

信息科学技术学术著作丛书

IDL程序 设计与应用

卞小林 邵 芸 编著

*Interactive Data Language
Programming and Application*



科学出版社

IDL 程序设计与应用

Interactive Data Language
Programming and Application

卞小林 邵 芸 编著

科学出版社

北 京

内 容 简 介

本书以 IDL 8.2 为基础,系统介绍利用 IDL 进行程序设计的基础知识与程序设计方法。全书 10 章,主要讲述 IDL 概述、语法基础、面向过程的程序设计、面向对象的程序设计、输入与输出、高效程序设计、图形用户界面设计、图形图像程序设计、应用程序发布与部署和应用程序设计实践等内容。全书以面向过程的程序设计为切入点,从编写简单的程序开始,循序渐进,由面向过程到面向对象,逐步深入。

本书可作为高校计算机、地理信息系统、遥感、图像处理及相关专业本专科生和研究生的教材,也可以供从事计算机相关工作的科技工作者和广大编程爱好者自学与参考。

图书在版编目(CIP)数据

IDL 程序设计与应用/卞小林,邵芸编著. —北京:科学出版社,2019.6
(信息科学技术学术著作丛书)

ISBN 978-7-03-061248-9

I. ①I… II. ①卞… ②邵… III. ①软件工具-程序设计
IV. ①TP311.56

中国版本图书馆 CIP 数据核字(2019)第 094205 号

责任编辑:魏英杰 / 责任校对:郭瑞芝

责任印制:吴兆东 / 封面设计:铭轩堂

科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencecp.com>

北京中石油彩色印刷有限责任公司 印刷

科学出版社发行 各地新华书店经销

*

2019 年 6 月第 一 版 开本:720×1000 1/16

2019 年 6 月第一次印刷 印张:18

字数:363 000

定价:108.00 元

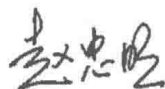
(如有印装质量问题,我社负责调换)

序

IDL 自 1977 年发布以来,已经广泛应用于地球科学、海洋科学、医学、天文学、遥感工程、信号处理、科研教育等众多领域。在国外,IDL 已经被设为大学的标准课程,其功能和应用效果完全可以和 MATLAB 等其他同类科学计算应用软件相媲美。在国内,IDL 还处在推广和应用阶段。随着 IDL 的应用和市场的广泛推进,越来越多的人将成为 IDL 的用户。

近年来,在国家自然科学基金、国家高技术研究发展计划、中国科学院知识创新工程等项目的支持下,中国科学院遥感与数字地球研究所邵芸研究员及其团队,围绕微波遥感开展了机理、方法与应用研究,并基于 IDL 研发了多个应用软件。作者在多年微波遥感应用和 IDL 相关的研发实践基础上,针对 IDL 程序设计中面向过程的程序设计与面向对象的程序设计,系统介绍 IDL 程序设计与应用。该书着力于使读者对 IDL 有一个较为系统、全面的认识,用容易理解的方法讲清楚基本概念和基本方法,减少罗列 IDL 语法中的各种烦琐细节,使读者既能利用 IDL 快速编写自己的算法,又能利用 IDL 提供的过程与函数快速构建自己的应用。

该书出版之际,我非常乐意将其推荐给广大读者。工欲善其事,必先利其器。希望该书能帮助更多的初学者顺利迈入 IDL 程序设计的大门,让已经具有编程基础的读者快速掌握和使用 IDL,促进 IDL 的发展。同时,期待该书对加速科学研究与软件开发进程发挥特有的作用。



2018 年 5 月

前 言

交互式数据语言(interactive data language, IDL)是第四代计算机语言,具有功能丰富、表达能力强、使用灵活方便、目标程序高效与可移植性好等优点,是进行科学数据分析、可视化表达和跨平台应用开发的理想工具。IDL 不是纯粹的面向过程的程序设计语言,它既可以编写面向过程的程序,也可以编写面向对象的程序。

作者参阅了国内为数不多的 IDL 程序设计相关书籍,并结合多年在微波遥感应用和 IDL 相关的研发实践经历,以及我国的大学程序设计课程开设情况,设计并确定了本书的体系结构。全书以面向过程的程序设计为切入点,从编写简单的程序开始,循序渐进,由面向过程到面向对象,逐步深入。读者可以根据需要选择学习 IDL 面向过程的程序设计和面向对象的程序设计,提高解决实际问题的能力。

程序设计是一门实践性很强的课程,要学好程序设计,仅仅看懂是不够的,必须多编程,多上机实践。许多读者在学习 IDL 之前已学过其他程序设计语言(如 C 语言),不同程序设计语言语法各异,但算法思想是相通的。多上机实践对掌握多门程序设计语言往往起到事半功倍的效果。

为了使读者较好地掌握 IDL 程序设计,本书根据作者 IDL 程序设计使用心得,从不同角度精心设计示例程序来增加对主要知识的理解。读者在学习和使用示例代码时,可以手工输入,也可以使用示例代码文件。所有示例代码均在 IDL 8.2 下编写,并且在 Win64 平台上测试通过。如果没有特殊说明,其中大部分程序与其他 IDL 版本兼容。示例程序按章节建立工程,示例数据存放在 Data 文件夹中。在使用过程中,将 IDLprogram 文件夹直接拷贝到硬盘中。对部分涉及示例数据文件名的程序,将文件路径改为示例数据所在路径即可正确执行。

本书的出版得到国家自然科学基金重点项目“可控环境下多层介质目标微波特性全要素测量与散射机理建模”(41431174)资助,作者在此表示诚挚的谢意。同时,感谢中国科学院遥感与数字地球研究所董庆研究员、张风丽博士、谢酬博士、宫华泽博士、田维博士、张春燕博士等同事与国家海洋技术中心王世昂博士提供的帮助。在编写过程中,本书得到中国科学院遥感与数字地球研究所、中科卫星应用德清研究院(浙江省微波目标特性测量与遥感重点实验室)等部门和单

位的支持，在此一并致谢。

限于作者水平，书中不足在所难免，敬请读者不吝指正。

作 者

2018 年 12 月

目 录

序

前言

第 1 章 IDL 概述	1
1.1 IDL 简介	1
1.2 IDL 的特点	1
1.3 IDL 集成开发环境	2
1.3.1 工程与工作空间	3
1.3.2 启动工作台	3
1.4 IDL 程序的编写与运行	4
1.4.1 简单的 IDL 程序介绍	4
1.4.2 IDL 程序编写	9
1.4.3 IDL 程序断点与调试	12
1.4.4 帮助系统	13
第 2 章 语法基础	16
2.1 数据类型	16
2.2 常量与变量	17
2.2.1 常量	17
2.2.2 变量	19
2.2.3 系统变量	20
2.2.4 变量创建	22
2.2.5 变量存储	23
2.2.6 变量类型转换	26
2.3 数组	27
2.3.1 数组创建	27
2.3.2 数组存储	30
2.3.3 数组使用	32
2.3.4 数组运算	32
2.3.5 数组函数	35
2.4 字符串	38

2.4.1	字符串创建	38
2.4.2	字符串连接与转换	39
2.4.3	字符串函数	40
2.5	结构体	42
2.5.1	结构体创建与访问	42
2.5.2	结构体函数	44
2.6	指针	44
2.6.1	指针创建与访问	44
2.6.2	指针函数	46
2.7	对象	47
2.7.1	对象创建与调用	47
2.7.2	对象函数	50
2.8	链表	51
2.8.1	链表创建与访问	51
2.8.2	链表操作	52
2.9	哈希表	53
2.9.1	哈希表创建与访问	53
2.9.2	哈希表操作	54
2.10	表达式	55
2.10.1	数值型表达式	55
2.10.2	字符型表达式	56
2.10.3	关系型表达式	56
2.10.4	逻辑型表达式	57
2.10.5	条件表达式	57
2.10.6	赋值表达式	58
2.10.7	运算符的优先级	58
2.11	编译规则	59
第 3 章	面向过程的程序设计	61
3.1	算法概述	61
3.2	语句	62
3.2.1	基本语句	62
3.2.2	控制语句	63
3.2.3	异常处理语句	65
3.3	顺序结构程序设计	70

3.4 选择结构程序设计	70
3.4.1 IF 语句	71
3.4.2 CASE 语句	72
3.4.3 SWITCH 语句	74
3.5 循环结构程序设计	75
3.5.1 FOR 语句	75
3.5.2 FOREACH 语句	77
3.5.3 WHILE 语句	78
3.5.4 REPEAT 语句	78
3.6 过程与函数	79
3.6.1 过程	79
3.6.2 函数	80
3.7 参数与关键字	81
3.7.1 参数	81
3.7.2 关键字	81
3.7.3 参数与关键字传递	82
3.7.4 参数与关键字检测	83
3.8 程序调用	86
3.8.1 程序调用的形式与方式	86
3.8.2 程序的嵌套调用	87
3.8.3 程序的递归调用	88
3.9 全局变量与局部变量	89
3.9.1 局部变量	89
3.9.2 全局变量	90
3.10 程序实例分析	91
第 4 章 面向对象的程序设计	98
4.1 面向对象的程序设计概述	98
4.1.1 面向对象的概念	98
4.1.2 面向对象程序设计的特征	99
4.2 IDL 类	100
4.2.1 类的创建和调用	100
4.2.2 构造函数和析构函数	102
4.3 类的继承	104
4.4 类的多态性	110

第 5 章 输入与输出	113
5.1 数据输入与输出	113
5.2 文件输入与输出	115
5.2.1 文件打开与关闭	117
5.2.2 文件的读写	121
5.3 常用文件格式读写	127
5.3.1 TIFF 格式	127
5.3.2 Shapefile 格式	129
5.3.3 NetCDF 格式	133
5.3.4 XML 格式	135
5.3.5 Excel 格式	139
第 6 章 高效程序设计	146
6.1 时间优化	146
6.2 空间优化	152
第 7 章 图形用户界面设计	156
7.1 常用组件	156
7.1.1 BASE 组件	156
7.1.2 文本组件	159
7.1.3 按钮组件	161
7.1.4 图形显示组件	163
7.1.5 列表选择组件	164
7.1.6 对话框组件	165
7.2 应用程序界面设计	169
7.2.1 用户图形界面设计方法	169
7.2.2 组件的控制	169
7.2.3 组件事件和事件处理	171
第 8 章 图形图像程序设计	176
8.1 直接图形法	176
8.1.1 直接图形法显示设备	176
8.1.2 直接图形法显示模式	177
8.1.3 直接图形法显示窗口	177
8.1.4 直接图形法显示区域	179
8.1.5 直接图形法显示字体	179
8.1.6 直接图形法图形绘制	181

8.1.7 直接图形法图像显示	190
8.2 对象图形法	193
8.2.1 对象图形法层次结构	194
8.2.2 对象图形法显示模式	195
8.2.3 对象图形法显示窗口	195
8.2.4 对象图形法显示区域	197
8.2.5 对象图形法显示字体	202
8.2.6 对象图形法图形绘制	202
8.2.7 对象图形法图像显示	216
8.2.8 对象图形法对象交互	219
8.3 直接图形法与对象图形法的比较	224
8.4 图像处理	224
8.4.1 图像类型转换	225
8.4.2 图像几何变换	228
第 9 章 程序发布与部署	234
9.1 IDL 程序发布	234
9.1.1 SAV 文件	234
9.1.2 EXE 文件	236
9.2 IDL 程序部署	237
9.2.1 Runtime 方式	237
9.2.2 虚拟机方式	237
第 10 章 应用程序设计实践	238
10.1 简易计算器程序设计	238
10.2 辐射计 AMSR-E 风速可视化	244
10.3 辐射计 AMSR-E 产品动画制作	254
10.4 SAR 图像分块并行处理	262
10.5 SAR 风场反演结果可视化	265
参考文献	276

第 1 章 IDL 概 述

1.1 IDL 简 介

IDL 是应用程序开发和数据分析与可视化表达的理想工具。IDL 作为第四代面向矩阵运算的计算机语言，语法简单，拥有丰富的分析工具包，采用高速的图形显示技术，是集可视化、交互数据分析、大型商业开发为一体的高级集成环境，可以使用户快捷有效地实现数据处理、科学研究与商业开发。

IDL 集成了各种工程所需的可视化和分析工具，自带大量的功能函数，用户使用少量的代码就能快速实现需要的功能，加速科学研究与软件开发进程。由于其强大的功能和独特的特点，IDL 可以满足众多领域的三维数据可视化、数值计算、三维图形建模、科学数据处理与分析等需求。

IDL 自 1977 年商业版正式发布以来，版本不断升级。本书以 Windows 操作系统平台下 IDL 8.2 为例介绍 IDL 程序设计与应用。

1.2 IDL 的特点

一种程序设计语言之所以能存在和发展，总是具有自身的优势和与其他语言不同的特点。IDL 的主要优势和特点如下。

1. 应用开发

IDL 语法简单，与其他常用的编程语言相比有很多相似之处，却又没有过多的严格限制，容易上手，初学者可以快速建立自己简单的应用程序。IDL 拥有面向对象的编程语言技术，用户可以完成复杂的功能和应用。

IDL 同时支持多种工作方式，不同的用户可以根据需要配合使用“命令交互式”、“程序执行”和“菜单操作”。IDL 程序编写、编译和执行时提供即时的交互性反馈。

IDL 具有 Windows 风格的图形用户界面，多功能集成开发环境、项目管理器和图形用户界面设计等多种应用程序环境和开发工具，可以运行在 Windows、

Macintosh、UNIX 等多种操作平台，方便进行跨平台开发应用程序。

IDL 提供了与 Visual C、Visual Basic 和 Fortran 等多种语言的标准接口工具，以及 ActiveX 接口，可以轻松实现 IDL 与其他语言的互相集成。IDL 虚拟机和 Runtime 两种方式可以方便地向团队和个人发布 IDL 程序，并提供便捷的跨平台程序发布工具。

2. 数据快速处理和分析与可视化表达

IDL 支持面向矩阵运算与并行计算，对整个矩阵的处理无需循环，数据量越大，速度优势越明显，可以简化交互分析、减少编程时间。

IDL 具有灵活的输入/输出能力，提供了大量的数据读写工具，支持常见数据格式的直接读写，可以读取任何类型的自定义格式数据，并且可以查看支持的文件格式细节。

IDL 支持基于 Open GL 的硬件加速图形技术，可以快速实现二维与多维图形绘制、体积可视化、图像显示与动画，并且可以即时观察程序的计算结果。

IDL 提供许多数值和统计分析软件包，有大量功能丰富的命令、函数和程序模块，并且采用多进程设计，能充分发挥多处理器效能，使数据分析和可视化更加简单、灵活、快捷、方便。IDL 的智能工具可以自订业务、数据分析流程，创建可视化环境。

1.3 IDL 集成开发环境

IDL 集成开发环境(integrated development environment, IDE)又称工作台(Workbench)，是进行代码编写、管理、编译、调试和运行的图形化操作环境。自 IDL 7.0 版本起，IDL 工作台基于全新的 Eclipse 框架运行。因此，在各种操作系统(如 Windows、Linux、Macintosh 或 Solaris 等)下均具备同样的操作界面和快捷键，便于在不同操作系统的平台下进行程序开发。IDL 8.2 支持的平台如表 1-1 所示。

表 1-1 IDL 8.2 支持的平台

平台	硬件	操作系统
Windows	Intel/AMD 32-bit Intel/AMD 64-bit	Windows
Macintosh	Intel 64-bit	OS X
UNIX	Intel/AMD 32-bit Intel/AMD 64-bit	Linux
	SPARC 64-bit Intel/AMD 64-bit	Solaris

1.3.1 工程与工作空间

工程是过程(pro)与函数(function)的集合,可以通过相互调用实现复杂的功能。IDL 的工程项目使用 Project 来组织和管理 IDL 源代码,以及工程资源文件。每个工程是当前工作空间的路径或工作空间的子路径,源代码存放在对应工程路径下,为工程使用中的文件搜索、书签创建等带来便利。

IDL 下的工作空间是包含一个或多个工程的空间,可以包含源代码文件和资源文件。每次启动工作台时,都会提示选择工作空间。IDL 工作台可以创建多个工作空间,但当前工作台只能有一个活动的工作空间。如果需要切换工作空间,可以通过“文件”菜单项下“切换工作空间”进行切换。

1.3.2 启动工作台

启动 IDL 工作台有多种方式,下面以 Windows 7 操作系统下的 IDL 8.2 工作台为例介绍 IDL 工作台启动。点击“开始”-“所有程序”-“ENVI5.0”-“IDL”, IDL 显示如图 1-1 所示界面,设置工作空间位置。



图 1-1 选择工作空间界面

选择“确定”进入 IDL 工作台环境,主要由菜单栏、工具栏、项目资源管理器、代码编辑区域,控制台和状态栏组成,各组件都可以根据需要任意改变大小和位置,如图 1-2 所示。为避免出现乱码现象,需要设置当前工作台的参数,选

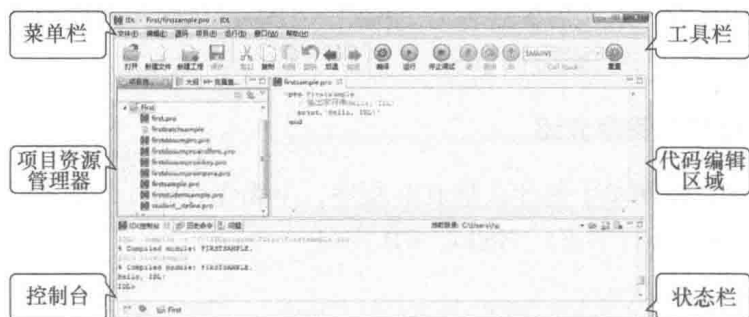


图 1-2 工作台界面

择菜单“窗口”-“首选项”，然后选择“常规”-“工作空间”，将“文本文件编码”设置为 GB2312，如图 1-3 所示。

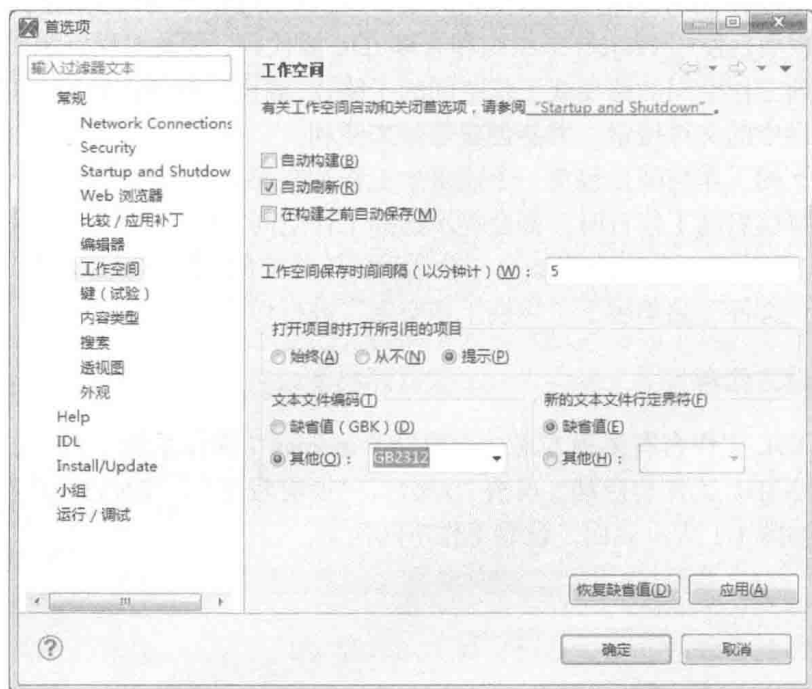


图 1-3 文本编辑编码界面

1.4 IDL 程序的编写与运行

所谓程序，就是一组计算机能识别和执行的指令。通常把编写程序的过程称为程序设计。程序设计好之后可以长期保存，并根据需要对其进行编辑、修改和反复调用。

程序的运行过程一般分为建立、编辑、调试、编译和运行等。IDL 编写的程序文件称为源程序，对应的扩展名为“`.pro`”。IDL 提供多种程序操作模式，用户可以根据需求组织与管理程序代码。

1.4.1 简单的 IDL 程序介绍

为了使读者能初步了解什么是 IDL 程序，下面先介绍几个简单的程序。

例 1.1 输出一行字符：“Hello, IDL!”。

```
pro Firstsample
    ;输出字符串 Hello, IDL!
```

```
Print, 'Hello, IDL!'
```

```
end
```

本程序是一个过程文件，作用是输出指定字符串。其中“pro”作为过程必不可少的部分，它是一个可执行语句的开始，后空一格接过程名，中间部分为可执行语句，最后必须以“end”结束。本过程只有一个输出语句，print 是 IDL 的输出过程。在 IDL 程序中，语句不区分大小写，没有结束符，分隔符为“;”。IDL 的注释符为“;”，该行注释符之后对应的所有内容不参与编译运行。注释内容可以为注释语句，也可以写在命令行的末端，用于对该行内容进行说明。将上述代码保存为程序文件“firstsample.pro”，注意此处文件名必须与过程名一致，不区分大小写。在命令行里键入 firstsample，示例如下。

```
IDL> firstsample
```

```
% Compiled module: FIRSTSAMPLE. ;如果已经编译，则无此行内容
Hello, IDL!
```

例 1.2 求两数 a 和 b 之和并输出。

示例 1:

;无输入参数求和过程

```
pro Firstdosumpro
```

```
;按提示信息依次输入两个数，分别保存到变量 a 与变量 b 中
```

```
read, a, b, prompt='请依次输入数 a 与数 b: ' ;read 为 IDL 输入过程
```

```
print, 'a + b = ', a+b
```

```
end
```

本程序为一个过程文件，无输入参数，其作用是按提示信息输入两个数并输出两数之和。在 IDL 程序中，变量名不区分大小写，变量不需要预先定义，数据类型可以在运行过程中更改。程序运行示例如下。

```
IDL> firstdosumpro
```

```
请依次输入数 a 与数 b: 3, 4
```

```
a + b = 7.00000
```

示例 2:

;有输入参数求和过程

```
pro Firstdosumproinpara, a, b
```

```
print, 'a + b = ', a+b
```

```
end
```

本程序为一个过程文件，有输入参数，其作用是输出两个输入参数之和。在 IDL 程序中，参数用于调用数据之间的传递，多个参数间以“,” 隔开。程序运行示例如下。

```
IDL>firstdosumproinpara,3,4
```

```
a + b =7
```

示例 3:

;有输入关键字求和过程

```
pro Firstdosumproinkey,a=a,b=b
```

```
    print,'a + b = ',a+b
```

```
end
```

本程序为一个过程文件，有输入关键字，其作用是输出两个输入关键字对应的数据之和。在 IDL 程序中，关键字用于调用数据之间的传递，多个关键字间以“,” 隔开，关键字一般为可选项，且顺序是任意的。程序运行示例如下。

```
IDL> firstdosumproinkey,a=3,b=4;或 IDL> firstdosumproinkey,  
b=4,a=3
```

```
a + b =7
```

示例 4:

;有输入参数求和函数与函数调用

```
function Firstdosumfunc,a,b
```

```
    ;返回 a 与 b 之和
```

```
    return,a+b
```

```
end
```

```
pro Firstdosumproandfunc
```

```
    read,a,b, prompt='请依次输入数 a 与数 b : '
```

```
    ;调用有输入参数求和函数 Firstdosumfunc
```

```
    c=Firstdosumfunc(a,b)
```

```
    print,'a + b = ',c
```

```
end
```

本程序包含一个函数与一个主过程，其作用是通过主过程 Firstdosumproandfunc 调用函数 Firstdosumfunc 实现输入两数，并输出两数之和。其中“function”作为函数必不可少的部分，是一个可执行语句的开始，后空一格接函数名，中间部分为可执行语句，最后必须以“end”结束。函数与过程不同；函数可执行语句必须具有返回语句，用“return”表示。函数调用运行示例如下。

```
IDL> c=firstdosumfunc(3,4);调用函数 firstdosumfunc, 将结果  
;返回给 c
```