

软件工程师丛书



Visual Basic 6

高级编程策略与范例

—— 错误处理编码与分层技术

[美]Tyson Gill 著
段来盛 宋黎松 等译



电子工业出版社
Publishing House of Electronics Industry
URL: <http://www.phei.com.cn>

软件工程师丛书

Visual Basic 6

高级编程策略与范例

——错误处理编码与分层技术

Visual Basic 6:Error Coding and Layering

^{T. 格}
[美] Tyson Gill 著

段来盛 宋黎松 等译

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

Visual Basic 往往被人们认为是一种易学、易用,但不稳定、不可靠的编程语言,而本书的作者根据自己的多年编程经验,提出了灵巧编码三角和安全编程框架的理论,可大大改善这一局面。本书分为错误处理编码和分层应用程序开发技术两大部分。第一部分通过介绍可重用性、标准化和错误处理编码这三部分构成的灵巧编码三角,来达到实现有效错误处理编码和预防错误的目的,第二部分介绍的分层的开发策略,大大提高了编程的效率和质量,降低了出错的可能性。本书提出的理论来源于实践,实用且有效。不仅对 Visual Basic 的初学者、分析人员、管理人员,而且对使用各种编程语言的编程高手来说,都是其提高编码质量,改善开发策略的良师益友。

Authorized translation from the English language edition published by Addison-Wesley Copyright ©1999.
SIMPLIFIED CHINESE language edition published by Publishing House of Electronics Industry. Copyright 1999.

本书中文简体专有翻译出版版权由美国艾维朗公司授予电子工业出版社,并可在全球出版发行。该专有出版版权受法律保护。

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,翻版必究。

图书在版编目(CIP)数据

Visual Basic 6 高级编程策略与范例/(美)吉尔(Gill,T.)著;段来盛等译。—北京:电子工业出版社,2000.5
(软件工程师丛书)

ISBN 7-5053-5934-7

I.V... II.①吉...②段... III.BASIC 语言—程序设计, IV.TP312

中国版本图书馆 CIP 数据核字(2000)第 08269 号

丛 书 名: 软件工程师丛书

书 名: **Visual Basic 6 高级编程策略与范例——错误处理编码与分层技术**

原 书 名: **Visual Basic 6:Error Coding and Layering**

著 者: [美] Tyson Gill

译 者: 段来盛 宋黎松 等

特约编辑: 张月萍

印 刷 者: 北京天竺颖华印刷厂

出版发行: 电子工业出版社 URL:<http://www.phei.com.cn>

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: 787×1092 1/16 印张: 16 字数: 326 千字

版 次: 2000 年 5 月第 1 版 2000 年 5 月第 1 次印刷

印 数: 5000 册

定 价: 28.00 元

书 号: ISBN 7-5053-5934-7/TP·3099

著作权合同登记号 图字: 01-2000-0089

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。

若书店售缺,请与本社发行部联系调换。电话 68279077

出版说明

近年来,随着计算机技术,特别是 Internet 网络的高速发展,计算机应用已经深入到人类社会的各个行业,各个领域,甚至千家万户。信息化社会、网络时代已经离我们不远了。

然而,计算机技术的发展,计算机应用的普及和深入,引发了对计算机软件的要求更广、更精、更加简便,以解决层出不穷的实际应用的问题。这样一来,非但没有解决十余年前专家们曾经预见的软件危机,反而使这一危机愈演愈烈。在世界范围内,软件开发人员的匮乏,高水平的操作系统、开发语言及应用软件的不足,各种补丁程序满天飞,已经成为计算机应用继续发展的一大障碍。

在我国,软件危机也同样存在,解决这一问题是目前刻不容缓的大事。

更值得引起我们重视的是,中国软件业要走自主创新之路。国际几大软件公司雄厚的技术和资金优势,已经垄断了重要的软件领域,并不断地推出新版本。后来者既要学习借鉴他人的技术,又绝不能永远跟着他人走下去。只是热衷于在他人的新版本上做一些表层开发工作是短视的,长此下去会逐渐丧失自主开发软件的能力。因此,在软件的学习和应用上,要学会掌握核心技术,即软件开发的思路和基本方法,并根据实际工作中提出的问题开发有自主知识产权的创新软件。

正是基于这种形势和认识,为发展我国的计算机软件开发事业尽一份出版工作者的责任,我们推出了这套《软件工程师丛书》。

这套丛书是为所有软件工程师和学习软件开发的计算机用户编写的,内容涵盖计算机软件开发的方方面面,其中既有国内作者编著的书籍,也有从国外精选引进的外版书。

为保证丛书的质量,我们选择的作者都是工作在计算机应用第一线,具有丰富软件开发和应用经验的学者、专家和高级工程师,外版书的译者都有多年计算机图书的翻译经验。

我们出版这套丛书的指导思想是帮助软件开发人员提高技术水平,解决他们在软件开发和应用过程中遇到的各种问题。

这套丛书大致可分为四种类型:

一是实例型,如《Visual Basic 6.0 中文版编程实例详解》,通过大量有用的实例说明如何使用某种流行语言开发自己的应用系统。

二是实用型,如《Windows 2000/NT 疑难问题详解》,回答在应用某种操作系统中遇到的各种疑难问题。

三是技术型,如《软件测试自动化技术和实例详解》,介绍有关软件测试技术及其在实践中的应用。

四是手册型，如《Visual C++ 6.0 类库大全》，是软件工程师的必备手册，可从中随时查阅所需的内容。

我们相信这套丛书对软件工程师和学习软件开发、应用的读者会有所帮助，我们希望听到读者宝贵的建议和意见。同时，希望更多的作者和我们联系，出版更多更好的书籍，来充实这套软件工程师丛书。

我们曾经努力，我们正在努力，我们仍将努力。

電子工業出版社

译者的话

有关 Visual Basic 的书籍市场上已经“泛滥”，但是正如本书前言中所指出的，“本书不是又一本‘学习使用 Visual Basic’之类的书籍，它并不是简单地对帮助文件和计算机书籍中能够找到的材料进行重新叙述和重新包装。”

实际上，本书并没有对 Visual Basic 语言进行全面深入的介绍，而是从使用 Visual Basic 开发应用程序时，经常出现的棘手问题或者常常导致开发任务失败的事实入手，逐步推出了一套全新而有效的软件开发策略。

我们常常听到一些使用 Visual Basic 开发应用程序的程序员抱怨 Visual Basic 的不稳定和不可靠，译者本人也有同样的体会，似乎 Visual Basic 成了一种真正的“入门”语言，使用它只能是“玩一玩”而已。果真如此吗？阅读完本书就会知道事实并非如此。对于上述所提出的问题本书给出了成功的解决方案。

本书实际上可以分为两个部分：错误处理编码和分层应用程序开发技术。在第一部分中，作者从软件开发的任务入手，进而对完成任务所存在的各种障碍进行了深入分析，从而对克服这些障碍、实现有效错误处理编码、预防各种可能的错误给出了有效的预防策略。在第二部分中，作者根据自己的实际工作经验和第一部分中的分析，提出了安全编程框架和应用程序分层开发的技术和策略。可以说本书不仅是作者多年来实际工作经验的完整总结，而且通过作者的归纳和扩展使之进而上升成为一种“理论”。

本书所涉及的问题以及解答这些问题所使用的语言是 Visual Basic，但其所提出的思想和策略实际上适用于整个软件开发业。也就是说无论程序员使用的是 Visual Basic、Visual C++、Delphi，还是 Java 等，都可以从本书中找出自己以前编程中所存在的那些根深蒂固的问题的原因所在，并建立起一种严格、有效、全新的编程思维模式。

本书原著语言生动，对问题的分析深入、透彻，确是一本目前图书市场上少见的好书。遗憾的是，由于译者水平有限，译稿中难免存在不妥之处，望读者原谅并指正。

本书的前言及第 1 章到第 6 章由段来盛同志主笔翻译，第 7 章到第 12 章及附录由宋黎松同志主笔翻译。另外，郑城荣、白云、韩珊、舒志勇、张永军、罗为、范庆年等同志也参加了本书的翻译和整理工作，在此表示感谢。

译者

2000 年 2 月于北京

前 言

灵巧编码

本书将帮助读者提高其软件开发工作的质量和效率。读者阅读本书后，将了解到如何更灵巧地编写代码，以及如何在团队中更灵巧地编写代码等内容。对于与 Visual Basic 编程有关的问题和缺陷，以及工程失败的一般原因，读者将会有更正确的评价。对于如何在一个长期及短期工程中避免这些问题，以保证工程的成功，读者也将会有更清楚的认识。

本书的对象

本书是为从事与计算机编程工作的有关人员而编写的。无论是分析员、程序员、管理人员，还是相关的专业人员，都将在本书中找到有价值的内容。无论读者的经验和水平如何，从一名学生到一名高级开发人员，本书的内容都适合于他们。无论读者正在参加的工程是何种类型，均可从本书中得到帮助。

本书所涉及的范围非常广泛，大至一般的哲学问题，小至特定的代码行。虽然本书中所提供的许多一般性方法和观点确实与读者开发所使用的语言无关，但所讨论的特定问题和解决方案则是为 Visual Basic 开发而设计。

本书不是又一本“学习使用 Visual Basic”之类的书籍，它并不是简单地对帮助文件和计算机书籍中能够找到的材料重新叙述和重新包装，而是为了解决实际上困扰着所有 Visual Basic 程序的编码问题，提供了新颖、独特且有价值的高级编程策略。本书并非是一本编程手册，但也讲述了编程技术，它所涉及的是 Visual Basic 编程的过程，而不是语言的细节。

本书不是一本管理手册，但也讨论到了管理方面的问题。本书不是一本关于测试的书籍，但也涉及到了测试问题。简而言之，本书涵盖了对于一个能够成功地进行集体编程的开发团体至关重要的所有问题。它为 Visual Basic 应用程序设立了一个雄心勃勃并且非常出色的崭新标准，为取得效率和质量的新标准提供了简单（回想起来也是显而易见的）技术。

它不是一本讨论特定技术、提供最新 API 或对象模型文档的一般技术性书籍，相反，它充满激情，涉猎到了技术和非技术两方面的问题，有时候还会涉及到家庭问题。它可能揭露读者已经使用多年的方法和技术方面所存在的缺陷。如果确实是这样的话，本书的任务就已经完成了一半。如果它成功地为读者指出了解决这些问题的方法，那么本书就完全取得了成功。

从本书获得最大收益

为了从本书获得最大收益，如同阅读小说一样通读本书非常重要。如果读者习惯于阅读技术性书籍，通过浏览目录或索引从一个主题跳到另一个主题，这种方法对于本书不起作用。概念和实用性思想在本书的论述中都是逐步建立的，每一部分都与前面的内容有关。如同一本小说，它也有一个高潮。在高潮处，读者可能会感到惊奇，面对所有可怕的敌人如何才能改进编码的准则。然而，在结尾处它给出了答案，指出了如何才能采用相对简单的编码方法，解决所有的问题。

另外，笔者推荐至少要阅读两边。在第一边阅读期间，读者可能会遇到令人怀疑的建议。很好，把它们记录下来继续阅读。然后回过头来再阅读一边。在第一边阅读时读者可能反对或怀疑的那些建议，在看了它们是如何地适合于整个工作之后，可能会感到更具魅力。在第二次阅读期间，读者可以有选择地将这些建议结合到自己的工作中，如果必要，则可针对自己的特殊情况进行调整。

致谢

感谢 J. Kevin Meadows、Ed Stegman、Edward Mickelic、Marcie Gill 以及 Bill Munro 复查手稿。也感谢 David Johnson 为图表所做的工作。最诚挚地感谢 WinResources 的全体工作人员所给予的支持和鼓励。最后，感谢 Waterside Productions 以及 Prentice Hall 使本书得以出版。

目 录

第1章 软件开发的任务	1
1.1 草拟任务	2
1.2 保留公司的知识.....	4
1.3 创建过程的标准化.....	5
1.4 错误处理编码.....	6
1.5 灵巧编码	7
1.6 鉴别这种可能性.....	8
1.7 实现这种可能性.....	9
1.8 灵巧编码三角.....	10
1.9 实现任务的障碍.....	11
第2章 理解任务的障碍	13
2.1 Visual Basic 错误处理编码.....	14
2.2 为什么好的错误处理编码很少能实现.....	15
2.2.1 样本代码将注意力放在了功能上	15
2.2.2 错误处理编码不具魅力	15
2.2.3 错误处理编码难以掌握	16
2.2.4 错误处理编码难以实现	16
2.2.5 错误处理编码被看作是辅助性的	16
2.2.6 错误处理编码被认为是不言而喻的	16
2.2.7 错误处理编码需要大量的代码	16
2.2.8 错误处理编码不可能正确地分别实现	16
2.2.9 错误处理编码在最终产品中是不可见的	16
2.2.10 错误处理编码是首先被节省的部分	17
2.2.11 如果确实节省了错误处理编码部分会怎样?	17
2.2.12 即使后来出现了错误, 当它们出现时总是可以被修复	17
2.2.13 管理设立了障碍	17
2.2.14 认为代码是一次性的	17
2.3 “我们将适应”	18
2.4 获得好的错误处理编码.....	19
2.5 错误处理编码的障碍.....	19
2.6 评估错误处理编码.....	20

2.7	代码标准化的障碍.....	21
2.8	代码重用的障碍.....	22
2.9	消除障碍.....	22
第3章	有效错误处理编码的实现.....	23
3.1	提高我们的期望.....	24
3.2	尽早处理错误.....	24
3.3	随时进行错误处理编码.....	26
3.4	预料错误.....	28
3.5	预防错误.....	29
3.6	处理错误.....	30
3.7	捕获错误.....	30
3.8	报告错误.....	31
3.9	避免假设.....	33
3.9.1	我将再也不需要使用这些代码.....	33
3.9.2	我是唯一接触到这些代码的人.....	33
3.9.3	我仅仅为一个特殊的情况而设计它.....	33
3.9.4	一般编码假设.....	33
3.10	为重用设计函数.....	34
3.11	重用错误处理编码.....	34
3.12	系统化错误处理编码.....	35
第4章	显式编码.....	37
4.1	显式变量的用法.....	38
4.1.1	永远使用Option Explicit.....	38
4.1.2	显式地指出变量类型.....	40
4.1.3	避免DefType语句.....	41
4.1.4	使用指定的数据类型.....	42
4.1.5	初始化所有变量.....	42
4.1.6	一行使用一个变量.....	45
4.1.7	使用TypeName、VarType及TypeOf.....	46
4.1.8	使用枚举.....	46
4.2	参数.....	47
4.2.1	永远使用ByVal或ByRef.....	47
4.2.2	显式地指明参数的类型.....	48
4.2.3	为可选参数设置显式缺省值.....	49
4.2.4	验证所有参数.....	50
4.2.5	使用命名的参数.....	51

4.3	数组.....	52
4.3.1	永远不要假设数组的下界.....	52
4.3.2	不要硬编码数组的界.....	53
4.3.3	避免使用Option Base.....	55
4.4	编码建议.....	55
4.4.1	永远包括Else.....	55
4.4.2	避免使用缺省属性.....	57
4.4.3	避免在表达式中混合使用数据类型.....	57
4.4.4	使用常量(Constant).....	58
4.4.5	避免使用操作符优先级.....	58
4.4.6	检查字符串长度.....	59
4.4.7	关闭所有打开的对象.....	61
4.4.8	设置对象为Nothing.....	61
4.4.9	永远显式地关闭错误捕获.....	62
4.4.10	永远不要对外部世界做任何假设.....	64
4.4.11	不要剪切并粘贴.....	64
4.4.12	正确使用+和&.....	65
4.4.13	伪代码.....	65
4.4.14	在运行时设置属性.....	65
4.5	基本原则：显式编码.....	66
第5章	错误处理编码机制.....	67
5.1	错误处理编码并非是一种假设.....	68
5.2	Visual Basic 错误处理.....	69
5.3	没有错误处理.....	69
5.4	错误处理器.....	70
5.5	On Error Resume Next.....	70
5.6	错误抑制.....	71
5.7	On Error Goto.....	72
5.8	恢复程序执行.....	73
5.8.1	Resume.....	73
5.8.2	Resume Next.....	74
5.8.3	Resume line.....	75
5.9	多个错误处理器.....	76
5.10	检查错误.....	76
5.11	检查 Err.Number.....	78
5.12	处理错误.....	80

5.13	清除 Error 对象	80
5.14	关闭错误处理器	82
5.15	错误处理的范围	83
5.16	错误冒泡	87
5.17	错误中的错误	88
5.18	修改错误处理器	90
5.19	错误捕获	91
5.20	错误捕获处理器	92
5.21	内联处理错误	93
5.22	引发错误	94
5.23	错误捕获块与内联错误处理器	94
5.24	何时使用错误捕获块	95
5.25	何时使用内联错误处理	96
5.26	完全避免错误处理	97
第6章	错误预防	99
6.1	错误类型	100
6.1.1	程序错误	100
6.1.2	逻辑错误	102
6.2	预防的种类	102
6.3	预防编码错误	102
6.3.1	长远考虑	103
6.3.2	为他人编写	103
6.3.3	防御性编码	103
6.3.4	攻击性编码	105
6.3.5	避免错误抑制	107
6.3.6	雅洁性预防技术	107
6.3.7	同一个地方不能跌倒两次	107
6.3.8	不要修复同一个错误两次	107
6.3.9	重用	108
6.3.10	标准化	109
6.3.11	包装系统函数	109
6.3.12	不要对预防使用错误捕获	110
6.4	预防用户错误	111
6.4.1	程序设计的三条基本原则	112
6.4.2	使用显式方式	113
6.4.3	精炼所做的设计	113

6.4.4	使用用户界面清晰	114
6.4.5	使消息清晰	115
6.4.6	过滤用户输入	116
6.4.7	确认用户输入	116
6.4.8	使用控件数组	118
6.4.9	选择正确的控件	118
6.4.10	包装控件	118
6.5	窗体预防习惯	119
第7章	安全编码框架	121
7.1	可重用性例程	122
7.2	安全过程	122
7.3	安全函数	125
7.3.1	忽略错误	125
7.3.2	报告错误	126
7.3.3	传回错误	126
7.3.4	返回一个新的错误	127
7.3.5	添加检查跟踪	127
7.3.6	处理错误	128
7.4	安全错误消息	128
7.5	防御函数	129
7.6	防御子例程	131
7.7	安全类	131
7.8	重用 SPF 程序	133
7.9	自完备程序	133
7.10	代码块	133
7.11	命名约定	134
7.12	参数	134
7.13	限定范围	136
7.14	计数器变量	136
7.15	修订版编号	136
7.16	可重用性文档	137
7.16.1	过程头	137
7.16.2	注释头示例	138
7.16.3	版本历史注释	139
7.17	清除	139
7.18	使用 SPF	139

7.19 实现标准.....	140
第8章 SPF实例.....	141
8.1 通用结构.....	142
8.2 安全错误应用.....	144
8.2.1 创建安全错误消息	144
8.2.2 在安全错误消息中统计错误	145
8.2.3 分析安全错误消息	146
8.2.4 报告安全错误消息	148
8.3 数组处理.....	150
8.3.1 获得数组下界	150
8.3.2 获得数组的两个界	151
8.3.3 获得数组计数	152
8.4 类型转换和数据验证.....	153
8.4.1 转换字符串	153
8.4.2 转换日期	154
8.4.3 转换数字	155
8.4.4 验证数字	156
8.5 字符串处理.....	157
8.5.1 安全Len包装程序	157
8.5.2 SSN格式	159
8.6 窗体和控件.....	162
8.6.1 判断窗体是否载入	162
8.6.2 卸载所有窗体	163
8.6.3 设置焦点	164
8.6.4 确定调整大小安全	165
8.6.5 拷贝List控件	166
8.7 数据库例程.....	167
8.7.1 格式化SQL字符串	167
8.7.2 检查游标的位置	169
8.7.3 编辑域	171
8.8 使安全过程正常工作.....	173
第9章 团体策略.....	175
9.1 灵巧编码团队.....	176
9.2 合作的竞争.....	176
9.3 开发自己的标准.....	178
9.4 创建安全过程.....	179

9.5	动机催化.....	179
9.6	认证可重用.....	180
9.7	共享被认证的过程.....	181
9.8	使用超级库.....	181
9.9	奖励永久的贡献.....	181
9.10	通过认证检查代码.....	182
9.11	适应性开发.....	182
9.12	消除非技术的障碍.....	183
9.13	还差一步!.....	184
第10章	编程体系.....	185
10.1	程序失败的方式.....	186
10.1.1	结束失败.....	186
10.1.2	执行失败.....	186
10.1.3	维护失败.....	186
10.2	串珠程序.....	186
10.3	可维护性.....	187
10.4	维护噩梦.....	189
10.5	隐式事务逻辑.....	190
10.6	体系结构的量度.....	190
10.7	叶和层.....	190
10.8	普遍的分层结构.....	192
10.8.1	用户层.....	192
10.8.2	事务层.....	193
10.8.3	数据层.....	193
10.8.4	用户连接层.....	193
10.8.5	数据库层.....	194
10.8.6	数据连接层.....	194
10.9	可重用层.....	194
10.10	分层的流.....	195
10.11	分层与绑定.....	196
10.12	分层与类.....	197
10.13	分层与级.....	197
10.14	层打包.....	197
10.15	配置层.....	198
10.16	分层的优点.....	198
10.17	学习分层实例.....	199

10.18	数据集.....	200
10.19	不受技术变化的影响.....	201
10.20	实现分层的应用程序.....	202
第11章	设计分层的应用程序.....	203
11.1	数据库.....	205
11.2	规划数据层.....	205
11.3	映射控件.....	206
11.4	创建层.....	208
11.5	伪编码用户层.....	208
11.5.1	Form_Load.....	209
11.5.2	Students_Click.....	210
11.5.3	Customers_Click.....	210
11.5.4	编辑域.....	211
11.5.5	保存变化.....	211
11.5.6	添加和删除.....	212
11.6	伪编码事务层.....	213
11.6.1	显示数据集.....	213
11.6.2	显示被计算的域.....	215
11.6.3	改变数据集中的数据.....	217
11.7	数据层的方法.....	218
11.8	用户连接层的方法.....	219
11.9	数据连接层的方法.....	219
11.10	分层应用程序的优点.....	220
11.11	创建一个安全的分层库.....	220
11.12	使用层包装控件.....	221
11.13	实际应用.....	222
第12章	完成任务.....	223
12.1	获得全方位视图.....	224
12.2	评价成功.....	226
12.3	进行下一步.....	226
12.4	继续前进!.....	227
附录A	命名约定.....	229
附录B	SPF检查表.....	233
附录C	认证评价单.....	237

第 1 章

软件开发的任务

- 1.1 草拟任务
- 1.2 保留公司的知识
- 1.3 创建过程的标准化
- 1.4 错误处理编码
- 1.5 灵巧编码
- 1.6 鉴别这种可能性
- 1.7 实现这种可能性
- 1.8 灵巧编码三角
- 1.9 实现任务的障碍