HZ BOOKS
华章教育

计 算 机 科 学 丛 书

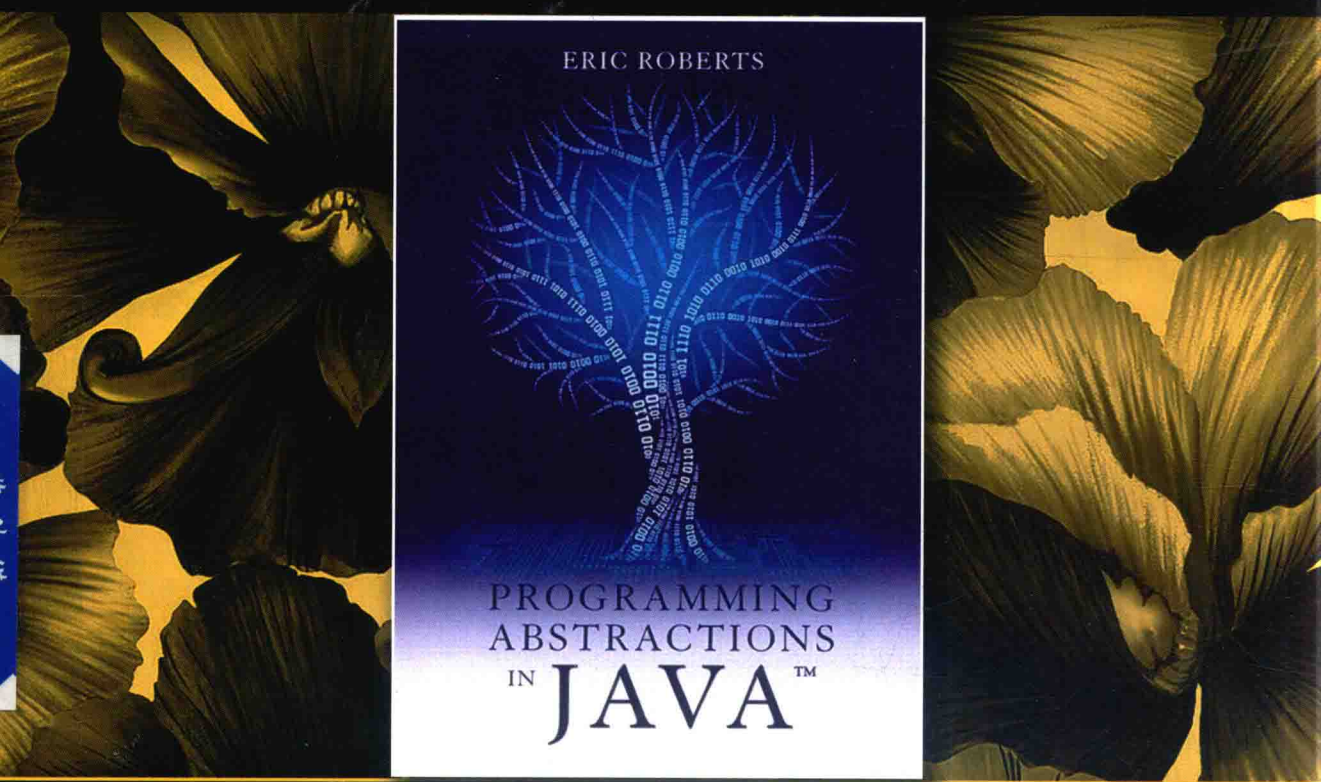
Pearson

Java程序设计

基础、编程抽象与算法策略

[美] 埃里克 S. 罗伯茨（Eric S. Roberts） 著
斯坦福大学
陈昊鹏 译
上海交通大学

Programming Abstractions in Java



机械工业出版社
China Machine Press

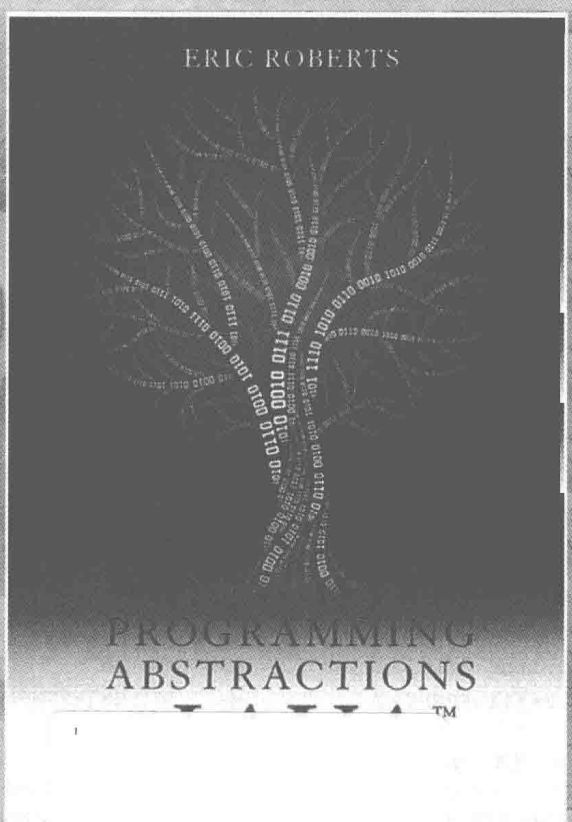
计 算 机 科 学 丛 书

Java程序设计

基础、编程抽象与算法策略

[美] 埃里克 S. 罗伯茨（Eric S. Roberts） 著
斯坦福大学
陈昊鹏 译
上海交通大学

Programming Abstractions in Java



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Java 程序设计：基础、编程抽象与算法策略 /（美）埃里克 S. 罗伯茨（Eric S. Roberts）著；陈昊鹏译 . —北京：机械工业出版社，2017.8
（计算机科学丛书）
书名原文：Programming Abstractions in Java

ISBN 978-7-111-57827-7

I. J… II. ① 埃… ② 陈… III. JAVA 语言—程序设计 IV. TP312.8

中国版本图书馆 CIP 数据核字（2017）第 202881 号

本书版权登记号：图字：01-2016-3033

本书介绍如何使用 Java 语言编写程序，旨在通过介绍编程过程中遇到的难点和问题来拓宽读者的视野。本书结合具体的示例代码，由浅入深介绍解决编程问题的策略和方法，有助于读者快速入门 Java 语言编程。同时，每章后面都有配套的复习题和习题，便于读者理论联系实践，通过编程实践查漏补缺，温故而知新。

本书适合作为计算机专业的教材，也适合希望学习 Java 语言的各个层次的读者阅读。

出版发行：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：朱秀英　　责任校对：李秋荣
印　　刷：北京市荣盛彩色印刷有限公司　　版　　次：2017 年 9 月第 1 版第 1 次印刷
开　　本：185mm × 260mm　1/16　　印　　张：35
书　　号：ISBN 978-7-111-57827-7　　定　　价：99.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自 1998 年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum，Bjarne Stroustrup，Brian W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com
电子邮件：hzjsj@hzbook.com
联系电话：（010）88379604
联系地址：北京市西城区百万庄南街 1 号
邮政编码：100037

华章科技图书出版中心

译者序

Programming Abstractions in Java

本书针对的是Java程序设计的入门者。与大多数有关Java编程语言的教材不同，本书没有针对Java语言的各种特性展开各个章节，而是从程序抽象的角度，围绕着编程思想来展开其内容。在针对具体问题的解决过程中，水到渠成地揭示Java语言的各种特性，使读者不仅知其然，而且知其所以然。哪怕你已经是熟练的程序员了，阅读本书你仍然会有一种耳目一新的感觉，你会发现从另一种维度来对你掌握的知识进行梳理时所呈现出来的不一样的视界。

本书介绍的程序抽象包含了常用的数据结构和常见的简单算法，因此，它不但可以作为Java语言的学习材料，还可以作为数据结构的参考教材。事实上，这本书就是斯坦福大学的第二门编程课程的教材，有兴趣的读者可以在该课程的网站上下载更多的参考材料，以拓展对本书内容的理解。

虽然我从事计算机类书籍的翻译工作已经十多年了，对翻译工作越来越得心应手，但是面对本书这样经典的教材，还是有很大的压力。在翻译过程中，我努力地将原文的意思按照中文的习惯进行表达，使本书读起来更加通畅。但是，就像人们常说的："我才刚刚上路，而且永远在路上。"因此，书中难免会存在缺陷和错误，请广大读者见谅，并积极反馈意见，我将在后续印刷的版次中不断地纠正错误和弥补缺陷。

感谢机械出版社华章公司的各位编辑，她们对译稿进行了仔细校对，提出了宝贵的修改意见，是她们的辛勤工作确保了本书得以顺利出版。

陈昊鹏

致学生

在过去的 10 年中，计算领域的发展激动人心。人们日常随身携带的各种网络设备变得速度更快、价格更便宜、能力更强。利用像 Google 和 Wikipedia 这些基于网络的服务，人们滑动指尖就可以获得世界上众多的信息。社交网络将全世界的人联系到了一起。流技术和更快的硬件使得人们可以随时随地下载音乐和视频。

但是，这些技术不会凭空出现，人们需要构建它们。幸运的是，至少对那些研究这个令人激动且变化万千的领域的人来说，具备必需的软件开发技能的人供不应求。这里是硅谷的高科技经济中心，能够将各大公司的技术愿景转化为现实的天资聪慧的工程师十分短缺。各大公司甚至不敢奢求找到更多懂开发和维护大型系统的软件开发人员——他们需要理解诸如数据表示、效率、安全性、正确性和模块化等问题。

尽管本书并不会教给你了解这些主题以及更广阔的计算机科学领域所需的所有知识，但是它会给你一个良好的开端。在斯坦福大学，每年有超过 1200 名学生选修教授本书内容的课程。其中许多学生的知识背景仅限于本书，但是他们都找到了暑期实习或在业界工作的岗位。更多的学生会继续选修更高级的课程，以便为把握这个快速发展的领域中的无限机会做好准备。

除了为从业提供机会，本书中的主题还充满了智力上的刺激。你在本书中学到的算法和策略，有些是在过去 10 年中发明的，而有些则已经有超过 2000 年的历史了。它们难以置信地灵巧，就像是一座座矗立着的人类创造力的丰碑。它们还非常实用，可以帮助你变成经验丰富的程序员。

在阅读本书时，请记住，编程永远都是实践出真知。读过有关某种算法技术的内容并不表示你就能够将其应用到实践中，真正的学习是在完成练习和调试为了解决这些问题而编写的程序时才开始的。尽管编程时不时会让你感到挫败，但是在发现最后一个 bug 并看到程序可以工作时的激动心情是无与伦比的，它让你可以将一路走来碰到的所有困难都抛之脑后。

致教师

本书旨在作为一般大学或学院的第二门编程课程的教材。它涵盖了传统的 CS2 课程的内容，CS2 是在美国计算机学会（ACM）制定的 Curriculum'78 中定义的课程。因此，它包含了 ACM/IEEE-CS 联合计算课程设置 2001（Joint ACM/IEEE-CS Computing Curricula 2001）定义的 $CS102_0$ 和 $CS103_0$ 课程中规定的大部分主题，以及计算机科学课程设置 2013（Computer Science Curricula 2013）中有关基础数据结构和算法部分的内容。

乍一看，本书中这些主题出现的顺序似乎很常规。典型情况下，传统的 CS2 课程大纲会对基础数据结构逐一按照顺序介绍。在这种模式中，学生会学习如何使用特定的数据结构，如何实现它，以及它的性能特性等，所有知识点会同时学习。这种方式的主要缺点是学生需要在掌握如何使用某种结构之前，就先理解它是如何实现的。例如，如果学生一开始不知道为什么某个应用要使用映射表，那么就很难让他们理解为什么可以优选某种实现模型而

不是另一种实现模型。

在斯坦福大学，我们采用了一种不同的策略——*客户优先方式*。学生在被要求思考任何实现问题之前，会先学习如何使用集合类的全集。他们还有机会去完成有趣的作业，在这些作业中他们会作为客户来使用这些集合类。在这个过程中，学生会对底层的数据模型和每种结构的用法获得更深刻的理解。一旦学生了解了客户端的视角，那么他们就已经准备好了去探索各种可能的实现及其对应的计算特性了。

客户优先方式被证明非常成功。在我们将这种改变引入 CS2 课程中之后，在所有教师教授的班级中，期中考试成绩的中位值提升了大约 15%，而期末考试的成绩则提升了超过 5%。课程等级和学生满意度都随着学生对课程内容理解程度的提高而不断增长。现在，我们每年向超过 1200 名学生教授 CS2，我们相信客户优先方式是产生这种变化的关键。

我撰写本书是为了让许多用 Java 来教授 CS2 课程的学校一起分享斯坦福大学的成功经验。我们自信地认为，你将会和我们一样，对于学生对知识的理解和运用程度的提升而感到惊讶。

补充材料

为学生提供的材料

本书的所有读者都可以在 Pearson 的网站（http://www.pearsonhighered.com/ericroberts）上获得下面各项材料：

- 书中每个示例程序的源代码。
- 样例运行的彩色 PDF 版本。
- 复习题的答案。

为教师提供的材料[⊖]

所有具有资质的教师都可以在 Pearson 的网站（http://www.pearsonhighered.com/ericroberts）上获得下面各项材料：

- 书中每个示例程序的源代码。
- 样例运行的彩色 PDF 版本。
- 复习题的答案。
- 编程习题的解决方案。
- 每一章的 PowerPoint 讲座幻灯片。

致谢

感谢斯坦福大学的同事，首先是 Julie Zelenski，感谢她开创性地开发了客户优先方式。我的同事 Keith Schwarz、Marty Stepp、Stephen Cooper、Cynthia Lee、Jerry Cain、Chris Piech 和 Mehran Sahami 都在教学策略和支撑材料这两方面做出了宝贵的贡献。还要向数任本科生部的领导和多年来的许多学生表达谢意，他们鼎力相助使教授这门课变得如此令人兴奋。

⊖ 关于本书教辅资源，只有使用本书作为教材的教师才可以申请，需要的教师请填写本书最后一页“教学支持申请表”，并通过邮件同时方送给培生与我方。——编辑注

此外，向 Pearson 出版社的 Marcia Horton、Tracy Johnson 和其他成员表示感谢，感谢他们数年来对本书及各个前期版本的支持。

一如既往，最诚挚的谢意要献给我的妻子 Lauren Rusk，她再次作为我的开发编辑完成了魔幻般的工作。Lauren 运用她的专业知识对本书的文字进行了仔细的打磨，如果没有她，就压根不会有本书。

Eric S. Roberts
斯坦福大学

目 录

Programming Abstractions in Java

第 1 章
Programming Abstractions in Java

Java 概览

程序脱胎于各种各样的实验。我们的经验是：程序并非出自我们这样的一两个人的思想，而是源自日复一日的辛苦工作。

——Stokely Carmichael 和 Charles V. Hamilton，《Black Power》，1967

在刘易斯·卡罗尔的《爱丽丝梦游仙境》中，国王让白兔“从起点处出发，一直走到终点处，然后停下来”。这是个好建议，但前提是你必须正在起点处。本书是为计算机科学的第二门课程设计的，因此，我们假设你已经在学习编程的道路上。同时，因为第一门所覆盖的课程内容变化相当大，所以对于教科书作者而言，很难认定你已经掌握的这样或那样的具体知识。例如，你们中有些人将从之前类似语言的经验中很容易地理解 Java 的控制结构，而有些人将会发现并不熟悉这些 Java 结构，正是由于这种背景知识上的差异性，本章将采纳“国王”的意见，介绍 Java 语言中编写简单程序所需的各个部分。

1.1 你的第一个 Java 程序

Java 的设计借鉴了多种设计源，包括 20 世纪 70 年代早期出现的 C 编程语言。在作为 C 的定义文档的著作《C 程序设计语言》[⊖]中，Brian Kernighan 和 Dennis Ritchie 在第一页上就给出了下面的建议：

> 学习一种新语言的唯一途径就是用它编写程序。对于所有语言的初学者而言，编写的第一个程序几乎都是相同的。
>
> 打印单词
>
> **hello, world**
>
> 尽管这个练习很简单，但对于初学者来说，仍然可能是一个巨大的障碍，因为要实现这个目的，首先必须编写程序文本，然后成功地进行编译，并加载、运行，最后输出到某个地方。掌握这些机制上的细节之后，其他事情就比较容易了。

如果你打算用 Java 编写最简单版本的“Hello World”程序，那么最终会产生看起来像图 1-1 中所示的代码。

```
/*
 * File: HelloWorld.java
 * ---------------------
 * This file is adapted from the example on page 1 of The C Programming
 * Language by Kernighan and Ritchie.
 */

public class HelloWorld {

   public static void main(String[] args) {
      System.out.println("hello, world");
   }

}
```

图 1-1 最小的“Hello World”程序

⊖ 该书已由机械工业出版社出版，书号为 978-7-111-12806-0/TP.2869。——编辑注

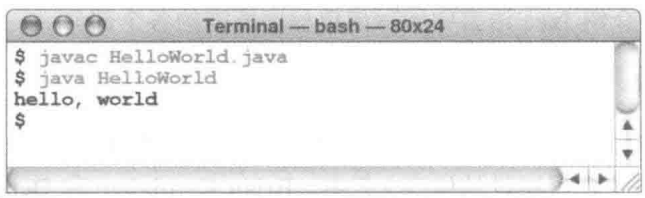
此刻，准确地理解这个程序中每一行代码的含义并不重要，以后我们有的是时间去掌握这些细节。你的任务应该是让 Hello.java 程序运行起来，即按照图 1-1 的样子准确地输入程序，然后搞清楚你需要做些什么才能让它运行起来。

你需要遵循的确切步骤根据创建和运行 Java 程序的编程环境的不同而不同。如果你的计算机支持命令行界面，例如 Mac 操作系统中的 Terminal 工具、Windows 机器上的 Console 应用，或者 Linux 中各种各样的 Shell 程序，那么在包含 HelloWorld.java 文件的目录下输入下面的命令就能运行“Hello World”程序：

```
javac HelloWorld.java
java HelloWorld
```

第一行命令将文件 HelloWorld.java 从图 1-1 中人类可阅读的形式转译为计算机可以更高效地执行的二进制文件。这个过程被称为*编译*。第二行命令会在该程序编译后的版本上运行 Java 解释器。

在命令行环境中，程序的输出会与用来编译和运行程序的命令显示在同一个窗口中。例如，在我的 Macintosh 机器上，终端窗口的会话看起来像下面这样：



尽管命令行模型可以完成任务，但是专业的 Java 程序往往会使用更加复杂的工具来集成编辑、编译、运行和调试程序的整个过程。这种工具被称为*编程环境*。有多种支持 Java 的编程环境，详细描述它们是不可能的。如果你正在某门课程中使用本书，那么你的老师可能会为你提供你想使用的编程环境的参考资料。如果你正在自己阅读本书，那么需要参考你所使用的编程环境的文档。

3 即使你正在使用专业的编程环境，从“Hello World”程序开始并运行它也不失为一种好的做法。不同的环境会以不同的方式显示控制台输出，但是你应该能够找到程序输出的令人感到亲切的“Hello World”问候语。尽管“其他事情就比较容易了”可能未必正确，但是你已经跨过了重要的里程碑。

1.2 Java 的历史

尽管本书阐述的是超越某种特定语言细节的有关编程策略的内容，但是我们必须选择某种语言，使得读者可以用这些技术来进行实验。毕竟，编程是一门动手实践的学科，仅仅靠阅读本书是无法成为成功的程序员的，即使你在纸面上解决了所有的练习题也是如此。学习编程是一项需要动手实践的工作，需要你用真实的编程语言去编写和调试程序。因为本书使用了 Java 编程语言，所以对 Java 的发展历史及其设计方案中体现的思想有所了解会对你大有帮助。

1.2.1 编程语言

在计算技术发展的早期岁月中，程序是用*机器语言*编写的，它们由机器能够直接执行的

基础指令构成。用机器语言编写的程序难以理解，主要是因为机器语言的结构反映的是硬件的设计，而不是程序员的需求。更糟糕的是，每种类型的计算硬件都有其自己的机器语言，这就意味着为一种机器编写的程序无法在另一种类型的硬件上运行。

20 世纪 50 年代中期，在 IBM，一组程序员在 John Backus 的领导下产生了一种深远地改变了计算本质的思想。Backus 和他的同事想知道：是否有可能编写出类似于他们正在计算的数学公式的程序，然后让计算机将这些公式翻译为机器语言？ 1955 年，这个团队开发出了 FORTRAN（这个名字是从 formula translation 这两个词中各抽取一部分构成的）的初始版本，它是第一种使得程序员可以用人类易于理解的高层概念来工作的语言。这种语言被称为*高级语言*。

自那时起，许多新的编程语言被不断地发明出来，大部分都是在之前语言的基础上以不断演进的方式而构建的。Java 在其演进中将两条分支汇聚在了一起。其前辈语言之一是被称为 C 的语言，这种语言是在 1972 年由贝尔实验室的 Dennis Ritchie 设计出来的，之后在 1989 年，美国国家标准学会（ANSI）对其进行了修订和标准化。但是 Java 还传承自另外一族语言，这族语言被设计用来支持一种风格完全不同的编程技术，这种风格在近年来已经极大地改变了软件开发的本质。 4

1.2.2 面向对象范型

在过去几十年中，计算机科学和编程技术经历了一场革命。与大多数革命——无论是政治巨变还是像 Thomas Kuhn 在他 1962 年出版的著作《科技革命的结构》中所描述的概念重建——一样，这种变革是挑战既有正统观念的思想萌发所驱动的。起初，这两种思想是相互竞争的，至少在一段时间内，旧秩序维持着它的主导地位。随着时间的推移，新思想的力度和推广程度都在不断增长，直至它开始以 Kuhn 所称的*范型转变*的形式替代旧思想。在编程技术中，旧秩序是由*过程范型*所支配的，在这种范型中，程序是由一组操作于数据之上的过程和函数构成的。新模型被称为*面向对象范型*，在这种范型中，程序看起来就像是一组数据对象，这些对象体现了特定的特性和行为。

面向对象编程思想并非全新的思想。第一种面向对象编程语言是 SIMULA，这是一种在 1967 年由来自斯堪的纳维亚半岛的计算机科学家 Ole-Johan Dahl 和 Kristen Nygaard 设计的语言，用来编写仿真代码。因为有着超越其所处时代的设计，SIMULA 预见到了许多后来成为编程技术常识的概念，包括抽象数据类型和现代的面向对象范型。事实上，用来描述面向对象语言的许多术语都来自于 1967 年有关 SIMULA 的最初报告。

遗憾的是，SIMULA 诞生后并没有引起人们很大的兴趣。第一种在计算业界内获得大量追随者的面向对象语言是 Smalltalk，它是在 20 世纪 70 年代晚期由施乐公司位于帕洛阿尔托的研究中心开发出来的。按照 Adele Goldberg 和 David Robson 所著的《Smalltalk-80：语言与实现》一书中的描述，Smalltalk 的目的是使编程成为更多人可以胜任的工作。

尽管具有许多很吸引人的特性以及简化了编程过程的高度可交互的用户环境，Smalltalk 在商业上还是一直不太成功。整个业界只是在面向对象编程的核心思想融入 C 的各种变体中之后，才开始对其感兴趣，因为 C 已经是工业标准了。尽管有多种并行展开的基于 C 设计面向对象语言的尝试，但是最成功的面向对象语言是 C++，它是 Bjarne Stroustrup 于 20 世纪 80 年代在 AT&T 贝尔实验室设计出来的。通过使面向对象技术与既有的 C 代码集成起来成为可能，C++ 使得大量的程序员能够以循序渐进的方式来采用面向对象范型。 5