



基于VBA的ArcObjects编程

——面向任务的方法

(第二版)

● [美] Kang-tsung Chang 著
● 陈健飞 蔡 砥 黄婷薇 译

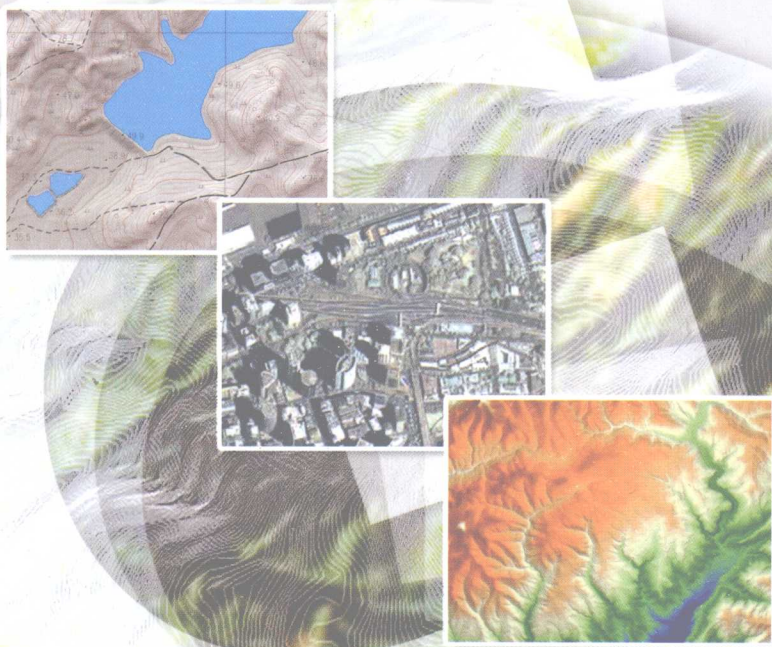


Image © 2008 TerraMetrics



科学出版社
www.sciencep.com

10

基于 VBA 的 ArcObjects 编程

——面向任务的方法

(第二版)

〔美〕Kang-tsung Chang 著

陈健飞 蔡 砥 黄婷薇 译

科 学 出 版 社

北 京

P208
2/30

图字：01-2009-5563 号

内 容 简 介

随着 ESRI 公司 ArcGIS 软件的发展, ArcObjects 已成为整个 ArcGIS 系列软件的核心基础, 学习掌握 ArcObjects 是地理信息系统开发课程的必然趋势。本书是美国 CRC 出版社于 2008 年出版的张康聪 (Kang-tsung Chang) 教授编著的 *Programming ArcObjects with VBA: A Task-Oriented Approach* (美国 CRC 出版社, 2008 年·第二版) 的中文翻译版。全书共 14 章, 内容涉及 ArcObjects 的所有方面, 以 ArcGIS 9.2 的二次开发为任务导向, 通过示例程序, 讲解编程要点并提供运行范例。随书附光盘, 包含 128 个示例宏程序及其案例数据, 涉及从最基本的添加图层到高级的地理操作等方面。本书有三大特色: 一是采用面向任务的方法来学习 ArcObjects 编程, 采用 VBA 讲解 ArcObjects 编程; 二是通过大量的示例练习开展渐进的编程训练; 三是系统性强。

本书可作为 GIS 专业本科高年级课程和相关领域研究生的教材, 也可作为 GIS 专业技术人员的参考书。

Kang-tsung Chang

Programming ArcObjects with VBA: A Task-Oriented Approach, Second Edition
ISBN 978-1-58488-580-1

Copyright © 2008 by Taylor & Francis Group LLC.

All Rights Reserved.

Authorized translation from English language edition published by CRC Press,
part of Taylor & Francis Group LLC.

本书封面贴有 Taylor & Francis 集团防伪标签, 未贴防伪标签属未获授权的非法行为。

图书在版编目 (CIP) 数据

基于 VBA 的 ArcObjects 编程: 面向任务的方法 / (美) 张康聪著; 陈健飞, 蔡砥, 黄婷薇译. —2 版. —北京: 科学出版社, 2010

ISBN 978-7-03-026495-4

I. ①基… II. ①张…③陈…③蔡…④黄… III. ①地理信息系统-程序设计-应用软件, Arcs IV. ①P208

中国版本图书馆 CIP 数据核字 (2010) 第 014711 号

责任编辑: 韩 鹏 赵 冰 / 责任校对: 张小霞

责任印制: 钱玉芬 / 封面设计: 王 浩

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

源海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

*

2010 年 1 月第 一 版 开本: 787×1092 1/16

2010 年 1 月第一次印刷 印张: 20 1/4

印数: 1—2 400 字数: 458 000

定价: 59.00 元 (含光盘)

(如有印装质量问题, 我社负责调换)

译者序

随着 ESRI 公司 ArcGIS 软件的发展, ArcObjects 已经成为整个 ArcGIS 系列软件的核心基础。每个想把自己的 ArcGIS 知识水平提升一个层次的人, 都需要学会使用 ArcObjects。美国 CRC 出版社于 2005 年出版了张康聪 (Kang-tsung Chang) 教授编著的 *Programming ArcObjects with VBA: A Task-Oriented Approach* (第一版), 广受读者欢迎, 2008 年修订出版了第二版。本书系原著第二版的中文翻译版。

该书有以下三大特色。

一是采用面向任务的方法来学习 ArcObjects 编程。面向任务的方法至少有三个主要优点: 首先, 它将 ArcObjects 与已有知识联系起来; 其次, 面向任务的方法用一种符合 ArcGIS 用户逻辑的方法来组织 ArcObjects; 最后, 面向任务的方法还可以帮助我们用 ArcObjects 的新知识更好地理解 ArcGIS。该书采用 Visual Basic for Application (VBA) 讲解 ArcObjects 编程。VBA 已经内嵌至 ArcMap 和 ArcCatalog, 因此用 VBA 进行 ArcObjects 编程对于 ArcGIS 用户来说非常方便。

二是通过大量的示例练习开展渐进的编程训练。该书共附有 128 个示例宏程序, 涉及从最基本的添加图层到高级的地理操作等方面, 每个练习都相互独立, 同时又有内在联系, 可使读者由浅入深地学习 ArcObjects 编程。大量的练习, 不仅在章节中作详细剖析, 而且与案例数据一起刻录成光盘附在书后, 大大方便了读者。

三是系统性强。14 章内容分别涉及了 ArcObjects 的各个方面, 不仅如此, 在每章节的编排上, 除了围绕任务讲解编程技巧外, 还将练习中所涉及的主要对象、接口等列举出来并作简要介绍, 使读者对各章节所涉及的技术内容有全面宏观的把握, 也方便读者按需查阅。

本书是针对 GIS 专业高年级学生以及 GIS 软件开发人员、ArcGIS 用户所编写的一本从入门到精通 ArcObjects 的编程教材。ArcObjects 是一个庞大的类、对象、接口体系, 各个对象和接口之间关系复杂, 采用知识点列举的教学方式显然不能让读者快速系统地掌握编程技术。国内外已出版的 ArcObjects 相关书籍多为原理方法介绍, 较少联系应用任务和实例, 缺乏循序渐进的系统引导, 也未附程序脚本和练习数据的光盘。本书以 ArcGIS 9.2 的二次开发为任务导向, 通过示例程序, 讲解编程要点并提供运行范例, 可作为 GIS 专业本科高年级课程和相关领域研究生的教材, 是我国 GIS 人才分层次培养进程中所急需的教学资源。无论读者的编程经验如何, 本书将提供一把钥匙, 让读者掌握 ArcObjects 强大且多样的功能, 用于协助管理 GIS 活动。

本书的翻译出版, 得到广州大学教材建设基金的资助。译校工作由陈健飞、蔡砥和黄婷薇合作完成。限于译者水平, 译文不足之处在所难免, 欢迎读者批评指正。

陈健飞

2009 年 10 月

引 言

本书的读者应是想要快速学习 ArcObjects 编程的 ArcGIS 用户。ArcObjects 和 ArcGIS 都是美国环境系统研究所公司 (Environmental Systems Research Institute Inc., ESRI) 所开发和分发的产品。ArcObjects 是 ArcGIS 的开发平台; ArcGIS 是管理地理信息系统 (geographic information systems, GIS) 的软件包。用户在使用 ArcGIS 之前学习 ArcObjects 会更加理想, 但实际情况却不是这样: 他们学习使用 ArcGIS 的工具栏和命令。通过用户界面来学习总是比较简单, 而通过从代码中找出对象、属性和方法来学习则相对困难。只有当我们意识到 ArcObjects 编程能减少大量的重复劳动, 使工作更流程化, 而且能够创造 ArcGIS 中不易获得的功能时, 才会把 ArcObjects 提出来讨论。

如何快速有效地学习 ArcObjects 编程? 答案是结合我们对 ArcGIS 的已有认识。这也许让一些人感到吃惊。

面向任务的方法

GIS 活动是面向任务的: 我们用 GIS 进行数据整合、数据管理、数据显示、数据分析等。因此, 采用面向任务的方法来学习 ArcObjects 编程是有效的。面向任务的方法至少有以下三个主要优点。

首先, 它将 ArcObjects 与我们已有的知识联系起来。以 QueryFilter 为例, 本书首先将 QueryFilter 对象 (与数据探查) 任务相联系。当我们知道该对象所执行的操作与 ArcMap 中的 Select by Attributes (按属性选择) 命令所执行的操作相同时, 就很好理解该对象的属性和方法了, 因为 Select by Attributes 命令是我们常用的。

其次, 面向任务的方法用一种符合 ArcGIS 用户逻辑的方式来组织 ArcObjects。对于初学者来说, 要掌握 ArcObjects 模型图中的成千上万个对象、属性和方法, 虽然是可能的, 但实际上却非常困难。而运用面向任务的方法, 我们就可以根据任务的组织渐进地学习 ArcObjects。

最后, 面向任务的方法还可以帮助我们用 ArcObjects 的新知识更好地理解 ArcGIS。例如, SpatialFilter 类是 QueryFilter 的一种, 除具有继承 QueryFilter 类的属性之外, 它还具有自身的几何属性和空间关系属性。类之间的关系解释了为什么 ArcMap 中的 Select by Location (按位置选择) 命令在进行数据查询时, 可以同时接受属性约束和空间约束。也许, Select by Location 命令改名为 Select by Attributes and Location 更合适。

关于本书

本书有 14 章。前 3 章介绍 ArcObjects、编程基础和定制。本书采用 Visual Basic for Application (VBA) 讲解 ArcObjects 编程。由于 VBA 已经内嵌至 ArcMap 和 ArcCatalog, 用 VBA 进行 ArcObjects 的编程对于 ArcGIS 用户来说非常方便。下面对前 3 章所涉及的主要内容进行概括。

第 1 章 ArcObjects 概述——Geodatabase、ArcObjects、ArcObjects 的组织 and ArcObjects 的帮助资源以及 Geoprocessing 对象。

第 2 章编程基础——基本要素、代码的编写、调用子程序和函数、VB 编辑器和代码调试。

第 3 章用户界面的定制——用已有命令创建工具条、添加新的按钮和工具、添加窗体、制作基本的模板。

第 4~14 章介绍如何用 ArcObjects 编程解决一般的 GIS 任务。每一章都围绕着一个中心主题, 分成三个部分: 第一部分是关于某主题的 ArcGIS 命令的概览; 第二部分讨论与该主题相关的对象; 第三部分介绍该主题下解决一般任务的示例宏和示例 Geoprocessing 宏。将 ArcGIS 命令、ArcObjects 和示例宏结合在一起使我们能够将使用 ArcGIS 的经验和 ArcObjects 编程本身的功能有效地联系起来。

随书附带的 CD 包含 95 个示例宏, 按章节存放在 VBA_programs 文件夹里。每个示例宏的起始部分有一段关于其用法的简短介绍, 并列出其主要接口和主要成员 (即属性和方法), 接着是代码及其解释。许多宏被分成两个或多个部分, 这样更容易对代码行进行解释。这些示例宏以文本文件的形式存储, 很容易导入 ArcMap 和 ArcCatalog 的 Visual Basic Editor 中进行查看和运行。

附带的 CD 还包括 33 个示例 Geoprocessing 宏, 这是第二版新增的内容。这些宏按章节存放在 GP_programs 文件夹里。Geoprocessing 对象是 ArcObjects 的新组件, 用于支持在诸如 VBA 或 Python 这样的脚本语言中运行数百个 Geoprocessing 工具。这些工具和 ArcGIS 桌面软件中的 ArcToolbox 应用是一样的。Geoprocessing 对象是一种“粗颗粒”的对象, 比那些“细颗粒”的对象要易用, 因此让那些对 ArcObjects 的细节不够了解的用户也可以运行宏。在第 4~7 章以及第 9~14 章里, 为了与常规的 VBA 宏相区别, 这些 Geoprocessing 宏是用专栏框起来的。

本书中的所有示例宏都在 ArcGIS 9.2 上成功运行。附带的 CD 包括了测试用的数据集, 按章存放在 Data 文件夹里。必须注意两个问题: 第一, 宏的运行需要 ArcGIS9.1 或 ArcGIS9.2; 第二, Data 文件夹必须在 C 驱动器下 (如 c:\data\chap4), 这已经编码到示例宏中。如果 Data 文件夹存放到其他驱动器 (如 g:\data\chap4), 则在运行宏之前, 必须修改路径。

下面对 4~14 章所涉及的主要任务进行概括。

第 4 章数据集和图层管理——将数据集添加为图层、管理图层、管理数据集和报告地理数据集的信息。

第 5 章属性数据管理——列举字段集和字段属性、添加或删除字段、字段值的计算、表的连接和关联。

第 6 章数据转换——Shapefile 转换为 geodatabase、coverage 转换为 geodatabase 和 shapefile、栅格化和矢量化、添加 XY 事件。

第 7 章坐标系——操作动态 (on-the-fly) 投影、定义坐标系、地理变换、投影数据集。

第 8 章数据显示——矢量数据、栅格数据、制作页面布局。

第 9 章数据探查——属性查询、空间查询、空间查询和属性查询的结合、计算描述性统计值。

第 10 章矢量数据操作——缓冲区 (buffer) 分析、叠置 (overlay) 分析、基于位置的数据连接、要素 (feature) 运算。

第 11 章栅格数据操作——管理栅格数据，逻辑操作 (local operations)、邻域操作 (neighborhood operations)、区带操作 (zonal operations)、距离量算 (distance measure) 操作。

第 12 章地形制图和分析——计算等值线 (contour)、坡度 (slope)、坡向 (aspect) 和地貌晕渲 (hillshade)、视域 (viewshed) 分析、流域 (watershed) 分析、创建并编辑 TIN。

第 13 章空间插值——创建空间插值、插值方法比较。

第 14 章二值模型和指数模型——建立二值模型和指标模型。

排 版 凡 例

下面列出本书中采用的排版凡例：

- (1) VBA 示例宏与正文是分开的，以不同字体表示。
- (2) 示例 Geoprocessing 宏用专栏框起来，以不同字体表示。

目 录

译者序

引言

第 1 章 ArcObjects 概述	1
1.1 Geodatabase	1
1.2 ArcObjects	4
1.3 ArcObjects 的组织	7
1.4 ArcObjects 帮助资源	8
1.5 Geoprocessing 对象	10
参考文献	11
第 2 章 编程基础	12
2.1 基本元素	12
2.2 代码的编写	17
2.3 调用子程序和函数过程	23
2.4 Visual Basic 编辑器	25
2.5 调试代码	26
第 3 章 用户界面的定制	30
3.1 用 ArcMap 命令创建工具栏	30
3.2 添加新按钮	32
3.3 添加新工具	34
3.4 存储新工具栏至模板	37
3.5 添加窗体	38
3.6 将窗体存储到模板中	43
第 4 章 数据集和图层管理	45
4.1 ArcGIS 中数据集的使用	45
4.2 数据集和图层相关的 ArcObjects	46
4.3 将数据集添加为图层	47
4.4 管理图层	55
4.5 管理数据集	56
4.6 报告地理数据集信息	59
第 5 章 属性数据管理	61
5.1 ArcGIS 中的属性数据管理	61
5.2 属性数据管理的 ArcObjects	62
5.3 列举字段集和字段的属性	65

5.4	添加或删除字段	71
5.5	字段值的计算	73
5.6	表的连接和关联	76
第 6 章	数据转换	86
6.1	ArcGIS 里的数据转换	86
6.2	用于数据转换的 ArcObjects	87
6.3	Shapefile 转换为 GeoDatabase	89
6.4	Coverage 转换为 GeoDatabase 和 Shapefile	97
6.5	栅格化和矢量化	101
6.6	添加 XY 事件	108
第 7 章	坐标系	112
7.1	在 ArcGIS 中管理坐标系	112
7.2	坐标系的 ArcObjects	113
7.3	操作动态投影	116
7.4	定义坐标系	119
7.5	地理变换	123
7.6	投影数据集	129
第 8 章	数据显示	139
8.1	在 ArcGIS 中显示数据	139
8.2	数据的 ArcObjects 对象	140
8.3	矢量数据	144
8.4	栅格数据	153
8.5	制作页面布局	162
第 9 章	数据探查	171
9.1	ArcGIS 中的数据探查	171
9.2	数据探查的 ArcObjects	172
9.3	属性查询	174
9.4	空间查询	179
9.5	空间查询和属性查询的结合	188
9.6	计算描述性统计值	193
第 10 章	矢量数据操作	197
10.1	ArcGIS 中的矢量数据分析	197
10.2	用于矢量数据分析的 ArcObjects	198
10.3	缓冲区分析	199
10.4	叠置分析	202
10.5	基于位置的数据连接	207
10.6	要素运算	209

第 11 章 栅格数据操作	217
11.1 在 ArcGIS 中分析栅格数据	217
11.2 栅格分析的 ArcObjects	218
11.3 管理栅格数据.....	220
11.4 逻辑操作.....	228
11.5 邻域操作.....	232
11.6 区带操作.....	234
11.7 距离量算操作.....	236
第 12 章 地形制图和分析	245
12.1 在 ArcGIS 中进行地形制图和分析	245
12.2 地形制图和分析相关的 ArcObjects	246
12.3 计算等值线、坡度、坡向和地貌晕渲.....	247
12.4 视域分析.....	259
12.5 流域分析.....	261
12.6 创建和编辑 TIN	265
第 13 章 空间插值	271
13.1 在 ArcGIS 中进行空间插值	271
13.2 空间插值相关的 ArcObjects	271
13.3 创建空间插值.....	272
13.4 插值方法比较.....	279
第 14 章 二值模型和指数模型	283
14.1 在 ArcGIS 中建模	283
14.2 GIS 建模相关的 ArcObjects	283
14.3 建立二值模型和指数模型.....	284

第 1 章 ArcObjects 概述

环境系统研究所 (ESRI, Environmental Systems Research Institute Inc.) 公司的 ArcGIS 采用单一的可伸缩的构架。ArcGIS 的三个版本, 即 ArcView、ArcEditor 和 ArcInfo 共享相同的应用程序, 包括 ArcCatalog 和 ArcMap。Geodatabase 数据模型和 ArcObjects 是 ArcCatalog 和 ArcMap 这两个桌面应用的基础。它们也向本书的读者提供了在 ArcGIS 中使用 VBA (Visual Basic for Applications) 定制应用程序的基础。

Geodatabase 数据模型取代了 ESRI 公司过去用于 coverage 的 shapefile 的 georelational 数据模型。这两个数据模型的不同点就在于如何对地理数据和属性数据进行存储。Georelational 数据模型把地理数据和属性数据分别存储在不同的系统中: 地理数据 (geo) 存储为图形文件; 属性数据 (relational) 存储为关系数据库。Georelational 数据模型一般采用要素标签或者 ID 来链接两个部分。这两个部分必须同步, 以保证能够统一地进行查询、分析和显示。相反, geodatabase 数据模型将地理数据和属性数据共同存放在唯一的系统中, 其中地理数据存放为几何字段。

Geodatabase 数据模型的另一个特色是使用了面向对象技术。面向对象技术把空间要素视为一个对象并把同一类空间要素组成一个类。类, 或者类中的对象, 拥有属性和方法。属性描述对象的特征或性质, 方法则实现对象的行为。ArcGIS 的开发者在 ArcGIS 的数以百计的类上已经实现了属性和方法。因此, 当我们在使用 ArcCatalog 和 ArcMap 的时候, 实际上是与这些类以及它们的属性和方法的互动。

本章的重点是 geodatabase 数据模型和 ArcObjects。为了在程序中使用 ArcObjects, 我们必须了解在 geodatabase 数据模型下空间数据的结构和存储方式, 以及 ArcObjects 中的类是如何设计和组织的。1.1 节介绍 geodatabase 数据模型的基础, 包括模型中所涵盖的数据的类型; 1.2 节介绍 ArcObjects 的基础, 包括类、类与类之间的关系、接口、属性和方法; 1.3 节介绍 ArcObjects 的组织; 1.4 节介绍 ArcObjects 的帮助资源。

1.1 Geodatabase

地理信息系统 (geographic information system, GIS) 管理地理空间数据 (geospatial data)。地理空间数据用于描述空间要素的位置信息和特征信息, 地面上诸如公路、地块及植被等都是空间要素。空间要素的位置信息是用地理坐标 (经度和纬度值) 或者投影坐标 (例如全球横轴墨卡托投影, 即 UTM 坐标) 来测度的。空间要素的特征信息被表示成数值型和字符型属性。本书中使用地理数据 (geographical data) 这个术语描述那些包含了空间要素位置的数据, 而非地理数据 (non-geographical data) 来描述那些仅包含空间要素属性的数据。

Geodatabase 用表 (table) 储存地理数据和非地理数据。因此区别表的不同类型是很重要的。表有行和列。每个行对应着一个要素, 每个列 (字段) 表示属性。包含地理数据的表有一个几何字段, 用于同仅包含非地理数据的表相互区分。接下来的章节将介绍可存放于 Geodatabase 的地理数据和非地理数据。

1.1.1 矢量数据

Geodatabase 数据模型表现了点、复线 (polyline) 和多边形等矢量型的空间要素^[1]。点要素可以是单点要素 (simple point), 也可以是包含一系列点的多点要素 (multipoint feature)。复线要素是一组线段, 可以是连续的也可以是不连续的。多边形要素可以由一个或多个环组成。一个环是一组连续的、闭合的、不相交的线段。

Geodatabase 将空间要素分为要素类 (feature class) 和要素数据集 (feature dataset)。要素类是一组具有相同几何类型的空间要素的集合。因此要素类可能包含单点要素、线要素或多边形要素。要素数据集是具有相同坐标系和空间范围的要素类的集合。因此, 要素数据集可以用于管理相同研究区的不同的要素类, 或用于管理相互具有拓扑关系的要素类, 比如几何网络或同平面 (二维) 拓扑结构。拓扑结构是一系列的关系, 用于定义一个或多个要素类中的要素如何共享几何空间。

要素类与 shapefile 相似, 拥有简单要素; 要素数据集与 coverage 相似, 拥有基于相同坐标系的多个数据集。但是这种相似并不能说明传统的数据模型与受计算机技术进步驱动的 geodatabase 数据模型之间没有其他区别。

在 geodatabase 中, 要素类可以是单独的要素类或者是要素数据集中的一部分。不论在哪种情况下, 要素类都是以表的形式存储的。一个要素类至少有两个字段: 一是对象 ID 或要素 ID; 另一个是几何 (geometry) 或形状 (shape)。要素类也可以有其他属性字段, 但是几何字段使一个要素类与其他的表区分开。

ArcGIS 的用户把要素类视作一个要素属性表。在 ArcMap 中打开要素层的属性表的时候, 可以看到上文两个默认字段。而且我们只需通过一张要素属性表就能在地图上定位并且高亮显示空间要素。

要素类中的要素可以进一步细分为子类型。例如, 道路要素类可以基于日均车流量来划分子类。Geodatabase 数据模型为对象分组提供了 4 条广泛认定的规则: 属性域 (attribute domains)、默认值 (default values)、连通性规则 (connectivity rules) 和关系规则 (relationship rules)^[1]。属性域将属性值限定为值的有效范围或是一组有效值的集合。默认值设定了预期属性值。连通性规则控制几何网络中的要素如何彼此相连接。关系规则决定了有多少要素是和另一个要素有联系的。

1.1.2 栅格数据

Geodatabase 数据模型把栅格数据描述成等大小的空间单元的二维数组^[1]。数组和单元的概念与 ESRI 网格模型 (grid model) 的概念相同。

对 GIS 而言,大部分的栅格数据是可用的,包括卫星影像、数字高程模型 (DEM)、数字正射影像、扫描文件、图形文件和特定软件的栅格数据 (如 ESRI 网格)^[2]。Geodatabase 模型将它们视为栅格数据集 (raster dataset)。不过,栅格数据集可以是单波段 (single band),也可以是多波段 (multiple bands) 的。一般 ESRI 网格包含单波段,而多光谱卫星影像则包含多波段。

一个多波段栅格数据集可以是栅格数据操作的输出。例如,耗费距离量算操作 (参见第 11 章) 可以生成最小累计耗费距离 (least accumulative cost distance)、后向链 (back link) 以及配置 (allocation)。首先可以把这几个不同的输出结果存储在多波段栅格数据集中,每个波段存储一个输出结果,而后被提取出来建立适当的栅格数据集。

1.1.3 不规则三角网

Geodatabase 数据模型用不规则三角网 (TIN) 数据集来储存一系列互不重叠的三角形来近似表示一个表面。高程值连同 x 、 y 坐标一起存储在组成三角形的结点上。在地形制图与分析的许多情况下,TIN 数据集是栅格数据的一种新选择。究竟选哪一个取决于数据的灵活性和计算效率^[2]。

TIN 的输入包括数字高程模型 (digital elevation model, DEM)、等值线、全球定位系统 (global positioning system, GPS) 数据、光达 (light detection and ranging, LiDAR) 数据和测量数据。我们也可以借助线要素 (如水流和道路) 和面要素 (如湖泊和水库) 来修改和改善 TIN。因此,数据灵活性是使用 TIN 的重要优势。此外,相比高程栅格来说,TIN 的三角形面更易于建立一个鲜明的地形图像。

计算效率是使用栅格数据集的主要优点。数组和单元的简单数据结构使坡度、坡向、表面曲率、视域、流域等运算变得相对容易。

1.1.4 位置数据

位置数据 (location data) 指的是能被转换成点要素的数据。通常位置数据的例子是包含 x 、 y 坐标或者街道地址的表。我们可以把有 x 、 y 坐标的表直接转换成点要素类,其中每个要素对应一对 x 、 y 坐标。用街道网络为参照,我们还可以把街道地址序列转换成一组点要素。

1.1.5 非地理数据

储存非地理数据的表是没有几何字段的。Geodatabase 数据模型把这样的表定义为一个对象类 (object class)。对象类的例子包括由逗号分隔的文本文件和 dBASE 文件。这些文件或表包含空间要素的属性,并包含键值 (即关联字段),以便在相关的数据库环境中链接地理数据。

1.2 ArcObjects

ArcObjects 是 ArcGIS Desktop 的开发平台,也是 ArcGIS Engine、ArcGIS Server 的开发平台(本书仅涵盖 ArcGIS Desktop)。ArcObjects 是对象的集合,隐含于我们执行 ArcGIS 任务的菜单和图标。这些对象还使得软件开发者得以访问数据并编程执行任务。

1.2.1 对象和类

ArcObjects 由对象和类组成^[3]。一个对象代表一个空间要素,如道路或植被。在 geodatabase 中,一个对象对应着表中的一行,而表中的列则记录对象的属性。一个类是一组具有相似属性的对象的集合。在 ArcObjects 中,类可以有内建的接口、属性和方法。

ArcObjects 包括三种类型。

(1) 最常见的类是组件类(coclass)。组件类可以用来产生新的对象。例如,FeatureClass 是组件类,它可以用于创建一个新的作为组件类实例的要素类对象。

(2) 第二种是抽象类。抽象类不能够用来创建新的对象。抽象类的存在使得其他的类(即子类)可以使用和共享某个抽象类所支持的属性和方法。例如,GeoDataset 是一个抽象类。GeoDataset 类的存在使诸如要素类、栅格数据集等地理数据集能全部地共享 GeoDataset 所支持的空间范围属性和空间参照属性。

(3) 第三种是类。类不能直接用来创建新的对象,类的对象必须由其他类来产生。例如,EnumInvalidObject 是一个不可创建的类,因为 EnumInvalidObject 只能从另一个对象获得,如数据转换对象等。例如,当我们将 Shapefile 从一种坐标转换到另一种坐标时,数据转换对象会自动产生一个 EnumInvalidObject 对象来跟踪转换失败的对象。

1.2.2 类间关系

面向对象技术引入了多种类间关系类型。ArcObjects 的开发者通常继续使用这些关系。Zeiler 编著的关于 ArcObjects 的类间关系是一本很好的参考书^[3]。同时还有许多讨论有关面向对象技术的分析与设计的书籍,如 Larman^[4]的著作。必须理解类间关系的基本概念,这对于浏览对象模型图以及 ArcObjects 编程都是非常重要的。

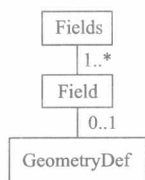


图 1.1 Fields 和 Field 之间的关系是“一对一”或“一对多”关系,而 Field 和 GeometryDef 之间的关系是“一对 0”或“一对一”关系

Association 描述了两个类之间的关系。Association 用多重性表达(multiplicity expressions)来明确一个类能有多少个实例与另一个类是关联的。常见的多重性表达是“一对一”或“一对多”

(1..*)。例如,图 1.1 表示 fields 和 field 之间的关联关系以及 field 和 GeometryDef 之间的关联关系。图 1.1 的多重性表达表明:

(1) 一个 fields 对象可以和一个或多个 field 对象关联。fields 表示表中字段的集合。

(2) 一个 field 对象可以与 0 个或一个 GeometryDef 对象关联,其中 GeometryDef 代表几何定义(geometry definition)。

与几何定义关联的字段是几何字段,一个表最多只能有一个几何字段。

类型继承(type inheritance)关系定义了父类和子类之间的关系。子类是父类的成员之一,并且继承了父类的属性和方法。同时子类也可以拥有其他的属性和方法以区别于父类的其他成员。如图 1.2 所示,GeographicCoordinateSystem 类是 SpatialReference 类的子类。GeographicCoordinateSystem 类、ProjectedCoordinateSystem 类和 UnknownCoordinateSystem 类都继承了 SpatialReference 类所支持的属性和方法。但 GeographicCoordinateSystem 类拥有额外的、关于地理坐标系的特定属性和方法。

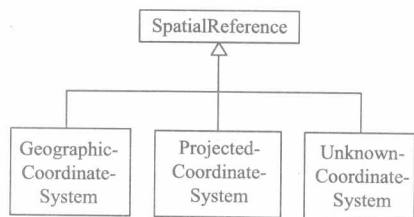


图 1.2 SpatialReference 及其三个子类

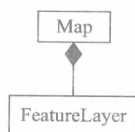


图 1.3 Map 对象由 0 个、一个或多个 FeatureLayer 对象组成

组成(composition)关系描述类之间的整体-局部的关系。组成关系属于关联(association)关系的一种,不过,其组合端的多重性一般是 1,另一端的多重性则可以是 0 或任意正整数。例如,Map 类和 FeatureLayer 类是组成关系(图 1.3)。在 ArcMap 中,地图对象(map object)表示一张地图或者一个数据框架(data frame),而

一个要素图层对象(feature layer object)表示地图中的要素型图层。一张地图能与多个要素图层关联。换言之,要素图层是地图的一部分。

聚合(aggregation)关系又称为共享聚合,描述类间的整体-部分的关系,但与组成关系不同,聚合关系的组成端的多重性一般超过 1。例如,图 1.4 表示 SelectionSet 对象可以由 QueryFilter 对象和 Table 对象创建。组合端的表和查询筛选器(query filter)可以创建另一端的查询集(selection set)。

实例化(instantiation)关系是指类的对象可以由另一个类的对象来创建。例如,图 1.4 显示查询集对象可以由查询筛选器和表创建。另一个例子是我们前面提到的 EnumInvalidObject,它可以由 FeatureDataConverter 对象来创建(图 1.5)。

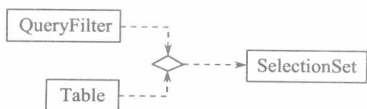


图 1.4 QueryFilter 对象和 Table 对象可共同创建 SelectionSet 对象

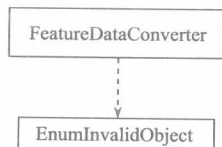


图 1.5 EnumInvalidObject 对象由 FeatureDataConverter 对象来创建

1.2.3 接 口

当我们用 ArcObjects 中的对象来编程时，可能不直接使用对象，而是通过其某个接口来访问。接口提供了一系列外部可见的操作。例如，RasterReclassOp 对象实现了



图 1.6 RasterReclassOp 对象支持 IReclassOp 接口和 IRasterAnalysisEnvironment 接口

IRasterAnalysisEnvironment 接口和 IReclassOp 接口（图 1.6）。我们可以通过 IRasterAnalysisEnvironment 接口或 IReclassOp 接口来访问 RasterReclassOp 对象，而不是直接访问对象本身。

一个对象支持两个或者更多的接口。另外，对象还可以继承父类接口。如果是多个接口，则可以由一个接口访问另一个接口，或从一个接口跳转至另一接口。这个技术被称作 QueryInterface（查询接口）或简称 QI。QI 简化了编程过程。假设我们要运用一个 RasterReclassOp 对象来进行栅格数据分类。首先，我们用 IRasterAnalysisEnvironment 接口建立分析环境。然后通过 QI 转到 IReclassOp 接口进行数据再分类。在第 2 章中介绍编程基础时，我们将详细讨论 QI 技术。

一些 ArcObjects 的对象有两个或多个相似接口。例如，FeatureDataConverter 对象实现了 IFeatureDataConverter 接口和 IFeatureDataConverter2 接口。这两个接口都提供了 shapefiles、coverages 和 geodatabases 之间的格式转换方法。但是 IFeatureDataConverter2 接口在数据子集方面有附加的功能。面向对象技术让 ArcObjects 的开发者在不需删除或更新现存接口的情况下仍可以为类添加新的接口。

1.2.4 属性和方法

接口提供了一系列外部可见的操作。更确切地说，接口允许程序员使用该接口提供的属性和方法。属性描述对象的一个性质或特征。方法也称为行为，即进行一个特定的动作。例如，图 1.7 显示了 IRasterAnalysisEnvironment 接口的属性和方法。这些属性和方法都被称作该接口的成员。

属性可以是只读的、只写的或可读写的。可读属性也被称为可取属性（get property），可写属性也被称为可置属性（put property）。在图 1.7 中，IRasterAnalysisEnvironment 接口的 Mask 属性和 OutWorkspace 属性带有杠铃状符号。左侧带方块表示可取属性，右侧带方块则表示可置属性。如果右侧方块是空心的，说明该属性是通过参数写入的；如果是实心的，则表示可直接写入值。这两种可置属性的区别在于赋值的方式不同。通过参数写入的可置属性必须使

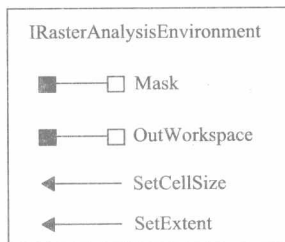


图 1.7 IRasterAnalysisEnvironment 接口的属性和方法

（属性用哑铃符号表示，方法用箭头符号表示；本书的一些场合中，属性和方法用双冒号表示，如 IRasterAnalysisEnvironment::Mask）

用关键词 Set，而直接写值的则不用。例如，通过 IRasterAnalysisEnvironment 接口的 Mask 属性来指定一个分析掩模，我们需要用到类似下列的语句：Set pEnv.Mask=pMaskDataset，其中，pEnv 代表分析环境，而 pMaskDataset 代表分析掩模对象。

要完成一个动作，接口的方法需要一些参数，并可能返回一个或多个值。在图 1.7 中，箭头符号表示 IRasterAnalysisEnvironment 接口上的 SetCellSize 方法和 SetExtent 方法。SetCellSize 方法的语法是 object.SetCellSize (envType [, cellSizeProvider])。该方法有两个参数：第一个参数是必需的；第二个是可选的。

接口可能不会兼有属性和方法。有的接口只有属性，而有的接口只有方法。例如，IReclassOp 接口只有方法。图 1.8 列出了 IReclassOp 接口的 5 个方法中的 2 个。这些方法都对栅格数据进行重分类，但机制不同。ReclassByASCIIFile 方法使用的是 ASCII 文件，而 ReclassByRemap 方法则利用编程建立的重映射 (remap)。

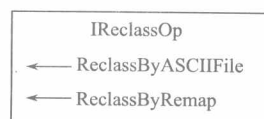


图 1.8 IReclassOp 接口的方法

1.3 ArcObjects 的组织

ArcGIS9.2 有成千上万个组件类和接口。ESRI 公司将 ArcObjects 对象划分为超过 65 个的库。诸如 ArcCatalog、ArcCatalogUI、ArcMap、ArcMapUI、Carto、Display、Geodatabase 和 Geoprocessing 等是核心库；而 3D Analyst、Spatial Analyst 和 Network Analyst 等是扩展库。每一个库所包含的对象都具有可绘制的类间关系。例如，Carto 库拥有 map document、map 和 page layout 等对象。

采用库和子系统来组织 ArcObjects 类似于通过应用模块和功能来组织 ArcGIS。这种组织方式对于熟悉 ArcGIS 操作的人来说，是一个很好的起点。例如，空间分析器 (Spatial Analyst) 扩展库根据栅格数据操作的类型来组织对象。因此，库中的对象模型列出了 RasterConditionalOp、RasterExtractionOp、RasterLocalOp、RasterMapAlgebraOp、RasterNeighborhoodOp、RasterZonalOp 以及其他与空间分析器扩展模块各种功能有类似名称的对象。熟悉 ArcGIS 空间分析器扩展模块的用户在使用这些对象时就不会有问题了。而其他子系统，如 Geodatabase、ArcMap 和 ArcCatalog 中的许多对象采用了新的面向对象的概念和方法，因此难以与以前的有所联系。

要使用库中的对象，必须引用该库。这意味着要引用的库对于用户必须是可用的。各个 ArcObject 库的可用性依赖于其有效许可。例如，如果没有 ArcScene 扩展模块的许可证，则用户不能访问 ArcScene 库及其对象。

对于 ArcGIS9.2 来说，ArcObjects 核心库、3D 分析器扩展库、空间分析器扩展库等自动地加载到 VBA 中。换句话说，我们不需要引用这些库就可以使用其中的对象。对于那些没有自动加载的库，在 Visual Basic 编辑器里有一个引用选项，该选项打开一个列有所有有效对象库的列表，允许用户引用它们。

ArcObjects 包含了 ESRI 公司所开发的对象。近来的发展趋势表明，对象也可以按行业分类的模式来归类。因为真实世界中的对象都具有不同的属性和方法，无法相互应