。
- (6) 可以创建类的继承层次。子类可以从一个父类 (单继承) 和多个父类 (多继承) 那里进行继承。利用继承，可以共享父函数。
- (7) 可以用组合创建类，其中一个对象包含另一个对象。这适用于一种对象是另一种对象的一部分的情况。

1.1.2 MATLAB 的数据类层次

在面向对象编程中，所有 MATLAB 数据类都作为类设计成函数。图 1-1 显示了 MATLAB 中定义的 15 种基本数据类型 (或类)。可以通过扩展类层次来给 MATLAB 添加一个以结构类继承的用户类。创建的所有类都基于结构。

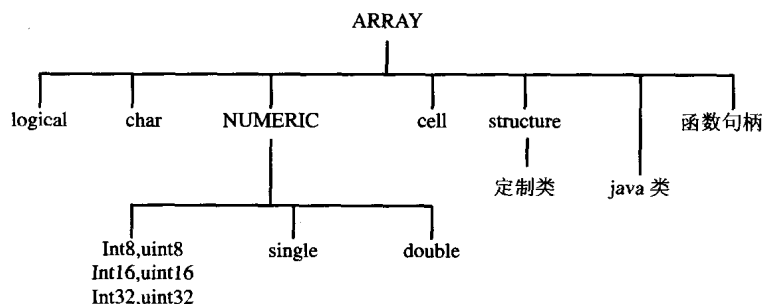


图 1-1 MATLAB 的数据类层次

1.1.3 创建对象

通过调用类的构造函数并传递合适的输入参数，可以创建对象。在 MATLAB 中，构造函数的名称与类名相同，例如，下面的语句

```
p=polynom([1 0 -2 -5]);
```

创建一个 `polynom` 类的对象 `p`。创建一个 `polynom` 对象以后，就可以用 `polynom` 类中定义的方法来操作对象。

1.1.4 调用对象的方法

类方法为 `M` 函数，该函数将一个对象作为输入参数之一。特定类的方法必须位于该类的类目录中（`@classname` 目录）。这是 MATLAB 查找类方法的第 1 个位置。

调用对象方法的语法与函数调用类似。通常类似于下面的格式：

```
[out1,out2,...]=methodName(object,arg1,arg2,...)
```

例如，自定义类 `polynom` 具有 `char` 方法，该方法将 `polynom` 对象转换为字符串并返回该字符串。下面的语句调用 `polynom` 对象 `p` 的 `char` 方法。

```
s=char(p);
```

使用 `class` 函数，可以确定返回值 `s` 是否字符串。使用 `methods` 命令，可以生成为类定义的所有方法的列表。

1.2 在 MATLAB 中创建自己的类

1.2.1 MATLAB 类的方法集合

设计 MATLAB 类时，应该提供一个使类能够在 MATLAB 环境中稳定运行的标准的方法集合。利用这些方法，应该可以实现创建类、设置属性以及对对象的引用、赋值、索引和转换等操作。表 1-1 列出了 MATLAB 类中包括的基本方法。

表 1-1 MATLAB 类中的基本方法

方 法	描 述
类的构造函数	创建一个类对象
<code>display</code>	MATLAB 是否显示对象的内容
<code>set</code> 和 <code>get</code>	设置和获取类属性

续表

方 法	描 述
subsref 和 srbsasgn	用户对对象的索引引用和赋值
end	在使用了对象的索引表达式中支持 end 语法, 如 A(1:end)
subsindex	支持在索引表达式中使用对象
转换器方法 (如 double,char)	将对象转换为一种 MATLAB 数据类型

1.2.2 类目录

1. 安装类目录

定义类方法的 M 文件集中在一个目录中, 该目录名由类名前加字符 “@” 组成。例如: 本章中用到的一个示例是与单变量多项式有关的一个类。类名和构造函数的名称都是 `polynom`, 定义 `polynom` 类的 M 文件位于 `@polynom` 目录下。类目录是 MATLAB 搜索路径上目录的子目录。例如, 新的 `@polynom` 目录可以是 MATLAB 工作目录的子目录或已经添加到搜索路径中的个人目录。

2. 添加类目录到 MATLAB 路径

创建类目录以后, 需要更新 MATLAB 路径, 这样, MATLAB 才能找到类的源文件。类目录不应该直接把父目录添加到 MATLAB 路径。例如, 如果 `@polynom` 类目录的位置是 `c:\my_classes\@polynom`, 用 `addpath` 命令把类目录添加到 MATLAB 路径中:

```
addpath c:\my_classes;
```

1.2.3 构造函数

构造函数是对类进行操作时第一个运行的函数, 它常常用于对象数据的初始化。可以通过调用构造函数并给它传递合适的输入参数来创建类。在 MATLAB 中, 构造函数与类具有相同的名称。例如: 声明 `P=polynom(10 -2 -5)`; 创建一个属于 `polynom` 类的名为 `P` 的对象。一旦创建了 `polynom` 对象, 就可以用类定义的方法操作对象。

特定类的 @ 目录必须包含一个作为类的构造函数的 M 文件, 除了 @ 前缀和 .m 扩展名以外, 构造函数的名称与定义类名目录的名称相同。构造函数通过初始化数据结构和实例化类对象来创建对象。

构造函数必须运行一定的函数。这样, 对象才能在 MATLAB 环境中正确运行。通常, 根据输入参数的个数和类型的不同, 构造函数有 3 种, 分别对应于没有输入参数、同一个类的一个对象作为输入参数和某种类型的数据作为输入参数的情况。

1. 没有输入参数

如果没有输入参数, 构造函数将创建一个默认对象。因为没有输入参数, 没有数据可用于创建对象, 所以用空值或默认值简单地初始化对象的数据, 调用类函数实例化对象, 并返回对象作为输出参数。在下面两种情况下需要使用本语法:

- 在工作空间中载入对象时, `load` 函数调用没有参数的构造函数。
- 创建对象数组时, MATLAB 向数组中添加对象。

2. 对象输入参数

如果参数列表中的第一个输入参数是同一类的对象，构造函数将简单地返回对象，使用 `isa` 函数确定参数是否为该类对象。

3. 数据输入参数

如果有输入参数，但没有同一类的对象，则构造函数使用输入的数据，常用于使用 `varargin` 输入参数有开关控制的程序中。

1.2.4 设置和访问对象数据

需要为类编写方法来提供访问对象数据的途径。`set` 和 `get` 方法提供了访问对象数据的便捷途径。

1.2.5 类方法

1. 在 MATLAB 中设置类方法

类方法实际上是 M 文件函数，只是该函数的输入参数是对象。类方法必须位于该类的类目录（`@class_name` 目录）中。这是 MATLAB 查找类方法的第一个位置。调用类方法的语法与调用函数的方法相同，一般地，它与下面的语句行类似：

```
out1 ,out2,...=method_name(object,arg1,arg2,...)
```

例如，名为 `polynom` 的类有一个 `char` 方法。该方法把 `polynom` 对象转换为字符串，并返回该字符串。下面调用 `polynom` 对象 `p` 的 `char` 方法：

```
s=char(p);
```

使用 `class` 函数，可以确认返回值 `s` 是一个字符串。

```
class(s)
ans =
    char
s
s =
    x^3-2*x-5
```

可以使用 `methods` 方法来生成类中的所有方法的列表。

2. 私有方法

私有方法只能被类中的其他方法调用。将有关的 M 文件放到一个 `@class_name` 目录的 `private` 子目录中，可以定义私有方法。例如 `@class_name\private\update_obj.m`。 `update_obj` 方法的使用范围局限于 `class_name` 类。这意味着 `update_obj` 可被 `@class_name` 目录中定义的方法调用，但它不能被 MATLAB 命令行或目录路径以外的方法，包括父方法所调用。私有方法和私有函数之间的区别在于，私有方法有一个输入参数是对象，而私有函数却没有。

3. 调试类方法

可以采用与调试其他 M 文件相同的方法，用 MATLAB 的调试命令调试对象的方法。唯一的区别在于需要在方法名前添加类的目录，就像下面例子中用 `dbstop` 显示的那样。

```
dbstop @polynom/char
```

调试类方法时，已经获得了为类定义的所有方法，包括继承方法、私有方法和私有函数。