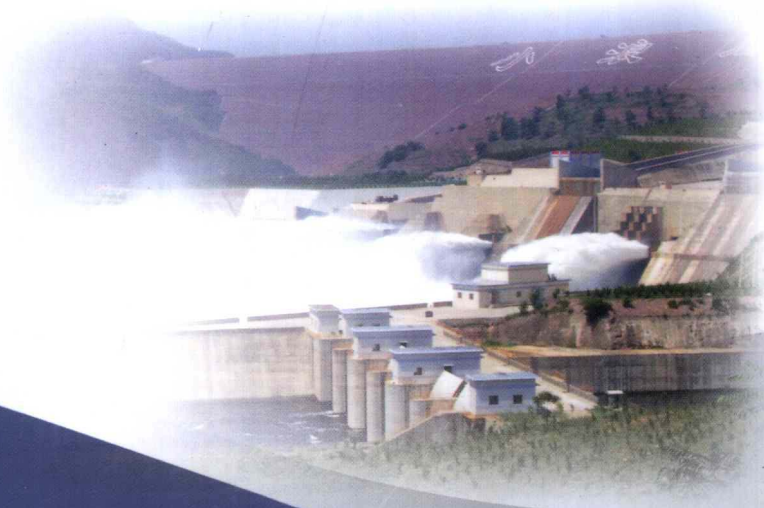


河流水沙科学分析系统 Excel 解决方案

孙赞盈 曲少军 著



黄河水利出版社

河流水沙科学分析系统 Excel 解决方案

孙赞盈 曲少军 著

黄河水利出版社
· 郑州 ·

内 容 提 要

本书主要介绍一个基于 Excel 解决方案的河流水沙科学分析系统,共分 10 章。内容包括:水文泥沙数据的特点;水文泥沙数据库管理程序应具备的特点;电子表格软件及其二次开发技术的发展史;Microsoft Office 作为解决方案的平台;基于 Excel 平台的黄河水沙科学分析系统解决方案的对象模型和图形用户界面;大量的自定义水文泥沙专业的工作表函数;辅助的水沙科学分析工具;小浪底水库泥沙多年调节和黄河下游河道冲淤计算的水文学数学模型;数据库的设计和数据库内容;部分源代码等。在附录中给出了作者提出的标称流量的物理意义,以及实体模型断面内插计算方法的详细介绍。

本书可供从事河流泥沙、河床演变、水资源、数学模型开发和水利信息研究,以及 Office 软件开发等方面的研究者、工作者和高等院校相关专业的师生参考使用。

图书在版编目(CIP)数据

河流水沙科学分析系统 Excel 解决方案/孙赞盈,曲少军著. —郑州:黄河水利出版社,2008. 12

ISBN 978 - 7 - 80734 - 381 - 3

I. 河… II. ①孙…②曲… III. 电子表格系统, Excel - 应用 - 河道 - 泥沙运动 - 系统分析 IV. TV142

中国版本图书馆 CIP 数据核字(2008)第 202911 号

策划编辑:岳德军 电话:0371 - 66022217 E-mail:dejunyue@163.com

出 版 社:黄河水利出版社

地址:河南省郑州市金水路 11 号 邮政编码:450003

发行单位:黄河水利出版社

发行部电话:0371 - 66026940、66020550、66028024、66022620(传真)

E-mail:hhsclbs@126.com

承印单位:黄河水利委员会印刷厂

开本:787 mm × 1 092 mm 1/16

印张:7.75

字数:180 千字

印数:1—1 000

版次:2008 年 12 月第 1 版

印次:2008 年 12 月第 1 次印刷

定价:20.00 元

前 言

黄河流域是中华民族的摇篮,孕育了光辉灿烂的华夏文明,同时黄河又是一条曾给中华民族带来深重灾难的河流。从公元前 602 年至 1938 年的 2 540 年间,黄河下游决溢达 1 590 多次,因决口导致的大改道 26 次,即所谓“三年两决口,百年一改道”。在人类长期与自然的斗争中,特别是在防御黄河水害、兴修水利的过程中,黄河水文泥沙测验也逐步发展起来。黄河上建立水文站等观测站网比较早,长期以来取得了大量的水文泥沙测验资料,借助这些宝贵数据,广大水利专家深入开展了黄河水沙变化、河床演变及库区冲淤变化等规律的研究和探索,为黄河治理开发和管理事业的发展作出了重大贡献。

随着信息技术和现代遥感技术的迅速发展,流域水文测验技术发生了重大突破,借助现代化的测验方法和手段,可获得更准确、更实时、更丰富的数据和资料,这将会给治河专家探索与揭示黄河的自然规律提供极大的方便,甚至会大大缩短揭示黄河自然规律的过程,但总体上看,由于黄河实测水沙资料十分丰富,而黄河科研工作的分析手段还较为单一,现代化水平不高,这些资料中还蕴含着大量尚未发现的规律。随着黄河治理的深入,越来越多的定量问题需要快速及时的回答,信息处理技术、网络技术的迅速发展为解决这些问题提供了可能。本书的作者是治黄科研一线的科研工作者,深谙直观、简单和易用的软件对提高科研分析工作效率的必要性以及获得准确结果的重要性,认为使用新的和更加有效的研究手段为治黄的迫切需求,为解决这些问题,本书对黄河分析研究中使用频率最高的常规水文资料,建立河床演变数据库,紧扣黄河跟踪研究的资料分析,提出了利用 Excel 软件的二次开发技术的解决方案。本书所提出的研究方法和成果,对于水文资料分析研究具有重要作用。

在本书撰写的过程中,黄河水利科学研究院的姜乃迁副院长,科技处苏运启副处长,泥沙所李勇所长、张原锋副所长和余欣副所长给予了大力支持,齐璞和刘月兰等教授给予了指导,尚红霞、彭红和郑艳爽等参与了图片整理、资料收集等部分工作,金孝华撰写了第 3 章、第 4 章,王富昌撰写了本书的第 8 章及附录 A,苏运启审阅了部分书稿,在此一并表示感谢。

水沙科学分析系统涉及到数据库的设计、对象的建模、用户界面的布局设计、数学模型的计算方法、帮助文档的撰写,以及之后的测试、优化和调整修改,是一个非常庞杂的系统工程,工作量之大、工作强度之高,靠一两个人完成实在是一件十分艰辛的事,加上资料所限和问题本身的复杂性,虽然我们尽了很大努力,但不足之处仍在所难免。各位读者如果有好的意见和建议,希望通过电子邮件(sunzanying@hotmail.com)和我们联系,以便再版时加以完善。

作 者

2008 年 10 月

目 录

前 言

第 1 章 水文泥沙数据的特点	(1)
1.1 逻辑上的二维性和三维性	(1)
1.2 资料的公益性	(1)
1.3 资料的规范性	(1)
1.4 时间上的延续性(积累性)	(1)
1.5 数据的成块利用	(1)
第 2 章 水文泥沙数据库管理程序应具备的特点	(2)
2.1 应该符合人们查阅《水文年鉴》的习惯	(2)
2.2 应该侧重于利用(而不是管理)	(2)
2.3 应该高度透明	(2)
2.4 输入数据的方式不同	(2)
2.5 保存为 Excel 和 Access 关系数据库两种格式	(3)
第 3 章 电子表格软件及其二次开发技术的发展史	(4)
3.1 电子表格的鼻祖 VisiCalc	(4)
3.2 成熟的 DOS 软件 Lotus 1-2-3 和 Quattro Pro	(5)
3.3 后来居上的 Excel	(6)
3.4 二次开发的程序设计语言	(7)
3.5 未来 Office 软件的发展方向	(15)
第 4 章 Microsoft Office 作为解决方案的平台	(20)
4.1 各类解决方案的优缺点	(20)
4.2 Microsoft Office 平台简介	(21)
第 5 章 基于 Excel 平台的黄河水沙科学分析系统	(25)
5.1 对象模型	(25)
5.2 工具条界面	(25)
5.3 三维活表格界面	(29)
5.4 地图上存取	(33)
第 6 章 自定义加载宏函数	(38)
6.1 河流横断面分析计算	(38)
6.2 概化断面、概化水沙过程	(43)
6.3 统计各流量级的水沙	(43)
6.4 基于逐日流量和逐日含沙量的统计函数	(44)
6.5 实时水情统计函数	(46)

6.6	同流量水位计算	(46)
6.7	线性插值	(51)
6.8	其他工作表函数	(52)
6.9	工作表函数常见错情及其处理	(54)
第7章	辅助分析工具	(56)
7.1	插入图表集	(56)
7.2	用图表辅助工具实现可视计算	(56)
7.3	为图表添加数据标签	(60)
7.4	判读图片式图表数据	(61)
第8章	水文学数学模型	(63)
8.1	黄河泥沙数学模拟存在的主要难点	(63)
8.2	小浪底水库泥沙多年调节数学模型	(64)
8.3	黄河下游河道水文学数学模型	(68)
第9章	数据库	(82)
9.1	数据库设计	(82)
9.2	黄河水文泥沙数据库的内容	(86)
第10章	源代码	(92)
10.1	Borland Quttro Pro 宏	(92)
10.2	河流水沙科学分析系统	(101)
附录A	标称流量的物理意义	(109)
附录B	考虑形态的实体模型的断面内插计算方法	(113)
参考文献		(118)

第 1 章 水文泥沙数据的特点

水文泥沙数据的特点是它在逻辑上的二维性和三维性以及资料的公益性、规范性等,因此它与其他行业尤其是商业上的数据有很大的不同,有自己的特点。

1.1 逻辑上的二维性和三维性

从检索出发,我们所需要的资料大体上可以从时间、空间位置 and 资料数据类型三个方面来逐步缩小范围。例如,对于某一年(时间),我们要查找的是某一站(空间位置)的某一类数据(数据类型,如施测流量成果表、洪水水文要素摘录表、实测大断面成果表),这样逐步缩小范围常常就能找到我们所要的资料。

所以,从时间、空间位置 and 资料数据类型三个方面就很容易查找到数据,这种查找资料的办法符合我们查阅水文年鉴的习惯。

1.2 资料的公益性

水文泥沙数据针对公益性的流域治理以及防治江河洪水和泥沙灾害的需要,测验、计算和整理都是在国家事业费的支持下建设的,其公益性的特点决定了这些数据没有保密性可言。公益性的另一个特点是这些数据由专人收集、整理、管理、维护和发布。

1.3 资料的规范性

水文资料是按照国家水文资料的规范严格整编出来的,为了统一全国的水文资料,从水利部到各地方流域管理机构,不仅数据有严格的、标准一致的格式,而且测站代码的制定以及测站发生变动时和测站代码如何随之变动也有相应的技术规范。

1.4 时间上的延续性(积累性)

所谓时间上的连续性是指水文资料一旦整编完毕,编入《水文年鉴》,就极少再去修改它,并且随着时间的延续不断增加资料。水文资料的这种特点决定了我们将很少对资料进行很多数据库意义上的排序等编辑操作,而只需追加数据。

1.5 数据的成块利用

数据的成块利用是指我们常常对角录集进行操作,很少对单个记录进行操作。例如,对于实测大断面数据,我们只是将整个起点距和对应的高程数据同时利用,单独的一个数据和一个记录是没有意义的。

第2章 水文泥沙数据库管理程序应具备的特点

既然水文泥沙数据与其他行业的数据有很大的不同,因此我们认为管理水文泥沙数据库的管理程序也应该有自己的特点。

2.1 应该符合人们查阅《水文年鉴》的习惯

在查阅刊印的《水文年鉴》时,我们常常是按年份→哪一个水文站(水位站或大断面)→何种数据的顺序进行查阅的。黄河水文网的历史水文资料数据库的数据访问页就是这样组织的。

2.2 应该侧重于利用(而不是管理)

黄河历史水文资料数据库的建设是一项工作量很大的工作,黄河水利科学研究院泥沙所作为科研单位,不应该而且也没必要对历史资料进行重复建设,实际上也做不好。黄河水文网已经公开历史水文资料,对于其他单位,重要的是如何充分利用这些资料。另一方面,由于黄河问题的复杂性,追踪研究又特别重要,整编的水文资料滞后显然无法及时满足跟踪研究的需要,对于科研单位,就存在一个如何管理近期数据的问题,但如果像管理历史资料数据库那样将所有的数据保存为关系数据库,不但使用不便,还会加大工作量。

2.3 应该高度透明

与社会上其他行业的数据不同,水文资料的安全性仅限于使数据库不受破坏,因此不但没有保密性,而且应该尽可能易于使用,这就要求资料越透明越好,历史资料数据由于被制作为数据访问页而较容易使用,近期资料如果全部被保存为关系数据库的形式,会降低透明度,因为多数人并不熟悉读写关系数据库。

2.4 输入数据的方式不同

水文资料数据具有成块的特点,如果用 TextBox 控件输入数据,往往意味着低效率(更何况更多的数据实际上是从电子表格直接导入到数据库的),因此我们常常愿意在电子表格中输入数据。其实即使不用电子表格,TextBox 也不是唯一的靠窗体输入数据的途径。

2.5 保存为 Excel 和 Access 关系数据库两种格式

水文泥沙数据具有时间上的延续性(积累性)的特点,对于大多数数据,我们很少进行关系数据库管理中的“插入”操作,一般的维护工作是“追加”,因此选择合适的保存形式,高度透明的 Excel 文件完全可以满足大多数数据的要求。我们可以利用关系数据库容易保存一对多关系的优点,将小浪底水库实测大断面这样的数据,保存为关系数据库。

第3章 电子表格软件及其二次开发技术的发展史

3.1 电子表格的鼻祖 VisiCalc

说到电子表格,必须提到发明人美国人丹尼尔·布莱克林(Daniel Bricklin)。丹尼尔·布莱克林 1951 年出生于美国费城,1969 年考入了麻省理工学院的数学专业,但大一还没上完,他就转到了计算机专业,大学毕业后,已是计算机技术专家的布莱克林顺利地进入了 DEC 公司。后来,布莱克林重返校园,进入哈佛大学攻读 MBA 课程。在哈佛大学学习的过程中,他发现当时的公司只能依靠纸、笔、计算器来完成财务数据的计算,这种计算方式又笨又慢;后来,虽然有了计算机,但要编写很多小程序来完成计算,虽然速度提高了,但只要有一个数据发生改变仍必须重新计算,计算过程很不直观。他想到应该编写一套具有实时数据处理功能的应用程序,于是就开发了名字叫 VisiCalc 的软件(见图 3-1),VisiCalc 的意思是看得见的计算,从此电子表格软件诞生了,虽然该软件只有 25 kB 大小,却是世界上第一个支持实时数据计算的电子表格程序。VisiCalc 首先获得了 Apple II 型电脑的认可。许多人甚至为了能用上 100 美元一套的 VisiCalc 而不惜花 2 000 多美元去买一台 Apple II 型电脑,能够使用上 VisiCalc 成为人们购买 Apple II 型电脑几乎唯一的原因。随后的几年,VisiCalc 程序不断完善,并相继获得了 TRS-80、Atair 800、惠普、IBM 等个人电脑的支持,其每月的销量达到了 3 万套。到 1986 年,Software Arts 一共销售了超过 700 万套 VisiCalc 软件。

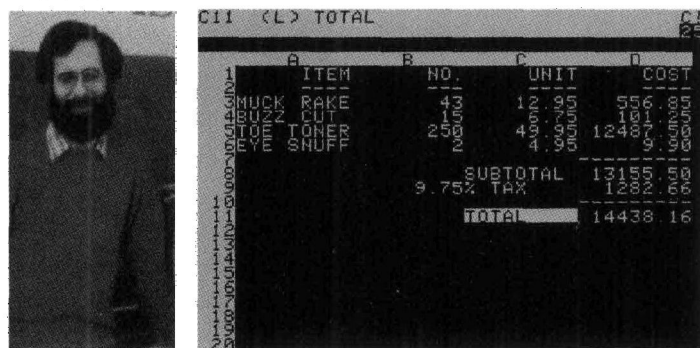


图 3-1 丹尼尔·布莱克林(左)和他的 VisiCalc 软件(右)

现在看来,电子表格之所以如此受人欢迎,主要有两大特点:其一是它将信息以一个二维表格的形式展示在人们面前,信息显示最为直观;其二是它利用了计算机能够根据数据变化自动重算公式的自动更新的特点。这是一个巨大的创新,其革命性的信息分析和

处理方式,使它成为迄今为止最没有争议的计算机软件。前不久在《PC Magazine》评出的全球 10 大软件中,VisiCalc 名列榜首(见表 3-1)。

表 3-1 《PC Magazine》评出的全球 10 大软件

排名	软件	推出时间	软件特色
1	电子表格软件 VisiCalc	1979 年	这是世界上第一款电子表格软件,VisiCalc 最初是基于 Apple II 型个人电脑而开发的,苹果公司采用该软件后得到广大商业用户的青睐,不到一年工夫其变成个人电脑历史上第一个最畅销的应用软件。它不但改变了人们以往处理和分析数据的方式,更使人们更直接地认识到计算机软件的巨大魔力。VisiCalc 以及包括 Lotus1-2-3、Quattro、Microsoft Excel 在内的杰出产品将现代电子表格软件及微型计算机带到了办公室中。电子表格软件的革命性信息处理方式,使它成为迄今为止最没有争议的计算机软件
2	字处理软件 WordStar	1978 ~ 1979 年	WordStar,更早的 Electric Pencil 以及后来的 Microsoft Wordfor-Windows 也应当入围该名单。更早的字处理软件有 WordPerfect,但 WordStar 是第一种真正受欢迎的字处理软件,自 20 世纪 70 年代以来,垄断字处理市场长达十多年
3	Apache	1995 年	一个开放源代码软件
4	Mosaic		Mosaic 是一个浏览器软件
5	Macintosh	1984 年	Macintosh 这款由 Apple 公司开发的操作系统开启了现代操作系统图形用户界面的先河,它早于微软的 Windows 视窗操作系统
6	Photoshop	1990 年	美国 Adobe 公司生产的图形图像处理软件
7	dBASE II	1980 年	一个大概是最古老的低端商业数据库的数据库软件。但在当时 dBASE II 是一款了不起的产品,因为它普及了关系数据库的概念
8	Aldus Pagemaker	1980 年	支持 WYSIWYG(所见即所得)的桌面排版印刷软件
9	Sendmail	约 1983 年	由 Sendmail 公司支持的一种电子邮件系统
10	Microsoft Basic	1976 年	微软公司开发的一款具有真正创新意义的软件。它是首批独立封装的编程语言之一。Microsoft Basic 开启了现代编程时代,并普及了封装软件的概念

3.2 成熟的 DOS 软件 Lotus 1-2-3 和 Quattro Pro

VisiCalc 这个当时最热门的计算机软件很快就因 Lotus 1-2-3 的出现而遭淘汰。

莲花发展有限公司(Lotus Development Corp.)成立于 1982 年,总部位于美国麻省波士顿,是全球领先的 Intranet 平台与通信软件供应商,它于 1983 年推出电子表格软件 Lotus 1-2-3。Lotus 1-2-3 的功能比 VisiCalc 要强大得多。和 VisiCalc 相比,Lotus 1-2-3 还增加了多种功能,同时还将表格计算、绘图、数据库分析等功能集于一身,获得了巨大的成功,在 IBM 及其兼容机系统上得到了非常广泛的应用,风行一时。

电子表格软件是如此的受欢迎,吸引了其他软件厂商参与角逐,如美国的宝兰公司(Borland International Inc.)和微软公司。

Borland 公司本来是软件开发工具生产商,它的 Turbo Pascal、Turbo C、Borland C/C++3.1、C++ Builder 以及 Delphi 开发工具是非常著名的软件。Borland 公司曾经主宰着当时的软件开发工具市场,Borland 公司推出的电子表格软件 Quattro Pro 就是用自己

所开发的软件工具 Borland C++ 开发的。Quattro Pro 也是一个著名的电子表格软件,当时,由 Borland C/C++ 编译器所开发的 Quattro Pro 在执行效率上几乎是最好的,Quattro Pro 曾经好于当时的 Lotus 1-2-3,但在功能上却不如当时微软的 Excel 强大,再后来,Quattro Pro 只能卖给 Novell 公司了。

3.3 后来居上的 Excel

关于微软的电子表格软件 Excel。据说在 Lotus 1-2-3 的鼎盛时期,Microsoft 一直在觊觎这个市场。1985 年,微软推出了其电子表格软件——Excel,向当时的竞争对手该领域的霸主 Lotus 发起了挑战,起初影响不大,但微软花了大量精力不断改进 Excel,不断为 Excel 增加新功能,使其最终超越了 Lotus 1-2-3。1993 年,微软正式推出了 Excel 5.0,该产品确立了微软在电子表格软件领域的重要地位,使其在与 Lotus 的竞争中的优势进一步加强,从此 Lotus 的市场份额越来越小。

关于 Excel 是如何胜出 Lotus 1-2-3 的,有这样一个小插曲:微软意识到 Office 软件是有巨大市场的,但苦于无法开发出一个能够和 Lotus 1-2-3 相竞争的产品。有一次 Lotus 举办了一个 Lotus 1-2-3 的技术研讨会,由当时 Lotus 1-2-3 的首席工程师主讲。Microsoft 知道了这个技术研讨会之后,立刻派出了最好的程序设计师,现场询问 Lotus 是如何开发 Lotus 1-2-3 的,并且趁机询问这位首席工程师如何克服 Lotus 1-2-3 在许多技术方面的难点,而这些困难之处正是 Microsoft 的工程师无法克服的。当时,在现场中的 Lotus 首席工程师虽然知道这些人是 Microsoft 派来的,而且询问的问题正是 Lotus 1-2-3 许多关键的技术点。但是这位首席工程师凭借着多年的开发经验,认为 Microsoft 不可能在短期之内追上 Lotus 1-2-3,因此就没有多作保留地回答了许多重要的问题。没有想到 Microsoft 的这些程序员也是非常优秀的人才,一经指点之后,立刻畅然全通,在短短的 1、2 个版本之后不但马上追上了 Lotus 1-2-3,许多功能方面更是青出于蓝而胜于蓝, Lotus 1-2-3 便逐渐失去优势了。我想这位 Lotus 1-2-3 的首席工程师一定很后悔当时回答了关键的技术问题吧。❶

微软在 Excel 5.0 中允许用户在一个文件内创建含多张工作表的工作簿;在对用户习惯悉心研究后,工具栏和菜单栏焕然一新;充分改善了帮助系统;并首次将广受大众欢迎的 Basic 可视开发 IDE 集成到 Excel 中,首次应用 Visual Basic for Application (VBA);扩展了工作表函数。自此,微软的 Excel 成为公认的世界功能最强大的电子表格软件。1994 年 6 月,微软推出了 Excel 5.0 简体中文版。

此后,微软不断改进 Excel,相继推出了 Office 95、Office 97。在 Office 2000 及其后续版本中,增加很多更加强大的功能,例如:

- 考虑到 Internet/Intranet 的发展,允许用户直接打开和保存 HTML(超文本标记语言)文件、XML(可扩展文本标记语言)文件,把各类文件在 Web 服务器或 ftp 服务器上保存或编辑,就如同在本地计算机上保存和编辑一样简单。允许用户将 Excel 的默认文件

❶ 李维,《Borland 传奇》(第二版)。

格式设置为 HTML 格式或 XML 格式,可以像编辑. xls 文件一样编辑此类文档。

- 吸收了统计分析软件 SPSS 的可视数据分析功能——数据透视表,增加了数据透视图的功能,允许用户仅仅通过拖放行字段、列字段或页字段和/或建立相应的数据透视图,对位于电子表格或数据库中的大量信息进行可视分析。

- 增加查询表,可以建立和编辑一个来自数据库、文本文件或其他 URL(统一资源定位符)的其他数据格式的查询表,并可设定这种数据表进行自动、按一定频率或手动刷新,从而使一些重复的数据导入工作变得简单而快捷。

- 提供可用于 Web 和开发的三大 Web 组件。①电子表格组件(Spreadsheet)使向 Web 浏览器传递交互式工作簿成为可能。用户可以利用 Excel 公式计算数据,在计算中还可以引用 Web 页的其他元素。工作簿组件中的数据在浏览器上是可编辑的,并包括数据分析特性,如排序和筛选。该组件还包括表示数据的特性,如字体和单元格格式、可调整的列和行。电子表格组件拥有 262 144 行、18 278 列,分别是 Excel 工作表的 4 倍和 71 倍。②数据透视表组件(PivotTable)提供查看和分析数据库信息的动态方法。它可以使用户浏览报表数据、对其进行动态排序和筛选、将其按行或按列分组、创建总计和聚焦总计后的细节。它还可以使用户更高效地处理大量数据。数据透视表组件可以显示工作簿、数据库(例如 Access 服务器或 SQL 服务器)或联机分析处理(OLAP)服务器的数据。③图表组件(Chart)提供用来连接到数组常数、电子表格组件和数据库(例如 Access 服务器或 SQL 服务器)的图表显示,提供图表类型 12 个大类、每个大类又分多个子类共 57 个图表类型,图表类型包括柱状图、条形图、折线图、平滑线图、饼图、 $X \sim Y$ 散点图、气泡图、面积图、圆形图、雷达图、股价图和极坐标图等,种类繁多,可基本满足各行业的数据分析之用。另外,这些组件还提供事件编程,使其应用更加灵活。这些组件的对象模型和 Office 中的对象模型在很多方面非常相似,熟悉 Office 对象模型者很容易使用上述组件进行开发。

3.4 二次开发的程序设计语言

借助计算机进行数据统计和计算等科学分析是自然科学研究者的研究手段。商业软件虽然功能强大,但它是从很多专业的通用性的角度出发来设计的,不一定能够完全满足本专业的要求,很多情况下需要进行编程(包括软件的二次开发),使其能够从大量的数据中发现有用的数据和剔除错误的数据,从而大大减轻单调复杂的工作和提高工作效率,提高数据分析结果的正确性和可靠性。

数学模型计算也是现代自然科学工作必不可少的研究途径之一。随着计算机硬件技术突飞猛进的发展,数学模型也得到了前所未有的发展,而且使用得越来越普遍,成为计算和比较不同方案优劣的廉价而快捷的手段。

现在,对于非计算机专业的科研人员来说,开发界面友好的软件化的程序不仅是大势所趋,而且也成为可能,但选择一种好的程序语言显得十分重要,特别是在一开始就注意到这个问题,将会大大节省工作量。可以编制数学模型的程序语言很多,但并不是都适合非计算机专业的人员使用。计算机程序设计语言五花八门,但随着计算机软硬件日新月

异的发展,常用的计算机程序设计语言也在不断的发展完善,各种常用的程序设计语言在功能和性能上差别越来越不明显,但在易用性上却有明显差别。

FORTRAN 语言曾经是最受科研人员欢迎的程序设计语言,和当时的 Basic 语言相比它的优势在于:①容易学习;②结构化;③使用的人多,使用的时间长。有很多特别适合数据运算功能的特点,还拥有大量的通用子程序。

然而,FORTRAN 语言也有很多难以克服的缺点,例如 FORTRAN 语言的最初设计过于追求简单易用,在容易维护方面埋下了很大的隐患。例如,FORTRAN 语言提供的 COMMON 语句,它把数据暴露给很多可能根本不需要的函数,用户很难区分本地变量和传递过来的变量,COMMON 语句在为程序员提供易用性的同时,也带来了难以维护的缺陷。FORTRAN 语言编译器的革新者显然意识到这一问题的严重性,在后续版本中,向用户建议,尽量不要使用 COMMON 语句,同时受面向对象的思路影响,在后来的 FORTRAN 语言编译器中,增加了 Module 的功能,但显而易见,它充其量向面向对象只迈出了半步,FORTRAN 语言仍然不是面向对象的,这使它和其他面向对象的语言相比,不但可维护性差,而且调用一些图形功能麻烦,不符合人的思维习惯,不好用。FORTRAN 语言在科学计算方面立下了汗马功劳,然而时过境迁,由于 FORTRAN 在面向对象方面存在的问题,它无法和诸如 C++ 语言和 Delphi 语言等面向对象的程序设计语言相比,已经很难适应大型的科学分析和数学模型程序设计;很多科学计算领域还在使用 FORTRAN 语言的一个主要原因是遗留的 FORTRAN 程序修改为面向对象的程序的工作量太大,另一部分原因是图形界面的必要性不大,或者可在计算完成后用某些软件对计算结果图表显示。

C++ 语言和 Java 语言在形式上相似,具有语言简洁的特点,非常受专业的计算机编程者喜爱。C++ 语言适合平台类软件的开发,而 Java 语言适合跨平台类软件的开发。C++ 语言和 Java 语言的思想是非常好的,但它们也存在很多缺点:

(1) 形式和概念很不友好。

代码 4-1 是一个在 Borland C++ 3.1 语言编译器下通过的 C++ 语言的源代码的头文件(*.h 文件),其中含有一个描述河流的横断面的类 CrossSection 和一个以 CrossSection 对象为成员的、用来描述构成一条河流的横断面的集合(包括相应的断面间距)的对象 River。CrossSection 对象有一个 CrossSection 方法,River 对象有一个 River 方法,分别是这两个对象的构造函数,相应地,CrossSection 对象有一个~CrossSection 方法,River 对象有一个~River 方法,分别是这两个对象的析构函数。说 C++ 语言形式很不友好表现在它用一个和类的名字一样的方法作为起初始化功能的构造函数(就如上面的 CrossSection 方法和 River 方法),用一个类的名字前面加波浪号的方法作为析构函数的名字,这本身就让初学者不易理解;再加上给这两个函数起一个构造函数和析构函数这样令人费解的名字,即使对于计算机专业的初学者也不容易很快领会其真正的含义。

(2) 它的源代码用太过简洁的符号表示,不容易阅读。

例如,它用“{”和“}”来划分程序块,“{”和“}”确实使代码书写更简洁和更灵活,但它模糊了各种程序块的差别。程序员必须联系上下文来判断程序块是那一种,但阅读起来就很辛苦,特别是有多个程序块嵌套的时候。又如用惊叹号“!”代表逻辑否定,也不太利于阅读。

代码 4-1

```
// #define DEBUG_DELETE //display prompt "delete..." when delete
#include "math.h" //for adjusted_section_area()
//No. 1 a class
class CrossSection //description of ONE cross section
{
    int nxy;
    int no;
    double dL;
    double L;
    double *px;
    double *py;
public:
    double dist();
    double area(double z);
    double Zmin();
    void setnxy(int n)
    {
        nxy = n;
        if( ! (px = new double[nxy]) )
        {
            cerr << "new err in class CrossSection, px = new double [" << nxy << "]" \n";
            system("mem");
            exit(1);
        }
        if( ! (py = new double[nxy]) )
        {
            cerr << "new err in class CrossSection, py = new double [" << nxy << "]" \n";
            system("mem");
            exit(1);
        }
        return;
    }
    ~CrossSection(void)
    {
        #ifdef DEBUG_DELETE
        cout << "\ndelete [" << px; delete [ ] py; nxy = " << nxy << " \n";
        #endif
        delete [nxy] px;
        delete [nxy] py;
    }
}
```

```

friend istream & operator >> ( istream & stream, CrossSection &o );
friend ostream & operator << ( ostream & stream, CrossSection o );
};

istream & operator >> ( istream & stream, CrossSection &o )
{
    double * px = o. px, * py = o. py;

    stream >> o. no;
    stream >> o. dL;
    stream >> o. L;
    o. L * = 1000.0;

    for( int i = 0; i < o. nxy; i ++ , px ++ , py ++ )
    {
        stream >> * px >> * py; //read ( x,y ) one by one
//        cout << " * px = " << * px << " * py = " << * py << endl;
    }
    return stream;
}

ostream & operator << ( ostream & stream, CrossSection o )
{
    double * px = o. px, * py = o. py;
    cout << "\"CrossSection: nxy = \"";
    stream << o. nxy << " ";
    stream << o. no << " ";
    stream << o. dL << " ";
    stream << o. L << endl;

    for( int i = 0; i < o. nxy; i ++ , px ++ , py ++ )
    {
        stream << * px << " " << * py << endl;
    }
    return stream;
}

double CrossSection::dist( void )
{
    return L;
}

```



```

    }

double CrossSection::area( double z)
{
//      return adjusted_section_area( px,py,nxy,z,0);
      return section_area( px,py,nxy,z);
}

double CrossSection::Zmin( void )
{
    double zmin = 8848.0, * p = py;
    for( int i = 0; i < nxy; i ++ )
    {
        if( * p < zmin)
            zmin = * p;

        p ++ ;
    }

    return zmin;
}

//No. 2 Note: the function isn't class's member
double VolumeBetween( CrossSection * p1, double z1, CrossSection * p2, double z2)
{
    double a1 = p1->area( z1 ), a2 = p2->area( z2 );
    double v;
    v = ( a1 + sqrt( a1 * a2 ) + a2 ) / 3.0 * fabs( p2->dist() - p1->dist() );
//    v = ( a1 + a2 ) / 2.0 * fabs( p2->dist() - p1->dist() );
    return v;
}

//No. 2 Note: the function isn't class's member
//seek the meeting point of two line
//line1 ( x1,y1 ), ( x2,y2)
//line2 ( x3,y3 ), k
//return the x of the meeting point
double XmeetPoint( double x1, double y1, double x2, double y2, \
    double x3, double y3, double k)
{
    return ( y1 - y3 + k * x3 - ( y2 - y1 ) / ( x2 - x1 ) * x1 ) / ( k - ( y2 - y1 ) / ( x2 - x1 ) );
}

```