

LEARNING TO PROGRAM with ALICE

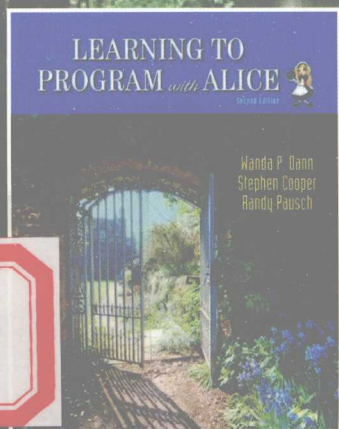
Second Edition

用Alice 学编程

原书第2版

Wanda P. Dann
(美) Stephen Cooper 著
Randy Pausch

付永刚 译



机械工业出版社
China Machine Press

TP311.1-43
D152

LEARNING TO PROGRAM with ALICE

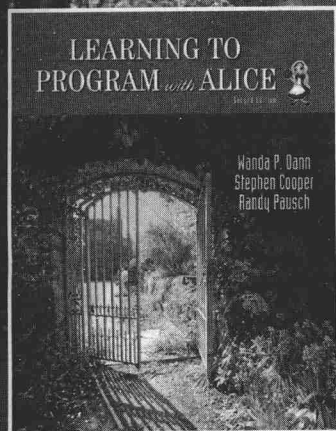
Second Edition

用Alice 学编程

原书第2版

Wanda P. Dann
(美) Stephen Cooper 著
Randy Pausch

付永刚 译

机械工业出版社
China Machine Press

本书是一种全新的面向对象的编程方式的程序设计课程的入门教材。主要内容包括：程序的设计与实现、内置函数与表达式、事件与事件处理、函数和If/Else语句、循环、递归、列表、变量等。在各章的后面都有提示与技巧、小结及练习，让学生在实际操作中掌握编程的技巧。

书中包含大量丰富生动的示例、练习，很容易激发学生的兴趣。本书可作为大学计算机、数字媒体、动漫等专业学生程序设计课程的入门教材，也可供对编程感兴趣的其他读者参考。

Simplified Chinese edition copyright © 2009 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Learning to Program with Alice, 2nd ed* (ISBN 978-0-13-208516-8) by Wanda P. Dann, Stephen Cooper, Randy Pausch, Copyright © 2009.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall.

本书封面贴有Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2009-1339

图书在版编目（CIP）数据

用Alice学编程（原书第2版）/（美）丹（Dann, W. P.）编著；付永刚译. —北京：机械工业出版社，2009.7

（数字媒体专业规划教材）

书名原文：Learning to Program with Alice, Second Edition

ISBN 978-7-111-27462-9

I. 用… II. ① 丹… ② 付… III. 程序设计—高等学校—教材 IV. TP311.1

中国版本图书馆CIP数据核字（2009）第103846号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：李俊竹

北京京师印务有限公司印刷

2009年7月第1版第1次印刷

184mm×260mm · 17.5印张

标准书号：ISBN 978-7-111-27462-9

定价：39.00元

凡购本书，如有缺页、脱页、倒页，由本社发行部调换
本社购书热线：（010）68326294

译者序

2001年，我由于师从戴国忠老师学习人机交互而了解到Randy Pausch——Alice项目的负责人，同时也是CMU娱乐技术中心（ETC）和Stage3实验室的创始人之一，他领导的团队致力于创新的三维交互技术及其应用研究。十余年来，Alice从最初的虚拟场景快速原型工具，逐步演变成为了一个面向程序设计入门课程的教育软件。在美国大概有300多所大学采用这种简单有趣的方法来讲授程序设计中的基本概念，甚至有不少高中也在用它进行初级的逻辑思维训练。在其他国家，越来越多的教育机构也开始使用Alice开设程序设计入门的课程。

我一直觉得将Alice引入到现在的教学体系中是一件很有意义的事情。在成为一名教师后，这种愿望更加强烈。2006年和2007年，我尝试着在多媒体技术基础课程的实验中，简短地介绍Alice。有趣的是，我所执教的班级女生偏多，学生对Alice表现出了极大的兴趣。2008年，我尝试着给全院的大一新生开设程序设计基础的课程，借鉴国外的教学内容和方法，用Alice来讲解概念，而不是某种计算机语言。令人鼓舞的是，在每周2个小时的上课时间内，绝大部分学生都能乐在其中，全神贯注地创作自己的“作品”。接下来的学期，当他们继续学习C和C++语言时，接受程度和学习效果都有很大提升。

从我个人的一点经验来看，把Alice纳入现有信息类专业的课程体系中，最佳的时间是第一学期，因为它既能调动学生的学习积极性，又能避免由于语法和开发环境的原因而影响对核心概念的学习。在讲授方式上，我不太赞同对象早行（object-early）和对象先行（object-first）的观点，而觉得应该采用Y. Daniel Liang教授在《Introduction to Programming Using C++》一书中所用的基础先行（fundamental-first）的方法，即先介绍程序设计的基本概念和方法，如控制语句、循环、函数、数组以及逐步细化的设计方法等，然后再介绍面向对象的设计方法。程序设计的本质实际上是问题求解，不具备基本的问题分析和程序设计能力的学生是不可能学好面向对象的设计方法的。

除了能够纳入现有信息类课程体系中以外，Alice还可作为其他专业的学生普及计算机知识的教学工具，甚至是为中学生开设的一门计算机课程。

关于Alice软件的几点说明

Alice并不是一个面向对象的编程语言，而是用面向对象的语言（Java）开发的交互式虚拟场景创作工具。Alice并不能用来讲解面向对象设计中的核心概念：封装、继承和多态性。Alice并没有实现严格意义上的封装，即没有访问控制；它所实现的继承也很有限，父类的改变并不会影响子类；它更没有实现多态性。即便如此，Alice依然是一个优秀的讲解程序设计和面向对象概念的教学软件，它是一种把程序可视化的方法。Alice甚至直接提供了数组可视化对象来形象地表现数组的操作原理。Alice还可以作为一个动画或者游戏的快速原型工具，基于丰富的模型库，用鼠标拖拽就可以迅速表达设计创意和程序过程。

关于翻译风格

本书保留相当分量的英文术语，经常中英并陈。一方面，对于一些关键的程序设计概念，

如函数、递归、数组、类等词语保留英文可以方便学生转向某种程序设计语言（如C、C++、Java）时不至于感到陌生；另一方面，由于Alice软件目前还没有中文版，其用户界面和代码中皆用英文表述，因此保留英文词语以方便学生查阅。本书还有诸多地方采用这种方式，意在告诉读者，译者深知自己的不足与局限，惟恐造成读者对中译名词的误解和不习惯，所以附上原文。

为保证插图质量，文中绝大部分插图都用Alice 2.0实际操作后截图得到，同时修正了原文中的几个错误。

最后，我要特别感谢我的学生白浩，他完成了本书初稿并验证了每章的练习和工程，为了让译稿更加符合中国人的阅读习惯，他花费了很大精力反复校对和推敲。我还要感谢我去年刚刚执教过的学生：张贞艳、张纷纷、朱枝红和丁伟峰，他们重新制作了全书的插图。

付永刚

2009年6月8日

北京语言大学

序

编程的入门课程一直都是让学生最头痛的，最近又有把面向对象的编程放在第一学期的趋势，这更是加重了学生的学习负担。近期的调查表明，选择计算机专业的学生人数下降了23%，这真是让人担忧。我们不该让学生在刚刚接触编程时就给他们不必要的打击。

Alice系统是面向对象编程教学的一个突破：在Alice中，所有的对象都是直观可见的，它们被具体化为三维的人、动物、家具等。这些对象的状态可以通过调用诸如“向前走1米”或“向左转1/4圈”这样的方法来改变，这些方法对学生来说都是简单易懂的。计算的过程以动画的方式，通过改变对象的状态来表现，在表达概念和计算过程时很难想出比这种方法更自然的表达方式了。Alice的优点之一就是能把抽象的概念用形象的方式呈现给初学编程的人。

老师们都知道，如果学生没有学习动力，任何教学方法和教学技巧都将毫无用处，只有学生有内在动力时才能达到最好的效果。虽然可以采用奖励和惩罚的方式（如分数）来激发学习动力，但Alice使用的却是一种更直接的激励方式：它用讲故事的方式来讲授程序设计，这对所有人都很有吸引力，就像在洛杉矶每个人都会说：“人人都想当导演”。

以三维图形作为创作手段，Alice系统能让那些玩游戏和看皮克斯动画片长大的年轻人很容易接受它，把计算机程序设计比喻成讲故事的过程，是少数几个能让大一新生迅速理解程序设计概念的方式之一。

配合一些高级概念，如直观可见的对象，同时受到Squeak系统中编辑器的启发，Alice设计了一个拖拽式的用户界面，使用它学生可以在屏幕上拖动程序中的组件而不会出现语法错误。

有些人说Alice是近20年来面向计算机入门课程的最有创新性的系统之一，而且它来得正是时候！

Alan Kay

Kay博士是最早研究面向对象编程、个人计算和图形用户界面的先驱之一。他曾获得美国国家工程院颁发的德雷珀奖（与Robert Taylor、Butler Lampson和Charles Thacker一同获奖）、美国计算机协会颁发的图灵奖和日本稻盛基金会颁发的京都奖。

前言

“一本书如果没有插图和对话还有什么用呢？”Alice想到。

本书及配套的Alice系统采用一种创新的方式来介绍程序设计入门课程。过去的30年里，这些课程对学生来说是相当头疼的，但其教学方式却几乎没有创新。我们用这种全新的教学方式的目标就是让传统的编程概念更容易讲解和接受。Alice系统是免费的，可以在网站www.alice.org下载。

程序设计课程应该教些什么

许多人都讨论过这个问题，大部分人都认为学生通过这门课程应该学到以下内容：

- 算法式的思维及表达方式：能够读写一种正式的编程语言。
- 抽象能力：能简单表达复杂的概念，并有逻辑性地分解一个问题的能力。
- 尽善尽美：认识到虽然解决一个问题有多种方式，但肯定有一些是更好的。

我们的教学方式有什么不同

我们的教学方式是让学生创作电影和游戏，通过这种方式让程序中“对象”的概念变为切实可见的三维物体。在Alice中，屏幕上的物体构成了一个微型的三维世界。学生用鼠标在编辑器中拖放程序的组成元素（if/then语句、循环、变量等）来编程，能避免语法错误。Alice是一个功能强大的现代化编程环境，支持方法、函数、变量、参数、递归、数组和事件。我们用这个强大的可视化环境来辅助“对象先行”（object-first）或“对象早行”（object-early）的教学方式（在ACM和IEEE2001年关于计算机专业课表的研究报告中有述）。在Alice中，每一个对象都是一个实实在在可见的物体！我们在本书开始的章节中就会介绍对象的概念。

我们认为，在程序设计入门课程中必须解决以下四个问题：

1) **编程中零散易错的规则，尤其是语法。**Alice的编辑器根除了让学生头疼的语法错误问题，它让学生建立起对语法的直观认识，每当拖拽一个程序元素到编辑器中时，所有有效的匹配位置都会高亮显示。

2) **程序运行时不能看到计算结果。**虽然文本式的调试器和变量监视器比没有它们要好一点，但在Alice中，程序中的状态在本质上就都是可见的。从某种意义上说，我们是把学生的心理负担从认知系统转移到了感知系统。对学生来说看到一个物体向后运动而不是向前运动，比观察一个“sum”变量是减少而不是增加要容易得多。Alice让学生可以看到动画程序怎样运行，这样就能在程序结构与表现的动画之间建立一种简单易懂的关系。现在的学生的生活中充满了交互式的三维图形，我们希望直接利用这一点，而不是强加给他们这种学习方式。

3) **缺乏编程动力。**多数学生学习程序设计的入门课程仅仅因为学校的要求。没有什么能比一个好老师更能激发学生的动力，但一个好的学习环境的确会有很大帮助。对使用Alice的学生的一项调查研究表明，相比那些用传统方式学习的学生而言，使用Alice的学生会去做更

多的练习，也倾向于去学习第2门程序设计课程。关于Alice的早期版本我们收到最多的意见就是希望能分享他们的成果，因此我们已经添加了在网页内运行Alice的新功能，以便学生把程序发布到他们自己的网页上。在使用Alice之后所有学生的编程动力都有所提高，尤其是在女同学身上我们看到了非常令人鼓舞的结果。

4) **理解复合逻辑与学习设计技巧的困难。** Alice环境当然鼓励你去创建小规模的方法和函数。更重要的是，Alice把程序设计类比为电影创作的过程，这样，我们可以借鉴其中故事板的概念，而学生们可以把它理解为一个设计好的电影制作过程。我们用简单的草图和屏幕截图来说明程序设计技巧，而建议使用文本式的故事板，逐步细化最终设计出程序的伪代码。

如何使用本书

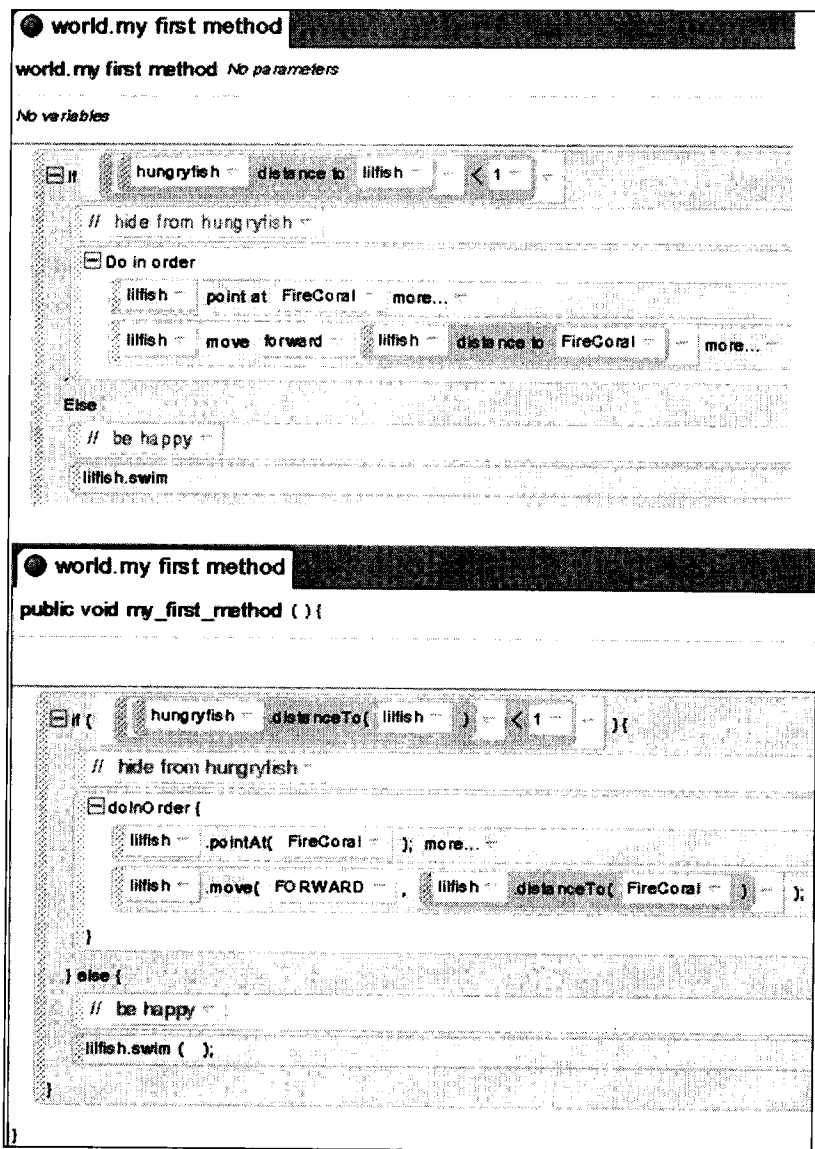
当然，使用本书的老师可以选择最适合自己的方式来授课。下面列举了我们能想到的四种方式：

作为短期或长期程序设计课程唯一的教科书。 这种方式的结果是学生在期末时能够写出相当复杂的程序（如300行）。这门课程可能适合那些非计算机专业的学生，只是想学习程序设计而不需要学习一种具体的程序语言。它也可以作为准备选择计算机专业但之前缺乏编程经验的学生的预备课程。在一个国家自然科学基金项目（NSF-0126833）研究中我们发现，那些没有或有很少编程经验的人直接去学习第一门程序设计课程（如CS1）时都会有相当的受挫感，而以Alice作为第一门程序设计课程的预备课程（pre-CS1）时，这种情形就有很大改善。

作为传统的“程序设计入门”课程的第一部分，例如CS1。Seymour Papert开发的Logo语言和Rich Pattis开发的Karel the Robot系统都采用了这种教学方式，它们给了我们很大启发。不同的是，Alice的功能很强大，这使它可以作为几个学期的学习工具（如Carnegie Mellon大学计算机专业的高年级学生通常会用Alice写出3 000行左右的程序）。然而，许多程序设计的入门课程不仅要介绍程序设计的概念，同时还要学生用传统的编程语言如Java写程序。而先学习Alice，学生就可以先熟悉程序设计中的基本概念，然后当过渡到一门具体程序语言中详细的语法规则时，他们就能很快掌握。Alice中也可以用类似Java语法的方式显示代码，如图P-1-1所示。

作为“计算机文化”课程中程序设计的部分。在许多学校，计算机文化课程都是向非计算机专业的学生很宽泛地介绍关于计算机或信息技术的内容。大部分这类课程都去掉了程序设计的部分，更像是去学习电子表格和字处理等办公自动化软件的实验课。Alice作为计算机文化课程中程序设计的部分有较大潜质。

作为高中“程序设计入门”课程的教材。在高中，有趣的能激发学生学习动力的环境是老师最好的朋友，在这里Alice有更大的发展空间。这本书可以作为某门课的一部分，或者是为高考而作计算机类课程的预备学习。



图P-1-1 Alice中的代码（上）也可以用Java语法的方式显示（下）

教辅材料

读者可以登录华章网站（www.hzbook.com）下载Alice的安装包和模型库。使用本书授课的教师还可以下载书中的例子文件、不同授课时长的课表安排及每周的讲义（ppt）、考卷范例及答案和书中的练习、工程及答案。

本书结构

第1章给出了几个学习计算机编程的原因，消除学生在学习编程时可能产生的畏惧心理（这对那些计算机文化课程尤其有帮助），然后介绍了Alice中的基本概念。传统的纸式教程见附录A。关于计算和编程的术语（如程序、类、对象）在第一次出现和定义时都用楷体字。

在接下来的每章，开始都是主题概述，每章结尾会有练习题、工程和小结。下面按章节顺序简要列出本书所包含的一些主要概念。很明显，本书的重点就是介绍程序设计中的基本概念。

第1章 入门

第2章 设计、逐步细化和第一个程序

第3章 内置函数、表达式和简单的控制语句

第4章 面向对象编程：方法、参数和继承

第5章 交互式程序：事件和事件处理

第6章 函数和条件执行（If/Else）

第7章 确定循环（Loop）和不确定循环（While）

第8章 递归

第9章 列表

第10章 变量、继承的再探讨和可视化数组

第11章 Alice中基本概念的总结

我们建议以较快的速度讲解本书的前3章或前4章（第1、2章的内容用两三次课就能讲完）。老师可以从每章结尾的大量练习中选择一部分作为作业，提供大量的练习题是为了方便老师从中选择最合适自己学生的部分。在后面的几章结尾还有几个较大的“工程”（一个工程是一个比较复杂的，需要更多时间的练习题）。

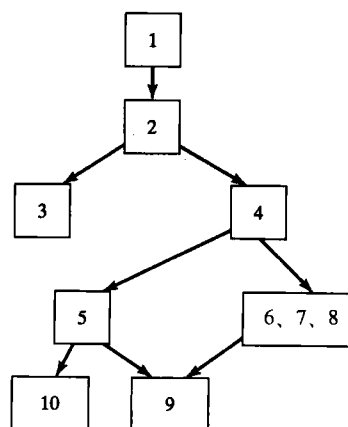
工程是为了激发学生的创造性，而不是为了给学生出难题。可行的话，我们推荐开放式的工程，就是让学生自己去设计故事情节和选择对象来创作一个动画。当然，这种工程也要满足一定的要求，例如“一个交互式的虚拟场景，可以用鼠标进行两个以上的交互操作，至少有三个方法，要用到一个判断语句，要通过添加类级方法定义两个新类并在场景中有这种类的对象。”在项目时间截止前，可以让学生用Alice把他们做的结果向班级其他学生展示。

每章都有一小节叫做“提示与技巧”。如果加在一起的话，这些小节和附录A、B就是一个简易的Alice用户手册。提示与技巧中介绍的是Alice中关于动画制作的内容，而不是像在章节中着重介绍的那些传统的程序设计概念。我们有规划地把这些技巧分布在全书中，为在后续的例子中使用它们打下基础。提示与技巧为那些想通过Alice学习更多关于动画知识的人提供指导。附录A是一个入门教程，附录B介绍如何使用Alice的用户界面，这些关于界面操作的例子在本书的例子中遇到时也会介绍。提示与技巧中还有选择地介绍了一些“怎样做”的主题来增加本书的趣味性。

内容筛选和授课顺序

内容筛选和授课顺序完全由老师决定！图P-1-2描述了各章节间的依赖关系，可以辅助老师选择授课顺序。

简单来说，第1~4章是有先后顺序的，而第5~11章则相对较独立，因此可以按照很多不同的顺序讲授。从第1章顺序讲到第4章是“对象早行”的教学方法，而如果跳过第3章则是“对象先行”的教学方法。第4章和第5章中的例子和第3章没有关系，因此“对象早行”的方法可以放心地跳过这一章，等到后面讲第6、7章的时候再来介绍这些内容。另



图P-1-2 依赖关系图

一种灵活的授课顺序是颠倒第7、8章的顺序，先介绍递归再介绍Loop和While控制结构。

在与不同大学和学院的的老师沟通后，我们发现内容筛选和授课顺序往往受到某个特定的课程结构、教学法和教学观的限制。如果本书只占一门综合课程中三到四周的时间，你也许只会布置练习题，而只在结束时布置一个工程作业。而且，你可以把提示与技巧中的内容布置为课下的阅读作业，这样不占用课堂时间。下面的顺序表列出了几个授课顺序。

顺序	教学方法/时间跨度	课程
1-2-3-4（也可以接着讲第5章）	对象早行（3~4周）	计算机文化 (CS1的前3周)
1-2-3-4-5-6-7-8	对象先行（短期课程）	程序设计基础 Pre-CS1
1-2-3-4-5-6-8-7	对象先行，先讲递归 (短期课程)	Pre-CS1
1-2-3-4-5 加上6-11章中挑选部分	对象早行（整个学期）	程序设计基础 Pre-CS1
1-2-3-4-5-6 加上7-11章中挑选部分	对象先行（整个学期）	程序设计基础 Pre-CS1

本书中某些需要注意的细节

如果你用本书讲授或者学习Alice时不关心程序设计的内容，那么可以跳过第2.1节。然而，文本式的故事板和逐步细化的方法将贯穿书中剩下的章节，以提供一种从算法和问题求解的角度讨论程序设计的框架。应很多讲授逻辑编程的老师的要求，本书第2版第1章中增加了对流程图符号的简要介绍，同时在1~3章中增加了几个流程图。这些流程图比较简单，主要是为了解释其对应的例子程序的内部逻辑。我们把在这个基础上设计复杂例子流程图的工作交给老师去做。你也许会选择不同的设计方法（也许是统一建模语言或更传统的伪代码形式），这种做法和本书的内容也不会有冲突。

在第4章，我们会提到Alice并不是完全意义上的继承。当在Alice中创建一个新类时，它会继承基类的属性和方法，并存储在新的三维模型文件中。然而，之后对基类的修改对这个新类是没有影响的。

在面向对象的程序设计语言中，继承是通过两种机制实现的：(a)添加方法（行为），(b)通过添加可变变量添加额外的状态信息。本书分开介绍这两种机制，在第4章中介绍添加行为，而直到第10章才会介绍添加状态信息。之所以这样安排，是由于可变变量在Alice的环境中不是直观可见的，因此在学生已经掌握了程序设计中的其他概念后才介绍。

我们发现对学生来说交互式程序更有趣味，也更能激发他们的学习兴趣。然而，从教学法的角度看，第5章也可以略去。虽然后面练习和工程中有些是交互式的，但是老师可以有选择地布置作业。

Alice

Alice的最新版本和三维模型库都可以在网站www.alice.org上下载。本书的教辅材料中提供两个Alice版本，PC版和苹果机版，而且都包含2.0版和2.2版（测试版）。如果你的PC使用的是Linux系统，可以在www.alice.org上找到相应的版本。Alice网站上还提供了安装指南、常见问题、提交bug的链接和面向Alice教学者的在线论坛。

当本书出版时，Alice 2.2仍在进行广泛的测试工作。本书包含这个软件是面向个人使用的，而不是为了在网络上传播，请访问www.alice.org免费下载更新版本。

Alice是一个对内存要求很高的三维图形应用。Alice开发团队对运行它有一些最低的要求和推荐配置。请注意，许多老式的笔记本电脑是不满足这些要求的。为了确保Alice能正常运行，一定要在你要使用的机器上进行测试。

操作系统要求：

PC	
Alice 2.0	Windows ME、2000、XP或Vista
Alice 2.2	Windows XP或Vista
Mac	
Alice 2.0	OS X 10.3、10.4或10.5
Alice 2.2	至少OS X 10.4

最低硬件配置：

处理器频率500MHz或以上
能显示16位颜色的显卡
256MB内存
显示器分辨率1024 × 768
声卡

建议硬件配置：

处理器频率1.0GHz 或以上
16MB显存的3D显卡（详细资料见www.alice.org）
500MB内存

Alice在课堂演示时输出到投影仪运行得很好，800 × 600分辨率的投影仪能正常运行，但1024 × 768的分辨率下效果是最好的。

本书配套教辅材料中包含Alice 2.2。由于广大师生的强烈要求，我们也希望该版本能够早日发布。Alice 2.2修正了2.0中的几个错误，更为重要的是，Alice 2.2增加了视频输出功能（在File菜单中）。此外，大部分的三维人物模型都实现了一个“行走”的方法，在模型库People文件夹中有一个“people who know how to walk”的文件夹，其中包含有这种模型。

第2版修订说明

1. 修正了第1版的一些错误。
2. 交换和扩展了第3章和第8章“提示与技巧”中的内容。
3. 扩展了附录B中“创建你自己的人物模型”的部分，并替换到第5章中“提示与技巧”的部分。
4. 将每章的小结部分放到了练习与和工程之前。
5. 在每章中添加了关于术语和概念的问答题。
6. 在第1章中增加了对流程图符号的简要介绍，在第1~3章的几个例子中添加了流程图。
7. 在附录B中添加了一小节，介绍如何将一个程序导出为QuickTime格式的文件。

Wanda P. Dann
Stephen Cooper
Randy Pausch

致 谢

正如之前提到的，Seymour Paper设计的Logo语言和Rich Pattis的Karel the Robot系统在利用直观可见的微型世界方面起了极大的推动作用。Alan Kay和Squeak小组激发了我们创作一个基于鼠标操作的程序编辑器的灵感，同时我们也受到了Tim Teitelbaum语法导向的编辑器的启发。我们还要感谢George Polya、Mike Clancy以及Doug Cooper在问题求解方法上给予我们的帮助。

我们由衷地感谢对本书和教辅材料进行测试的人员和用户，他们提出了很多宝贵意见和建议：Susan Rodger (Duke大学)、Rick Zaccane (Bucknell大学)、Bill Taffe (Plymouth州立大学)、Angela Shifflet (Wofford 学院) 以及William Taylor (Camden County 学院)。另外，我们还要感谢这些学生的帮助：Toby Dragon (Ithaca学院)、Kevin Dierzler (Saint Joseph's 大学)、Patricia Hasson (Saint Joseph's 大学) 以及Kathleen Ryan (Saint Joseph's 大学)。我们还要由衷感谢来自Calhoun 社区大学的Sue Mitchell，她为第2版中添加流程图和实现策略图提供了例子和咨询。

Alice软件的诞生要归功于Carnegie Melon 大学一群充满创造力、精力充沛、有奉献精神的研究生、本科生及员工。没有他们，就不会有Alice，也就不会有这本书。Alice 2.2的主要作者包括：Ben Buchwald、Dennis Cosgrove、Dave Culyba、Cliff Forlines、Jason Pratt以及Caitlin Kelleher，详细名单见www.alice.org。Carnegie Melon大学的艺术家们都慷慨地为模型库提供作品。做出贡献的还有数百人，我们列出其中几位谨表感谢：Sarah Hatton、Mo Mahler、Shawn Lawson以及Tiffany Pomarico。Tommy Burnette、Kevin Christiansen、Rob Deline、Matt Conway和Rich Gossweiler对Virginia大学开发的Alice的早期版本做出了原创性的贡献，在此我们要感谢Virginia大学对Alice早期开发的支持和鼓励。

我们非常感谢Carnegie Melon大学的Madeleine Pitsch，他完成了Alice 2.2版本的更新工作。我们也很感谢Dennis Cosgrove，他为我们提供了很多帮助和指导，还有Georgia 理工学院的Barbara Ericson，她无私地为视频导出功能的实现提供了很多代码和建议。还要真心感谢Jennifer Mung'Au，她为修改后的人物模型库中的大部分人物模型实现了“行走”方法。

本书的内容基于国家自然科学基金的资助，这些基金编号是：0302542、0339734、01268833、0624528、0623808、0624479、0624642和0624654。本书中的任何观点、发现、结论或建议仅代表作者立场，与自然科学基金会无关。

我们要感谢本书第1版的评阅人：De Anza 学院的Mary Ann Amy-Pumphrey、San Diego州立大学的Leland Bech、Colorado州立大学的Elizabeth Boese、Wisconsin 州立大学的Deb Deppeler、Utah州立大学的Kendra Dinerstein、Harvard学院的John Dougherty、Virginia理工的Odis Hayden Griffin, Jr.、Georgia理工学院的Mark Guzdial、Carnegie Mellon大学的 Richard Pattis、Winsconsin大学的Sally Peterson、Duke大学的Susan Rodger、Camden County学院的William Taylor、Arizona大学的Suzanne Westbrook、Bucknell大学的Rich Zaccane。

我们还要感谢为本书第2版提出宝贵意见和建议的评阅人：Western Virginia社区学院的Gilbert Armour、North Carolina Wilmington大学的Marni Ferner、Pacific Lutheran大学的

Michelle Folsom、Rowan 大学的Stephen J.Hartley、Indiana大学的Christopher Haynes、Northern Virginia社区学院的Barbara Holt、Kalamazoo学院的Kelly Schultz、Charleston学院的Roxanne H. Stalvey、Lehman学院的Katherine St. John。

我们要感谢Alan Apt和Prentice Hall出版社的Tracy Dunkelberger为本书的发行做出的贡献。在过去的十年中，Electronic Arts、Google、DARPA、NASA、Apple、Ford、Intel、Microsoft Research和SAIC都对Alice系统的开发提供了帮助，对此我们深表感谢。

关于作者

Wanda P. Dann是Carnegie Mellon 大学的副教授和Alice项目的负责人。她的研究领域是：程序及程序设计语言的可视化、程序设计入门课程的创新性方法。

Stephen Cooper是Saint Joseph大学数学和计算机科学系的副教授。由于在Alice项目上的工作，他获得了该校历史上最大的国家自然科学基金资助项目。

Randy Pausch是Carnegie Mellon 大学计算机科学、人机交互和设计系的教授，同时也是Carnegie Mellon 大学娱乐技术中心的创始人之一，并且是Google、Walt Disney Imagineering和Xerox PARC的顾问。

目 录

译者序
序
前言
致谢

第一部分 Alice简介

第1章 Alice入门	2
1.1 Alice简介	2
1.2 Alice中的概念	6
提示与技巧1 特效：文本与二维图像	12
小结	14
练习	15
第2章 程序的设计与实现	17
2.1 剧情与故事板	17
2.2 第一个程序	22
提示与技巧2 方位指令与运动指令	33
小结	41
练习	43
第3章 编程：把代码片段集中到一起	45
3.1 内置函数与表达式	45
3.2 简单的控制结构	49
提示与技巧3 相机和动画的控制	56
小结	59
练习	60

第二部分 面向对象和事件驱动编程的概念

第4章 类、对象、方法和参数	64
4.1 全局方法	66
4.2 参数	73
4.3 类级方法与继承	81
提示与技巧4 可见和不可见对象	91
小结	96
练习和工程	97
第5章 交互：事件和事件处理	106
5.1 交互式编程	106
5.2 参数和事件处理方法	111
提示与技巧5 创建你自己的人物模型	118
小结	120
练习和工程	121

第三部分 使用函数和控制语句

第6章 函数和If/Else语句	130
6.1 函数	130
6.2 用If/Else及布尔函数控制程序执行	136
提示与技巧6 随机数和随机动作	146
小结	148
练习和工程	150
第7章 确定循环和条件循环	159
7.1 Loop	159
7.2 While——条件循环	164
提示与技巧7 事件和循环	168
小结	170
练习和工程	171
第8章 递归	177
8.1 递归简介	177
8.2 另一种形式的递归	181
提示与技巧8 工程学角度的观感	190
小结	192
练习和工程	193

第四部分 高级主题

第9章 列表和列表处理	200
9.1 列表	200
9.2 列表搜索	205
提示与技巧9 姿势	208
小结	209
练习和工程	210
第10章 变量和继承的再探讨	214
10.1 变量	214
10.2 在可视化数组对象中使用索引变量	223
提示与技巧10 使用监视和文本输出进行调试	229
小结	232
练习和工程	233
第11章 接下来学什么	237
附录	241
附录A：使用Alice	241
附录B：Alice界面操作	257

第一部分

Alice简介



第1章 Alice入门

第2章 程序的设计与实现

第3章 编程：把代码片段集中到一起

第1章

Alice入门



"Let's pretend there's a way of getting through into it, somehow, Kitty. Let's pretend the glass has got all soft like gauze, so that we can get through. Why, it's turning into a sort of mist now, I declare! It'll be easy enough to get through—"

1.1 Alice简介

为什么要学习计算机编程

也许你学习本书是因为你想学习计算机编程或者你选修的一门课程要求你学习计算机编程。无论哪种情况，我们先来看看为什么学习编程对你可能还是很有价值的。

首先，我们要澄清的是：**学习编程不会把你变成一个电脑谜**。我们知道人们对于程序和写程序的那些人有很多固有的看法。我们保证你不会因为学习这个而突然变得像那些书呆子一样，口袋里装个防水塑料袋^①，不经常洗澡或者满口都是晦涩难懂的计算机术语。实际上，本书使用的Alice系统采取的是一种全新的编程方式，你将不再需要对着计算机输入难以理解的计算机语言，指望它能做出某种奇怪的运算；相反，你会成为一出戏的导演，看着屏幕上的对象按你的剧本进行演出！不过让我们先别讲得那么远，还是回到为什么要学习计算机编程的主题上。

我们可以找到许多理由来学习编程。对有些人而言，编程可以带来极大的乐趣——他们享受的就是编程本身。但对更多的人而言，编程是一种解决问题的有效方式，这些人有重要的事要做，而计算机是一个好帮手。事实上，计算机已经渗透到社会的方方面面，如果你出生在未来的话，那么你的一生将有可能自始至终都与计算机相关。在你出生那天，医院会在你的脚踝上绑上小小的计算机芯片，以确保他们时时刻刻都知道你在哪儿。在你临终那天，可能有一台计算机陪伴你，监测你的生命特征。在你的一生中，你会活得更久也会更健康，因为计算机科学在辅助医学研究方面的发展，汽车里有计算机控制的刹车片和安全气囊，计算机建模还能帮助我们设计对付各种疾病（比如AIDS）的药物。计算机程序员使所有这些科技进步都成为可能。

计算机和程序员编写的软件给娱乐业带来了巨大的变革，计算机游戏越来越流行。互联网调查机构Pew Internet & American Life Project在2003年做出的调查显示，有70%的大学生一星期至少玩一次在线游戏。《星球大战》(Star Wars)系列电影以及其中的所有特效都是因为计算机的存在才成为可能。顺便说一句，Alice系统开发成员中的一个本科生，在毕业之后去了ILM

① 古代的书呆子在衣兜里放防水塑料袋，以防止笔漏水把衬衫弄脏。——译者注