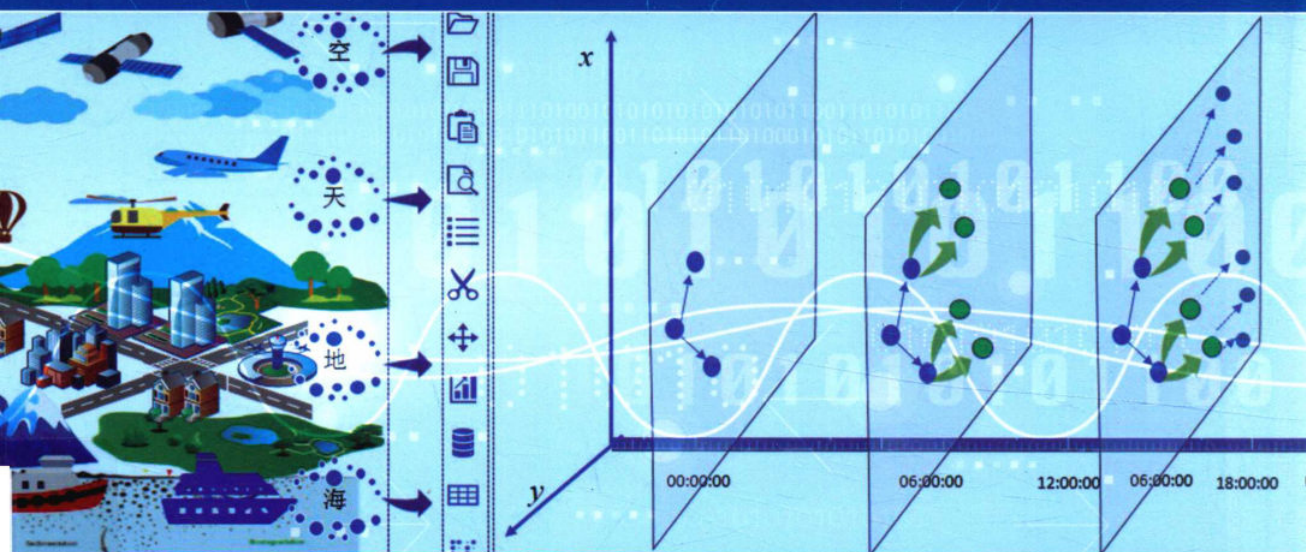


测绘程序设计 (上册)

主编 李英冰



WUHAN UNIVERSITY PRESS

武汉大学出版社

- 责任编辑 / 鲍 玲
- 责任校对 / 李孟潇
- 版式设计 / 马 佳
- 封面设计 / 王荆强

Introduction

I n t r o d u c t i o n

《测绘程序设计》共分为5篇，分为上下两册，本书是上册，包含第1篇和第2篇。第1篇是教学篇，共9章内容，简要介绍了控制台应用程序开发、桌面应用程序开发和网络程序开发，并且包含文件读写、图形图像处理、数据库操作等内容。在每章后面给出了一些教学视频，便于学生实战模仿，并提供往届参赛学生的优秀作品，供参考学习。第2篇是基础篇，共12章，考查文件读写、简单测绘算法实现等编程能力，难易程度相当于夏令营优秀营员选拔、研究生复试，以及测绘程序设计课程期末上机测试。

本书可作为测绘地理信息专业本科学生的参赛实践教材，也可作为其他相关专业对测绘编程感兴趣的学生的参考用书。



武汉大学出版社
官方微信平台



武汉大学出版社
天猫旗舰店

ISBN 978-7-307-21090-



9 787307 210905 >

定价: 38.00元

高等学校测绘工程系列教材

测绘程序设计

(上册)

主 编 李英冰

副主编 邹进贵 车德福 戴吾蛟 吴杭彬



武汉大学出版社

图书在版编目(CIP)数据

测绘程序设计.上册/李英冰主编.—武汉:武汉大学出版社,2019.9
高等学校测绘工程系列教材
ISBN 978-7-307-21090-5

I.测… II.李… III.工程测量—高等学校—教材 IV.TB33

中国版本图书馆 CIP 数据核字(2019)第 169325 号

责任编辑:鲍玲 责任校对:李孟潇 版式设计:马佳

出版发行:武汉大学出版社 (430072 武昌 珞珈山)
(电子邮箱: cbs22@whu.edu.cn 网址: www.wdp.com.cn)

印刷:湖北民政印刷厂

开本:787×1092 1/16 印张:14.5 字数:341千字

版次:2019年9月第1版 2019年9月第1次印刷

ISBN 978-7-307-21090-5 定价:38.00元

版权所有,不得翻印;凡购我社的图书,如有质量问题,请与当地图书销售部门联系调换。

编 委 会

主 编 李英冰 武汉大学
副主编 邹进贵 武汉大学
车德福 东北大学
戴吾蛟 中南大学
吴杭彬 同济大学

编 委 (排名不分先后)

李英冰 武汉大学
邹进贵 武汉大学
车德福 东北大学
戴吾蛟 中南大学
吴杭彬 同济大学
李 军 武汉大学
王同合 解放军战略支援部队信息工程大学
黄劲松 武汉大学
詹总谦 武汉大学
韩 亮 山西大同大学
马明舟 大连理工大学
孙佳伟 西京学院
陈艳红 河北地质大学
赵兴旺 安徽理工大学
张云生 中南大学
李阳腾龙 成都理工大学
廖振修 安徽建筑大学

梁 丹 浙江农林大学
温扬茂 武汉大学
高 祥 安徽农业大学
钱如友 滁州学院
张 瑞 华南农业大学
刘国栋 重庆交通大学
王红梅 山西工程技术学院
张金亭 武汉大学
雷 斌 华北水利水电大学
闻道秋 东南大学
朱晓峻 安徽大学
王中元 中国矿业大学
隋 心 辽宁工程技术大学
肖海红 河南工程学院
蔡来良 河南理工大学
刘 宁 长安大学
王胜利 山东科技大学

中第59章至第61章选自武汉大学的学生作品(第59章是硕士生黄孝等同学的研究成果,第60章是本科生白晓斌等同学的作品,第61章是本科生何朝博等同学的作品),本卷还向测绘新技术新方法的编程实践、应用案例、工程应用、教学和研究生科洪选题。第1章



WUHAN UNIVERSITY PRESS

武汉大学出版社

序 言

随着现代科学与技术的飞速发展，特别是移动互联网、云计算和大数据等现代技术的兴起，测绘数据获取手段越来越多样，需要处理的数据类型越来越复杂，计算机已经成为测绘数据处理的基本工具，程序设计已经成为测绘工程专业学生所必备的基本能力。由于测绘地理信息专业所需知识的理论性很强，在程序设计时不仅需要很强的编程能力，还必须具备正确的测绘理论思维。在测绘编程实践学习中，如何设计合适的数据输入方式，以及难度适中的实践算法，这对当前测绘地理信息专业学生来说是比较困难的，但是，一旦攻克这个突破口，学生的编程实战能力将得到快速提升。

测绘程序设计已经受到测绘地理信息专业教育主管部门的高度重视，许多院校开设了相关课程，测绘程序设计在全国大学生测绘技能大赛中占 30% 的比重，计划将来还准备推出围绕测绘、遥感、地理信息与导航等方面开发相关软件系统或软硬件集成系统的测绘创新开发比赛。江苏、河南等省也推出了相关的测绘程序设计比赛。武汉大学的测绘技能大赛设有程序比赛专项，并在大学生夏令营优秀营员选拔，以及硕士生和博士生复试等都设置了编程环节，为优秀人才的选拔起到了积极作用。

为了进一步提高全体测绘人员的编程水平，我们组织了来自武汉大学、东北大学、中南大学、同济大学、中国人民解放军战略支援部队信息工程大学等 27 所高校的 34 位教师共同编写本书。在 2019 年 3 月 29 日于武汉召开了一次专门研讨会，来自全国大学生测绘技能大赛工作组的邹进贵教授、翟翊教授、宋卫东教授、程效军教授、车德福教授、邹峥嵘教授，以及本书编委会相关成员共 13 人，对本书的选题、组织形式、内容等进行了广泛的研讨。

本书分为上下册，共 5 篇。第 1 篇是教学篇，共 9 章，内容涵盖了控制台应用程序开发、桌面应用程序开发和网络程序开发，并且包含文件读写、图形图像处理、数据库操作等内容。在每章后面给出了一些教学视频，便于学生实战模仿，并提供往届参赛学生的优秀作品，供参考学习。第 2 篇是基础篇，共 12 章，主要是一些培养学生文件读写、简单测绘算法实现等编程能力的实例，难易程度相当于夏令营优秀营员选拔、研究生复试，以及测绘程序设计期末上机测试。第 3 篇是进阶篇，共 12 章，内容包括考查文件读写、用户界面设计、较为复杂的测绘算法的实现等能力的实例，可用于测绘程序教学和实习、竞赛人才选拔等。第 4 篇是竞赛篇，共 18 章，内容包括考查复杂测绘程序开发和团队合作能力的实例，服务于全国、省级、院校级测绘编程竞赛。第 5 篇是创新篇，共 10 章，其中第 59 章至第 61 章选自武汉大学的学生作品（第 59 章是博士生黄奎等同学的研究成果、第 60 章是本科生白璐斌等同学的作品、第 61 章是本科生何雨情等同学的作品），本篇面向测绘新技术新方法的编程实现，服务于大学生创新创业训练和研究生科研选题。第 1 篇

和第2篇属于上册内容。第3、4、5篇属于下册内容。

本书的编撰得到同事、同学和朋友的大力支持，在此感谢大家给予本书的各种贡献，感谢武汉大学出版社王金龙的帮助。需要感谢的人很多，限于篇幅不一一列出。本书涉及的内容庞杂，难免存在错误，欢迎批评指正。

编著者

2019年5月

目 录

一、教学篇	1
第 1 章 C#概述	3
第 2 章 类型、运算符、表达式	18
第 3 章 语句与方法	34
第 4 章 类与对象	50
第 5 章 窗体应用程序	76
第 6 章 流与泛型	98
第 7 章 ADO. NET 数据库操作	117
第 8 章 网络编程基础	136
第 9 章 ASP. NET 网络编程	158
二、基础篇	181
第 10 章 出租车轨迹数据计算	183
第 11 章 反距离加权插值	187
第 12 章 线状要素数据的压缩算法	190
第 13 章 最短路径计算	193
第 14 章 时间系统转换	196
第 15 章 面积计算	199
第 16 章 滑坡体的变形速度与应变计算	201
第 17 章 矩阵卷积计算	204
第 18 章 空间直角坐标转换为站心直角坐标	207
第 19 章 电离层改正计算	210
第 20 章 对流层改正计算	215
第 21 章 矩阵基本运算	219

一、教学篇

(C#语言)

负责人：李英冰、李军

本篇内容可用于测绘专业本科生的编程教学。本篇分为9章内容，涵盖了控制台应用程序开发、桌面应用程序开发和网络程序开发。在综合练习中，特别提供了一些视频学习内容，可以帮助学生快速学会编程。

相关的资源包括：

1. 教学视频：<https://pan.baidu.com/s/1c60VJkzoOHJzrExMF1o9ug>。

2. 学生的大作业：

(1)2017：https://pan.baidu.com/s/1nRV8bLVYIY8n_em6s-Ig5_w。

(2)2016：<https://pan.baidu.com/s/1MtnfGbhAQ2v3MYCqt9ipqg>。

(3)2015：https://pan.baidu.com/s/1_EyOVqNPzWfGbFBa8g7SHQ。

3. 课件：本篇内容的教学课件可免费提供给授课老师，请联系李英冰(ybli@sgg.whu.edu.cn)。

第1章 C# 概述

一、基本知识

























1. 计算机语言排行榜

电脑每一次动作、每一个步骤，都是按照用计算机语言编好的程序执行的。程序是计算机要执行的指令集合，而程序全部都是用我们所掌握的计算机语言编写的。人们要控制计算机一定要通过计算机语言向计算机发出命令。

计算机语言的种类非常多，表 1-1 是 TIOBE 网站 2019 年 3 月给出的计算机语言使用排行榜。

表 1-1

计算机语言使用排行榜

排名	语言	比率	适用性
1	Java	14.880%	  
2	C	13.305%	  
3	Python	8.262%	 
4	C++	8.126%	  
5	Visual Basic. NET	6.426%	  
6	C#	3.267%	  
7	JavaScript	2.426%	 
8	PHP	2.420%	
9	SQL	1.926%	
10	Objective-C	1.681%	  

注：表示桌面应用，表示网络应用，表示移动设备开发，表示硬件驱动开发。

2. .NET 框架、编程语言和开发工具

C#是运行于 .NET 框架的设计语言，它是一种类型齐全、现代、简单、面向对象的编程语言。 .NET 框架是由微软开发的多语言组件开发和执行环境，是跨语言、跨平台的统一编程环境。

.NET 框架、编程语言和开发工具关系如图 1.1 所示。公共语言运行时 (CLR) 定位、加载和管理 .NET 类型，是应用程序的执行引擎类库。公共类型系统 (CTS) 是关于“公共类型”的一个文档说明书。CTS 体现了核心编程思想：面向对象的语言 (如 C#) 把类型推广到整个程序中。基类库 (BCL) 封装了各种基本类型，每个类型都对应着一些功能，如线程、文件输入/输出、图形绘制、数据访问、Web 窗体以及与各种外部硬件设备的交互。BCL 通过各种名字空间为开发者提供了所需的各种服务。例如，Collections 名字空间包括链表、哈希表等集合类型；System.IO 名字空间就包含输入/输出基本类。主要功能最强的开发工具是微软的 Visual Studio，当前最新版本是 2019。



图 1.1 .NET 框架

托管是融于 CLR 中的一种新的编程理念，托管代码是编译器编译的代码。编译器把代码编译成中间语言(IL)，而不是能直接在用户的电脑上运行的机器码。托管代码在公共语言运行库 (CLR) 中运行，这个运行库为运行代码提供了各种各样的服务。随着程序集的运行，运行库会持续地提供各种服务，例如安全、内存管理、线程管理等，这个程序被“托管”在运行库中。

非托管代码不能享受一些运行库所提供的服务，例如安全和内存管理等。如果非托管代码需要进行内存管理等服务，就必须显式地调用操作系统的接口。

3. 类型

类型 (type) 是一般性的术语，它指的是 {类、接口、结构、枚举、委托} 里的任意一个成员。C# 的编程思想是：一切都是类型、对象或值，而对象或值也都源自类型。

CTS 分为两个大类：值类型和引用类型，同时这两种类型之间还可以进行强制转换，这种转换被称为 Boxing (装箱) 和 UnBoxing (拆箱)。值类型继承自 ValueType 类，变量直接

存储数据，实例是被分配在栈中，并且永远不可能为空；引用类型继承自 `Object`，存储的是数据内存的地址，实例是被分配在可以进行垃圾回收的堆中。

二、程序结构

1. 第一个样例程序

【例 1】计算圆的周长。

```
1 using System;
2
3 //第一个样例程序
4 namespace Demo
5 {
6     class Program
7     {
8         static void Main(string[] args)
9         {
10            double r=10;
11            Console.WriteLine(2 * 3.14 * r);
12        }
13    }
14 }
```

(1) 注释行。注释代码有两种方法，第一种是 `/*` 和 `*/` 之间的内容都被注释掉，C 程序可采用这种注释方式，在 C# 中不推荐。第二种方法是 `//` 后的内容被注释掉。

(2) 命名空间。在第 1 行中，`System` 是命名空间，用 `using System` 使该命名空间中的所有成员都可直接使用，例如 `Console` 是该命名空间中的类。第 4 行定义命名空间 `Demo`，命名空间使用 `namespace` 声明，并使用 `{ }` 来界定命名空间的作用域。

(3) 类。第 6 行声明了类 `Program`，`class` 是类关键字。`Program` 是系统定义的类，默认命名为 `Program`，如果需要的话我们完全可以修改这个名字。类是一种构造，通过使用该构造，可以将变量、方法和事件组合在一起。类和对象是面向对象语言的核心思想。

(4) `Main` 方法：`Main` 方法是 C# 控制台或窗口应用程序的入口点，是执行程序 (.exe) 的入口点，程序控制流在该处开始和结束。`Main` 必须是静态方法，需要在类或结构内声明，但不要求封闭类或结构是静态的。`Main` 的返回类型有两种：`void` 或 `int`。

(5) 输出信息到控制台。`Console` 类提供从控制台读入、向控制台写出等方法，如第 11 行向控制台输出计算结果。

(6) 程序保存与编译。C# 的源文件通常以 `.cs` 结尾，如可以保存为 `hello.cs`。编译程

序: csc hello. cs, 将生成一个 hello. exe 执行文件。

2. 源程序和程序集

C#的源程序结构如图 1.2 所示。源程序通过命名空间进行代码组织, 在命名空间内, 用大括号界定范围, 大括号中的内容称为类型(类、接口、结构等)。类型中定义的内容称为成员(方法、属性、成员变量等)。

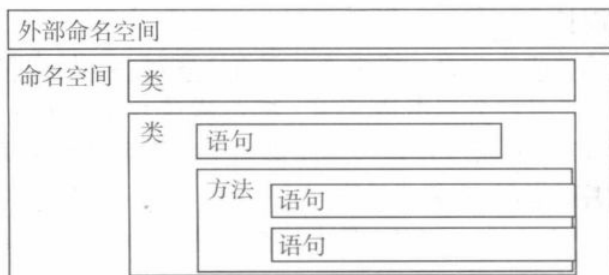


图 1.2 程序结构示意图

程序集是指经由编译器编译得到的, 供 CLR 进一步编译执行的中间产物, 在 Windows 系统中, 它一般表现为 .dll 或者是 .exe 的格式。一个程序集可以包含任意个命名空间, 每个命名空间又可以包含多种类型, 每个类型可包含多个成员。

3. 命名空间

命名空间是一种代码组织和重构形式, 是相关类型的分组符号。命名空间名称要唯一。BCL 中常用的命名空间有:

System	内建数据、数学计算、垃圾收集器
System. Drawing	处理图形和绘图, 包括打印
System. Data	处理数据存取和管理, ADO. NET
System. IO	管理对文件和流的同步和异步访问
System. Windows	处理基于窗体的窗口的创建
System. Reflection	包含从程序集读取元数据的类
System. Threading	包含用于多线程编程的类
System. Collections	包含定义各种对象集的接口和类

三、集成开发环境

1. Visual Studio 集成开发环境(IDE)

图 1.3 是 Visual Studio 2015 的集成开发环境(IDE), 主要包含菜单栏、工具栏、代码/视图编辑区、解决方案管理器、工具箱、属性窗口、输出窗口。IDE 将软件开发项目

中涉及的任务合并到一个集成开发环境中，同时提供创新功能，便于高效开发任何应用程序。

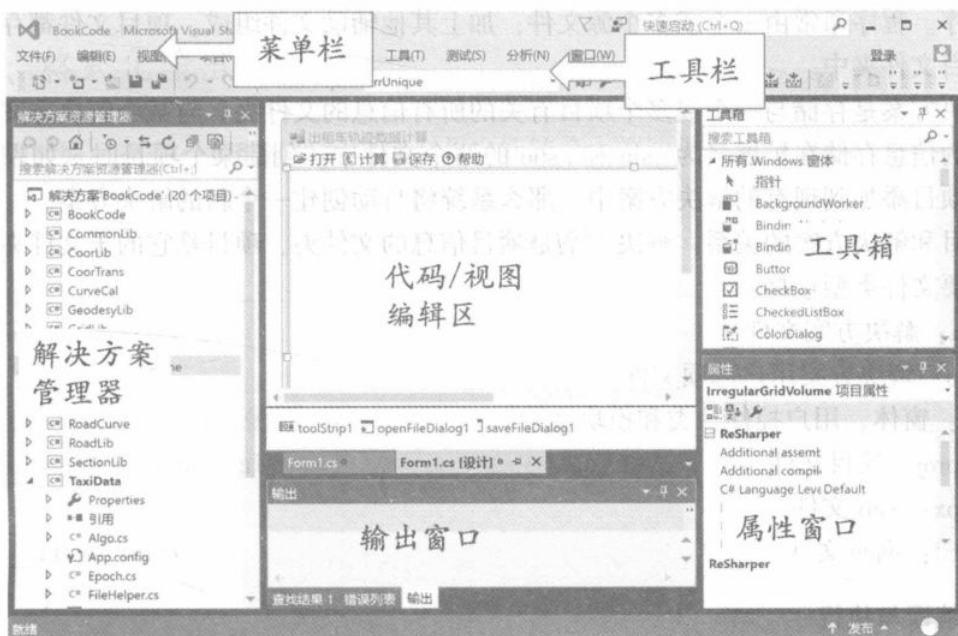


图 1.3 集成开发环境

1) 统一且可定制

IDE 将所有开发任务合并到一个工具中，通过功能强大的集成调试器、IntelliTrace、性能与诊断中心以及分析工具，提供生成和优化应用程序所需的全部功能。IDE 具有深度可扩展性，迄今已有数千项扩展，允许开发人员与合作伙伴集成自己的工具和软件开发工具包。

2) 代码编辑器

代码编辑器支持 C#、VB.NET、C++、HTML、JavaScript、Python、SQL 等语言，全部都具有语法突出显示与 IntelliSense 代码完成功能。使用代码映射，可以通过直观显示以更轻松地了解更复杂的源代码，其他高效功能还包括：查看定义(用于行内引用检查)、强大的代码重构工具以及检测重复代码的功能。

3) 开发平台支持

IDE 提供了统一的开发环境构建针对全部 Microsoft 平台(如桌面、Windows 应用商店和 Windows Phone 应用)以及移动 Web 应用、Web 应用程序和云服务等的应用程序。无论选择哪种编程语言、针对哪种应用程序，是现代化现有客户端/服务器应用程序还是跨设备的云服务，Visual Studio 都可以提供所需的开发环境。

4) 调试与诊断

IDE 有一组可以对所有应用程序进行调试和诊断的现代工具。通过设置断点，优化代码，解决问题，可实现在本地或远程设备上调试项目，通过丰富的可操作信息了解异常。

2. 解决方案和项目

项目是构成某个程序的全部组件的容器，该程序可能是控制台程序、窗口程序或某种别的程序。程序通常由一个或多个源文件，加上其他辅助文件组成。项目文件都存储在相应的项目文件夹中。

解决方案是存储与一个或多个项目有关的所有信息的文件夹。与某个解决方案中的项目有关的信息存储在扩展名为 .sln 和 .suo 的两个文件中。创建某个项目时，如果没有选择将该项目添加到现有的解决方案中，那么系统将自动创建一个新的解决方案。

项目和解决方案的关系：解决方案是项目信息的文件夹，项目是它的子文件夹。

主要文件类型包括：

- .sln：解决方案文件
- .suo：解决方案用户选项文件
- .cs：窗体、用户控件、类和模块文件
- .csproj：项目文件
- .aspx：web 文件
- .html：网页文件

3. 编译与执行

程序的编译与执行过程如图 1.4 所示。编译时使用 .NET 编译器生成 dll 或 exe 文件，即将源程序转换为由 CPU 执行的计算机代码。

第一次编译：使用 .NET 编译器生成 dll 或 exe 文件时，生成的模块会被打包成一个程序集。第二次编译：程序集包含公共中间语言(CIL)代码，它只有在绝对必需的情况下才会编译为特定平台的指令，第二次编译称为即时编译(JIT)。即时编译器将 CIL 代码转换为可以直接由 CPU 执行的计算机代码。

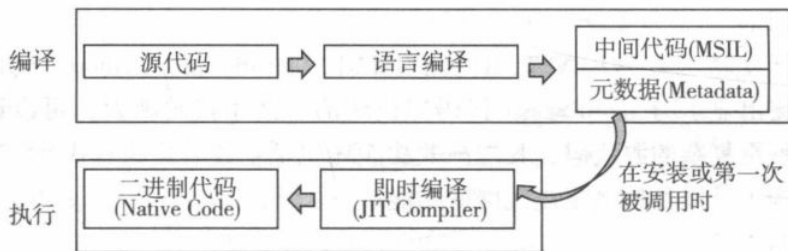


图 1.4 程序的编译与执行

生成应用程序时创建的文件包括：

(1) 生成解决方案(或项目)后，方案文件夹中出现了一个 debug 文件夹，该文件夹包含构建项目时产生的输出。其中：*.exe 表示可执行文件；*.ilk 表示在重新构建项目时被连接器使用；*.pdb 表示包含在调试模式中执行程序时要使用的调试信息。在调试模式(Debug 版)，可以动态检查程序执行过程中所生成的信息。

(2)程序的 Debug 版本和 Release 版本。Debug 版本：包括帮助用户调试程序的信息，使用 Debug 版，可以在出现问题时单步执行代码，以检查程序中的数据值。Release 版本：不包括调试信息，并且打开了编译器的代码优化选项，以提供最高效的可执行模块。

四、格式化输出

【例 2】编写程序，实现极坐标向笛卡儿坐标转换，方法见公式(1.1)。

$$\begin{cases} x = r\sin\theta \\ y = r\cos\theta \end{cases} \quad (1.1)$$

```

1  using System;
2
3  //极坐标转换为直角坐标
4  namespace ConsoleApp1
5  {
6      class Program
7      {
8          static void Main(string[] args)
9          {
10             double r = 1000.0, theta = 45.0;
11             Console.WriteLine("r = {0:E};theta = {1:F} ", r, theta);
12             theta *= Math.PI / 180;
13             double x = r * Math.Sin(theta);
14             double y = r * Math.Cos(theta);
15             Console.WriteLine("x = {0:0.000};y = {1:0.000}", x, y);
16             Console.ReadKey();
17         }
18     }
19 }

```

程序编译后执行，会在控制台上输出以下结果：

```
r=1.00000E+003; theta=45.00
```

```
x=707.107; y=707.107
```

以上结果的输出格式分别是源程序中第 11 行和第 15 行相关语句的结果。

格式化输出见表 1-2 和表 1-3，输出的一般格式：

```
{ N [, M ] [ : 格式码 ] }
```

N：指定参数序列中的输出序号，比如 {0}，{1}，{2} 等。

M：指定参数输出的最小长度。如果参数长度小于 M，则空格填充；若大于等于 M，