



普通高等教育“十二五”规划教材

PUTONG GAODENG JIAOYU "12·5" GUIHUA JIAOCAI

GIS软件SharpMap源码 详解及应用

陈 真 何 津 余 瑞 编著



冶金工业出版社
Metallurgical Industry Press



普通高等教育“十二五”规划教材

GIS 软件 SharpMap 源码 详解及应用

陈 真 何 津 余 瑞 编著



北 京
冶 金 工 业 出 版 社
2012

内 容 提 要

本书对目前基于 C#语言开发的 GIS 开源项目——SharpMap 进行详细剖析、解释,以便 GIS 专业学生及 GIS 的初学者能学习和掌握 GIS 底层开发技术。全书分为 3 部分共计 11 章。第 1 部分讲解 SharpMap 源码,主要内容包括地图、地图控件、图层、绘制、样式、数据、几何对象等;第 2 部分介绍基于 SharpMap 的应用开发,主要内容包括两个 SharpMap 下载包中附带的 Windows 应用程序;第 3 部分介绍 SharpMap 系统扩展,主要内容包括数据源对象扩展及图层对象扩展。

本书可作为地理信息系统相关专业本科生“GIS 开发与设计”等课程的教学用书,也可供对 GIS 感兴趣的初学者及 GIS 工程技术人员阅读参考。

图书在版编目(CIP)数据

GIS 软件 SharpMap 源码详解及应用/陈真,何津,余瑞编著.
—北京:冶金工业出版社,2012.3
普通高等教育“十二五”规划教材
ISBN 978-7-5024-5869-0

I. ①G… II. ①陈… ②何… ③余… III. ①地理信息
系统—应用软件, SharpMap GIS—程序分析—高等学校—教材
IV. ①P208

中国版本图书馆 CIP 数据核字(2012)第 022779 号

出 版 人 曹胜利

地 址 北京北河沿大街嵩祝院北巷 39 号,邮编 100009

电 话 (010)64027926 电子信箱 yjcbbs@cnmip.com.cn

责任编辑 廖 丹 美术编辑 李 新 版式设计 孙跃红

责任校对 石 静 责任印制 李玉山

ISBN 978-7-5024-5869-0

三河市双峰印刷装订有限公司印刷;冶金工业出版社出版发行;各地新华书店经销
2012 年 3 月第 1 版,2012 年 3 月第 1 次印刷

787mm×1092mm 1/16;19 印张;455 千字;291 页

39.00 元

冶金工业出版社投稿电话:(010)64027932 投稿邮箱:tougao@cnmip.com.cn

冶金工业出版社发行部 电话:(010)64044283 传真:(010)64027893

冶金书店 地址:北京东四西大街 46 号(100010) 电话:(010)65289081(兼传真)

(本书如有印装质量问题,本社发行部负责退换)

前 言

在过去的十年间，GIS（Geographic Information System，地理信息系统）已经从一个高度专业化的领域发展为影响我们生活方方面面的技术，从道路导航到自然灾害管理，随处可见 GIS 的应用。直到几年前还只有少数研究人员、决策人员和政府职员在使用 GIS，现在几乎每个人都可以通过互联网，利用众多 GIS 开发商提供的各种免费工具创建个性化地图或对 GIS 数据进行叠置分析。随着智能手机的普及，移动 GIS 应用技术也将迅速融入人们的生活，为普通百姓日常生活提供便利。

近年来 GIS 发展势头迅猛，新技术层出不穷，软件频频升级，设备不断更新。要跟上 GIS 发展的潮流，就必须不断了解和掌握 GIS 最新的发展动态。这也对 GIS 开发人员提出了更高的要求，不仅要知其然（掌握基于商用 GIS 平台的二次开发技术），还要知其所以然（了解 GIS 底层开发技术），从而具备更高的综合素质及竞争能力。

当前 GIS 软件有商业 GIS 系统及开源 GIS 系统之分。GIS 商用软件功能强大，有完善的技术支持，提供封装好的、功能强大的类库，基于商用 GIS 库进行的二次开发效率高、难度低、资源丰富。但对于小型 GIS 开发人员，商用 GIS 价格过高，对于 GIS 学习者来说，由于商用 GIS 产品的源代码是保密的，无法深入学习 GIS 底层技术，开发技术含量低，难以深化对 GIS 的理解；而开源 GIS 是完全免费的，其源代码完全公开，可作为非商业用途使用，是开发小型 GIS 系统或 GIS 开发人员学习的宝贵资料。但开源 GIS 项目的帮助资料通常很少，且没有强大的技术支持。此外在众多 GIS 开源代码中，C 语言及 Java 平台的项目居多，.NET 平台的 GIS 开源项目较少。SharpMap 是基于 .NET 2.0 平台用 C# 语言开发的 GIS 开源项目，是一套简单易用的小型 GIS 平台，扩展性强，可以用于开发网络或桌面 GIS 应用程序。它支持多种 GIS 数据格式，支持空间查询，可渲染出精美地图。本书针对 SharpMap 的核心模块进行详尽的讲解，目的是为广大 .NET 平台的 GIS 开发人员提供服务，尽可能解决 SharpMap 开发文档匮乏的问题。

由于编者水平和经验有限，书中欠妥之处在所难免，恳请专家和广大读者不吝赐教。最后向为本书提出宝贵建议的专家、学者和同仁表示感谢。

编 者
2011 年 12 月

目 录

第 1 章 概述	1
1.1 SharpMap 的特性	2
1.2 SharpMap 支持的 GIS 数据格式	2
1.3 SharpMap 名称空间概述	3
1.4 SharpMap 中用到的第三方库	3
1.5 SharpMap 源代码下载	4
复习思考题	4
第 2 章 地图	5
2.1 Map 类概述	5
2.2 SharpMap 执行过程	5
2.3 SharpMap 创建地图示例	5
2.4 Map 类	7
2.4.1 Map 类的数据成员	7
2.4.2 Map 类的属性	7
2.4.3 Map 类的方法	14
2.4.4 Map 类的事件	19
复习思考题	19
第 3 章 地图控件	20
3.1 Tools 枚举	20
3.2 MapImage 类	20
3.2.1 MapImage 类的数据成员	20
3.2.2 MapImage 类的属性	21
3.2.3 MapImage 类的方法	23
3.2.4 MapImage 类的事件	31
复习思考题	32
第 4 章 图层	33
4.1 图层对象概述	33
4.2 ILayer 接口	34
4.2.1 ILayer 接口的属性	34

4.2.2 ILayer 接口的方法	34
4.3 Layer 抽象类	34
4.3.1 Layer 抽象类的属性	34
4.3.2 Layer 抽象类的方法	36
4.3.3 Layer 抽象类的事件	37
4.4 ICanQueryLayer 接口	37
4.5 矢量图层 VectorLayer	38
4.5.1 VectorLayer 的属性	38
4.5.2 VectorLayer 的方法	40
4.5.3 VectorLayer 的事件	45
4.6 注记图层 LabelLayer	45
4.6.1 LabelLayer 的属性	45
4.6.2 LabelLayer 的方法	49
4.7 Layer 集合	56
4.7.1 LayerCollection 的索引器	56
4.7.2 LayerCollection 的方法	57
复习思考题	58
第5章 绘制	59
5.1 ClipState 枚举	59
5.2 LabelBox 类	60
5.3 Label 类	60
5.3.1 Label 类的数据成员	60
5.3.2 Label 类的属性	61
5.3.3 Label 类的方法	61
5.4 LabelCollisionDetection 类	62
5.4.1 LabelCollisionDetection 类的方法	62
5.4.2 LabelCollisionDetection 类的代理	63
5.5 矢量数据渲染类 VectorRender	63
5.5.1 VectorRender 类的数据成员	63
5.5.2 VectorRender 类的属性	64
5.5.3 VectorRender 类的方法	64
5.6 主题	73
5.6.1 ITheme 接口	73
5.6.2 色彩混合类 ColorBlend	73
5.6.3 GradientThemeBase 类	79
5.6.4 GradientTheme 类	86
5.6.5 CustomTheme 类	87
复习思考题	88

第6章 样式	89
6.1 矢量图层样式 VectorStyle	89
6.1.1 VectorStyle 的数据成员	89
6.1.2 VectorStyle 的属性	90
6.1.3 VectorStyle 的方法	90
6.2 标注样式 LabelStyle	91
6.2.1 LabelStyle 的数据成员	91
6.2.2 LabelStyle 的属性	92
6.2.3 LabelStyle 的方法	92
复习思考题	93
第7章 数据	94
7.1 空间数据库连接池技术	94
7.1.1 数据连接对象 Connector	95
7.1.2 连接池管理 ConnectorPool	98
7.2 数据提供接口 IProvider	100
7.2.1 IProvider 的属性	100
7.2.2 IProvider 的方法	100
7.3 DbaseReader 类	102
7.3.1 DbaseReader 类的数据成员	102
7.3.2 DbaseReader 类的属性	103
7.3.3 DbaseReader 类的方法	104
7.4 数据提供者 ShapeFile	111
7.4.1 ShapeFile 的数据成员	112
7.4.2 ShapeFile 的属性	114
7.4.3 ShapeFile 的方法	116
7.5 MsSql 类	131
7.5.1 MsSql 类的数据成员	131
7.5.2 MsSql 类的属性	131
7.5.3 MsSql 类的方法	133
7.6 其他 Provider 类	144
7.7 FeatureDataSet 类	144
7.7.1 FeatureDataSet 类的属性	145
7.7.2 FeatureDataSet 类的方法	145
7.8 FeatureDataTable 类	148
7.8.1 FeatureDataTable 类的属性	148
7.8.2 FeatureDataTable 类的方法	148
7.8.3 FeatureDataTable 类的事件	150

7.9 FeatureDataRow 类	150
7.9.1 FeatureDataRow 类的属性	150
7.9.2 FeatureDataRow 类的方法	151
复习思考题	151
第 8 章 几何对象	152
8.1 几何对象抽象基类 Geometry	152
8.1.1 Geometry 的属性	152
8.1.2 Geometry 的方法	154
8.2 点对象 Point	159
8.2.1 Point 的数据成员	159
8.2.2 Point 的属性	159
8.2.3 Point 的方法	161
8.3 复合点对象 MultiPoint	166
8.3.1 MultiPoint 的属性	166
8.3.2 MultiPoint 的方法	167
8.4 线状几何形状的抽象类 Curve	170
8.4.1 Curve 的属性	170
8.4.2 Curve 的方法	171
8.5 多边形 Polygon	171
8.5.1 Polygon 的属性	171
8.5.2 Polygon 的方法	173
8.6 外包矩形框 BoundingBox	178
8.6.1 BoundingBox 的属性	178
8.6.2 BoundingBox 的方法	180
8.7 空间关系类 SpatialRelations	189
复习思考题	191
第 9 章 Windows 应用程序开发——WinFormSamples	192
9.1 数据	192
9.2 系统简介	193
9.3 代码分析	194
9.3.1 主窗体代码	194
9.3.2 数据访问代码	197
复习思考题	210
第 10 章 Windows 应用程序开发——DemoWinForm	211
10.1 数据	211
10.2 系统简介	211

10.3 代码分析.....	212
10.3.1 数据访问代码.....	212
10.3.2 主窗体代码.....	212
复习思考题.....	219
第11章 数据源扩展与图层对象扩展	220
11.1 DataTablePoint 类	220
11.1.1 DataTablePoint 类的数据成员	220
11.1.2 DataTablePoint 类的属性	220
11.1.3 DataTablePoint 类的方法	222
11.2 OgrProvider 类.....	227
11.2.1 OgrProvider 类的数据成员	228
11.2.2 OgrProvider 类的属性	229
11.2.3 OgrProvider 类的方法	232
11.3 GdalRasterLayer 类	243
11.3.1 GdalRasterLayer 类的数据成员	244
11.3.2 GdalRasterLayer 类的属性	244
11.3.3 GdalRasterLayer 类的方法	252
复习思考题.....	288
附录 书中多次引用的基本概念.....	289
参考文献.....	291

第 1 章 概 述

SharpMap 是一套简单易用的制图库，可以用于开发网络或桌面 GIS 应用程序。它支持多种 GIS 数据格式，支持空间查询，可渲染出精美的地图（见图 1-1）。SharpMap 是基

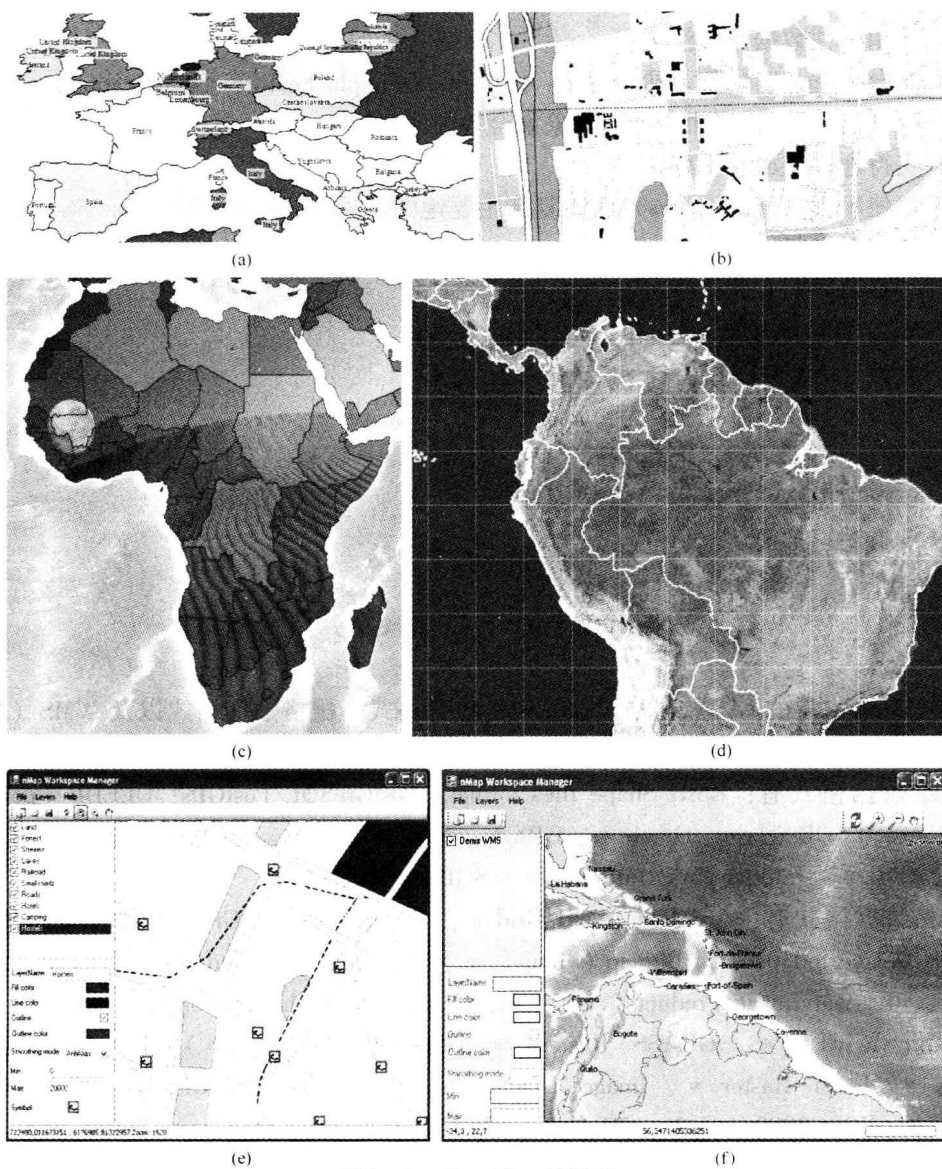


图 1-1 SharpMap 效果图

- (a) 人口密度分级地图；(b) 城市街区地图；(c) 采用纹理及半透明效果的地图；
(d) ECW 栅格影像；(e) 采用地图控件的桌面应用系统；(f) WMS 服务数据

于微软的 .NET 2.0 开发平台, 用 C# 语言开发的一套开源系统, 受到 GNUL (GNU Lesser General Public License) 许可保护。其网址为: <http://sharpmap.iter.dk/> 和 <http://www.codeplex.com/SharpMap>。

1.1 SharpMap 的特性

SharpMap 具有以下特性:

(1) 核心特性。

- 1) 支持多种 .NET 开发语言, 包括 C#、VB.NET、C++/CLI 和 J#等;
- 2) 使用属性数据作为注记;
- 3) 符合 OpenGIS 的简单要素规范 (OpenGIS Simple Features Specification), 支持点、线、面、多点、多线、多面、几何对象集合等要素类型;
- 4) 支持地图缩放及平移;
- 5) 可使用代理 (delegates) 编制复杂主题地图;
- 6) 能实时投影。

(2) 网络地图。

- 1) 支持使用 Http Handler 生成地图;
- 2) 支持 AJAX 地图控件;
- 3) 支持 WMS1.3.0 服务。

(3) 扩展功能。

- 1) 支持数据格式扩展;
- 2) 支持图层扩展。

1.2 SharpMap 支持的 GIS 数据格式

SharpMap 支持的 GIS 数据格式包括矢量数据格式、栅格数据格式以及 WMS (网络地图服务) 格式。

矢量数据格式有: ESRI Shape files format、PostgreSQL/PostGIS、OLEDB (points only)、Microsoft SQL Server、Oracle *、GPX *、MapInfo File *、TIGER *、S57 *、DGN *、CSV *、GML *、Interlis 1 *、Interlis 2 *、SQLite 和 ODBC *。

栅格数据格式有: Arc/Info ASCII Grid *、Arc/Info Binary Grid (.adf) *、Microsoft Windows Device Independent Bitmap (.bmp) *、ESRI .hdr Labelled *、ENVI .hdr Labelled Raster *、Envisat Image Product (.n1) *、Graphics Interchange Format (.gif) *、GMT Compatible netCDF *、GRASS Rasters *、TIFF/GeoTIFF (.tif) *、Erdas Imagine (.img) *、Idrisi Raster *、Image Display and Analysis (WinDisp) *、JPEG JFIF (.jpg) *、JPEG2000 (.jp2, .j2k) *、JPEG2000 (.jp2, .j2k) *、JPEG2000 (.jp2, .j2k) *、JPEG2000 (.jp2, .j2k) *、Erdas 7.x .LAN and .GIS *、Daylon Leveller Heightfield *、In Memory Raster *、NetCDF *、PCI .aux Labelled *、PCI Geomatics Database File *、Portable Network Graphics (.png) *、RadarSat2 XML (product.xml) *、

USGS SDTS DEM (CATD, DDF) *、Raster Matrix Format (.rsw, .mtw) *、SGI Image Format *、USGS ASCII DEM (.dem) * 和 X11 Pixmap (.xpm) * 等。

WMS (网络地图服务) 格式有: Version 1.0.0、Version 1.1.1、Version 1.3.0。网络地图服务 (Web Map Service, WMS) 是从地理信息动态产生, 具有地理空间位置数据的地图。地图本身并不是数据, WMS 产生的地图一般以图像格式提供, 如 PNG、GIF 或 JGPE; 或按 SVG (Scalable Vector Graphics) 或 Web CGM (Web Computer Graphics Metafile) 格式提供基于矢量的图形元素。

1.3 SharpMap 名称空间概述

- (1) SharpMap 名称空间只有一个类 (Map 类), 也是 SharpMap 的核心。
- (2) Converts 名称空间提供数据转换服务。
- (3) Forms 名称空间用于 Windows Form 编程, 包含 MapImage 控件。MapImage 空间是一个简单的 User Control (用户控件), 用于表达 Map 对象。
- (4) Geometries 名称空间包括了 SharpMap 要使用到的各种几何类及其接口类, 例如点、线、面等类, 是 SharpMap 的基础之一。所有几何对象都继承自 Geometry 这个抽象类, 其中定义了几何对象应该具备的公共操作, 例如大小、ID、外接矩阵、几何运算等。
- (5) Layers 名称空间包括了 ILayer 接口、Layer 集合类等, 代表地图的图层。Layer 是一个抽象类, 实现了 ILayer 接口。Layer 目前有 3 个子类, 分别是 VectorLayer、LabelLayer 和 WmsLayer, 分别代表 3 种不同数据类型的图层。
- (6) Data 名称空间包含了与空间要素相关的类, 如 FeatureDataSet、FeatureDataTable、FeatureDataRow、FeatureTableCollection 等。
- (7) Data.Providers 名称空间包括了 IProvider 接口和 Shape 文件、PostGIS 数据访问实现代码。该名称空间为 SharpMap 提供数据读 (写) 支持, 通过面向接口的设计, 可以方便地扩展各类数据格式。
- (8) Rendering 名称空间用于提供绘制空间数据的功能, 目前包括矢量绘制对象和几个专题图绘制对象, 该类将几何对象根据其 Style 设置, 绘制到 System.Drawing.Graphics 对象上。
- (9) Styles 名称空间主要提供了图层的样式类, 例如线样式、点样式、填充样式等。
- (10) Utilities 名称空间包括 Algorithms 类 (目前仅实现了一个方法)、Providers 类 (是数据提供对象 Provider 的一个 Helper 类, 应用了反射机制)、Surrogates 类 (主要用于系统的 Pen 和 Brush 的序列化)、Transform 类 (提供了从图片坐标到地理坐标的互相变换, 即桌面 GIS 的二次开发中经常使用的屏幕坐标和地理坐标的转换, 主要用于地图的渲染、交互操作等)。
- (11) Utilities.SpatialIndexing 名称空间用于对象的空间索引。
- (12) Web 名称空间实现了 HttpHandler 和 Caching 类, 用于网络环境。Web.UI.Ajax 提供对 Ajax 的支持。

1.4 SharpMap 中用到的第三方库

SharpMap 中用到的第三方库如下:

- (1) ProjNET。这是一套基于微软 .NET2.0 的开源地图投影库, 支持基准面转换; 支持大地坐标系、地心坐标系以及投影坐标系。其网址为: <http://projnet.codeplex.com/>。

(2) BruTile。这是一套 C#语言开发的支持 GIS 瓦片 (tile) 数据服务的开源代码库。其网址为: <http://brutiles.codeplex.com/>。

(3) NetTopologySuite。这是一套快速、可靠的基于 .NET 的平台,用于处理 GIS 空间数据的拓扑关系,支持对二维线性几何形状进行拓扑运算。其网址为: <http://code.google.com/p/nettopology>。

(4) nunit.framework。这是一套基于 .NET 的单元测试平台。其网址为: www.nunit.org/。

1.5 SharpMap 源代码下载

打开网站 <http://www.codeplex.com/SharpMap>, 在“Source Code”页面下,是项目各个版本的列表,显示各个版本的更新集编号 (Change Set)、发布日期、下载次数以及更新说明。点击页面右边“Latest Version”文字下的链接 Download,可下载最新的项目打包文件。页面中点击更新集编号,可下载相应版本。下载后文件名为 sharpmap - x x x x x.zip,其中 x x x x x 是一个 5 位数字,表示当前下载文件的更新号。本书使用版本的更新集编号是 78712,这是一个较稳定的版本,会与最新版有所不同,请读者注意。

下载的压缩文件中,有三个目录:一个是 Branches,为 SharpMap 的一些分支版本;一个叫 BuildProcessTemplates,是一些预定义的项目编译模板;第三个目录 Trunk,是包含项目代码的主目录,也是本书讲述的内容。

Trunk 目录下,有三个 Visual Studio 项目文件,即 SharpMap.sln、SharpMap.VS2008.sln、SharpMap.VS2010.sln,分别对应 Visual Studio 2005、Visual Studio 2008、Visual Studio 2010 三个版本。本书使用的是 SharpMap.VS2010.sln,建议使用 VS2010。

Trunk 目录还包含如下子目录:

- (1) ExternalReferences。它是项目用到的第三方库文件。
- (2) SharpMap。它是核心项目,包括 Map、Layer、Data、Geometries、Rendering、Style 等主要内容。
- (3) SharpMap.UI。它包括地图控件。
- (4) SharpMap.Extensions。它包括一些数据、图层扩展对象。
- (5) UnitTests。它用于管理单元测试。
- (6) SharpMap.SqlServerSpatialObjects。空间数据库,底层使用 MS SQL Server 的空间对象机制。
- (7) PostgisDemoDb、MsSqlSpatialDemoDb。它们是使用 PostGIS、MsSqlSpatila 空间数据库的例子项目。
- (8) DemoWebSite、DemoWinForm、WinFormSamples。它们是三个综合实例,第一个为 Web 应用,其他两个为桌面系统。
- (9) ExampleCodeSnippets。它是 SharpMap 的代码段项目。

复习思考题

- 1-1 SharpMap 的功能和特性是什么,它支持的主要数据格式有哪些?
- 1-2 SharpMap 主要名称空间的功能是什么?

第 2 章 地 图

2.1 Map 类概述

Map 是 Sharp 的核心类之一，位于 SharpMap 名称空间下。Map 是装载地图图层 Layer 的容器，通过创建 Map 类的实例来生成地图，它由包含 Layer 对象的集合组成。用户可以通过创建 Map 对象的实例来得到地图，并通过 GetMap 方法来得到渲染（render）后的地图图形。

2.2 SharpMap 执行过程

在分析 SharpMap 之前，应该知道它的执行逻辑过程。典型的例子就是加载一个 Shape 文件的过程以及当地图的显示范围变化时，系统将要执行什么操作来更新地图视图。

Shape 文件只是数据源的一种，所以有一个访问数据源的接口来支撑这个数据访问层，接口对外界暴露一些访问数据的函数。毕竟在系统的高层（地图和图层层上）并不会关注数据源来自哪里。这些我们会在第 7 章数据中做详细的分析。

2.3 SharpMap 创建地图示例

我们来看一下如何构造一个地图对象，完整代码如下：

```
SharpMap.Map myMap = new SharpMap.Map();
myMap.MinimumZoom = 100;
myMap.BackgroundColor = Color.White;
SharpMap.Layers.VectorLayer myLayer = new SharpMap.Layers.VectorLayer("My layer");
string shpLayerFullPath = @"C:\data\Lotlines.shp";
myLayer.DataSource = new ShapeFile(shpLayerFullPath);
myLayer.FillStyle = new SolidBrush(Color.FromArgb(240,240,240));
myLayer.OutlineStyle = new Pen(Color.Blue, 1);
myLayer.Style.Line.Width = 2;
myLayer.Style.Line.Color = Color.Black;
myLayer.Style.Line.EndCap = System.Drawing.Drawing2D.LineCap.Round;
myLayer.Style.Line.StartCap = layRailroad.LineStyle.EndCap;
myLayer.Style.Line.DashPattern = new float[] { 4.0f, 2.0f };
myLayer.Style.EnableOutline = true;
myLayer.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.AntiAlias;
```

```
myLayer.MaxVisible = 40000;  
myMap.Layers.Add(myLayer);  
myMap.Center = new SharpMap.Geometries.Point(725000, 6180000);  
myMap.Zoom = 1200;  
myMap.Size = new System.Drawing.Size(300, 200);  
System.Drawing.Image imgMap = myMap.GetMap();
```

从上面的简单示例中我们可以看到, 要构造一个完整的地图对象, SharpMap 需要经过以下几个步骤:

第一步, 实例化一个 Map 对象, 作为地图的容器。

```
SharpMap.Map myMap = new SharpMap.Map();
```

第二步, 设置地图显示属性。

```
myMap.MinimumZoom = 100; // 最小地图宽度为 100  
myMap.BackgroundColor = Color.White; // 背景颜色设置为白色
```

第三步, 实例化图层, 将这个图层加入到地图对象中。

```
SharpMap.Layers.VectorLayer myLayer = new SharpMap.Layers.VectorLayer("My layer");
```

第四步, 设置图层的数据源。

```
string shpLayerFullPath = @"C:\data\Lotlines.shp";  
myLayer.DataSource = new ShapeFile(shpLayerFullPath);
```

第五步, 设置图层的渲染样式。

将图层的填充样式设置成灰色 Color.FromArgb(240, 240, 240), 用的是 .NET 的画刷类 SolidBrush。

```
myLayer.FillStyle = new SolidBrush(Color.FromArgb(240, 240, 240));
```

将外框线设置成蓝色, 使用的是 .NET 的画笔类 Pen。

```
myLayer.OutlineStyle = new Pen(Color.Blue, 1);
```

设置线条的宽度为两个像素。

```
myLayer.Style.Line.Width = 2;
```

设置线条的颜色是黑色。

```
myLayer.Style.Line.Color = Color.Black;
```

设置线的终止线头为圆头, 使用 System.Drawing.Drawing2D 名称空间下的线头。

```
myLayer.Style.Line.EndCap = System.Drawing.Drawing2D.LineCap.Round;
```

设置线的起始线头为圆头。

```
myLayer.Style.Line.StartCap = myLayer.Style.Line.EndCap;
```

设置虚线样式。

```
myLayer.Style.Line.DashPattern = new float[] { 4.0f, 2.0f };
```

允许外包框。

```
myLayer.Style.EnableOutline = true;
```

设置平滑模式为抗锯齿。

```
myLayer.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.AntiAlias;
```

最大显示比例尺度为 40000。

```
myLayer.MaxVisible = 40000;
```

第六步，加载图层，将图层载入地图中。

```
myMap.Layers.Add(myLayer);
```

第七步，设置显示地图属性。

设置地图中心为点 (725000, 6180000)。

```
myMap.Center = new SharpMap.Geometries.Point(725000, 6180000);
```

地图的宽度为 1200。

```
myMap.Zoom = 1200; //Set zoom level
```

设置地图的屏幕大小。

```
myMap.Size = new System.Drawing.Size(300,200);
```

第八步，渲染地图，调用 Map 对象的 GetMap() 方法渲染地图。

```
System.Drawing.Image imgMap = myMap.GetMap();
```

以上代码中的用法将在本书后面的内容中进行详细的解释。

2.4 Map 类

2.4.1 Map 类的数据成员

NumberFormatEnUs 是 Map 类的数据成员，是 Map 类中定义的一个静态变量。它的值反映了当前 SharpMap 使用的数字格式信息。完整定义代码如下：

```
public static NumberFormatInfo NumberFormatEnUs = new CultureInfo("en - US", false)
    .NumberFormat;
```

它把数字格式设置为“美国”数字，并使用默认的格式（useUserOverride 设置为 false）。

2.4.2 Map 类的属性

2.4.2.1 Layers

Layers 获取或者设置地图图层 Layer 对象的集合，它是一个 LayerCollection 的集合，具

有增加、删除、定位图层的功能，代码如下：

```
private LayerCollection _Layers;
public LayerCollection Layers
{
    get { return _Layers; }
    set
    {
        int iBefore = 0;
        if (_Layers != null)
            iBefore = _Layers.Count;
        _Layers = value;
        if (value != null)
        {
            if (LayersChanged != null)
                LayersChanged();
            if (MapViewOnChange != null)
                MapViewOnChange();
        }
    }
}
```

从以上代码可以看到，get 访问器直接返回地图几何对象 LayerCollection；在 set 访问器中，若设置的值不为空，分别检查 LayersChanged、MapViewOnChange 事件是否为空，若不为空则分别触发图层变化事件 LayersChanged（）和地图视图变化事件 MapViewOnChange（），如用户注册了相应事件，相应事件处理函数会被调用。

2.4.2.2 MapTransform

MapTransform 为变换矩阵，其利用矩阵运算对地图图像进行变形，代码如下：

```
private System.Drawing.Drawing2D.Matrix _MapTransform;
internal System.Drawing.Drawing2D.Matrix MapTransformInverted;
public Matrix MapTransform
{
    get { return _MapTransform; }
    set
    {
        _MapTransform = value;
        if (_MapTransform.IsInvertible)
        {
            MapTransformInverted = _MapTransform.Clone();
            MapTransformInverted.Invert();
        }
        else
            MapTransformInverted.Reset();
    }
}
```