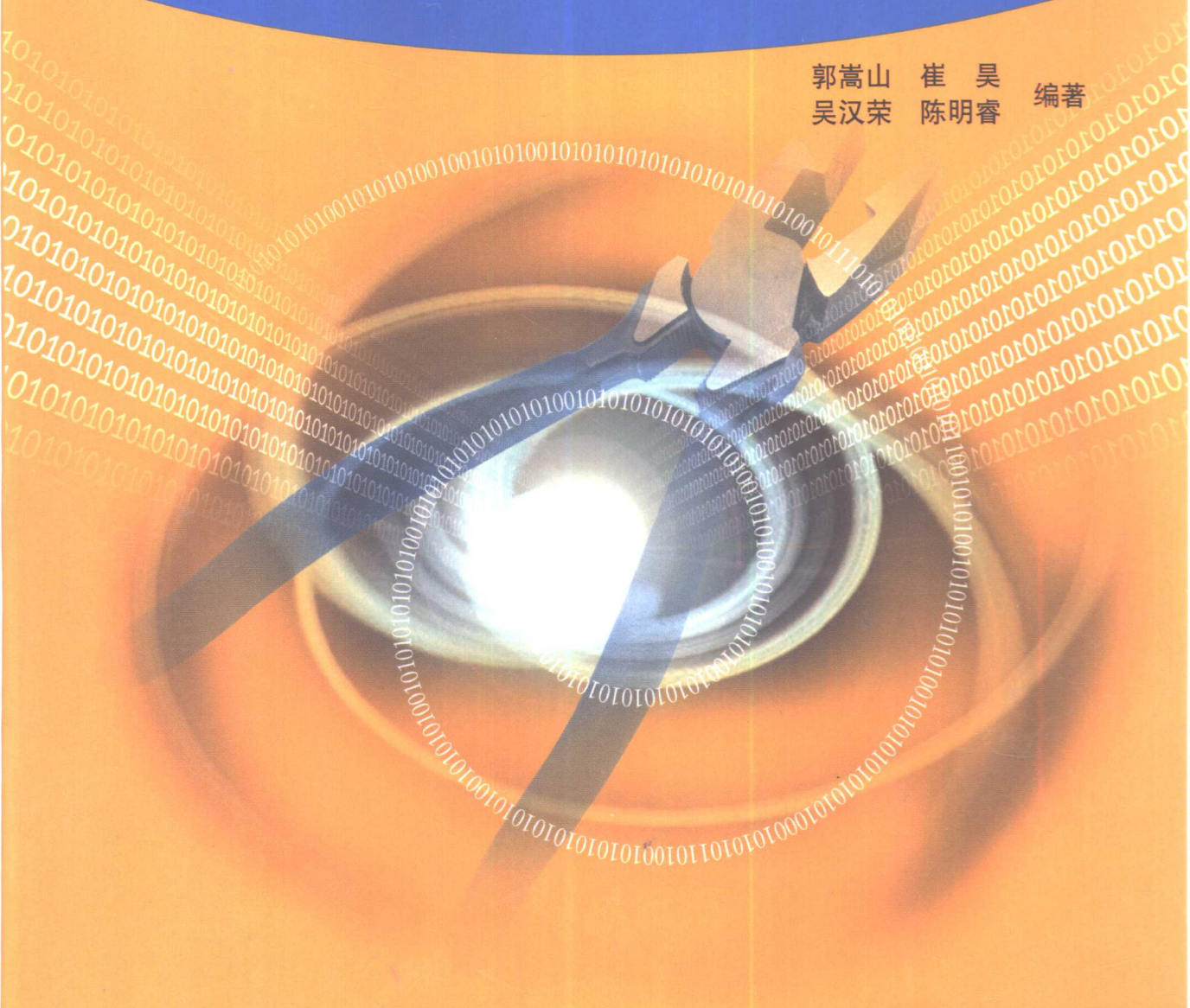


国际大学生程序设计 竞赛辅导教程

郭嵩山 崔 昊 编著
吴汉荣 陈明睿



北京大学出版社
<http://cbs.pku.edu.cn>

国际大学生程序设计竞赛 辅导教程

郭嵩山 崔昊 吴汉荣 陈明睿 编著

北京大学出版社
北京

内 容 简 介

本书介绍了 ACM 国际大学生程序设计竞赛(ACM/ICPC)世界决赛和各大洲区域预赛试题中所用到的几种常用的典型算法,并结合例题,对如何灵活地运用这些算法进行了比较详细的分析和深入浅出的讲解。本书以精讲多练为教学宗旨,并在书中最后两章选出一批有代表性的竞赛试题作为习题,备有题解和基本的测试数据,以便读者学习基本算法后作为竞赛训练之用。

本书可以作为高等院校有关专业的研究生和本科学生参加国际大学生程序设计竞赛的辅导教材,也可作为高等院校有关专业相关课程的教材和教学参考书,也比较适合作为中学青少年信息学奥林匹克竞赛省级及省级以上优秀选手备战信息学奥林匹克竞赛的培训教材及训练题集。

图书在版编目(CIP)数据

国际大学生程序设计竞赛辅导教程/郭嵩山,崔昊,吴汉荣,陈明睿编著.
—北京:北京大学出版社,2001.12

ISBN 7-301-05300-2

I. 国… II. ①郭…②崔…③吴… III. 程序设计 IV. TP311.1

中国版本图书馆 CIP 数据核字(2001)第 076922 号

书 名: 国际大学生程序设计竞赛辅导教程

著作责任者: 郭嵩山 崔 昊 吴汉荣 陈明睿

责任编辑: 黄庆生 汉 明

标准书号: ISBN 7-301-05300-2/TP·0621

出版者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

电 话: 编辑部(010)62765013 发行部(010)62750672

网 址: <http://cbs.pku.edu.cn>

电子信箱: xxjs@pup.pku.edu.cn

印刷者: 中国科学院印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米

16 开本 16 印张 406 千字

2001 年 12 月第 1 版

· 2001 年 12 月第 1 次印刷

定 价: 26 元

作者简介

郭嵩山，男，1945 年 4 月出生，1967 年毕业于上海交通大学无线电技术专业，中山大学信息科学与技术学院计算机科学系教授，主要从事计算机操作系统的教学和研究。主要著作有电子工业出版社出版的《BIO 结构分析教程》、《DOS 内核结构分析教程》、《COMMAND 结构分析教程》，人民邮电出版社出版的《数据结构及其应用》，中山大学出版社出版的《青少年信息学奥林匹克竞赛广东试题及解答（1986—1994）》等。近年来，担任 ACM 国际大学生程序设计竞赛中山大学队主教练，中山大学队在 1999—2001 年连续三年获得国际决赛权，并在国际决赛中取得较好的成绩。

前言

ACM 国际大学生程序设计竞赛 (ACM International Collegiate Programming Contest, 简称 ACM/ICPC) 是由国际计算机界历史最悠久、颇具权威性的组织 ACM 学会 (Association for Computer Machinery) 主办, 是世界上公认的规模最大、水平最高的国际大学生程序设计竞赛, 其旨在使大学生运用计算机来充分展示自己分析问题和解决问题的能力。该项竞赛从 1970 年举办至今已历 25 届, 因历届竞赛都荟萃了世界各大洲的精英, 云集了计算机界的“希望之星”, 而受到国际各知名大学的重视, 并受到全世界各著名计算机公司的高度关注, 成为世界各国大学生最具影响力的国际级计算机类的赛事, ACM 所颁发的获奖证书也为世界各著名计算机公司、各知名大学所认可。该项竞赛分区域预赛和世界决赛两个阶段进行, 各预赛区第一名自动获得参加世界决赛的资格, 世界决赛安排在每年的 3 月—4 月举行, 而区域预赛安排在上一年度的 9 月—12 月在各大洲举行。ACM/ICPC 的区域预赛是规模很大, 范围很广的赛事, 近几年, 全世界有 1000 多所大学, 近 2000 支参赛队在六大洲的 28~30 个赛站中争夺世界决赛的 60 个名额, 其激烈程度可想而知。

与其他编程竞赛相比, ACM/ICPC 题目难度更大, 更强调算法的高效性, 不仅要解决一个指定的命题, 而且必需要以最佳的方式解决指定的命题; 它涉及知识面广, 与大学计算机系本科以及研究生如程序设计、离散数学、数据结构、人工智能、算法分析与设计等相关课程直接关联, 对数学要求更高, 由于采用英文命题, 对英语要求较高, ACM/ICPC 采用 3 人合作、共用一台电脑, 所以它更强调团队协作精神; 由于许多题目并无现成的算法, 需要具备创新的精神, ACM/ICPC 不仅强调学科的基础, 更强调全面素质和能力的培养; 由于 ACM/ICPC 是采用 5 小时全封闭式竞赛, 参赛队员与外界完全隔离, 独立完成, 没有任何水份, 是其实际能力的真实表露, 其成绩可信度甚高; 但 ACM/ICPC 又是一种“开卷考试”, 可以带任何书籍、资料甚至源程序代码清单 (但不能带软盘), 不需要去死背算法, 而强调的是算法的灵活运用; 与其它计算机竞赛 (如软件设计, 网站设计等) 相比, ACM/ICPC 有严谨而客观的评判规则 (严格的数据测试), 排除了因评委的主观因素而造成评审不公平的现象, 所以, ACM/ICPC 对成绩的争议较少, 大家比较心服口服。

纵观近三年 (1998—2000) 中山大学共参加了 6 次地区预赛, 成绩全部在三甲之列: 连续三年夺得上海站季军 (1999—2000 年还连续两年夺得上海站第四名)、夺得台北站冠军、香港站亚军和日本站亚军; 并连续三年争得了世界决赛权。并在第 24 届 (2000 年) 在美国佛罗里达州奥兰多市举行世界决赛中夺得了第 11 名的好成绩, 在第 25 届 (2001) 在加拿大温哥华市举行的世界决赛中首获铜牌 (世界第 14 名)。

为了帮助高等院校的大学生们备战国际大学生程序设计竞赛, 帮助他们提高程序设计水平和培养更强的分析问题和解决问题的能力, 我们编写了这本辅导教材。本书所用的语言是 Pascal (Delphi)。全书共分六章, 第 1 章先介绍 Delphi 的运行环境, 以便于读者能更好地读通后面各章的程序; 第 2 章采用精讲的方式, 简明扼要、深入浅出地介绍了在国际大学生程序设计竞赛中经常用到的各种典型算法; 而在第 3 章中, 我们着重介绍了寻找最优解的算法, 诸如图论中的搜索算法和如何运用动态规划的思维来解决实际问题等方法; 在第 4 章

中，我们从 ACM/ICPC 世界决赛和区域预赛试题中精选了有代表性的 10 道例题，通过对例题的详细分析，力图让读者能更深刻地理解第 2、3 章中所介绍的基本算法；在第 5 章中，我们精选了一批有代表性的试题作为习题，并为这些习题设计了严格的、有梯度的测试数据，以便于读者检验自己编程的正确性；而在第 6 章中，我们给出了这些习题的详细分析和解答。为便于读者们学习和理解，本书的全部例题和习题都给出了我们自己编写的参考程序，而所有参考程序都有详细的注释。

参加编写本书的 4 位作者，第一位是国际大学生程序设计竞赛中山大学队的教练，其余 3 位都是参加过多次世界决赛和亚洲多个赛区区域预赛的中山大学队的主力队员。他们都是在读的研究生，我们期望能将自己的知识、经验、心得和体会，奉献给广大的程序设计爱好者，以便与大家共同探讨和交流。

本书可以作为高等院校大学生和研究生们准备参加各级国际大学生程序设计竞赛活动的辅导教材和试题集，也可以作为高等院校研究生和本科高年级学生学习相关课程的参考书，也可以作为省级及以上信息学奥林匹克优秀选手准备高层次程序设计竞赛的参考用书。

中山大学计算机科学系 97 级孔颖同学（也是参加过两次世界决赛和亚洲多个赛区区域预赛的中山大学队的主力队员）曾参与过本书的一些程序的编写工作，在此表示衷心的感谢。

由于我们水平所限，书中难免有不足之处，欢迎读者批评指正，谢谢！

作 者

2001 年 11 月

责任编辑：黄庆生 汉 明

封面设计：杨月静

国际大学生程序设计 竞赛辅导教程

本书介绍了 ACM 国际大学生程序设计竞赛 (ACM/ICPC) 世界决赛和各大洲区域预赛试题中所用到的几种常用的典型算法，并结合例题，对如何灵活地运用这些算法进行了比较详细的分析和深入浅出的讲解。本书以精讲多练为教学宗旨，并在书中最后两章选出一批有代表性的竞赛试题作为习题，备有题解和基本的测试数据，以便读者学习基本算法后作为竞赛训练之用。

本书可以作为高等院校有关专业的研究生和本科学生参加国际大学生程序设计竞赛的辅导教材，也可作为高等院校有关专业相关课程的教材和教学参考书，也比较适合作为中学青少年信息学奥林匹克竞赛省级及省级以上优秀选手备战信息学奥林匹克竞赛的培训教材及训练题集。

ISBN 7-301-05300-2



9 787301 053003 >

北京大学出版社

地址：北京市海淀区中关村北京大学校内

邮编：100871

电话：(010)62765013 (编辑部)

(010)62750672 (发行部)

E-mail: xxjs@pup.pku.edu.cn

http://cbs.pku.edu.cn

ISBN 7-301-05300-2/TP · 0621

定价：26.00 元

目 录

第 1 章 Delphi 简介	1
1.1 Delphi 的运行环境	1
1.1.1 Delphi 简介	1
1.1.2 Delphi 的 IDE 环境	1
1.1.3 Delphi 程序的编译、运行和调试	3
1.2 Delphi 常量	4
1.3 Delphi 变量	4
1.4 Delphi 类型	5
1.4.1 Delphi 预定义类型	5
1.4.2 枚举类型	7
1.4.3 子界类型	7
1.4.4 数组类型	7
1.4.5 字符串类型	8
1.4.6 集合类型	9
1.4.7 记录类型	9
1.4.8 指针类型	10
1.5 Delphi 的基本语句	11
1.5.1 赋值语句	11
1.5.2 复合语句	11
1.5.3 分支语句	12
1.5.4 循环语句	12
1.5.5 输入输出语句	14
1.6 Delphi 函数与过程	15
1.6.1 函数和过程的调用	15
1.6.2 常用的函数	15
1.6.3 自定义函数和过程	16
1.7 函数的递归	18
1.8 面向对象 Object Pascal	19
1.9 Delphi 中使用嵌入汇编	19
第 2 章 基本算法介绍	21
2.1 概述	21
2.2 常用数据结构在 Delphi 中的实现	21
2.2.1 线性表	21

2.2.2 栈	25
2.2.3 队列	26
2.2.4 二叉树	27
2.2.5 图	30
2.3 枚举算法.....	33
2.4 回溯算法.....	35
2.5 贪心算法.....	37
2.6 分治算法.....	40
2.7 数值计算.....	42
2.7.1 高精度计算	42
2.7.2 求解线性方程组	47
2.8 计算几何.....	50
2.8.1 线段问题	50
2.8.2 凸包问题	51
2.9 模拟题解法.....	55
第3章 寻找最优解的算法	56
3.1 动态规划.....	56
3.1.1 动态规划的常用名词	56
3.1.2 最优化原理	57
3.1.3 什么是动态规划	57
3.1.4 动态规划适于解决什么样的问题	58
3.1.5 用动态规划解题的好处	58
3.1.6 动态规划的逆向思维法	58
3.1.7 动态规划的正向思维法	63
3.2 最短路问题.....	64
3.2.1 有向无环图上的单对顶点间的最短路径问题	65
3.2.2 任意图上的单源最短路径问题	66
3.2.3 任意图上每对顶点间的最短路径问题	73
3.3 搜索算法.....	74
3.3.1 搜索的基本概念	75
3.3.2 搜索算法的一般模式	75
3.3.3 两种基本的搜索算法	76
3.3.4 搜索算法的优化手段	77
3.3.5 应用搜索算法求解最优化问题	78
3.3.6 广度优先搜索求单位耗散问题最优解的算法	79
3.3.7 广度优先双向搜索	84
3.3.8 最小耗散优先	91
3.3.9 搜索算法的总结	93

第4章 国际大学生程序设计竞赛 (ACM/ICPC) 试题及分析	95
4.1 生成字符串	95
4.2 模式识别的“中心”问题	100
4.3 划分凸多边形	104
4.4 防卫导弹	106
4.5 邮票问题	110
4.6 骨牌矩阵	113
4.7 师生树	120
4.8 旅游预算	125
4.9 正整数竖式除法	133
4.10 移棋子	138
第5章 习题	149
5.1 习题	149
5.1.1 电子表格 (Table)	149
5.1.2 DEL 命令 (DEL)	150
5.1.3 分割方格 (Divide)	151
5.1.4 信息编码 (Decode)	152
5.1.5 海上交通控制 (Lane)	153
5.1.6 投递最佳路线 (Best Deliver)	155
5.1.7 计算机网络连接 (Computer Network)	157
5.1.8 联系圈 (Circle)	158
5.1.9 球钟 (Ball Clock)	159
5.1.10 建筑物 (Buildings)	159
5.2 部分习题测试数据及参考答案	160
5.2.1 电子表格 (Table)	160
5.2.2 DEL 命令 (DEL)	163
5.2.3 分割方格 (Divide)	164
5.2.4 信息编码 (Decode)	168
5.2.5 海上交通控制 (Lane)	171
5.2.6 投递最佳路线 (Best Deliver)	173
5.2.7 计算机网络连接 (Computer Network)	176
5.2.8 联系圈 (Circle)	178
5.2.9 球钟 (Ball Clock)	181
5.2.10 建筑物 (Buildings)	181
第6章 习题解答	185
6.1 电子表格 (Table)	185
6.2 DEL 命令 (DEL)	189
6.3 分割方格 (Divide)	194

6.4 信息编码 (Decode)	202
6.5 海上交通控制 (Lane)	205
6.6 投递最佳路线 (Best Deliver)	213
6.7 计算机网络连接 (Computer Network)	222
6.8 联系圈 (Circle)	226
6.9 球钟 (Ball Clock)	229
6.10 建筑物 (Buildings)	232
附录 1997-2000 年 (第 22—25 届) ACM 国际大学生程序设计竞赛 (ACM/ICPC)	
亚洲区预赛成绩	243

第 1 章 Delphi 简介

1.1 Delphi 的运行环境

1.1.1 Delphi 简介

Delphi 是著名的 Borland 公司开发的可视化软件开发工具。“真正的程序员用 C，聪明的程序员用 Delphi”，这句话是对 Delphi 最经典、最实在的描述。Delphi 被称为第四代编程语言，它具有简单、高效、功能强大的特点。和 VC 相比，Delphi 更简单、更易于掌握，而在功能上却丝毫不逊色；和 VB 相比，Delphi 则功能更强大、更实用。可以说 Delphi 同时兼备了 VC 强大的功能和 VB 简单易学的特点。因此，它一直是程序员至爱的编程工具。

Delphi 具有以下特性：基于窗体和面向对象的方法，高速的编译器，强大的数据库支持，与 Windows 编程紧密结合，强大而成熟的组件技术。但最重要的还是 Object Pascal 语言，它才是一切的根本。Object Pascal 语言是在 Pascal 语言的基础上发展起来的，简单易学。

Delphi 提供了各种开发工具，包括集成环境、图像编辑（Image Editor），以及各种开发数据库的应用程序，如 Desktop DataBase Expert 等。除此之外，还允许用户挂接其他的应用程序开发工具，如 Borland 公司的资源编辑器（Resource Workshop）。

在 Delphi 众多的优势当中，它在数据库方面的特长显得尤为突出：适应于多种数据库结构，从客户机 / 服务器模式到多层数据结构模式；包括高效率的数据库管理系统和新一代更先进的数据库引擎；具备最新的数据分析手段和提供大量的企业组件。

Delphi 发展至今，从 Delphi 1、Delphi 2 到现在的 Delphi 6，不断添加和改进各种特性，功能越来越强大。本书将以 Delphi 6 为基础，介绍 Delphi 的开发环境、基本概念，让读者能在 Delphi 环境下编写竞赛程序。

1.1.2 Delphi 的 IDE 环境

启动 Delphi 后，屏幕上会出现如图 1.1 所示的界面，这是使用 Delphi 开发 Windows 应用程序和 Internet 应用程序常见的界面。

与上面两种应用不用，编写竞赛程序常常是使用控制台应用程序（Console Application），创建一个新的控制台应用程序可以使用如下的步骤：

- （1）使用菜单 File -> New -> Other，如图 1.2 所示。
- （2）在 New Items 对话框中选择新建“Console Application”（如图 1.3 所示）

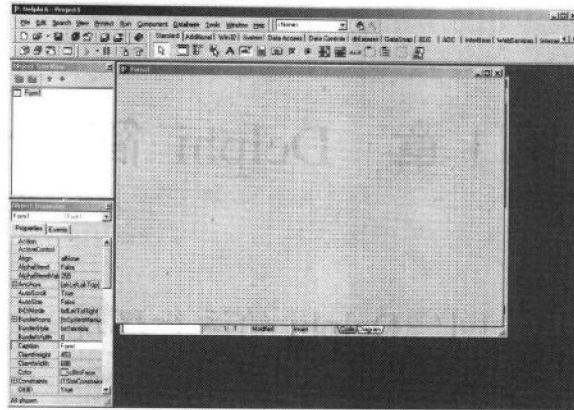


图 1.1

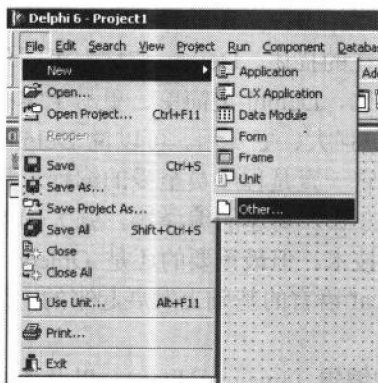


图 1.2

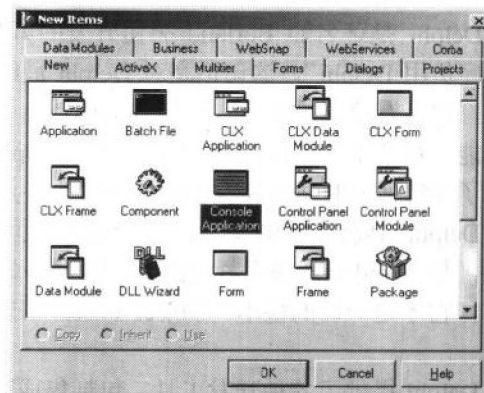


图 1.3

新建的控制台应用程序界面如图 1.4 所示。

下面将开始我们的第一个程序：Hello 程序，并且在 Delphi 环境中运行这个程序。程序在代码编辑窗口中编辑，例如可以在代码编辑窗口中输入 Hello 程序：

```

program Hello;

{$APPTYPE CONSOLE}

uses
  SysUtils;

begin
  { TODO -oUser -cConsole Main : Insert code here }
  WriteLn('Hello World. ');
  ReadLn;
end.
  
```

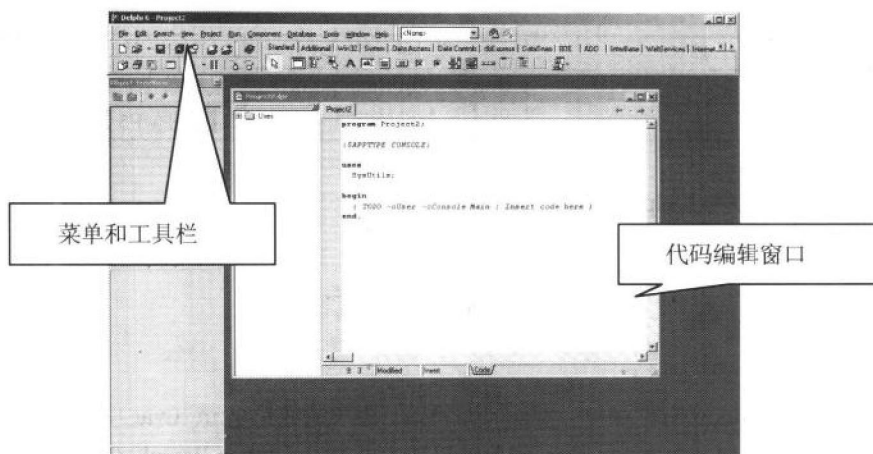


图 1.4

语句“WriteLn('Hello World.）”是输出“Hello World.”的字样；语句“ReadLn”是等待输入一个回车符号，以便观察程序输出结果。

然后选择菜单“Run->Run”运行程序。

这时，屏幕中会出现“Hello World.”的字样，如图 1.5 所示。

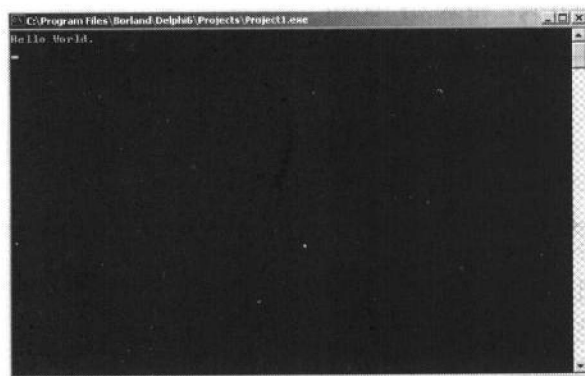


图 1.5

1.1.3 Delphi 程序的编译、运行和调试

表 1.1 列出了编译、运行和调试 Delphi 程序的常用命令。

表 1.1

功 能	菜 单 命 令	快 捷 方 式
编译 (Compile)	Project -> Compile	【Ctrl + F9】
运行 (Run)	Run -> Run	【F9】
单步跟踪 (Trace Into)	Run -> Trace Into	【F7】
单步执行 (Step Over)	Run -> Step Over	【F8】
执行到光标所在位置 (Run to Cursor)	Run -> Run to Cursor	【F4】

(续表)

功 能	菜 单 命 令	快 捷 方 式
设置 (Breakpoint)	Run -> Add Breakpoint	鼠标单击代码行左边的蓝点
观察窗口 (Watch)	Run -> Add Watch	【Ctrl + F5】

更多的命令请参考 Delphi 的联机帮助。

1.2 Delphi 常 量

标识符是 Delphi 应用程序中一些量的名称, 这些量包括变量 (var)、常量 (const)、类型 (type)、过程 (procedure)、方法 (Method) 及其他, Object Pascal 在应用标识符时, 必须首先说明它们。Object Pascal 是强类型语言, 它的编译器可以检查确保赋给变量或属性的值是正确的类型, 以便于您改正错误。由于 Object Pascal 是编译语言, 所以 Delphi 的执行速度要比使用解释语言快得多。在使用标识符前说明它们, 可以减少程序错误并增加代码的效率。有一点需要特别说明的是 Object Pascal 中的标识符是不区分大小写的, 这和 C/C++ 语言有很大的区别。例如: Delphi 和 delphi 被看成是同一个标识符。

常量说明是为一个标识符赋予一个值, 在程序执行过程中是不可改变的。常量说明使用保留字 “const” 作为开头。格式为: const 常量名=常量值; 下面的例子声明了三个常量:

```
const
  Pi = 3.14159;
  Age = 34;
  ProductName = 'Delphi';
```

上文的三个常量分别是实型、整型和字符串型常量。常量用 “=” 表示两边的值是相等的。和 C/C++ 语言不同, Object Pascal 的字符串常量是使用单引号 (') 作为定界符的, 例如上面的 'Delphi', 而不是使用双引号 (")。

1.3 Delphi 变 量

变量是程序代码中代表一个内存地址的标识符, 而该地址的内存内容在程序代码执行时可以被改变。在使用变量前必须对它进行说明, 即对它进行命名, 并说明它的类型。在所有变量说明以前加上保留字 var。变量说明左边是变量的名称, 右边则是该变量的类型, 中间用 (:) 隔开。即格式为:

var 变量名:变量类型;

例如:

```
var
```

```
Value ,Sum : Integer;
Name : String;
```

在 Delphi 中，可以在变量声明同时赋初始值，格式为：

const 变量名:变量类型=变量初始值;

注意：此时是使用 **const** 保留字开头，而不是 **var**。

例如：

```
const
Value: Integer:=10;
Name : String='Gates';
```

1.4 Delphi 类型

Delphi 有两大数据类型，一类是系统已经预定义的，另一类是用户自定义的。Object Pascal 有一些系统预定义的数据类型，这些类型包括有：整型、实型、布尔型、字符型、指针型；而用户自定义的数据类型有枚举型、子界型、数组型、集合型、记录型、对象型等，您可以利用这些用户预定义的数据类型来构造新的数据类型以满足程序的特定需要。

1.4.1 Delphi 预定义类型

Object Pascal 有多个预定义的数据类型，您可以说明任何这些类型的变量：

1. 整型

与 CPU 和操作系统相关的整型包括 **Integer** 和 **Cardinal**，在当前 32 位编译器下，取值范围如表 1.2 所示。

表 1.2

类 型	范 围	格 式
Integer	-2147483648..2147483647	signed 32-bit
Cardinal	0..4294967295	unsigned 32-bit

与 CPU 和操作系统无关的整型如表 1.3 所示。

表 1.3

类 型	范 围	格 式
Shortint	-128..127	signed 8-bit
Smallint	-32768..32767	signed 16-bit
Longint	-2147483648..2147483647	signed 32-bit
Int64	-2 ⁶³ ..2 ⁶³ -1	signed 64-bit
Byte	0..255	unsigned 8-bit
Word	0..65535	unsigned 16-bit
Longword	0..4294967295	unsigned 32-bit

2. 实型

表 1.4 列出了实型的范围和存储格式。

表 1.4

类 型	范 围	有 效 位	占用字节 (bytes)
Real48	$2.9 \times 10^{-39} \sim 1.7 \times 10^{38}$	11-12	6
Single	$1.5 \times 10^{-45} \sim 3.4 \times 10^{38}$	7-8	4
Double	$5.0 \times 10^{-324} \sim 1.7 \times 10^{308}$	15-16	8
Extended	$3.6 \times 10^{-4951} \sim 1.1 \times 10^{4932}$	19-20	10
Comp	$-2^{63+1} \sim 2^{63-1}$	19-20	8
Currency	-222337203685477.5808...922337203685477.5807	19-20	8

通用类型 Real 在当前的解释下，等价于 Double，如表 1.5 所示。

表 1.5

类 型	范 围	有 效 位	占用字节 (bytes)
Real	$5.0 \times 10^{-324} \sim 1.7 \times 10^{308}$	15-16	8

3. 布尔型

Boolean，只包含 true 或 False 两个值，占用 1 字节内存。

4. 字符型

Char，一个 ASCII 字符；字符型的常量形式上和字符串型常量一样，都是使用单引号 (') 作为定界符，例如：'A'。

5. 字符串类型

String 一串最长可达 2G 个 ASCII 字符。

6. 指针型

Pointer，可以指向任何特定类型。相当与 C/C++ 中的 “void*”。

类型的兼容性：

整型类别和实型类别都各有五种类型，同一类别中，所有的类型与其他同类别的都相容，您可以将一种类型的值赋给相同类别中不同类型的变量或属性，而只需要这个值的范围在被赋值的变量或属性的可能值范围内。例如，对于一个 Shortint 型的变量，可以接受在 -128 到 127 范围内的任意整数，例如，对于 Shortint 类型，您不能将 128 赋给它，因为 128 已经超出了 Shortint 的范围了；此时，如将范围检查功能打开（选用 Options\Project，并在 Compiler Options Page 中选择 Range Checking），将会检查出一个范围错误；如果 Range Checking 没有被打开，那么程序代码将可以执行，但被赋值的值将不是您期望的值。

在一些情况下，您可以进行不同类型的变量或属性的赋值。一般来说，可以将一个较小范围的值赋给一个较大范围的变量。例如，您可以将整型值 10 赋给一个能接受实型值的 Double 类型的变量，而使其值成为 10.0，但如果将一个 Double 类型的值赋给整型变量，则会出现类型错误。如果您不清楚类型的兼容性，可以参阅 Delphi 的在线帮助中 “Type Compatibility and Assignment Compatibility” 主题。