



WILEY

WILEY PROFESSIONAL COMPUTING

C++ 面向对象的 程序开发技术

构造CASE TOOLS

OBJECT-ORIENTED DEVELOPMENT

Building CASE Tools with C++

[美] David E. Brumbaugh 著

柏路 等 译
肖峰 校



電子工業出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

74 V

C++面向对象的程序开发技术

——构造 CASE TOOLS

在佳博
[美] David E. Brumbaugh 著

柏 路 等 译

肖 峰 校



電子工業出版社

内 容 提 要

本书是一本面向对象软件开发技术的实用指南。它既不是现有手册的翻版,也不是深奥的理论性书籍。它给出了面向对象开发领域所遇到的现实问题的实际、可操作的解答。当您拥有本书时,同时拥有了一个开发中使用的CASE工具组,以及在项目实施期间涉及的事例。

全书共分十三章及附录A~D。第一至三章介绍了面向对象技术的基础知识。第四至七章介绍了面向对象的开发原则及所使用的工具。第八章至十三章以实例分别讨论了三个CASE工具的设计和应用过程。附录A~C提供了一些参考资料和来源,附录D给出了全书讨论程序的源代码。

本书适合于想要进一步熟悉面向对象开发的PC软件开发人员,同时也适用于面向对象项目的管理人员、设计人员和编程人员。

Copyright ©1994 by John Wiley & Sons, Inc.

Chinese Version Copyright ©1995 by Publishing House of Electronics Industry.

本书中文版经 John Wiley & Sons 公司授予中国电子工业出版社独家出版,未经出版者许可,不得以任何手段复制或抄袭本书内容。

C++面向对象的程序开发技术

——构造CASE TOOLS

[美]David E. Brumbaugh 著

柏 路 等译

肖 峰 校

责任编辑:贾 蒙 王世忠

*

电子工业出版社出版

北京市海淀区万寿路173信箱(100036)

电子工业出版社发行 各地新华书店经销

北京科技印刷厂印刷

*

开本: 787×1092毫米 1/16 印张: 31.75 字数: 770千字

1996年1月第一版 1996年1月北京第一次印刷

印数: 3000册 定价: 60.00元

ISBN 7-5053-3441-7/TP·1348

著作权合同登记号 图字: 01-1995-442

译者序

本书是一本使用 C++ 语言编写面向对象开发设计工具的书藉。原著作者通过对其在面向对象开发设计方面几年以来工作和经验的总结,深入探讨了整个面向对象程序设计开发过程,并且从一位编程人员的角度,讨论了在面向对象开发这个相当新的领域中将会遇到的各种实际问题,且向读者提供了解决这些问题的可行方案。因而,本书并不是一本语言手册,而是实际问题解答。它把理论和实践看作一个整体。通过书中提供的实例和磁盘文件,向读者提供了自我培训的机会。为了推动我国面向对象开发设计领域的研究和工作,我们翻译了本书,希望广大读者能从中获益。

柏路翻译第一至五章,赵军翻译第六、七章,梁洁翻译第八章,陆珊翻译第九、十章,苏宝文翻译第十一至第十三章,罗成波翻译附录 A 至 C。另外,全书的录入、排版由刘芳、王怡、王立飞、章世嵩等人完成,在此表示感谢。

译者

1995 年 10 月

关于作者

David Brumbaugh 是伊利诺易州 Peoria 城的系统集成人员和高级信息服务的项目经理。他获得了 Elgin 城的 Jusdon 大学的计算机信息科学的学士学位。自从 1985 年以来,他一直专门从事程序开发的工作。并且从 1989 年至今,他已经在不同的专业杂志上发表过五篇文章。也就在 1989 年,他发现 C++ 和面向对象编程并且相互挂接在一起。从 1985 年专门开发程序以来,他活跃于软件开发的各个领域:管理、分析、设计和编程。其项目包括芝加哥的科学和工业博物馆的钱币中心的交互式展览、芝加哥银行的图像识别系统、为主要国际制造商的焊接计划系统以及 Windows 环境下的文档存储和检索系统。

在 1989 年,David 开始使用 C++ 编程。从那时起他经历了 100 多个小时的正规面向对象或 C++ 训练。他服务于 AIS 的面向对象筹划指导委员会。在本书编写之前,他已经使用 C++ 设计和编写了几个 OO 程序分别用于仿真、设计和规则。

David 于 1989 年开始以 ORACLE DBMS 报告的形式在《C 用户》期刊(C User's Journal)上发表技术文章。他已经在《C 用户》(C User's Journal)期刊上发表了三篇其它文章,它们包括如何使用 C 完成 OOP,如何使用类封装移 C 库至 C++,并且在 Zinc C++ 用户界面库上制作一份报告。在 1992 年,《PC 技术》(PC Techniques)发表了他的一篇有关实际软件设计方面的文章,本书是他的第一本专著。

前言

“工业就是准备从介绍和研究阶段进入实际应用,我们必需了解面向对象项目,从概念直至完成。

David Brumbaugh

我认为本书是“第二代”面向对象的书籍。第一代面向对象的书籍介绍“面向对象示例”。大多数示例是关于 OOP(面向对象编程, Object-Oriented Programming)的方面。许多示例由科技研究人员编写。工业正准备从介绍和研究阶段进入实际应用。我们必需了解面向对象项目,从概念直至完成。

对应面向对象编程出现了许多神话。如果用户发现有些神话并不真正是那么神奇的,那么可以试图全面拒绝 OOP。我的目标是在 OO 开发中寻找真正的优点和欠缺。

在本书中,在开发 CASE 工具的同时我们将探讨整个 OO 开发过程。本书的附盘包括了在本书中开发的源代码和可执行程序。

目标读者

本书适合那些想要进一步熟悉面向对象开发的 PC 软件开发人员以及想要更深入地了解本书给出理论的人员,但依据项目学习而非从头至尾地学习更易理解。

本书适用于面向对象软件项目的管理人员、设计人员和编程人员。通常这三个角色一、两个人承担。在其它情况下,这些角色由许多人共享。不论对应何种情况,本书都将对您有所帮助。

面向对象的传播者作出了许多允诺。本书的目标是那些想要实现这些允诺的开发人员。

为了最大限度地从本书获益,用户应该拥有 C++ 编程语言的工作知识。这里使用 Borland C++ 开发工具和源代码例子。用户也应该通晓传统结构分析和结构设计方法。如果曾经使用过传统设计方法,那会更好。

本书的特点是什么?

在讨论本书的内容以前,我应该告诉你它什么都不是。它不是现有手册的翻版。它不是深奥的理论性书籍。它提出了在较新的面向对象开发领域所遇到的现实问题,以及这些问题的实际、可操作的解答。它把理论和实践看做一个整体。当使用这本书工作时,将会拥有一个在开发中使用的 CASE 工具组,以及一个在项目实施期间涉及的事例。

本书解决什么问题?

有关面向对象设计有许多介绍性书籍。一个较为著名的例子是 Grady Booch 的《面向对象设计与应用》。依据 Booch 的说法,转向面向对象设计需要完成三件事情:培训、一个低风险

险初始项目和一组好的例子。他进而建议,软件开发工具是首选项目的最佳候选者。

本书帮助读者平稳地转向面向对象开发,因为它向读者提供了自我训练的机会。它也可以用做课堂学习的课文。它给出头一个项目的情况学习,创建软件开发工具,从头学习到完成,给出丰富的事例。读者按所读到的开发项目,或在自己的项目中使用那些例子。

阅读本书的收益是什么?

面向对象技术不仅仅是软件开发中最新的时尚。它是开发的一个恰当、实际的方法,并且使软件更可靠,使编程人员效率更高,并且降低开发期间变化的影响。可是,它不是一个魔法。开发人员需要理解它,并且要工具使用它。

本书立刻解决了几个问题:提供有价值的 CASE 工具,并且还拥有做为学习实例的 CASE 工具的完整设计。

包括的主题

本书以不同的深浅程度包含了下列主题:

- OO 设计、分析和编程的基础
- OO 开发项目的管理和文档化
- OO 开发所需的 CASE 工具
- 创建某个 CASE 工具

创建 CASE 工具

CASE 是计算机辅助软件工程的缩写。术语 CASE 带入人的大脑中的通常是图表和绘图程序。可是,在软件开发中任何节省时间、提高质量或增加交流的软件就是一个 CASE 工具。“CASE 的一个定义是在软件开发周期中某点处提供力量的工具的使用。它们通常用于分析人员、设计人员、编程人员和管理人员。我就职于一个小公司,在那里这些工作经常交织在一起,我必须同时身兼四职。这不太普遍,因而许多 CASE 工具解决了多种需要。此外,许多 CASE 工具假定用户使用结构化生命期,或某些类似于开发软件的东西。老式结构化技术没有考虑 OOD 中的许多问题,因而我们需要新的工具。既然某些 CASE 工具不能从软件销售商处获取,那么我们将按需要自己创建它。在本书中将讨论这些新 CASE 工具。没必要键入书中提及的代码,本书附盘含有可执行程序。

本书中实际包含四个完整的 CASE 工具。第一个是一个小的 DOS 命令行程序。其它三个需要 Microsoft Windows 3.1 或更高版本才能运行。

1. 在第六章中定义第一个 CASE 工具。这是代码样板生成器。它可以在任何 PC 或 MS-DOS 兼容机上运行。它已经使用 DOS 5 进行测试,并且 3.0 以后的任何 DOS 版本应该能够工作。包括全部源代码,还有一个 Borland MAKE 文件。
2. 第二个 CASE 工具是类库。它记录正在创建或已创建的所有类。它在第 8 至第 10 章中定义。这需要 MS Windows 3.1 或更高版本才能运行。它已在配备 4M RAM 的

386SX 上测试通过。包括所有创建的源代码。它使用了两个在本书中提到的商用库。也存在一个 Borland MAKE 文件。

3. 第三个 CASE 工具是一个类浏览器。它在第十一章中定义。这需要 MS Windows 3.1 或更高版本才能运行。它已在配备 4M RAM 的 386SX 上测试通过。包括所有创建的源代码。它使用了两个在本书中提到的商用库。也存在一个 Borland MAKE 文件。
4. 最后一个 CASE 工具是图形设计程序。清单定义的新实现之一使用了来自《C++ 中实用的数据结构》一书中的某些类,该本由 Bryan Flaming 编著,由 John Wiley & Sons 出版社出版。(读者也可以从该本得到程序代码或修改实现。)在此包含创建的所有源代码。并且使用了两个在本书中提到的商用库。也存在一个 Borland MAKE 文件。

如果对 CASE 工具做了修改,使之在 CompuServe 上有效。将把修改的 CASE 工具放在计算机语言(不久将称做软件开发)论坛的 OOP Alley 库中。(GO CLMFOR 或 SDFORUM)。使用下列关键字“SSOOT”、“BRUM BAUGH”、“CASE”、“OOBOOK”查找上装的文件。

我将查找这个论坛以了解有关本书的信息。除了私人事务,将使这些信息公开化——讨论可以帮助其它读者。有关 SSOOT CASE 改善或错误报告的建议也欢迎大家寄来。

附盘的安装程序也将在读者选择的盘上创建下列目录结构。(这当然假定在全安装情况下)。

```
\OOBOOK
\OOBOOK\CHAP06
\OOBOOK\CHAP08
\OOBOOK\CHAP09
\OOBOOK\CHAP10
\OOBOOK\CHAP11
\OOBOOK\CHAP12
```

每个子目录包含对于子目录名代表各章的所有文件。

CHAP08、CHAP10、CHAP11 和 CHAP12 中的代码在没有一个用户界面库和一个数据库的情况下不能链接成一个完整的程序。为了在不修改代码的情况下链接,必须购买 Zink User Interface Library 和 Pinnacle Relationship Engine。

我已经设计好了类,因而读者应该能够替代自己的用户界面库和数据库,而不会对程序的其余部分产生大影响。(使用图形工具,这样做的必要性就减少了。)

这个操作后面的详细内容、原因和体系在本书的 1.1.4、3.2.4、7.7 和附录 A 中讨论。

本书的组织结构

我试图使本书的每一节都做为一个独立单元存在,同时又保持着连贯的整体性。第一章是关于本书的一个较为详细的介绍。它给出了在本书中所盼求内容的一个完整的构思、引入方法论、OO 开发工具以及软件开发组件。

第二至七章解答诸如基础、分析、设计、编程和管理这类问题。这些并不是“理论”,代之

它们是开发面向对象系统时应具备的视野和经历。

第八至十二章给出某些 CASE 工具的需求、设构和构造。每个工具使用相同的通用策略开发。

- 断定想要做什么
- 决定如何做
- 完成它
- 完善

第十三章包括本书要点的概述以及进一步探索的某些构思。

附录用作有价值的参考工具。它们列举了面向对象开发的信息源和查找面向对象开发工具和组件的地点。附录 D 包含第六章实例和第十章库管理器的源代码。其余的代码和可执行文件在附盘上。附录 D 还包括了本书附带软件的详细安装指令。

存在一个标准索引帮助读者在本书中查找,还有一个源代码索引以便于读者准确地找到需要的类和函数定义。

我试图使本书具有个性并且正规化。有时技术性书籍非常深奥和难以理解,因而我试图使用一种直观、常见的方法完成我的解释。我希望读者喜欢本书,希望它能够帮助读者提高工作效率并且更加喜爱自己的工作。

关于软件

附盘包含的内容？

附盘中包含本书中引用的完整和最新的代码。每章的代码分别放在选择安装盘中自己的目录下。每个目录包含一个 README 文件，该文件包括目录内容的一个综述以及文件的重大修改。

硬件需求

- 386SX 或更好的 CPU
- 2Mb RAM
- 带有 3M 富余空间的硬盘(仅库管理器)
- MS-DOS 5.0 或更高
- 鼠标支持,但不要求

制作一个备份盘

在开始使用原盘以前,我们强烈建议读者制作一份原盘备份。可是,记住,备份盘仅适于你个人使用。备份盘的它人使用都违反版权法。请及时完成备份,使用下列指令。

1. 假定软驱是“A”,在计算机 A 驱中插入 DOS 盘。
2. 在 A:>处,键入 DISKCOPY A:A;并且按 Return 键。DOS 提示在 A 驱中放入要拷贝的盘。
3. 在 A 驱中放入“面向对象开发”盘。

遵循屏幕引导完成拷贝。当完成时,从 A 驱中取出新的拷贝盘并且立刻贴上标签。取出原盘并且存放在安全地点。

安装盘

原盘以压缩格式存放 79 个文件。为了使用这些文件,必须从盘中运行安装程序。遵循下列简单步骤可以在计算机中安装盘:

1. 假定软驱是“A”,在计算机 A 驱中插入“面向对象开发”盘。
2. 在 A:>处键入 INSTALL 并且按 Return 键。

调用安装程序,在出现标题目以后,将出现如图 P-1 所示的选项。

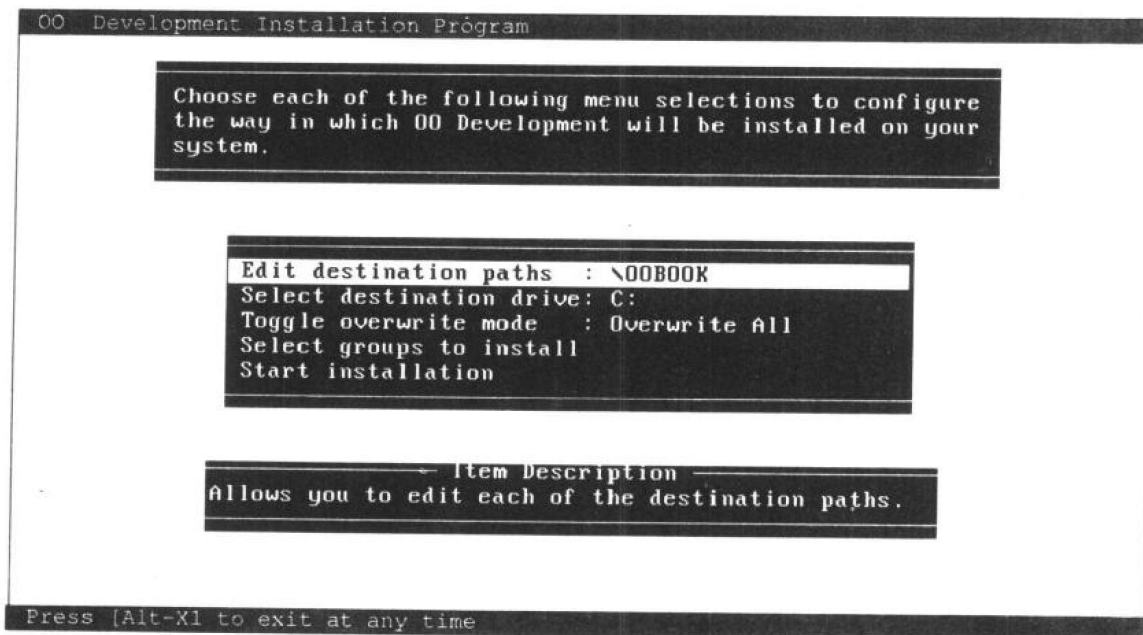


图 P-1 面向对象开发安装程序启动屏幕

为了修改缺省设定,键入突出显示字母或菜单条至所需选项并且按 Enter 键。

3. 为了启动安装,键入“S”或移动菜单条 Start Installation 选项并且按 Enter 键。

在完整安装以后,取出原盘并且把它放在安全的地方。

用户帮助和信息

John Wiley & Sons 出版公司非常高兴向用户提供这个软件包的帮助。读者对这个软件包的使用存有疑问,请打技术支持电话(212)8506194 与我们联系,工作时间为东部标准时间上午 9:00 至下午 4:00。

为了订购其它 Wiley 产品或了解有关信息,请打电话(800)8794539。

有关产品中使用的类库源代码的详细信息和订购信息,请参看用户文档的附录 A。

目 录

关于作者

前 言

关于软件

第一章 简介	(1)
1.1 为什么要用面向对象?	(1)
1.2 关于方法、战略、标记和类似的东西	(1)
1.3 所需面向对象工具	(1)
1.4 开发组件	(2)
1.5 小结	(3)
第二章 面向对象分析、设计和编程基础	(5)
2.1 简介	(5)
2.2 转向面向对象技术	(5)
2.3 面向对象术语	(6)
2.4 面向对象系统开发——对象工程	(7)
2.5 小结	(11)
第三章 面向对象开发的管理和文档	(12)
3.1 简介	(12)
3.2 选择一种方法	(12)
3.3 培训和人员	(19)
3.4 其他费用	(22)
3.5 计划和安排	(24)
3.6 效率	(27)
3.7 文档	(27)
3.8 小结	(28)
第四章 分析:面向对象和传统	(30)
4.1 简介	(30)
4.2 登记卡	(31)
4.3 结构化观点	(36)
4.4 OO 观点	(39)

4.5	使用者透视	(46)
4.6	另一种分析	(47)
4.7	有关分析的实际提示	(47)
4.8	小结	(51)
第五章	更详细地讨论面向对象设计.....	(52)
5.1	在何处设计适应开发	(52)
5.2	设计的目的	(53)
5.3	友好设计的原则	(54)
5.4	一个良好设计方案的基本要素	(57)
5.5	面向对象设计的具体事情	(60)
5.6	一个通用的设计策略	(68)
5.7	总结	(70)
第六章	C++ 中的 OOP	(72)
6.1	针对于 C++ 初学者的几点建议	(72)
6.2	C++ 中对 OOP 的支持	(72)
6.3	类和对象	(75)
6.4	多态性和函数重载	(78)
6.5	继承、抽象和专门化.....	(79)
6.6	C++ 流	(83)
6.7	组合	(85)
6.8	总结	(89)
第七章	概貌——SSOOT CASE	(90)
7.1	引言	(90)
7.2	“参与者”	(91)
7.3	支持的表示符号	(91)
7.4	平台	(92)
7.5	总体需求	(93)
7.6	SSOOT CASE 中的工具	(96)
7.7	软件组件——自己开发还是购买?	(97)
7.8	工作图	(101)
7.9	小结	(104)
第八章	第一个 CASE 工具——类库管理器.....	(105)
8.1	这个工具是谁来使用的?	(105)
8.2	功能性需求	(105)
8.3	系统需求	(109)

8.4	OOA	(110)
8.5	原型回顾	(120)
8.6	总结	(121)
第九章	设计管理器.....	(122)
9.1	设计计划	(122)
9.2	设计问题域	(125)
9.3	数据库类的设计	(130)
9.4	设计用户界面	(143)
9.5	总结	(157)
第十章	库管理器的编码工作.....	(158)
10.1	重复使用现有代码.....	(158)
10.2	新的模型代码.....	(159)
10.3	有关数据库的代码.....	(161)
10.4	用户界面.....	(161)
10.5	其它程序清单.....	(162)
10.6	总结.....	(163)
第十一章	分析和设计类浏览器.....	(164)
11.1	用户使用方案.....	(164)
11.2	有关浏览器的需求.....	(165)
11.3	用户界面模型.....	(166)
11.4	对问题域模型的更改和添加.....	(180)
11.5	对数据库模型的更改.....	(182)
11.6	源代码的指南.....	(182)
11.7	小结.....	(187)
第十二章	SSOOT CASE 图形工具的分析 and 设计.....	(193)
12.1	用户想要从一个图形工具中得到什么?	(191)
12.2	功能需求.....	(191)
12.3	系统需求.....	(192)
12.4	问题域.....	(194)
12.5	界面类.....	(200)
12.6	源代码指南.....	(220)
12.7	小结.....	(221)
第十三章	项目终点.....	(224)
13.1	当接受时我就知道这项工作是危险的.....	(224)

13.2 极其满意的用户.....	(224)
13.3 可以做哪些工作?	(226)
13.4 结束语.....	(226)
附录 A OO 和 C++ 来源	(228)
类库来源.....	(228)
附录 B 商业 OO CASE 工具	(232)
附录 C 推荐书目	(233)
C++ 编程	(233)
面向对象主题.....	(234)
附录 D	(236)

第一章 简介

“理论和实践。不仅是不可分的，它们是同一事物的两个方面：在一个产品的开发中，它们彼此映衬”。

Jeff Duntemam,《PC 技术》，1991,12/1 月

1.1 为什么要用面向对象？

我相信这个时代成功的软件开发者是那些掌握高效秘诀的人。其中一个秘诀就是面向对象，不仅是编程，还有分析和设计。

我们耳中塞满了面向对象编程高效率的承诺。通常，我们听到的提高都来自可重复使用性。还有一些没有被宣扬的来自面向对象开发的高效优点，其中之一就是通过将“现实世界”和程序在语义上的分歧最小化而减少工作量，这种来自减少语义分歧的高效率可以体现在开发中的各个活动中：分析、设计和编程。

面向对象不是一种编程语言。它不是解决编程中所有问题的神奇方法。它甚至不需要一种面向对象编程语言。面向对象是一种方式。对象工程师不是把程序看作是工作在上的一系列任务，而是把程序看作相互协作而又彼此独立的对象的集合。一句话，每个对象就象微型程序，有自己的数据、功能和目的。

1.2 关于方法、战略、标记和类似的东西

每月我要听说大约 100 次“方法”这个词。但我仍然不能就这个词的实质下一个定义，方法包括标记法、战略、开发模型、机理和技术。

因为面向对象是一个新观念，所以还没有一个流行的“面向对象方法”。在面向对象开发的“最佳”标记法、战略、开发模型和生命期方式上，计算机专家们还是有很多种选择。我不想给出“最佳”的或者“最真实的”面向对象方法，我想给出几种选择，从中可以找到更多的信息，并注意其相似的地方而不是差别。所以在本书中，我采用自己最喜欢的标记法，并在开发中用它们讨论战略。

1.3 所需面向对象工具

有很多面向对象设计的入门书，最著名的要数 Gray Booch 的《面向对象设计 and 应用》。根据 Booch 所说，平滑过渡到面向对象设计思想中需要三件事：训练、低风险的第一项目和

一套好的实例,它进一步建议软件开发工具是一个很好的第一项目候选对象。

Booch 还给出了一张在面向对象设计中有用工具的清单,他建议:

1. 一个支持面向对象标记的图形系统
2. 一个类浏览器
3. 一个增量编译器
4. 一个带有类知识的调试器
5. 一个版本控制工具
6. 一个类库管理者

当我开始写本书时,有些 PC 上的这类工具不是找不到,就是太贵,再就是太专业不合适,它们包括:

- 支持面向对象标记的图形系统
- C++ 的类浏览器
- 类库管理者

本书提供了解决这些问题的实例,并还提供了这三种工具的设计和开发。

1.4 开发组件

专业编程不可能凭空产生,对面向对象的一个早期批评就是缺乏软件工程支持。对象工程、面向对象开发的软件工程应用,是由类似于传统组件的组件组成的。

1.4.1 分析

分析,即规化出系统要做什么,是软件开发过程中成功的关键。我在软件开发中的一些失败的经验可以追溯到对系统究竟要干什么缺乏了解。

我对结构化分析的最大抱怨就是用户和同行专家必须要学会计算机上的数据流程图,要找到对系统要做什么的感觉必须要能看懂这些图示。面向对象分析(Object-Oriented Analysis, OOA)技术是使用一些“非计算机人员”能理解的词汇提出问题的。在面向对象分析领域中最著名的工作已由 Peter Coad 和 Ed Yourdon 完成了。

1.4.2 设计

在 1990 年,我去芝加哥,参加了的 Borland 的 OOP 世界巡游。在展示传统软件开发和面向对象开发的区别的同时,演讲人 David Intersimone 说,使用面向对象技术,应该把大约 50% 的时间花在设计上。但他不能给我们更多关于怎样做好设计的指导,他说这门技术还太年轻。

当我开始我的第一个面向对象项目时,我尽量阅读了手边关于面向对象设计(OOD)的资料,我有两个发现:

1. 每个人都基本上在做同一件事。
2. 他们都有表达所做工作的不同方法。