

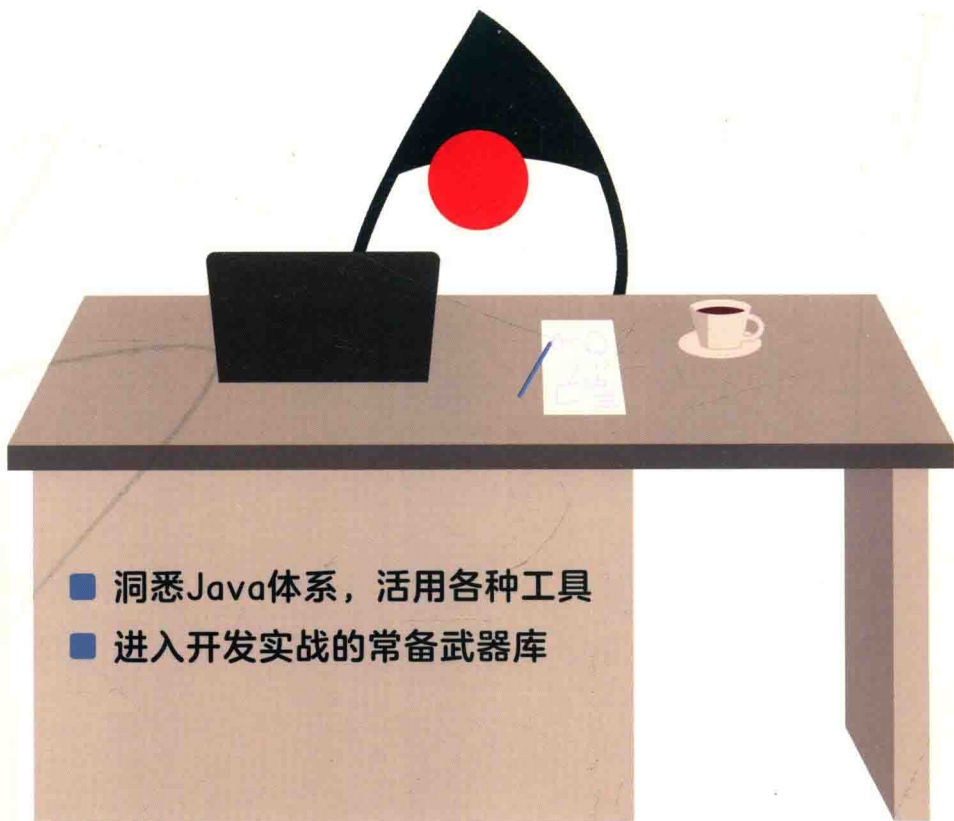
TURING

图灵程序设计丛书

Java for the Real World

JAVA 实践指南

[美] 菲利普·约翰逊 著 武传海 译



- 洞悉Java体系，活用各种工具
- 进入开发实战的常备武器库

 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

Java for the Real World

JAVA 实践指南

[美] 菲利普·约翰逊 著 武传海 译



人民邮电出版社
北京

图书在版编目 (C I P) 数据

Java实践指南 / (美) 菲利普·约翰逊
(Phillip Johnson) 著 ; 武传海译. — 北京 : 人民邮
电出版社, 2019.9
(图灵程序设计丛书)
ISBN 978-7-115-51786-9

I. ①J… II. ①菲… ②武… III. ①JAVA语言—程序
设计—指南 IV. ①TP312.8-62

中国版本图书馆CIP数据核字(2019)第172758号

内 容 提 要

Java 因其强大、易用等诸多优点而广受青睐、久盛不衰。本书是 Java 实践指南, 从实战角度指导读者快速上手 Java 编程。各章结合代码示例依次介绍了 JVM 环境搭建、Java 虚拟机、常用构建工具、编写并运行测试、Spring、Web 应用框架、Web 应用部署、数据库使用、日志和实用第三方库等内容。

本书适合所有对 Java 编程感兴趣的读者作为入门指导。

-
- ◆ 著 [美] 菲利普·约翰逊
 - 译 武传海
 - 责任编辑 岳新欣
 - 责任印制 周昇亮

 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京市艺辉印刷有限公司印刷

 - ◆ 开本: 800×1000 1/16
印张: 8.75
字数: 207千字 2019年9月第1版
印数: 1-3 000册 2019年9月北京第1次印刷
著作权合同登记号 图字: 01-2017-7886号
-

定价: 49.00元

读者服务热线: (010)51095183转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147号

版权声明

Authorized translation from the English language edition, entitled *Java for the Real World* by Phillip Johnson Copyright © 2017.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the author. CHINESE language edition published by Posts & Telecom Press, Copyright © 2019.

本书中文简体字版由Phillip Johnson授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书。

版权所有，侵权必究。

引 言

“所以，你打算把我调到开发岗位吗？”

“嗯，我是这样想的。”

从此，我成了一名 Java 开发者。不久，我便接手了一个写得很糟糕并且满是 bug 的 ETL 程序，它所依赖的框架早在我读高中的时候就已经废弃了，而且没有测试代码。我天真地想：我学 Java 差不多有一年了，对我来说，添加测试应该不难，只要仔细地重构代码就行了。但是，这些 XML 文件有什么用？独立文件中的 SQL 是怎么进入 DAO 和 DAOImpl 的？程序中为什么有 Ant 和 Maven 的构建脚本？（这个问题我一直没想明白。）Ant 和 Maven 又是什么？我又是 Google 搜索，又是请教专家，还动手做了试验，最终才好不容易搞定了这些问题。然而，写完自己第一个真正的 Java 程序时，想起这种语言的巨大反差带给我的折磨，我仍然惊魂甫定。

多年后，我晋升为高级开发工程师，团队决定新招一名初级开发人员。来了一个小伙子，他大学毕业刚一年，在之前的工作中主要使用 JavaScript。但是，他在学校学过 Java，并且很有天分。实际上，他的毕业设计是用 C++ 语言从零开始编写了一个 3D 图形渲染器。入职第一天，我给他展示了一个小的 Web 应用，这个应用以后就由他来做，并向他介绍了整个项目。很快，我就发现他对 Java 的理解只停留在语言层面，和几年前的我一模一样，而且他对 Maven、MyBatis 及 Tomcat 一概不懂。

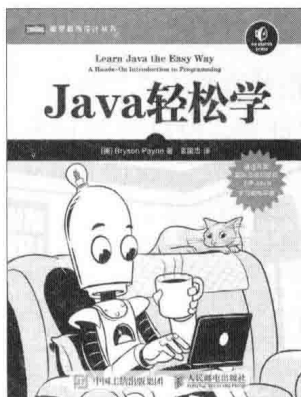
对我来说，为自己的无知找个借口很容易，比如没有系统地学习过计算机科学。这个小伙子虽然在学校学过编程，可还是被难住了。我们的求学道路截然不同，但结果都是一名不合格的 Java 开发者。事实证明，大多数 Java 教学只停留在对标准库的讲解上。我编写本书的初衷是希望自己当初开始学 Java 时能有这样一本书。我希望在你开始 Java 职业生涯时，本书能给你提供一些帮助。祝你编码快乐！

电子书

扫描如下二维码，即可购买本书电子版。



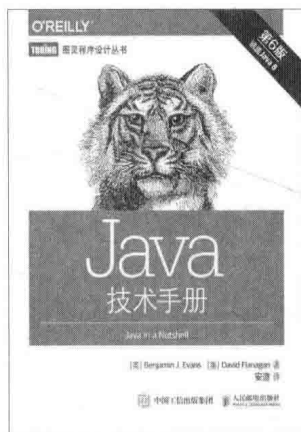
技术改变世界 · 阅读塑造人生



Java 轻松学

- ◆ 针对Java零基础读者
- ◆ 通过开发实际应用和游戏上手Java，学习曲线平缓

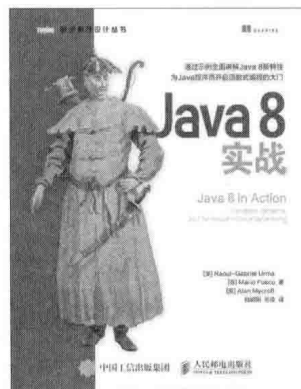
作者: Bryson Payne
译者: 袁国忠



Java 技术手册 (第6版)

- ◆ 帮助有经验的Java程序员充分使用Java 7和Java 8的功能，也可供Java新手学习
- ◆ 通过大量示例演示如何充分利用现代API和开发过程中的最佳实践

作者: Benjamin J Evans, David Flanagan
译者: 安道



Java 8 实战 (第2版即将上市)

- ◆ Java 8终极指南
- ◆ 通过新的Stream API及Lambda表达式等示例全面讲解Java 8新特性，为Java程序员开启函数式编程的大门

作者: Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft
译者: 陆明刚, 劳佳

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

目 录

第 1 章 入门介绍	1	3.4.1 Ant	25
1.1 目标读者	1	3.4.2 Maven	25
1.2 如何使用本书	1	3.4.3 Gradle	25
1.3 搭建环境	3	第 4 章 测试	27
1.3.1 安装 Java	3	4.1 向 IScream 应用程序添加服务	27
1.3.2 集成开发环境	3	4.2 编写测试	30
第 2 章 Java 虚拟机	5	4.2.1 JUnit	30
2.1 何为 Java 虚拟机	5	4.2.2 TestNG	31
2.2 JVM 版本	5	4.3 运行测试	33
2.3 JVM 种类	7	4.4 使用测试替身	34
第 3 章 构建工具	9	4.4.1 为可模拟服务修改 IScream	34
3.1 Ant	11	4.4.2 使用 Mocks 创建测试	35
3.1.1 构建文件	11	4.4.3 EasyMock	36
3.1.2 使用 Ivy 管理依赖	14	4.4.4 Mockito	37
3.1.3 小结	15	4.4.5 PowerMock	39
3.2 Maven	16	4.5 小结	40
3.2.1 Maven 任务	16	4.6 参考资料	40
3.2.2 项目对象模型文件	17	4.6.1 综合测试	40
3.2.3 插件	18	4.6.2 测试框架相关	41
3.2.4 仓库和发布	19	第 5 章 Spring	43
3.2.5 小结	20	5.1 Spring Core	43
3.3 Gradle	21	5.1.1 依赖注入	43
3.3.1 构建文件	21	5.1.2 属性	49
3.3.2 任务	22	5.2 Spring Boot	51
3.3.3 依赖管理	23	5.2.1 运行 Spring Boot 应用程序	51
3.3.4 Gradle 守护进程	24	5.2.2 配置	52
3.3.5 小结	25	5.3 小结	53
3.4 参考资料	25	5.4 参考资料	53

第 6 章 Web 应用程序框架	55
6.1 Java EE Web API	56
6.1.1 请求和响应	56
6.1.2 JavaServer Pages	56
6.1.3 servlet 容器	56
6.2 Spring MVC	57
6.2.1 模型	57
6.2.2 视图	59
6.2.3 控制器	60
6.2.4 配置	61
6.3 Spring Boot	64
6.3.1 Thymeleaf	64
6.3.2 运行 Spring Boot Web 应用 程序	65
6.4 JavaServer Faces	65
6.4.1 托管 Bean	66
6.4.2 JSF 视图	69
6.5 Vaadin	70
6.5.1 布局和组件	70
6.5.2 Vaadin UI	72
6.5.3 主题	73
6.5.4 运行应用程序	73
6.6 小结	74
6.7 参考资料	74
第 7 章 Web 应用程序部署	75
7.1 打包	75
7.2 部署	77
7.3 参考资料	79
第 8 章 使用数据库	81
8.1 Java 数据库连接	81
8.2 Spring JDBC 模板	82
8.2.1 IScream 新数据模型	83
8.2.2 查询数据	86
8.2.3 写数据	87
8.3 MyBatis	90
8.3.1 查询数据	90
8.3.2 写数据	93
8.3.3 动态 SQL	95
8.4 Hibernate	96
8.4.1 领域 POJO 调整	96
8.4.2 JPA 注解	97
8.4.3 XML 映射	98
8.4.4 写数据	100
8.4.5 读数据	101
8.5 小结	102
8.6 参考资料	103
第 9 章 日志	105
9.1 java.util.Logging	105
9.2 Log4j	107
9.3 Logback	111
9.4 SLF4J	112
9.5 JCL	114
9.6 小结	115
9.7 参考资料	115
第 10 章 有用的第三方库	117
10.1 JSON 支持	117
10.1.1 Google Gson	117
10.1.2 Jackson	119
10.2 实用工具库	121
10.2.1 Guava	121
10.2.2 Apache Commons	123
10.3 Joda Time 库	126
10.4 小结	127
10.5 参考资料	127
附录 A Docker	129
A.1 创建 Docker 镜像	129
A.2 部署 Docker 容器	130
A.3 注意事项	131
A.3.1 内存	131
A.3.2 JDK	131
A.4 参考资料	131



1.1 目标读者

如书名所示，本书针对的是在商业环境中使用 Java 的人士。根据我的个人经验，学习 Java 体系几乎和学习 Java 语言一样困难。对经验丰富的程序员来说，相比于学习 Java 体系，学习 Java 语言可能没什么难度。虽然学习 Java 语言有大量工具可利用，但是介绍 Java 体系的资源并不多。本书旨在介绍编写专业 Java 软件所需的各种框架、工具和库。

不管你是刚毕业，还是自学过编程，只要你想进入这个领域，本书都能为你提供大量实用知识，招聘主管会很看重你是否具备这些知识。实际工作中，你可能根本不需要写什么排序算法，但你肯定会遇到使用 Hibernate 实现持久化的 Spring MVC Web 应用。另一方面，如果你已经是一名专业开发者，并且理解了相关概念，那你更有可能问自己：“Java 是如何实现……的？”

本书不教授 Java 基础知识！阅读本书需要了解 Java 标准类库。如果你确实需要从头学习 Java，建议你首先阅读 *Head First Java*，然后再阅读一本比较新的深入讲解 Java 8 的书。

如果你准备好学习如何开发企业级 Java 应用了，那就开始吧！

1.2 如何使用本书

本书每章都会讲一个一般性概念，而且在某种程度上，各章相互依赖。所以，如果你有时间，建议从头到尾阅读本书。不过，如果你时间有限，可以只阅读感兴趣的章节。

文字解释的效果有限，所以本书会着重于代码呈现。相关代码都在正文中给出了，但简洁起见，省略了一些样板代码。访问本书中文版页面（<http://www.ituring.com.cn/book/2438>），可以找到完整的项目代码^①。

^① 你也可提交中文版勘误。——编者注

示例代码形式如下。

OrderService.java

```
27     public void save(Order order) {  
28         try(Session session = sessionFactory.openSession()) {  
29             Transaction tx = session.beginTransaction();  
30             session.persist(order);  
31             tx.commit();  
32         } // Session 自动关闭  
33     }
```

其他用于讨论或演示的代码形式如下。

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

本书篇幅不大，我们不会详细讲解任何工具。很多情况下，已有专门的图书详细讲解了某个工具。本书旨在简要介绍一些工具及其基本用法。如果你想学习更多相关内容，请进一步阅读每章后面的参考资源。

阅读本书过程中，你会见到如下标志。



这到底是什么？

框架开发者喜欢用一些花哨的词描述其工具。比如，“Apache Maven 是一个软件项目管理和理解工具”就很不容易理解。阅读这些部分，你可以快速了解某个工具的具体用途。



Java 痼

Java 是一门比较老的语言，向后兼容性较好。所以目前还存在许多过时的用法和大量废弃的标准库，本书称其为“Java 痼”。



落后警告

这是一个警告，提醒你注意在遗留系统中可能遇到的一些东西。你应该尽量避免在新项目中使用这些东西。



超前警告

这与上一个标志正好相反。它提醒你注意 Java 中一些新引入的“东西”，它们可能尚未被广泛采用。这不一定是坏事，只是提醒你注意。



更多内容

这里给出了更多相关信息，以补充正文中提到的内容。它们并不是特别重要，但是如果你感兴趣，可以把它作为延伸阅读。

1.3 搭建环境

1.3.1 安装 Java

安装 Java 的方法有很多种，请根据你所用的操作系统和个人喜好来选择。

Homebrew (macOS): `brew cask install java`

Chocolatey (Windows): `choco install jdk9`

Apt-Get (Linux): `sudo apt-get install default-jdk`

SDKMAN! (类 Unix): `sdk install java`

官方安装方法：访问 Oracle 官网，根据相关提示进行安装。请确保安装的是 JDK 而非 JRE。如果你选择了这种安装方法，请务必认真阅读 Oracle JDK 的许可协议（相关内容见下一章）。

这些工具会把 Java 放到你的 PATH 变量中。为了确认这一点，可在命令行中输入 `java -version`，输出信息如下所示。

```
java version "1.8.0_131"  
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)  
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
```

此外，你可能还需设置 `JAVA_HOME` 环境变量，许多使用 Java 的工具都会首先查看它。你也可以在一台机器上安装多个 Java 版本，只需把 `JAVA_HOME` 指向主版本即可，这在测试 Java 新版本时很有用。

1.3.2 集成开发环境

常用的 Java 集成开发环境 (IDE) 有三种，分别为 Eclipse、IntelliJ 和 Netbeans。我青睐 IntelliJ，因为它的功能集丰富，并且集成了大量开发框架。不过，这三者之中，只有 IntelliJ 推出了付费

版。它的免费社区版也不错，我家中计算机安装的就是社区版。对于工作项目，强烈建议使用付费旗舰版。

有些公司为了定制的插件和设置，明确要求开发者使用特定的 IDE 开发环境，实为多此一举。因为从技术上讲，在命令行中可以编译任何 Java 代码，所以开发者选用哪款 IDE 编写代码其实无所谓。

2.1 何为 Java 虚拟机

多年来，支持多平台一直是 Java 的卖点之一。比如，你在 Mac 上编写和测试 Java 代码，然后将其部署到 Windows 服务器上，它也能正常运行。这是因为编译好的代码和操作系统之间有一个 Java 虚拟机 (JVM)，它可以把 Java 转换成本地系统调用。准确地说，具体承担这个转换任务的是一个 JVM 实例。JVM 既可以指 Java 虚拟机规范，也可以指其某个实现。

2.2 JVM 版本

Java 维护者会定期更新 JVM 规范。通过这种方式，他们可以给 Java 添加新特性或对其进行改进。写作本书时，Oracle 公司发布的 Java 最新版本是 Java 1.9。方便起见，人们通常把 Java 1.9 简称为“Java 9”，把 Java 1.8 简称为“Java 8”，以此类推。



落后警告：过时的 JVM

然而有些公司对新技术并不怎么上心，金融行业尤为突出。有些 Java 学习资料、博客、新闻等设想读者用的是 Java 8 或 Java 9，实际上读者用的却是 Java 7，甚至是 Java 6。

明确程序将来在哪个 JVM 版本上运行十分重要，在着手开发之前就应该确定下来。JVM 版本不同，可使用的 Java 语言特性也不同。然而，不管你编写 Java 代码时使用的是哪个版本的 JVM，你都可以在自己的机器上安装最新的 JVM 来运行。这是因为 Java 有着良好的向后兼容特性，比如，针对 JVM 1.6 编写的 Java 代码可以在 JVM 1.8 上正常运行。

为了验证这一点，下面给出两个示例程序，它们的功能相同。

NamesOld.java

```

1 import java.util.ArrayList;
2 import java.util.List;
3
4 public class NamesOld {
5     public static void main(String[] args) {
6         List<String> names = new ArrayList<String>();
7         names.add("Foo");
8         names.add("Bar");
9         for(String name : names) {
10             System.out.println(name);
11         }
12     }
13 }

```

NamesNew.java

```

1 import java.util.ArrayList;
2 import java.util.List;
3
4 public class NamesNew {
5     public static void main(String[] args) {
6         List<String> names = new ArrayList<>();
7         names.add("Foo");
8         names.add("Bar");
9         names.forEach(System.out::println);
10    }
11 }

```

针对两个不同版本的 JVM，编译上面两段程序，结果如下所示。

	javac -source 1.6	javac -source 1.8
NamesOld.java	无错误	无错误
NamesNew.java	两个编译错误	无错误

**Java 疣：向后兼容性**

Java 维护者高瞻远瞩，让 Java 具备了良好的向后兼容性，这可能是促使 Java 今天无处不在的原因之一。然而，这样做也带来了许多“包袱”，如下所示。

- ❑ 自 1997 年以来，废弃了 `java.util.Date` 包中的一些方法。
- ❑ 在很大程度上，`java.time.*` 已经取代 `java.util.Date` 包。
- ❑ 泛型只用于编译时检查，运行时会被清除。
- ❑ `const` 和 `goto` 两个关键字被保留，但并未实现。

- 所有基本类型都有封装类，比如 `int` 和 `java.lang.Integer`，`boolean` 和 `java.lang.Boolean`，等等。
- 通常应该尽量避免使用 `Hashtable` 和 `Vector` 集合，而使用 `HashMap` 和 `ArrayList`。如果你需要的集合要用于并发编程，那么应该考虑使用 `java.util.concurrent` 包。

2.3 JVM 种类

其实 Oracle 推出了两款 JVM 产品，即 Oracle JVM 和 OpenJDK JVM。几乎在所有情况下，任选一个即可。建议选择在你的系统中最容易安装的那个。不过请注意，在 Oracle JVM 中存在一些许可限制，但 OpenJDK 没有。如果你在意这些许可条款，建议咨询公司的法务部门。

另外值得一提的是，由于 JVM 规范是公开的，所以任何人都可以打造自己的 JVM。事实上，有些人和有的公司也这样做了，比如 IBM 的 J9、Azul 的 Zing、Excelsior 的 JET 等。这些第三方 JVM 实现都有各自的宣传噱头，但通常可以归结为以下三个方面：针对不同的操作系统、性能提升或添加了新特性。

Oracle JVM 和 OpenJDK JVM 的区别

两者区别不大。Oracle 官方博客中写道：

……Oracle JDK 是在 OpenJDK 7 基础上发布的，添加了更多功能，比如部署代码，包含了 Oracle 对 Java Plugin 和 Java WebStart 的实现，以及一些闭源第三方组件（比如 Graphics Rasterizer）和一些开源第三方组件（像 Rhino），还有其他一些零碎的东西，比如额外的文档或第三方字体等。

Henrik Stahl, “Java 7 Questions & Answers”, 2011 年 8 月 11 日