

适合开发者 项目经理 CTO&CIO 编程爱好者阅读收藏

程序员

2003合订本(上)

《程序员》杂志社 编



《程序员》全年12期

《csdn开发高手》全年4期

超大增值内容: CSDN网站全年精华
百本书评
百家网站推荐
程序员工作手册



光盘: 全刊电子文件及源码下载
知名软件公司最新工具试用
10本最经典的技术书
50个最值得学习的源码
编程手册

www.csdn.net/magazine



程序员

2003合计本(上)

总策划：蒋 涛

编委会：曾登高

闫 辉

欧阳璟

常 可

唐 琦

徐 勉

关峻峰

徐 锋

孟迎霞

熊建国

孔祥利

周松奕

孟 岩

金旭亮

吴志民

刘如鸿

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

《程序员 2003 合订本》由程序员杂志社联合电子工业出版社最新推出。本合订本沿袭 2001、2002 年《程序员合订本》的风格，具有内容丰富、技术文章专业、信息资料全、价格实惠等特色。

全书为大 16 开，分上下两册，共 832 页，配 1 张光盘。在《程序员》杂志 2003 年 1 ~ 12 期精华文章合集的基础上，增加了近一半篇幅的增值专题，分别为 1 ~ 4 期《CSDN 开发高手》精华合集、专家推荐“最爱 100 本书”、本刊编辑推荐的 100 个最优秀技术开发网站评点、程序员手册（内含新名词术语集萃、工具集萃、常用模板集萃与技术规范集萃等）。

配套光盘包括 12 期《程序员》电子文件、4 期《CSDN 开发高手》电子文件及相关源码下载、Borland 等知名软件厂商最新的工具试用、10 本最经典的技术书、50 个最值得学习的源码以及编程手册（包含大量软件开发规范和标准、软件项目文档的模板大全）。

京海工商广临字 2003046 号



2003合订本（上）

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

程序员 2003 合订本（上） / 《程序员》杂志社编. —北京：电子工业出版社，2004.1
ISBN 7-5053-9420-7

I.程... II.程... III.程序设计—文集 IV.TP311.1-53

中国版本图书馆 CIP 数据核字（2003）第 116181 号

责任编辑：孟迎霞

印 刷：

出版发行：电子工业出版社

北京市海淀区万寿路173信箱 邮编100036

经 销：各地新华书店

开 本：850×1168 1/16 印张：52 字数：1600 千字

印 次：2004年1月第1次印刷

定 价：39.00元（上、下册+1CD）

读者订购或发现装订错误、缺损、破损，请与《程序员》读者服务部联系。

联系地址：北京市朝阳区北三环东路静安中心26层（100028）

电话：010-84480948/84540231~33转279

传真：010-84540263

家评刊

Programmer create the future of IT

——微软亚洲研究院院长兼首席科学家 张亚勤

《程序员》杂志力图为中国程序员的发展指明方向！

——金山公司总裁 求伯君

是一本非常专业的杂志，在倡导技术新潮流方面做了很大的努力，值得软件开发从业人员仔细阅读

——点击科技总经理 王志东

改版后的《程序员》更好更精美了。加上色彩之后，刊物的质感及技术文章的图表描述性有飞跃进展。《程序员》在技术内容的深广、技术报导的多样、版面水准的精整，目前已超越台湾所有技术性杂志，我个人确实如此认为。祝《程序员》日精月进。

——台湾著名技术作家，《深入浅出MFC》作者 侯捷

同我们培养适应市场需要专业人才一样，这本杂志在为软件开发与市场的结合上也起到了重要的作用。

——北京航空航天大学软件学院院长 孙伟

对于广大的开发人员来说，《程序员》应该算是每月必读的刊物之一。在众多杂志中，《程序员》的文章质量与编排方式均优于其他杂志，并且随时将最新的技术介绍给读者。即使是身为专栏作家的我，每个月也可以从中获取其他优秀程序员的开发心得，并以此为乐。

——台湾著名技术作家 何致亿

读者来信

如果你从未阅读过《程序员》杂志，

她将送给你：

- 一本最权威的技术杂志全年精华
- 12 期《程序员》杂志全年完整电子版
- 2003 年软件开发领域最前沿的技术文章
- 本年度软件企业最关注的管理与人才焦点
- 本年度最值得学习的图书、工具点评

如果你是《程序员》杂志的读者，

她将送给你：

- 4 本《开发高手》完整杂志
- 300 多页超值专题
- 10 本最经典的技术书
- 50 个最值得学习的源码
- 专家推荐 100 本书评
- 最全的规范、最实用的开发模板
- 一本最实用的开发技术手册

**如果你是 2002、2001 年《程序员增值合订本》
的老朋友，**

她将送给你：

- 12 期《程序员》完整杂志
- 4 本《开发高手》最新刊
- 一本焕然一新的 2003 年增值合订本
- 一张高容量、超丰富的光盘手册
- 一个最有价值的 2004 年新春礼物

王伟（学生）

我以前相继阅读过 2001 年、2002 年的增值合订本，非常喜欢这位老朋友。

《程序员增值合订本》从整体上讲，全书内容结构十分合理，各版块搭配很好，非常适合我学习。她在帮助提高我技术能力的同时，也指导我如何进入软件开发领域，所以我会一如继往地支持贵刊的工作。同时，也十分感谢你们对工作的认真负责！

闫英（系统架构师）

总体而言，《程序员增值合订本》有三大优点：技术领先；职业之师；经典收藏。

祝贵刊越办越好，我永远支持你们！

周京生（电气工程师）

在达州很难找到《程序员 2002 增值合订本》，为此我花了将近一天的时间去寻找。非常幸运，我找到了一家书店，并买下了那个书店的最后一套书！希望将来你们的销售渠道能更好一些。阅读后，对合订本的整体感觉非常好，尤其值得收藏！光盘里的内容丰富，不过我更关心的是教程和规范。最后祝贵刊越办越红火！

欧阳（在读研究生）

通过《程序员合订本》2002 上期的增值部分就可以看出编辑们的良苦用心。几乎全部的文章都是以前没有见过的，内容非常充实。作为一个计算机专业并热爱编程的学生，花大笔的钱购买专业书籍无疑是非常困难的，而且要在这其中找到真正适合自己学习的书，也并非轻易之事。《程序员》给了我们一个很好的了解软件开发世界的平台，她给了我很大的帮助。

《程序员 2003 合订本》配套光盘简介

- 杂志电子版：2003 年全年《程序员》和《CSDN 开发高手》电子版，PDF 文件
- 杂志配套源码：全年《程序员》和《CSDN 开发高手》配套源码
- 10 本经典技术书：经典的“Thinking In……”系列和侯捷先生的精彩著作，开放电子版
- 工具集萃：优秀的开发平台和工具，如 Together、Python、Eclipse……，特别提供李维先生制作的 Delphi 8.NET 开发精彩范例演示
- 50 个优秀项目源码：收录了 50 余个优秀的开源项目，如 Apache 组织、ASP.NET 范例、C++ 程序库……
- 标准、规范和模板集萃：包括软件项目开发的标准、规范以及大量常用的项目文档模板

人物 & 报道

名人堂

书写神话的布鲁克斯	1
汤姆·迪马可：六十耳顺	1
软件产业的设计大师：艾兰·库伯	2
JBuilder 第一功臣：Blake Stone	2
尼克劳斯·威茨：Pascal 及结构化程序设计的首创者	3
IBM 小子：John Cocke	4
自学成才的计算机编程先驱：Robert W. Floyd	4
杰拉尔德·温伯格——软件与系统领域的“醒客”	5
开源领袖 & 顶尖黑客 Eric Raymond	5
Ed Yourdon：近乎不朽	6
英国人改变世界——WWW 发明者 Tim John Berners-Lee	6
俄罗斯方块设计者：亚力克西·帕杰诺夫	7

调查

最爱 10 本书揭晓	8
新年电脑新选择	9
关心程序员（工作篇）	10
关心程序员（学习篇）	12
关心程序员（爱情篇）	13
程序员收藏网站	14
我的黑夜比白天多	16
程序员空间革命	18
武装数码程序员	19
程序员维权知多少	21
谁动了程序员的 money?	22
战胜疲劳	24

特别专题

马不停蹄又一年——2002 年开发技术大综述	26
从程序员到数据库管理员	28
从程序员到配置管理员	31

人物访谈

“C++ 之父”访谈	33
Bertrand Meyer——Eiffel 语言奠基人	35
一位 OO 专家的思考——Craig Larman 问答	38
关注程序员自己的文化——专访 Tom DeMarco	40
开源软件成功之本	42
C++ 之父：Bjarne Stroustrup	44
J2EE 高手袁红岗	45
EJB 设计模式——Floyd 访谈录	47
软件思想家 Gerald Weinberg 专访	49
听 Anders Hejlsberg 讲 C# 设计过程	51
“戴眼镜的蜘蛛”——James Gosling 访谈	54
Ruby 的设计思想	56
错位	58

管理版块

程序人生

在日本做程序员	59
一个 35 岁的美国程序员	60
情系互联网的程序员	61
能比武招亲就好——一个女程序员的心声	61
40 岁考上系统分析员	62
后生可畏——CSA 最年轻的系统分析员	62
来自摩尔多瓦的编程小子	63
飞翔最高的程序员	63
在微软亚洲研究院的日子	65
IT 写作“情”史	67

开发故事

为什么我喜欢 EJBs?	70
JBoss：偏执者的生存	72

PowerBuilder 传奇	74
光荣与梦想——Perl 传奇	78
PHP: 创造奇迹的网络脚本	81
Palm 之父 Jeff Hawkins 传奇	83
面试取经	
轻松面试, 搞定技术奇才	85
15 位业界专家的面试高招	87
软件创业	
开放源码的风险评估	90
一个软件小老板的创业法则	91
软件外包三要素	92
分工和对它的抵制: 看软件开发中的三种重要角色	93
每日构建是你的朋友	95
香港程序员大陆淘金	97
为什么我喜欢开放源码	99
台湾程序员的活法	100
JBoss 内幕故事	102
抓住市场那只手——论开发两要素: 最优化程序和易用性	106
绿色之旅: 史前 Java	107
程序员跳槽的分析及对策	109
解析程序员老化	111
项目管理	
项目启动三部曲	113
挣值分析法帮你考核项目费用	116
项目实施 步步为营	118
软件开发三部曲之项目收尾	122
游戏制作全接触	124
软件质量保障与软件企业发展	127
软件产品实施经验汇编	129
成功项目的杀手——领导意志	132
项目迷失的要因——目标不明确	133
一个微型软件开发的过程手记	134
软件工程论坛	
CRM 现状及发展	137
用项目管理法实施 SCM	138
EAI——企业集成的解决方案	140
数据仓库项目的十大重症	143
信息建模方法、工具及应用	144
那支写程序的笔——软件工业的修辞学	147
理解企业应用框架	148
软件测试之自动测试	150
利用 Ant 来完成 Daily Building 和 Testing	153
基于 J2EE 的产品框架设计最佳实践	156
体验特征驱动开发	158
实现异构	161
感悟测试驱动开发	164
CTO 论道	
微创的秘密——专访微创研发事业部经理蔡铭	167
情系软件开发管理工具——专访 Hansky (中国) 公司 GM&CTO 刘开阳 ..	168
解决方案	
石化金卡工程实践方案	170
电信系统架构方案	173
面向对象的医疗系统设计	175
组件化分布式开发系统在移动增值业务的应用	178
构建税务部门信息服务平台	181
软件开发与管理新视野	
企业·产品·人(上)	183

栏目	文章名	页码
	企业·产品·人(下)	187
	立项管理	191
	凌波微步 II	
	寻找“小李飞刀”	196
	从“魔力整合针线”谈起	199
	奇妙的二叉树	203
	设计“好看”的用户界面	207
	喜欢一种语言需要理由吗?	210
	侯捷观点	
	Java 的对象永续之道(上)	214
	Java 的对象永续之道(下)	217
	李维专栏	
	.NET 开发工具劲旅——Borland C#Builder	221
	C#Builder 的企业级快速数据库存取技术	224
	使用 C#Builder 和设计模式在 .NET 中管理 Unmanaged 程序代码	228
	VCL Framework 的演化——VCL.NET	232
	细说 Borland C++BuilderX	235
	王森专栏	
	Personal Java 程序设计入门	239
	PocketPC 程序设计	241
	Sun ONE Application Server 7 程序设计(一)	245
	Sun ONE Application Server 7 程序设计(二)	247
	JavaServer Faces 程序设计 JavaServer Faces 基本架构与基本程序设计	249
	胡百敬专栏	
	SQL Server 全文检索简介(上)	256
	SQL Server 全文检索简介(下)	260
	蔡学镛专栏	
	.NET 中间语言	263
	.NET 执行引擎	265
	.NET 安全性	268
	函数指针的进化论	270
	何致亿专栏	
	JBuilder 8 功能概览	274
	建构与执行 Java 应用程序	277
	数据库技术的新纪元——谈 Oracle10g 与企业网格计算	280
	Oracle 10g 的“数据闪回”机制	282
技术版块		
	电脑英语	
	垃圾收集算法	284
	关于 SQL	285
	EPOC OPL 开发介绍	286
	如何进行项目规划	288
	XML Web 服务基础	289
	JSF 简介	290
	Jabber 简介	292
	PoPFile	293
	献给所有热爱编程和游戏的人们——怀念图形魔术师 Seumas McNally · 作序的艺术	294
	JMSML: 基于 XML 的标记语言	296
	ACE 新名词解析	299
	编程擂台	
	搜寻恐怖分子题解	301
	高性能计算机问题	303
	蓄水池问题	305
	DNA 测序问题	306
	空中交通控制问题	308

栏目	文章名	页码
	招聘问题	309
	文件时间问题	311
	仓库问题	312
	WWW 搜索问题	314
	病毒问题	316
	飞弹问题的算法分析	317
增值专题		
	百本书评	
	最爱 100 本书	319
	.NET	319
	C/C++	321
	Java	322
	Delphi	324
	基础理论	325
	安全	326
	技术人文	327
	人工智能	328
	软件工程	329
	数据库	334
	系统开发	335
书评		
	好书推荐	
	2002 年软件开发类图书市场热点回顾	336
	2002 年亚马逊开发类获奖图书	338
	我的嵌入式书架	339
	我的 Delphi 书架	340
	我的 .NET 书架	342
	我的 EJB 书架	343
	新书推荐	345
	《程序员》好书榜: .NET 专题	346
	《程序员》好书榜: 软件工程专题	348
	好书榜: Java 专题	350
	好书榜: 算法与数学专题	351
增值专题		
	CSDN 2003 年精华帖	
	C/C++ 版	353
	VC/MFC 版	362
	BCB 版	371
	VB 版	376
	.NET 版	379
	Java 版	383
	Oracle 数据库版	386
	软件工程版	387
	硬件版	389
	SQL 版	390
增值专题		
	CSDN2003 年经典文章集萃	
	ASP.NET Web Service 是如何工作的?	391
	ASP.NET 页面对象模型	392
	运行 .NET 程序的秘密	394
	为什么选择 Delphi.NET?	395
	Java 语言 TSP 递归程序的优化	396
	用 Java 实现颜色透明值混算的一种方法	398
	Weblogic 服务器中的 JNDI 概述	399
	UNIX 与 Linux 内核信息输出过程比较	400
	设计模式之 C# 实现——ProtoType	401
	将 Java 程序作为 Linux 的 Daemon 进程以及防止程序多次运行	402
附录		
	《程序员》2003 全年目录	403

书写神话的布鲁克斯

20 世纪最后一年（也就是 1999 年）的图灵奖，授予了年已 69 岁的资深计算机科学家布鲁克斯（Frederick Phillips Brooks, Jr.）。在 60 年代初，布鲁克斯只有 29 岁时就主持并领导了被称为“人类从原子能时代进入信息时代标志”的 IBM/360 系列计算机的开发工作，取得辉煌成功，从而名噪一时。以后他作为硬件和软件的双重专家和出色的教育家始终活跃在计算机舞台上，在计算机技术的诸多领域中都做出了巨大的贡献。在计算机科学领域，布鲁克斯的名字俨然已经成了一个“神话”。

布鲁克斯 1931 年 4 月 19 日生于北卡罗来纳州的杜哈姆。1953 年毕业于杜克大学，并进入哈佛大学深造，1955 年取得硕士学位，1956 年取得博士学位。他的博士论文课题工作是在哈佛著名的计算实验室进行的，最终完成的博士论文题目为“自动数据处理系统的分析设计”。从博士论文开始，布鲁克斯的一生就与计算机结下了不解之缘。

在哈佛取得博士学位以后，布鲁克斯进入 IBM 公司设立在纽约波凯普茜的实验室当工程师，并在那里参加了 Harvest 和 Stretch 计算机的开发，任体系结构设计师。1959 年，布鲁克斯曾被调至 IBM 在约克郡高地的研究中心工作，但翌年又重新被调回波凯普茜的实验室，并出任 IBM/360 项目的主持人。IBM/360 的开发总投资 5 亿美元，达到美国研究原子弹的曼哈顿计划投资的 1/4。在研制期间，布鲁克斯率领着 2000 名程序员夜以继日地工作，单单 360 操作系统的开发就用了 5000 个人年。因此，在 IBM 公司纪念其成立 50 周年的庆祝大会上，360 系列计算机被称为“公司历史上发布的最重要的产品”。为此，布鲁克斯常常被称为“IBM/360 之父”。

360 成功以后，布鲁克斯离开 IBM 回到故乡，为北卡大学创建了计算机科学系，并担任系主任长达 20 年（1964 ~ 1984 年）。除了教学以外，他还致力于发展美国的计算机技术和计算机在国防等方面的应用，有许多社会兼职。1966 ~ 1970 年，他是 ACM 全国委员会的委员；1973 ~ 1975 年出任 ACM 体系结构委员会的主席；1977 ~ 1980 年布鲁克斯在美国国家研究院计算机科学技术部任职；1983 ~ 1984 年他是美国国防科学委员会人工智能攻关领导小组的成员，1986 ~ 1987 年是上述委员会另一个攻关领导小组“计算机模拟和训练”的成员；1985 ~ 1987 年他担任军用软件攻关小组组长。1987 年布鲁克斯当选为美国工程院院士，他同时也是英国皇家学会和荷兰皇家科学与艺术院的外籍院士。

在授予图灵奖之前，ACM 在 1987 年曾授予布鲁克斯“杰出服务奖”，1995 年曾授予他以纽维尔（A. Newell, 1975 年图灵奖获得者，1992 年去世）命名的 Newell 奖。加上这次的图灵奖，布鲁克斯成为继克努特（D.E. Knuth, 1974 年图灵奖获得者）之后的第二位同时拥有 ACM 三个奖项的计算机科学家。

IEEE 也先后向布鲁克斯授予了 McDowell 奖（1970 年）、计算机先驱奖（1982 年）和冯·诺伊曼奖（1993 年）。AFIPS 在 1989 年授予布鲁克斯 Harry Goode 奖。数据处理管理协会 1970 年授予他“计算机科学”奖，并命名他为该年度的风云人物。1985 年布鲁克斯因在开发 IBM/360 上的杰出贡献而荣获全国技术奖章。物理学界的富兰克林学会（Franklin Institute）也曾授予布鲁克斯 Bower 奖。

布鲁克斯的著作不多，但影响都很大。1963 年他和依费逊（APL 发明人，1979 年图灵奖获得者）合著了《自动数据处理》一书。他还与荷兰特文德理工大学的勃芬夫教授合著了《计算机体系结构：概念与发展》。除了学术性著作外，1995 年，他与苏泽兰特（“计算机图形学之父”，1988 年图灵奖获得者）等还合编了一本书，书名是 Evolving the High Performance Computing and Communications Initiative to Support the National Information Infrastructure，论述了有关高性能计算机计划及信息基础设施（也就是所谓“信息高速公路”）建设的一系列问题。

而布鲁克斯最引人关注的著作无疑是《人月神话》（The Mythical Man-Month: Essay on Software Engineering）。1975 年，他把他历年来的有关软件工程和项目管理方面的文章汇集成这本书，立即引起了轰动。由于本书是他领导 IBM/360 软件开发经验的结晶，内容丰富而生动，成为软件工程方面的经典之作。在出版 20 年之后，在 1995 年，他又对书中内容进行了修缮，并加上了后来的《没有银弹》和《再论“没有银弹”》这两篇文章，出版了一个《“人月神话”20 周年纪念版》。一本计算机科学专著能够在盛行 20 年之后再版，《人月神话》也成了个绝无仅有的“神话”。（2003.01）



汤姆·迪马可：六十耳顺

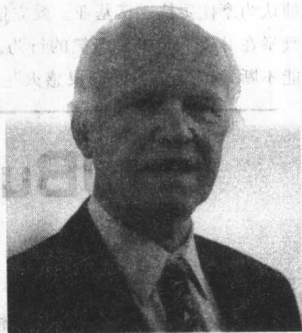
在汤姆·迪马可（Tom DeMarco）的脸上，似乎总有微笑挂著，就像在《最后期限》封底的那张照片一样。孔子云：三十而立，四十不惑，五十知天命，六十耳顺。当中国的大多数程序员还在为自己能不能“立”得起来而感到困惑迷惘的时候，迪马可用他标志性的微笑告诉他们：不管你在年轻时曾经怎样叱咤风云、怎样锋芒毕露，到六十岁的时候都会“耳顺”，会用宽容的快乐去面对世界。

1940 年 8 月 20 日，汤姆·迪马可出生于美国宾夕法尼亚州的黑泽尔顿。他在康奈尔大学读完本科，随后在哥伦比亚大学获得硕士学位，并在巴黎大学获得软件硕士学位。他的第一份工作是在贝尔电话实验室，参加的第一个项目就是传奇般的 ESS-1 项目。在后来的日子里，他曾经在法国的 La CEGOS Informatique 公司管理实时项目的开发，也曾经在瑞典、荷兰、法国、芬兰等国家负责在线银行系统的售前服务。他多次在世界各地发表演讲或给其他企业提供咨询服务，足迹遍及美洲、欧洲、非洲、澳洲和远东。现在，他是大西洋系统行会的负责人之一。

在那本著名的《自适应软件开发》中，James Highsmith 指出：软件

开发最初是“自在自为”的，随后有了偶像崇拜。在第一批被崇拜、被神化的“偶像”中，包括 E. Dijkstra、David Parnas、Edward Yourdon、Larry Constantine，当然，也有汤姆·迪马可。在 1968、1969 年连续召开的两次 NATO 会议上，正是他们提出了“软件工程”的概念。1978 年，E. Yourdon 和 L. Constantine 共同提出了结构化软件开发方法；1979 年，迪马可对结构化方法进行了改进。在后来的 20 年中，结构化方法

对整个软件产业产生了无法估量的巨大影响。在 1978 年，迪马可还推出了自己的第一部著作。这本书的名字大得能把人吓出一身冷汗：《结构化分析和系统设计》。凭着这本名字能吓死人（内容确实经典）的著作，



迪马可正式走上了神坛，成为不折不扣的“一代宗师”。

因为在结构化时代作出的巨大贡献，汤姆·迪马可在1986年荣获Warnier“计算机领域终身贡献奖”，在1999年又荣获Stevens“软件开发方法贡献奖”。尽管获得了众多的褒奖，迪马可也清楚地看到：当软件从军队、政府走进平常百姓家时，结构化方法和传统软件工程思想已经成了明日黄花。Fred Brooks在1986年和1995年两度强调“没有银弹”，彻底将软件工程赶下了神坛。人月是神话，银弹无处求，人们开始认识到：软件工程并不能解决一切问题，软件产业也不可能像硬件产业那样有一个神奇的“摩尔定律”。盲目乐观终于变成悲观。

但是，早已“知天命”、即将进入“耳顺”之年的迪马可却显得更加乐观了：当人们终于意识到软件工程并不是万灵药之后，他们就能在最适当的地方使用软件工程，并积极地去寻求通过其他途径来解决软件工程

所不能解决的问题。在1995年之后，他的作品都明显地带有这种乐观态度。1997年的《最后期限》描写了一位幸运的汤普金斯先生克服困难获得成功的故事；1999年的《人件》带有一种“加料过火的Dilbert式幽默”；2002年推出的新书《Slack》更被批评者贬为“无产者的呓语”。面对“过于乐观”的批评，迪马可借汤普金斯先生的故事告诉他们：既然你已经没有更好的办法，那么就雇一个最好的人，给他最好的待遇，然后祈祷上帝保佑吧——软件工程是救不了你的。

除了技术之外，汤姆·迪马可的生活还有另外一面：小说家。他写的主流小说《Dark Harbor House》受到读者的广泛好评，最近他又推出了第二部主流小说Lieutenant America and Miss Apple Pie。以技术专家的身份来写小说，这样的闲情逸致，怕也只有一位“耳顺”的老人才能拥有吧。(2003.02)

软件产业的设计大师：艾兰·库伯

“一个所谓‘优良的设计’是建立在对广泛的人类共通性和特殊群体个别意向的深刻理解之上，因此它可以对适当的信息、过程和技术进行排序，从而使那些特殊群体达到他们的专业和个人目的。”

出席艾兰·库伯(Alan Cooper)演说的听众好像被一致催眠了似地猛着头点。库伯不仅在网站开发人员，而且在每个IT人身上都找到了共鸣，这种能力加强了他对听众的深刻影响，并迫使设计者对原先的设想产生疑问。他在强调他的观点时并不是凭着激昂、热情的词汇，而是通过一些精心考虑、能够深刻影响听众思考方式的言词来完成。

在数字化人群中，库伯以“Visual Basic之父”著称。1992年，艾兰等人共同创建了库伯交互设计公司(www.cooper.com)。以公司为基础，这位具有洞察力的软件设计师与编程大家肩负的使命是：阐明当前软件开发过程中的“设计差距”——由于对使用产品的实际用户考虑不周，开发过程经常在接近完成的时候功亏一篑。

二十年来，艾兰·库伯设计并开发了许多消费类软件产品，包括SuperProject、MicroPhone II for Windows，以及用于Visual Basic的微软可视编辑用户界面等。他著有两本享有盛名的书：About Face: The Essentials of User Interface Design和The Inmates are Running the Asylum(后者的中译本名为《软件开发的创新思维》)。

作为Visual Basic的发明者，库伯借着发行畅销的书籍和主持专题演讲巩固了他的先知地位，并促成了交互式设计策略和个人化概念的普及。库伯现在心目中的三个基本设计原则是精简的设计、快速的页面下载和健全、无误的除错管理。

作为一个设计专家，库伯对现代信息社会的设计大都不以为然。他认为Windows、Linux和Mac的界面大同小异，一样地过时、灰暗而单调；他认为HTML是垃圾，它限制了信息的传播和人与人之间的交流；他认为摩托罗拉、诺基亚、爱立信的手机设计都同样糟糕，“说话时将键盘举在耳边”是一种可笑的行为。正是由于这种永不满足的态度，他才能不断地迸发出设计的灵感火花。

与所有有效设计专家相似，库伯不认为互联网上存在着天堂，他自己描绘了一幅在现在网页设计完全改变后网络浏览的景象。他告诉我们，通过遵循他为大家指引的道路前进，我们可以不再重复以前的软件开发者所犯下的错误。

1994年，比尔·盖茨授予库伯“视窗先锋奖”，以表彰他为软件产业所作的贡献。在1998年，他又接受了“软件梦幻奖”。作为让世界了解软件开发者的代言人，库伯活跃于一系列专业设计组织，包括协作设计基金会、美国工业设计学会、美国设计中心等。同时，他还是美国图形艺术专业研究所交互设计分组的核心成员之一，并主持了软件开发论坛。



艾兰·库伯的设计戒律

没有所谓“门户网站”这种东西，门户网站也不见得适合所有的造访者；
不要对您的造访者要求承诺或其它任何东西；
以多元化的形式提供大量详细、有用的信息；
不要让使用者通过一个复杂的、多层的网站导航而最后发现他们仍需要通过电话才能得到答案；
除了在不得已的情况下，尽量不要使用动画(电影除外)；
不要只为了吸引大量眼球而设计。使用者是公民，不是消费者；
不要提前发货，不要经常发货，不要拿使用者作实验；
就像一场恶梦，浏览器对互联网的支配最终会消失。(2003.03)

JBuilder 第一功臣：Blake Stone

JBuilder 已经成为公认的“最佳Java IDE”。但你是否知道，在JBuilder的发展历程中，有一位软件天才居功至伟，他就是JBuilder现任首席设计师Blake Stone。

1997年，Borland公司推出的第一款Java开发工具——Open JBuilder在市场上的反映并不如预期的好。为此，Borland内部面临一个抉择：如何继续Open JBuilder的发展，是用Delphi重写Open JBuilder的IDE，

还是尽量用纯Java来开发？作为Borland在Java领域的第一次尝试，Open JBuilder陷入了困境。

形势在Open JBuilder的新产品架构领导人Blake Stone加入后才逐渐明朗。Blake Stone原本是DSW Systems Corporation公司的技术主管，DSW公司一向和Borland互动良好，许多DSW公



司的人都曾在 Borland 的 Conference (BorCon) 中负责技术讲座。Blake Stone 先生也在 1997 年的 BorCon 中负责了一个讲座。也许是 Blake Stone 和 Borland 在 BorCon 中的合作愉快, Borland 很赏识 Blake Stone 的技术和才华。在 BorCon 结束之后不久, Borland 便有意和 Blake Stone 接触, 询问其是否愿意加入 Borland 的 Java 研发小组。也许是天意, 在 Borland 失去了 Anders Hejlsberg 这个天才之后, 老天又给了 Borland 一个难得的补缺机会。

在双方接触后没多久 Blake 便答应进入 Borland 作为 JBuilder 的 Architect, 使命是带领 JBuilder 成为最成功的 Java 开发工具。由于 Blake 惊人的天分, 他很快成为 JBuilder 的主要 Architect 以及技术的主领导者。此后, JBuilder 未来开发的 Java 技术都由 Blake 负责研发的工作。

Blake 进入 JBuilder 开发小组之后决定把 JBuilder 带向纯 Java 类开发工具, 把整个 JBuilder 使用 Java 语言本身来打造。这一策略很快被证明是正确的。后来, JBuilder 能够支持 Windows、Linux 和 Solaris 平台, 成为当时架构最大、最复杂的 Java 应用程序。更重要的是 SUN 充分肯定了 Borland 在 Java 方面卓越的技术, 进而采用 Borland 的 Baja 技术制定 Java Bean 规格, 并邀请 Borland 共同参与开发 JDK。Blake 在 JBuilder 早期选择了正确的方向, 奠定了 JBuilder 成功的基础。稍后 JBuilder 新的产品经理 Tony de la Lama 又成功制定了 JBuilder 的市场研发和竞争策略, 终于让 JBuilder 在 3.5 之后一飞冲天, 成为 Java 开发工具中的翘楚。

在 BorCon99 中, Blake 在讲座开始之后就拿出了 Notebook 专心写程序, 让人以为他参加讲座只是消磨时间。但在讲座结束之后, 主讲人开始接受询问问题, Blake 却不断地举手发问——Blake 看起来能够一心多用, 不但脑袋可以想东西, 手指可以敲键盘, 心思还能够倾听讲座的内容, 实在是一位少年天才。而且 Blake 身材纤细, 又留了一头长发,

不知情的人从身后看一定会认为这是一位美丽的小姐呢。

1999 年 7 月, Borland 首次在公司内部设立了首席科学家的荣誉职位, 以颁给 Borland 最优秀和重要的软件人员。Blake Stone 当年就和 Chuck Jazdzewski 以及 Andreas Vogel 同时获得了 Borland “首席科学家”大奖, Blake 对 JBuilder 的贡献也算是实至名归。Blake 在进入 Borland 短短的数年时间内, 就成为获此殊荣的最年轻者, 这也证明了他惊人的实力。

JBuilder 6 的一个“彩蛋”, 列出了开发团队的名单, Blake 高居榜首。他的荣耀史包括:

Borland 最年轻的首席科学家

成功带领 Jbuilder 成为世界第一的 Java 开发工具的软件天才

成功克服 Linux 上当初没有标准 JDK 的问题, 让 JBuilder 能够在 Linux 上执行

世界 Java 专业论坛的主讲人

主导 Borland Java 开发工具和技术的核心人物

现在 Blake 愈来愈受到 Borland 的重视, 除了原本的 JBuilder 产品之外, 后来 Borland 并购 OptimizeIt、进一步强化 JBuilder 整体竞争力也是 Blake 的主张。Blake 已经慢慢从 JBuilder 产品线转至负责 Borland 大部分 Java 技术的关键人物。日前, Borland 宣布 Blake 还将负责新的生命周期管理软件 (ALM) 的研发。看来 Borland 下一代 Java 产品也将由 Blake 贡献心力。

继续加油吧, Blake! (本文改编自李维先生《Borland 传奇》)(2003.04)

尼克劳斯·威茨: PASCAL 及结构化程序设计的首创

学过计算机的人都知道“算法 + 数据结构 = 程序”这一著名公式, 提出该公式的正是 1984 年的图灵奖获得者, 瑞士计算机科学家尼克劳斯·威茨(Niklaus Wirth)。到目前为止, 他是获得图灵奖殊荣的惟一位瑞士学者。

威茨于 1934 年 2 月 15 日生于瑞士北部的 Winterthur, 他从小就喜欢动手动脑, 他的最大爱好是组装飞机模型。1958 年从苏黎世工学院取得学士学位后, 他到加拿大的莱维大学深造, 之后进入美国加州大学伯克利分校获得博士学位。

上世纪 50 年代末到 60 年代初, 威茨设计了第一个语言 Euler。Euler 语言在实用性上考虑并不周全, 在学术上却非常优秀, 为编译器的系统设计创造了很好的基础。接下来, 威茨在斯坦福大学定义了另一种语言来描写 Algol W 的编译器, 由此催生了新的语言——PL360。PL360 是作为辅助工具开发的, 却出人意料地在许多地方获得应用。

Algol W 及 PL360 的成功奠定了威茨作为程序设计语言专家的地位。成名后的他拒绝了斯坦福大学的挽留, 于 1967 年回到祖国, 先在苏黎世大学任职, 第二年转到母校苏黎世工学院。在这里, 他在 CDC6000 上成功设计了 PASCAL 语言。说来有趣, 威茨开发 PASCAL 的初衷只是为了有一个适合于教学的语言, 没有考虑商业用途。PASCAL 一经推出, 由于它的简洁明了, 更由于它特别适合于由微处理器组成的计算机系统, 竟然广泛流传开来。在 C 语言问世以前, PASCAL 成了最受欢迎的语言之一。威茨的学生菲利浦·凯恩(Phillipe Kahn)在美国加州就先后卖出了 100 多万套 PASCAL 拷贝, 其影响可见一斑。

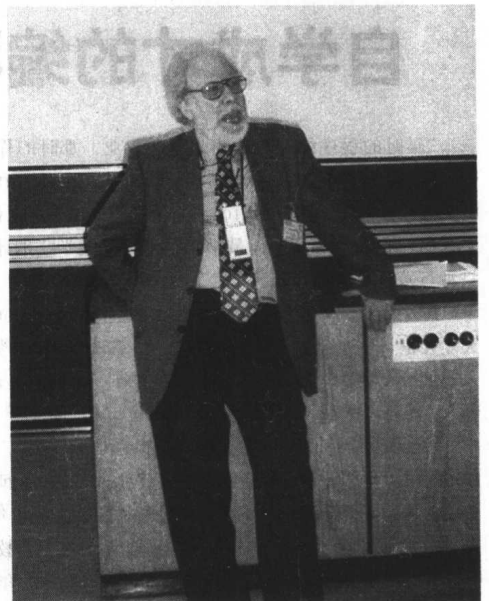
1971 年, 基于自己的开发程序设计语言和编程的实践经验, 威茨首次提出了“结构化程序设计”(Structured Programming)的概念。威茨提出的这种结构化程序设计方法又称为“自顶向下”或“逐步求精”法, 在程序设计领域引发了一场革命, 成为其中的一种标准, 尤其在后来发

展起来的软件工程中获得广泛应用。

PASCAL 的成功和结构化程序设计思想带来的巨大影响, 都没有停止威茨继续创造性的研究与开发工作。1970 年代中期, 为适应并发程序设计的需要, 威茨又成功开发了获得广泛应用的 Modula 语言。后来, 参考 Alto 的经验, 他还设计开发了 Lilith 个人计算机系统。

威茨的学术著作很多, 著名的包括《系统程序设计导论》、《算法 + 数据结构 = 程序》、《算法和数据结构》、《PASCAL 用户手册和报告: ISO PASCAL 标准》等书籍。

除得到了美国计算机制造商协会 (ACM) 颁发的图灵奖外, 他还得到过 ACM 授予的另一项大奖: 计算机科学教育杰出贡献奖。威茨现仍在苏黎世工学院任教, 他的电子信箱为: wirth@inf.ethz.ch。(2003.05)



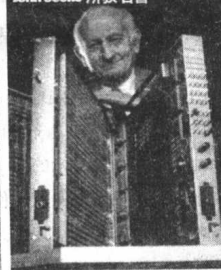
IBM 小子: John Cocke

“IBM 小子”，是 RISC (Reduced Instruction Set Computer, 精简指令系统计算机) 架构设计师——John Cocke，在 1972 年得到的 IBM 公司颁赠给内部员工的最高荣誉称号。同年，John Cocke 还获得了美国国家技术勋章和图灵奖。1991 年，当时的老布什总统亲自授予他国家科学奖（国家技术勋章和国家科学奖是美国最具荣耀的两项政府大奖）。在作为科学家的一生中，他在高性能系统设计中取得的革命性进步，Cocke 为信息技术的发展做出了独一无二的创造性贡献。在系统架构和编译器优化研究领域取得了大量进步，他当之无愧地成为编译器优化的奠基人和革新带头人。不幸的是，在与病魔长期搏斗后，John Cocke 于 2002 年 7 月 16 日与世长辞。

John Cocke，1925 年出生于北卡罗莱那州 (North Carolina) 的夏洛特 (Charlotte)。在杜克大学 (Duke University)，Cocke 先后取得了机械工程学士学位和硕士证书，1956 年他获得数学博士学位。第二年，Cocke 加盟 IBM 公司直到退休。他在退休后还依然为 IBM 公司发挥余热，直到病故。

Cocke 在 IBM 公司从事的第一个项目是研究 Stretch 计算机 (世界上第一个“超级计算机”型号)，他很快成为大型机专家。1974 年，Cocke 和他领导的研究小组开始尝试研发每秒能够处理 300 线呼叫的电话交换网络。为了实现这个目标，他不得不寻找一种办法来提高交换系统已有架构的交换率。1975 年，John Cocke 研究了 IBM370 CISC (Complex Instruction Set Computing, 复杂指令集计算) 系统，发现占总指令数 20% 的简单指令发出了 80% 程序调用，而占总指令数 80% 的复杂指令却只有 20% 的机会被用到。由此，他提出了 RISC 的概念，其中心思想就是简化硬件设计，硬件只执行一部分很有限的最常用的指令，大部分复杂的操作则使用成熟的编译技术，由简单指令合成。RISC 的最大特点是指令长度固定，指令格式种类少，寻址方式种类少，大多数是简单指令且都能在一个时钟周期内完成，易于设计超标量与流水线，寄存器数量

John Cocke 所获名譽



- 1994 年 获得国家科学奖章、计算机和通信奖金 (NEC 基金)
- 1993 年 当选美国科学院院士
- 1992 年 荣获美国年度发明奖
- 1991 年 受赠美国国家技术奖章
- 1989 年 受赠 IEEE 计算机学会先驱奖
- 1987 年 荣获 ACM 图灵奖
- 1985 年 荣获 ACM/IEEE 计算机学会 Eckert-Mauchly 奖
- 1979 年 当选美国工程院院士

多，大量操作在寄存器之间进行。

1980 年，Cocke 在 IBM 位于约克城 (Yorktown) 的华生研究中心 (Watson Research Center) 里开始研制 IBM801 (PowerPC 的前身)，第一台 RISC 机器也就作为 801 微电脑项目的一部分最终开发成功。

自 RISC 技术推出以来，由于其优化指令系统带来运算速度提高的优势，使得 RISC 技术在 1980 年代后期，逐渐在高端服务器和工作站领域中取代了 CISC 成为主流微处理器设计架构之一。各个具备一定技术实力的厂家开始在这个架构上研发出自己的处理器，经过近二十年的发展，各大型计算机和超级服务器都采用 RISC 架构的处理器。现在，RISC 处理器已经成为高性能计算机的代名词。

“从 1957 到 1992 年，John Cocke 在工作中把自己近 40 年的精力毫无保留地奉献给了 IBM，他取得了了不起 (amazing) 的成绩”，Cocke 的同事兼密友 Peter Capek 这样评价他：“他的经历非同一般。他因为计算机架构而闻名，但他对很多领域同样充满浓厚兴趣。包括逻辑模拟 (logic simulation)、编码理论 (coding theory) 和编译器优化 (compiler optimization)，他都可以把这些技术当成艺术来研究。” (2003.06)

自学成才的编程先驱: Robert W. Floyd

“早时的程序编写被认为是不务正业，那时的程序帮不了工作什么忙。”计算机程序与艺术协会名誉教授 Donald Knuth 回忆说，“是计算机工程师 Robert W. Floyd 证明了程序完全可以代替人工，是他改变了人们对程序的看法。他最早总结出来的数学联结计算机科学的言论是一种行业性革命”。

Floyd 于 1936 年 6 月 8 日出生在纽约，他曾在 6 岁的时候被人视为天才儿童，后连升 3 级结束高中课程。在取得 14A 奖学金后他被允许进入芝加哥大学学习。1953 年，17 岁的 Floyd 在芝加哥大学文学艺术学士学位，后于 1958 年他拿到双学士学位。

“他奉献给计算事业的一生是努力自学的一生”，他的第二位太太，德国汉堡大学的计算机科学家 Christiane Floyd 回忆说，“他利用夜间轮流值班的时间开始自学编程的，他通过阅读 1960 年代出版的程序书籍来学习如何编程。当他在卡内基·梅隆大学当上副教授进入计算学术界时年方 27 岁。他取得斯坦福大学全职教授之职时，年纪也不过 32 岁。”

虽然 Floyd 没有享受到博士学位带来的荣耀，但于 1968 年被任命为斯坦福计算机科学系副教授之前，他已经撰写了一组至少 12 篇，被视为比当时的博士论文更出色的计算机科学论文集。计算机编程语言是当时计算机科学中一个最热门的话题。“但关于这个话题的论文只有 4 篇论文，他们全部出自 Floyd 之手”。

Floyd 的研究方向包括以寻求网络最短路径的运算法则的设计和浮点运算、点阵打印机的灰度打印阴影、信息索引和随机交换与聚合的筛选等。其中，Floyd 的最重要的科学成就就是开创了程序检验的系

统方法论。1967 年，他的启蒙性论文 Assigning Meaning to Programs 开创了程序检验新方法。他的基本想法是在逻辑位置表中附加所谓的“标记”，以此来区别程序描述或分支，以定义基于常用程序语言的定义程序效果。世界范围内众多的计算方法研究人员采纳了他的观点。

同时，Floyd 还是代码重构——基本理念是重新改写程序，只保留程序基本概念——的第一位倡导者。重构是目前计算机编程中的基本操作。在持续探索事半功倍的努力中，Floyd 不仅致力于改善和优化程序，同时还寻求提高程序员的开发能力和理解力。

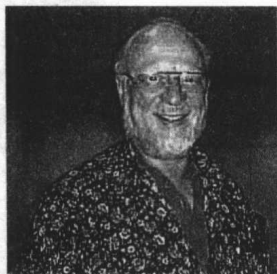
在斯坦福大学，他教授涵盖“索引和搜索”的运算法则课程，Floyd 还和他以前的学生 Richard Beigel 一起合力编撰了教材——《机器语言：可计算性和正式语言》(计算机科学出版社：1994)。自 1973 年至 1975 年，Floyd 担任斯坦福大学计算机科学系主任，这是计算机科学系高速增长为斯坦福重点学科的时期。

1978 年，“因为创建了有效和可信赖软件方法论，这大大方便了建立计算机科学的重要子域：剖析理论、编程语言语义学、自动程序检验、自动程序集成和算法分析”，Floyd 获得了 ACM 颁发的图灵奖。1991 年，因为早期编译器的研制工作，IEEE 计算机协会授予 Floyd 计算机先驱奖。1994 年 Floyd 从斯坦福大学退休时，他已经是美国科学与艺术学会会员。

1997 年 9 月 25 日，Floyd 在斯坦福大学医疗中心病逝，享年 65 岁。(2003.07)



杰拉尔德·温伯格——软件与系统领域的“醒客”



美国人杰拉尔德·温伯格 (Gerald M. Weinberg) 在软件和系统领域已经浸淫了不下 45 年。从 1956 到 1969 年间, 他先后担任过软件开发员、系统设计员、软件项目经理、大学教授、以及 IBM 公司软件课程的设计者。之后, 他和身为人类学家的妻子成立了 Weinberg & Weinberg 咨询公司, 来帮助软件工程师以更彻底的人性方式管理软件过程。

1997 年, 因温伯格在软件领域的杰出贡献, 他被美国计算机博物馆的计算机名人堂选为首批 5 位成员之一。这个名人堂至今只有 20 名成员。比尔·盖茨和迈克尔·戴尔等也是在他之后才获得这一计算机界至高无上的殊荣。

温伯格刚出生的时候非常弱小, 大约只有 9 磅左右。但他自小就精力旺盛、思想活跃, 11 岁的时候他开始对计算机产生兴趣, 并决定长大了要与之相伴。在这种想法指导下, 他学习了数学和物理 (那时学校还没有计算机课)。因此, 到目前为止, 温伯格还从未上过任何正式的计算教育课。

从 20 世纪 70 年代开始, 或独立创作, 或与人合作, 温伯格先后撰写了 30 多本书籍和数以百计的论文。他最早期的著作主要集中在操作系统和编程语言领域。随着经验的增长, 他发现如果人们在技术应用和构建之时对于人文方面的理解没有给予足够重视, 技术就变得毫无价值——甚至危险。于是, 温伯格决定在作品中加入人文领域的内容。1971 年, 温伯格成功出版了《程序开发心理学》(The Psychology of Computer

Programming) 一书, 其中研究了软件开发、测试和维护中的关于人的过程, 这被认为是研究软件工程作为人的行为的开端。该书现在已经是银版 (自第一次出版至今已经有 25 年), 这充分说明了人们对于理解其工作中人文部分的渴求。温伯格后期的著作实际上是对这本书中提出的许多问题的一个更详细的探索, 它们涉及到了软件生命周期里的各个阶段, 从问题和必要条件的定义、分析和设计、测试和测量, 一直到管理。在西方国家, 温伯格拥有大量忠实的读者群, 这些“追星族”阅读了他的每本重要著作, 他们甚至建立有专门的组织和网站来讨论和交流大师的重要思想。

温伯格的主要贡献集中在软件领域, 他是从个体心理、组织行为和企业文化角度研究软件管理和软件工程的权威和代表人物。在其职业生涯中, 温伯格从事过软件开发, 软件项目管理、软件管理教学和咨询。

温伯格喜欢创造那些帮助人们的新事物。他在自传中写道: “虽然我写了几百篇文章和 30 多本书, 我最满意的还是创造学习现实生活经验的环境——学校、培训基地、大学、研讨会或是其他种类的团体。如果我不是在发起一种新社会体系, 那么我总是在修理旧的体系; 在这个意义上, 我是组织的临床医生。还有, 我也喜欢与一个人的组织工作; 也许我最称职的是作为一名艺术作品的修补者。”

经过了将近 40 年的计算机工作, 温伯格仍然认为自己对于计算机的大多数领域知之甚少。他说: “我发现人几乎是所有问题的根本。我努力重新体会这一道理的含义, 并学习如何去做一些与之相关的事情。通过自学, 我还学会了第二个原理: 我是我大多数问题的根本。”

这就是他, 温伯格, 一位程序员, 一位设计师, 一位咨询师, 一位专业作家, 更是一位杰出的思想家。(2003.08)

开源领袖 & 顶尖黑客: Eric Raymond



埃里克·斯蒂芬·雷蒙 (Eric Steven Raymond) 是《大教堂与市集》的作者、《新黑客词典》的维护人、著名黑客。作为《新黑客词典》的主要编撰人以及维护者, Eric Raymond 很早就被认为是黑客文化的历史学家以及人类学家。但是在 1997 年以后, Eric Raymond 被广泛公认为是开放源代码运动的主要领导者之一, 并且是最为大众所知道 (并最具争议性) 的黑客。

Eric Raymond, 1957 年出生于美国马萨诸塞州的波士顿, 从小就跟随父母在世界各地东奔西走。1971 年他回到美国宾夕法尼亚州, 从 1976 年起开始接触黑客文化 (他以自己姓名的缩写 ESR 作为黑客代号), 1982 年完成了他的第一个开放源代码软件项目。

从 1978 年 Eric Raymond 开始真正涉足黑客文化, 最早的经验是原始的 ARPAnet, 那时的他涉猎甚广, 甚至短期接触过 ITS 机器。1984 年自由软件基金会成立以前, 他已经在写自由软件并投递到 Usenet, 这样顺理成章地他成为了自由软件基金会最早的撰稿人之一。Eric Raymond 是一个不注重物质生活的人。1985 年, 他辞去了最后一份工作, 此后就开始集中精力于编程、写书及发表技术评论这些他所热爱的、感兴趣的事情上, 并再也没有重新为金钱而工作。1990 年, Eric Raymond 编辑

了《新黑客字典》。1996 年, 他写成了《黑客道简史》。并把自己定位为“研究计算机黑客文化”的人类学家。

1993 年, Eric Raymond 接触到 Linux, 这对他是一个冲击。因为他认识到: 业余黑客尽管可能很有天分, 但不可能集聚所需的资源和技巧来建立一个可用的多用户操作系统。而 Linus Torvalds 和他的小组最终不光推出了 Linux 操作系统本身, 还有数百兆的各类应用程序、文档和其他资源、整套的 Internet 工具、桌面出版软件、图形支持、编辑器, 他们创造了或创造着你能说出的任何软件。这个冲击最终使 Eric Raymond 写出了《大教堂与市集》(The Cathedral and the Bazaar), 并于 1997 年 5 月在 Linux Kongress 上发表。《大教堂与市集》受到了空前的欢迎, 就如同《圣经》对基督教的作用一样, 它为开放源代码运动指明了方向。

1997 年以后, Raymond 成为了开放源代码运动的主要理论家, 以及开放源代码促进会 (Open Source Initiative) 的主要创办人之一。他还担任了开放源代码运动对媒体、商界以及主流文化的形象大使。他是一名优秀的演说家, 曾经到过六大洲的 15 个国家进行演说。他的话经常被主流媒体所引用, 也是所有黑客中曝光率最高的。

最近, Eric 刚刚完成了他的另一本书《The Art of UNIX Programming》。在这本汇集了他数十年编程经验的图书中, Eric 全面描述了大师级 UNIX 黑客的编程哲学与编程技巧。人们相信, 这本书将会是 UNIX 和黑客文化史上的又一本标志性著作。

此外, Raymond 还是一个彻头彻尾的自由主义者。他对科幻小说十分感兴趣; 是一名出色的业余音乐家, 还是空手道黑带选手。他坚持美国宪法修正案第二条中关于公民持枪械的自由, 并因此遭到很多黑客的反对, 但他自己却十分喜欢和人争辩该议题。(2003.09)

Ed Yourdon: 近乎不朽

Ivar Jacobson 曾经不乏自得地提到过一场“方法论之战”和它的休战。的确，上世纪90年代初，“软件工程”领域还处在群雄并起的局面——据OMG组织（UML、CORBA背后的那个帝国）统计，当时仅仅“面向对象”方法论，就有27种之多。“圣人出而黄河清”，Jacobson自诩，他本人、Grady Booch、Jim Rumbaugh这“三位老友（the three amigos）”联手之后，应该天下一统，战事早该结束、硝烟也应散去了。

统一的最大受益者也许是开发者们——我们再也不用面对方法论学者们层出不穷的争执和分歧。但硬币的另一面，这个领域的景色也变得略嫌无趣：RUP已经变得过于稳定，像中晚期的罗马帝国，高踞宝座、打着呵欠、和SEI的CMM评估师们有一搭没一搭的调着情，自己都有些厌倦自己了吧；而新兴的Agile联盟，在短短几年内也变得丰满、成熟，它的前途不再无可限量，而在它的限度之外，我们还有什么呢？

Ed Yourdon可能是上述方法论之战的最大牺牲者，因而也是我个人私下最怀念的声音之一。在RUP还未一统天下的史前时期，Yourdon属于软件工程领域最有影响力的顶尖神祇。1979年，他（和另一位大师Larry Constantine合著）的《Structured Design》一书，廓清了“结构化开发”的王国。该书的故事也很有意思：这本书的原型是1975年Yourdon在一个炎热的小岛上完成的手稿（整个写作只耗时26天）；而当该书由Yourdon自己的出版社YOURDON Press印行后，也就迅速成为当时软件界的圣经。人们根据该书的封面，给它起了“橙皮书”的雅号，这也是名著作者们能享有的最高荣誉之一。后来Yourdon把出版社卖给了大社Prentice Hall，后者重印该书时，不恰当地把封面改成了绿色——一个最能招致读者、作者愤恨的做法。时至今日，它还被残存的一些“结构化”开发人员喜爱。不过Yourdon自称，该书目前每年的版税，也就够他和Constantine享用些啤酒、比萨饼了。

如果Yourdon停留于此，他大概确实只剩下这点零食。但在随后的面向对象大潮中，Yourdon仍是最重要的先觉者之一。90年代初的黄金岁月里，他和Peter Coad大师合著的《Object Oriented Analysis》和稍后的《Object Oriented Design》两书又成了这个领域中经典参考资料。而前者更奠定了以“Coad-Yourdon”命名的OOA方法。今天Yourdon本人自己对OO领域的贡献时认为，OOD中的大部分内容基本上已经被UML覆盖了（是呀，即使是Agile联盟的各种实践，也同样以UML为重要工具），但《OOA》一书则还是他的宠儿。Yourdon的妙语是：一本介绍全新方法论的著作，应该能在纽约/芝加哥的一趟飞行中读完——这也是他的《OOA》、Martin Fowler的《UML Distilled》、Kent Beck

的《XP Explained》的最大好处。

所谓“食髓知味”，但感恩却不是开发者们的美德。当UML、RUP成为主流之后，Yourdon必须新的领域证明自己，否则就难逃被程序员们遗忘的下场。Yourdon的几部近作，也确实在另一个方向上对软件业产生了影响：人们发现，这不是一位单纯的技术专家，毋宁说，他变成了思辩者，变成了先知或社会分析家。名著《Death March》（中译《死亡之旅》，译名不够恰当）直面软件项目管理问题，被很多人认为是最新版的《人月神话》（事实上从书名上我们也能追索出接近的风格）；而Yourdon最受争议的两部著作又构成了名称上的对称：《Decline and Fall of the American Programmer》（美国程序员的衰退和失落）、《Rise and Resurrection of the American Programmer》（美国程序员的崛起和复兴）。试图扮演先知角色的Yourdon，致力于探究软件开发中的劳动力问题，在前一部书中，他认为IT外包将对北美程序员供求造成重大影响，开发者们应该早作准备——在经济膨胀的90年代末，这样的警告无异于危言耸听，不仅广大程序员不信，就连Yourdon自己也有些尴尬，连忙在后一部书中加以弥补，称赞RUP、CMM等实践的重要意义。

在警示者自己都怀疑的时候，狼还是来了——经济的衰退完美地符合了Yourdon的预言，但他在此前的一些过度言论（他认为Y2K问题非常严重，甚至为此从纽约搬到新墨西哥）仍在他的名誉上投下些许阴影。你也会说，Yourdon从“技术专家”到“人文主义者”的转型并不完美（他甚至出版过几部纯虚构小说），但他的正直、广博、他对多个领域的贡献仍使他近乎成为一个不朽者，一个人们最需要倾听的声音。（2003.10）



Ed Yourdon，麻省理工学院应用数学学士、布宜诺斯艾利斯CAECE大学名誉教授，1997年进入美国“计算机名人堂”，现为咨询顾问、技术作家。他正在筹划出版《Death March》的第二版。

“英国人改变世界”

——World Wide Web发明者Tim John Berners-Lee

早上起床，泡上一杯咖啡，坐到电脑前打开浏览器看新闻——这是很多人每天的例行公事。当我们陶醉于在链接之间自如跳转，是否也曾经偶尔好奇过：到底是谁想出了这聪明的信息发布方式？在我们迅速将这个“无聊”的问题抛诸脑后时，那可敬的发明者却仍然继续为WWW的未来努力 working。不过，他已不是孤身一人，而是领导着整个W3C。

他，就是Tim John Berners-Lee，World Wide Web的发明者。

Tim 1955年6月8日生于英国伦敦。在牛津大学女王学院读书时，他曾使用M6800处理器和旧电视自己攒了一台计算机。1976年毕业后，他先后受雇于Plessey Telecommunications Ltd、D.G Nash Ltd两家公司。在D.G Nash，Tim曾写过多个任务操作系统，但真正开启未来之门

的，却是之后在CERN工作的一年半。

CERN (European Particle Physics Laboratory，欧洲量子物理研究所) 位于瑞士日内瓦。在那里，为了工作需要，Tim写了一个从未发布过的工具Enquire——这个我们无缘得见的Enquire，正是World Wide Web的雏形。不过在此之后Tim就到John Poole's Image Computer Systems Ltd进行实时控制固件方面



的开发工作去了。1984年,当Tim获得奖学金,重新回到CERN之后研究实时系统时,他逐渐发现CERN内部沟通机制存在信息漏失的弊端。今天我们可以找到的关于WWW概念的第一份公开文件,是Tim于1989年3月写给CERN的建议书:Information Management:A Proposal。在这份文件中,Tim提出利用Hypertext(超文本)构造链接信息系统的设想。同样,我们也可以从文件中看到“Browser(浏览器)”概念的最初提出。

这时Tim已经开始在Enquire的基础上,基于NeXTStep系统开发第一个真正意义上的Web Server——httpd、第一个客户端浏览器——WorldWideWeb,之后又在1991年建立并开通第一个WWW网站http://info.cern.ch/ (该网站至今仍然是CERN的官方站点)。到1993年,Tim再接再厉,制定了URL、HTTP、HTML等的第一个规范。

1994年, Tim加入位于麻省理工大学的LCS(计算机科学实验室)。为了让World Wide Web不被少数人所控制, Tim组织成立了World Wide Web Consortium,即通常所说的W3C,致力于“引导Web发挥其最大潜力”。我们所熟知的HTML协议各个版本,都出自W3C会议。可贵的是,W3C的HTML规范是以“建议”的形式发布,并不强迫任何厂商或个人接受。至于微软利用HTML协议的开放性扩展自有标准,打败Netscape,应该是Tim始料未及的了。

《时代》杂志在评价Tim的贡献时这样写道——“和其它影响世界的发明不同,这项发明的确应该归功于一人……Berners-Lee设计了World Wide Web,然后就把它开放给世界。他比其他任何人都更努力地保持WWW的开放性、非营利性和自由性……很难对WWW做出适当评价,它几乎可以媲美古登堡印刷术。Berners-Lee一手把只有精英们掌握的通讯系统变成了大众媒体。”

这段话实实在在并非溢美之词。Eric Schmidt也曾说过:“如果计算机网络是一门传统科学,Berners-Lee早该获诺贝尔奖了。”我想, Tim本人不会在意这个。当他把上个世纪最伟大的传媒界发明完全无偿地奉献给全世界,天堂已准备好了一双洁白的翅膀。Tim Berners-Lee是一位具有极高创造力的程序员,但更是一位品格高尚的盗火者。

在《织网(Weaving the Web)》一书中, Tim提出三个观点:

◎ 编辑Web与浏览Web一样重要,Wiki是向正确方向踏出的一步。

◎ Internet的任何表现形式都应该呈网状,而非树形结构。域名系统例外。

◎ 计算机科学家在技术层面和道德层面都负有不可推卸的责任。

也许我们可以从这些论断中获益良多,也许只当它是无稽之谈。不管怎样, Tim Berners-Lee发明了World Wide Web,并继续在为之做出贡献。这,就已足够。(2003.11)

俄罗斯方块设计者: 亚力克西·帕杰诺夫

如果有人告诉你,曾设计过最有影响力游戏的设计师根本就没有从这个作品中拿到过一分钱,你会不会相信?

这位设计师就是莫斯科的一位数学家亚力克西·帕杰诺夫(Alexey Pajitnov),他所设计的最成名作品《俄罗斯方块》,被公认为有史以来最畅销的游戏,至今魅力不减。

亚力克西·帕杰诺夫曾在俄罗斯科学院电脑中心工作,当时他在Electronica 60的旧型俄罗斯主机上进行开发工作,这种机器不支持图形界面,和PC也不兼容。

虽然帕杰诺夫的主要工作是语音识别及人工智能开发,但他热爱电脑游戏,同时又是个拼图迷。他尝试将一种传统的拼图游戏Pentomino改写成电脑游戏。Pentomino有十二种形状,各以七个方块组合而成。帕杰诺夫试着写了一个程序,旋转并移动这十二种不同形状的方块,让它们拼成一大块。

“这太有趣了!”帕杰诺夫一直在寻找让人入迷的游戏,他知道,心目中构思许久的东西出现了!“如果可以即时互动那就几乎是完美的了!”七个方块十二种组合在当时要即时互动有些太复杂,帕杰诺夫决定构思一种四个方块七种组合的简易版本。

只用了两个星期,帕杰诺夫就写出这个游戏的第一个版本,并给这个游戏起了一个源自希腊字“四”(tetra)的名字——Tetris,俄罗斯方块诞生了!

同事们对这个小游戏大呼过瘾,但却只能到Electronica 60主机上玩。帕杰诺夫将改写成PC版本的工作交给了一位16岁的电脑高手伯丁。很快,俄罗斯方块的第一个PC版本开始在莫斯科流传了。几个星期之内,这个游戏就占领了莫斯科的每一台个人电脑。

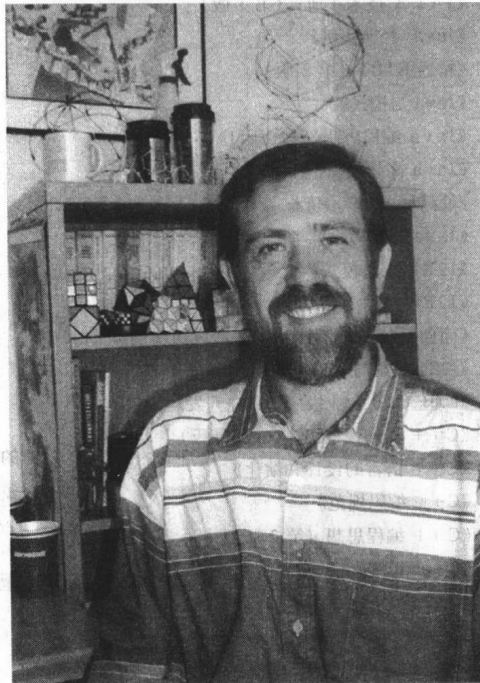
俄罗斯方块的影响毋庸置疑,当今全世界的每一台电脑、手机、PDA,到处都有俄罗斯方块的踪迹,同时它更是每个游戏平台的必备游戏,相关的复制品数不胜数,随之而来的版权官司也层出不穷。最后的结果是:律师、游戏发行商和俄罗斯的官僚们瓜分了本应属于帕杰诺夫的巨大的财富。

虽然有了一段辉煌的历史,但帕杰诺夫并未空闲下来。二十世纪九十年代初,帕杰诺夫到了美国,为任天堂等一些公司开发游戏。

有一天,微软公司突然打电话给他,决定投资让帕杰诺夫独立完成属于他自己的方案。这一行动的结果就是游戏《潘多拉的匣子》(Pandora's Box)。游戏从基本的构思到每个谜题的类型无不渗透着他的心血。他设置了不计其数的谜,让它们彼此独立而又相互关联,散布于各处。一个古典七巧板类的谜题,就有350多个变化。所有的

谜题都是可视性的,无论从排列组合还是从图形上的空间匹配上,比起那些字符、数字或逻辑谜题更胜一筹。“许多人不愿在玩游戏时投入时间,他们仅仅想要得到一种快速爆发的娱乐,”帕杰诺夫说:“因此,我们决定设计由数以百计的短谜题组成的游戏,但又将它们捆绑在一个情节中。”为了使游戏制作得更合理、更真实、更有趣,帕杰诺夫在故事情节的构思上几乎倾注了所有的时间和精力。因此,游戏中谜题的深度和内涵是空前的。

不过,至今依然没有哪个游戏能像《俄罗斯方块》那样,寓无穷的樂趣于简单的玩法中,而作为电脑游戏史上最具有影响力的游戏设计师之一,帕杰诺夫也将永远被记入史册。(2003.12)



程序员调查 4

喜爱系列

最爱 10 本书揭晓

作为软件开发人员、项目管理人员提升自己的一个最直接介质——图书，历来是大家最为关注的热点话题之一。为了激励相关作者和出版社在 2003 年出版更高品质的好书，《程序员》杂志联合 CSDN 网站举办了“2002 年度最佳开发类图书评选”调查活动。根据读者、网友的提名和专家推荐，经过严格评审，产生了以下候选书目：

获得提名资格的图书（共 85 本）

（评选范围：2002 年国内出版社出版的原创、翻译、影印版开发类图书）

《C# 程序设计》
《Microsoft c# Windows 程序设计》（上、下册）
《Applied Microsoft .NET Framework Programming》（影印版）
《Programming Windows with C#(Core Reference)》（影印版）
《C# COM+ 编程指南》
《精通.NET 核心技术——高级特性》
《精通.NET 核心技术——原理与构架》
《JAVA 技术手册》（第 3 版）
《Java 经典实例》
《精通 EJB》（第 2 版）
《Java 与模式》
《Java 编程思想（第 2 版）》
《Java 高效编程指南》
《最新 Java 2 核心技术 卷 1: 原理》（原书第 5 版）
《Java 深度历险》
《J2EE 核心模式》
《J2EE 技术内幕》
《面向对象程序设计——图形应用实例》
《J2EE 应用与 BEA WebLogic Server》
《EJB 2.0 企业级应用程序开发》
《C++ 程序设计语言》（特别版）
《C++ 语言的设计和演化》
《C++ 编码规范》
《C++ 编程思想（第 2 版）》
《C++ 沉思录》
《C++ Primer 中文版》（第 3 版）
《C++ Primer 题解》
《C++ 程序设计语言题解》
《C++ Builder 深度历险》
《STL 源码剖析》
《C++ Builder5 程序设计大全》
《C++ 标准程序库——自修教程与参考手册》
《More Exceptional C++ 中文版》
《C++ 代码设计与重用》
《C 陷阱与缺陷》
《Delphi6 应用开发指南》
《Delphi6 / Kylix2 SOAP / Web Service 程序设计篇》
《Delphi6 企业级解决方案及应用剖析》
《Delphi 第三方控件使用大全（II）》

《Java 编程思想》（第 2 版）
《C++ Primer 中文版》（第 3 版）
《C++ 程序设计语言》（特别版）
《精通 EJB》（第 2 版）
《精通.NET 核心技术——原理与构架》
《设计模式：可复用面向对象软件的基础》（英文版）
《人月神话》
《TCP/IP 详解 卷 1：协议》（英文版）
《C++ 编程思想（第 2 版）第 1 卷：标准 C++ 导引》
《STL 源码剖析》

《XSL 技术内幕》
《Visual C++ 编程深入引导》
《数据库系统导论》（英文版，第 7 版）
《Transact-SQL 权威指南》
《UML 数据库设计应用》
《Delphi/Kylix 数据库开发》
《算法导论》（第 2 版 影印版）
《软件开发的科学与艺术》
《计算机程序设计艺术》（英文影印版）
《计算机程序设计艺术》（中文版）
《从规范出发的程序设计》
《程序设计实践》（英文版）
《计算机体系结构：量化研究方法》（英文版，第 3 版）
《操作系统概念》（第六版 影印版）
《分布式系统——原理与范例》（英文影印版）
《计算机网络》（第 3 版）
《系统分析与设计》
《系统分析与设计》（英文版）
《软件工程：实践者的研究方法（第 5 版）》
《设计模式：可复用面向对象软件的基础》
《设计模式：可复用面向对象软件的基础》（英文版）

《分析模式》
《人月神话》
《编写有效用例》
《编写有效用例》（英文版）
《UML 和模式应用：面向对象分析与设计导论》
《统一软件开发过程》
《快速软件开发——有效控制与完成进度计划》
《UML 参考手册》
《UML 用户指南》
《CMM 实践应用——Infosys 公司的软件项目执行过程》
《个体软件过程》
《小组软件开发过程》
《软件需求管理：统一方法》
《程序调试思想与实践》
《UML 精粹——标准对象建模语言简明指南（第 2 版）》
《软件开发的滑铁卢——重大失败项目的经验与教训》
《ERP 原理·设计·实施》
《Win32 多线程程序设计》
《4.4BSD 操作系统设计与实现》
《TCP/IP 详解 卷 1：协议》（英文版）
《TCP/IP 详解 卷 2：实现》（英文版）
《TCP/IP 详解 卷 3：TCP 事务协议、HTTP、NNTP 和 UNIX 域协议》（英文版）
《Windows 图形编程》
《计算机图形学原理及实践：C 语言描述》（英文版，第 2 版）
《Windows 程序调试》
《UNIX 网络编程卷 2：进程间通信》（第 2 版）（英文影印版）
《最高安全机密（第 3 版）》
《黑客大曝光：网络安全机密与解决方案》（第 2 版）
《编写安全的代码》
《面向对象的软件测试》
《软件测试》

此次评选活动从策划到实施，再到结果揭晓，共历时一月左右，期间得到了广大读者和网友的极大关注和配合，在 CSDN 网站上引起了很大的反响，正式评选阶段网友投票数高达 2 万 4 千多张。评选步骤：1) 编辑、专家、网友提名；2) CSDN 网上投票。

调查结果如下

1、最受欢迎的 10 本书：

名次	书名	作者	出版社	ISBN	票数
1	《Linux 编程思想》（第 2 版）	Ernst Eicht (美) / 屠 洪强译	机械工业出版社	Jans	2417
2	《C++ Primer 中文版》（第 3 版）	Stacey E. Lipman/Stone Lauer (美) / 潘爱民 张松海译	中国电力出版社	C++	2361
3	《C++ 程序设计语言》（特别版）	Bjarne Stroustrup (美) / 高 伟 译	机械工业出版社	C++	2259
4	《精通 EJB》（第 2 版）	Ed Roman (美) / 雷 何能明译	电子工业出版社	Jans	2237
5	《精通.NET 核心技术——原理与构架》	Ericnie/Stone Lauer (美) / 潘爱民 张松海译	电子工业出版社	JNET	2095
6	《设计模式：可复用面向对象软件的基础》（英文版）	Erch Gamma/ Richard Helm/ Ralph Johnson/ John Vlissides (美) / 李 彦 译	机械工业出版社	软件工程	1912
7	《人月神话》	(美) Frederick P. Brooks, Jr. 著 汪 明译	清华大学出版社	软件工程	1673
8	《TCP/IP 详解 卷 1：协议》（英文版）	W. Richard Stevens (美) / 李 彦 译	机械工业出版社	系统开发	1495
9	《C++ 编程思想（第 2 版）》 第 1 卷：标准 C++ 导引	Bjarne Eicht (美) / 屠 洪强 黄兆山 潘秋明译	机械工业出版社	C++	1477
10	《STL 源码剖析》	侯捷/著	华中科技大学出版社	C++	1452

2、图书分类排行榜：

名次	书名	作者	出版社	票数
1	《精通.NET 核心技术——原理与构架》	屠洪强/著	电子工业出版社	2399
2	《Linux 编程思想》（第 2 版）	Ernst Eicht (美) / 屠洪强译	机械工业出版社	2473
3	《C++ Primer 中文版》（第 3 版）	Stacey E. Lipman/Stone Lauer (美) / 潘爱民 张松海译	中国电力出版社	2368
4	《Delphi/Kylix 数据库开发》	李彦/著	机械工业出版社	1575
5	《数据库系统导论》（英文版，第 7 版）	C.J Date (美) / 潘 彦 译	机械工业出版社	1186
6	《计算机图形学原理及实践：C 语言描述》（英文版，第 2 版）	Donald E Knuth (美) / 屠洪强译	清华大学出版社	783
7	《面向对象程序设计——图形应用实例》	Erch Gamma/ Richard Helm/ Ralph Johnson/ John Vlissides (美) / 李 彦 译	机械工业出版社	1842
8	《C++ 编程思想（第 2 版）》 第 1 卷：标准 C++ 导引	W. Richard Stevens (美) / 李 彦 译	机械工业出版社	1495
9	《编写安全的代码》	Michael Howard/ David LeBlond (美) / 李 彦 译	机械工业出版社	1387
10	《软件测试》	Eric Patton (美) / 潘 彦 译	机械工业出版社	1323