

Java编程思维

作者：艾伦 唐尼 (Allen B Downey)

版权信息

书名：Java编程思维

作者：[美] Allen B. Downey Chris Mayfield

译者：袁国忠

ISBN：978-7-115-44015-0

本书由北京图灵文化发展有限公司发行数字版。版权所有，侵权必究。

您购买的图灵电子书仅供您个人使用，未经授权，不得以任何方式复制和传播本书内容。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

091507240605ToBeReplacedWithUserId

[版权声明](#)

[O'Reilly Media, Inc. 介绍](#)

[业界评论](#)

[前言](#)

[编写理念](#)

[面向对象编程](#)

[附录](#)

[使用代码示例](#)

[排版约定](#)

[Safari® Books Online](#)

[联系我们](#)

[致谢](#)

[电子书](#)

[第1章 编程之道](#)

[1.1 何为编程](#)

[1.2 何为计算机科学](#)

[1.3 编程语言](#)

[1.4 Hello World程序](#)

[1.5 显示字符串](#)

[1.6 转义序列](#)

[1.7 设置代码格式](#)

[1.8 调试代码](#)

[1.9 术语表](#)

[1.10 练习](#)

[第2章 变量和运算符](#)

[2.1 声明变量](#)

[2.2 赋值](#)

[2.3 状态图](#)

[2.4 显示变量](#)

[2.5 算术运算符](#)

[2.6 浮点数](#)

[2.7 舍入误差](#)

[2.8 字符串运算符](#)

[2.9 组合](#)

[2.10 错误类型](#)

[2.11 术语表](#)

[2.12 练习](#)

[第3章 输入和输出](#)

[3.1 System类](#)

[3.2 Scanner类](#)

[3.3 程序结构](#)

[3.4 英寸到厘米的转换](#)

[3.5 字面量和常量](#)

[3.6 设置输出的格式](#)

[3.7 厘米到英寸的转换](#)

[3.8 求模运算符](#)

[3.9 整合](#)

[3.10 Scanner类的bug](#)

[3.11 术语表](#)

[3.12 练习](#)

[第4章 void方法](#)

[4.1 Math类的方法](#)

[4.2 再谈组合](#)

[4.3 添加方法](#)

[4.4 执行流程](#)

[4.5 形参和实参](#)

[4.6 多个形参](#)

[4.7 栈图](#)

[4.8 阅读文档](#)

[4.9 编写文档](#)

[4.10 术语表](#)

[4.11 练习](#)

[第5章 条件和逻辑](#)

[5.1 关系运算符](#)

[5.2 逻辑运算符](#)

[5.3 条件语句](#)

[5.4 串接和嵌套](#)

[5.5 标志变量](#)

[5.6 return语句](#)

[5.7 验证输入](#)

[5.8 递归方法](#)

[5.9 递归栈图](#)

[5.10 二进制数](#)

[5.11 术语表](#)

[5.12 练习](#)

[第6章 值方法](#)

[6.1 返回值](#)

[6.2 编写方法](#)

[6.3 方法组合](#)

[6.4 重载](#)

[6.5 boolean方法](#)

[6.6 Javadoc标签](#)

[6.7 再谈递归](#)

[6.8 姑且相信](#)

[6.9 再举一个例子](#)

[6.10 术语表](#)

[6.11 练习](#)

[第7章 循环](#)

[7.1 while语句](#)

[7.2 生成表格](#)

[7.3 封装和泛化](#)

[7.4 再谈泛化](#)

[7.5 for语句](#)

[7.6 do-while循环](#)

[7.7 break和continue](#)

[7.8 术语表](#)

[7.9 练习](#)

[第8章 数组](#)

[8.1 创建数组](#)

[8.2 访问元素](#)

[8.3 显示数组](#)

[8.4 复制数组](#)

[8.5 数组的长度](#)

[8.6 数组遍历](#)

[8.7 随机数](#)

[8.8 遍历和计数](#)

[8.9 生成直方图](#)

[8.10 改进的for循环](#)

[8.11 术语表](#)

[8.12 练习](#)

[第9章 字符串](#)

[9.1 字符](#)

[9.2 字符串是不可修改的](#)

[9.3 字符串遍历](#)

[9.4 子串](#)

[9.5 方法indexOf](#)

[9.6 字符串比较](#)

[9.7 设置字符串的格式](#)

[9.8 包装类](#)

[9.9 命令行实参](#)

[9.10 术语表](#)

[9.11 练习](#)

[第 10 章 对象](#)

[10.1 Point对象](#)

[10.2 属性](#)

[10.3 将对象用作参数](#)

[10.4 将对象作为返回类型](#)

[10.5 可修改的对象](#)

[10.6 指定别名](#)

[10.7 关键字null](#)

[10.8 垃圾收集](#)

[10.9 类图](#)

[10.10 Java类库的源代码](#)

[10.11 术语表](#)

[10.12 练习](#)

[第 11 章 类](#)

[11.1 Time类](#)

[11.2 构造函数](#)

[11.3 再谈构造函数](#)

[11.4 获取方法和设置方法](#)

[11.5 显示对象](#)

[11.6 方法toString](#)

[11.7 方法equals](#)

[11.8 时间相加](#)

[11.9 纯方法和非纯方法](#)

[11.10 术语表](#)

[11.11 练习](#)

[第 12 章 对象数组](#)

[12.1 Card对象](#)

[12.2 方法toString](#)

[12.3 类变量](#)

[12.4 方法compareTo](#)

[12.5 Card对象是不可修改的](#)

[12.6 Card数组](#)

[12.7 顺序查找](#)

[12.8 二分法查找](#)

[12.9 跟踪代码](#)

[12.10 递归版本](#)

[12.11 术语表](#)

[12.12 练习](#)

[第 13 章 数组对象](#)

[13.1 Deck类](#)

[13.2 洗牌](#)

[13.3 选择排序](#)

[13.4 合并排序](#)

[13.5 方法subdeck](#)

[13.6 方法merge](#)

[13.7 添加递归](#)

[13.8 术语表](#)

[13.9 练习](#)

[第 14 章 包含其他对象的对象](#)

[14.1 Deck和手里的牌](#)

[14.2 CardCollection](#)

[14.3 继承](#)

[14.4 发牌](#)

[14.5 Player类](#)

[14.6 Eights类](#)

[14.7 类之间的关系](#)

[14.8 术语表](#)

[14.9 练习](#)

[附录 A 开发工具](#)

[A.1 安装DrJava](#)

[A.2 DrJava Interactions窗格](#)

[A.3 命令行界面](#)

[A.4 命令行测试](#)

[A.5 运行Checkstyle](#)

[A.6 使用调试器进行跟踪](#)

[A.7 用JUnit进行测试](#)

[A.8 术语表](#)

[附录 B Java 2D 图形](#)

[B.1 创建图形](#)

[B.2 Graphics类的方法](#)

[B.3 绘图示例](#)

[B.4 术语表](#)

[B.5 练习](#)

[附录 C 调试](#)

[C.1 编译时错误](#)

[C.1.1 编译器显示大量的错误消息](#)

[C.1.2 编译器显示怪异的错误消息，怎么都消除不掉](#)

[C.1.3 怎么做都无法让程序通过编译](#)

[C.1.4 按编译器说的做了，但还是不管用](#)

[C.2 运行时错误](#)

[C.2.1 程序挂起](#)

[C.2.2 程序运行时出现异常](#)

[C.2.3 添加了很多打印语句，输出都泛滥成灾了](#)

[C.3 逻辑错误](#)

[C.3.1 程序不管用](#)

[C.3.2 冗长表达式的结果出乎意料](#)

[C.3.3 方法的返回值出乎意料](#)

[C.3.4 打印语句什么都不显示](#)

[C.3.5 陷入了绝境，无法自拔](#)

[C.3.6 必须得有人帮我](#)

[C.3.7 终于找到bug了！](#)

[作者简介](#)

[封面简介](#)

版权声明

© 2016 Allen B. Downey and Chris Mayfield.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2016. Authorized translation of the English edition, 2016 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版，2016。

简体中文版由人民邮电出版社出版，2017。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

O'Reilly Media, Inc. 介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 Make 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版、在线服务或者面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar 博客有口皆碑。”

——*Wired*

“O'Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——*Business 2.0*

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——*Irish Times*

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔视野，并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路遇到岔路口，走小路（岔路）。’回顾过去，Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——*Linux Journal*

前言

本书是针对初学者编写的计算机科学和编程入门教程。从最基本的概念入手，每个术语都在首次使用时给出详尽的定义；循序渐进地介绍新概念；将内容广泛的主题（如递归和面向对象编程）分成多个部分，并分多章介绍。

本书简明扼要，每章都只有十几页的篇幅，涵盖了一周的大学课程内容。本书无意全面介绍 Java，只是想让读者了解基本的编程结构和技巧。我们从小问题和基本算法着手，逐步过渡到面向对象设计，用计算机教学学术语讲，本书采取的是“迟来的对象”法。

编写理念

本书是基于如下的指导原则编写的。

- **每次一个概念。**对于可能给初学者带来麻烦的主题，将其分成多个部分，让读者无需熟悉整个主题就能将新学到的概念付诸实践。
- **兼顾 Java 和概念。**本书的主要目的并非介绍 Java，而是用代码示例诠释计算机科学概念。大多数章节以 Java 的语言特性开头，以概念结束。
- **简明扼要。**本书的一个重要目标是使篇幅够小，好让读者一个学期就能读完并搞懂本书内容。
- **突出术语。**尽可能少引入术语，并在首次使用时给出术语的详尽定义。在每章末尾，我们还将它们组织成了术语表。
- **程序开发策略。**程序编写策略有很多，包括自下而上、自上而下，等等。我们演示了开发程序的多种方法，让读者能够从中选择最适合的。
- **多条学习曲线。**要编写程序，得理解算法、熟悉编程语言，还要能够调试代码。本书始终在讨论这些内容，同时专辟了一个附录来总结调试建议。

面向对象编程

有些 Java 书一上来就介绍类和对象，有些则先介绍过程性编程，再逐步过渡到面向对象编程。

Java 的很多面向对象功能都旨在解决以前的语言存在的问题，因此，其实现受到了这些历史原因的影响。对于这些功能，如果你不熟悉它们所能解决的问题，就很难理解。

我们每次介绍一个概念，并尽可能将它讲清楚，让读者能够立即将学到的知识付诸实践。在这个前提之下，我们会尽早地介绍面向对象编程，因此，你不可能翻开本书就接触到这个主题。

然而，如果不使用面向对象功能，根本就无法编写 Java 程序，哪怕是简单的 Hello World 程序。对于有些功能，我们会在首次提及时简要地介绍一下，再在后面作更深入的讨论。

本书几乎涵盖了“AP Java subset”中的每个主题，非常适合用来备考 AP 计算机科学 A 考试（包括面向对象设计和实现）。我们的网站 <http://thinkjava.org> 中列出了本书各节与 AP 课程最新描述的对应关系。

附录

本书适合按顺序逐章阅读，因为每一章都以前一章的内容为基础。本书还有三个附录，你可在任何时间阅读。

- **附录 A（开发工具）**

编译、运行和调试 Java 代码的步骤随开发环境和操作系统而异，我们没有将这些细节放在正文中，因为这会分散读者的注意力。相反，我们专辟了附录 A，简要地介绍 DrJava——一个非常适合初学者使用的集成开发环境（interactive development environment, IDE），以及用于检查代码质量的 Checkstyle 和用于测试的 JUnit 等工具。

- **附录 B（Java 2D 图形）**

Java 提供了处理图形和动画的库，这些主题可能对学生很有吸引力。这些库涉及面向对象功能，读者可能阅读完前 11 章才能完全理解，但可以很早地使用它们。

- **附录 C（调试）**

有关调试的建议遍布全书，我们将这些调试建议收集到了附录 C 中。建议读者在阅读本书的过程中反复温习该附录。

使用代码示例

本书的示例代码大都在 Git 仓库 <https://github.com/AllenDowney/ThinkJavaCode> 中找到。Git 是一个版本控制系统，让你能够跟踪项目中的文件。受 Git 控制的文件集合称为“仓库”。

GitHub 是一种托管服务，为 Git 仓库提供存储空间，还提供了方便的 Web 界面。它提供了多种处理代码的方式。

- 单击 Fork 键可以在 GitHub 上创建仓库的副本。如果你没有 GitHub 账户，就需要创建一个。建立分支后，你便在 GitHub 上有自己的仓库了，可用它来跟踪你编写的代码。然后，还可以“克隆”这个仓库，即将文件的副本下载到计算机。
- 你也可以在不建立分支的情况下克隆仓库。这样就不需要 GitHub 账户了，但也无法将所做的修改保存到 GitHub 中。
- 如果你根本不想使用 Git，可用 GitHub 页面上的 Download ZIP 按钮下载 ZIP 格式的代码，也可通过链接 <http://www.tinyurl.com/ThinkJavaCodeZip> 下载。

克隆仓库或解压 ZIP 文件后，你将看到一个名为 ThinkJavaCode 的目录，其中包含与本书每章对应的子目录。

本书中的所有示例都是用 Java SE Development Kit 8 开发和测试的。如果你使用的是更新的版本，这些示例也能正确地运行；但如果你使用的是更早的版本，有些示例可能无法正确地运行。

排版约定

本书使用了下列排版约定。

- **楷体**

表示术语或重点强调的内容。

- **等宽字体（constant width）**

表示程序片段，以及正文中出现的变量、函数名、数据库、数据类型、环境变量、语句和关键字等。

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)

文档名称：《Java编程思维》艾伦 唐尼 (Allen B Downey) 著.epub

请登录 <https://shgis.cn/post/282.html> 下载完整文档。

手机端请扫码查看：

