累计销量
10万册

JWorld@TW技术论坛版主
Java权威技术顾问与专业讲师

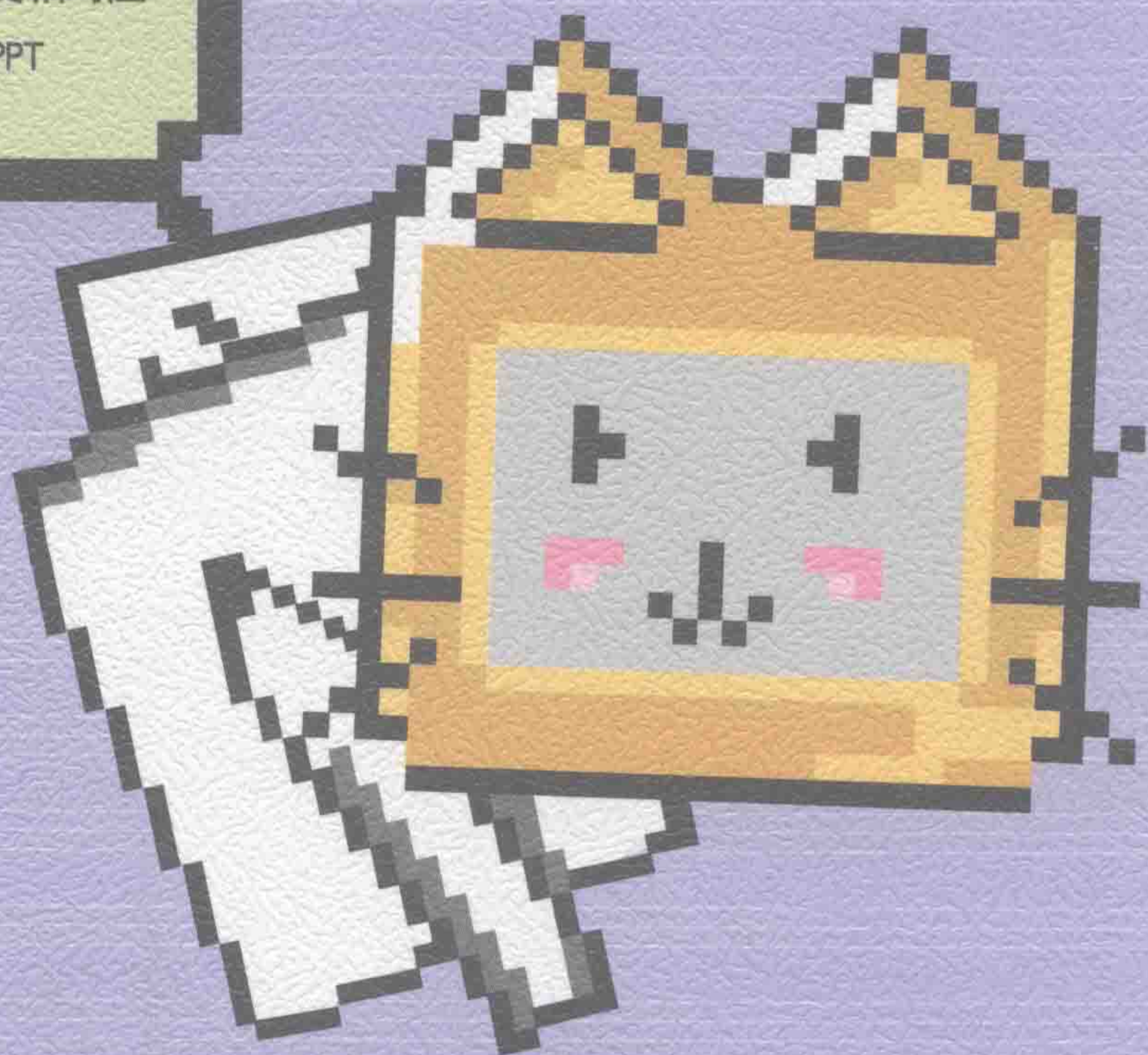
全新改版！

JDK 9

Java 学习笔记

- 分享作者学习Java心得
- 涵盖OCP/JP(原SCJP)考试范围
- 提供Java 9新功能快速索引
- 深入探讨Java模块平台系统
- JDK基础与IDE操作交相对应
- 提供Lab文档与教学PPT

林信良 编著

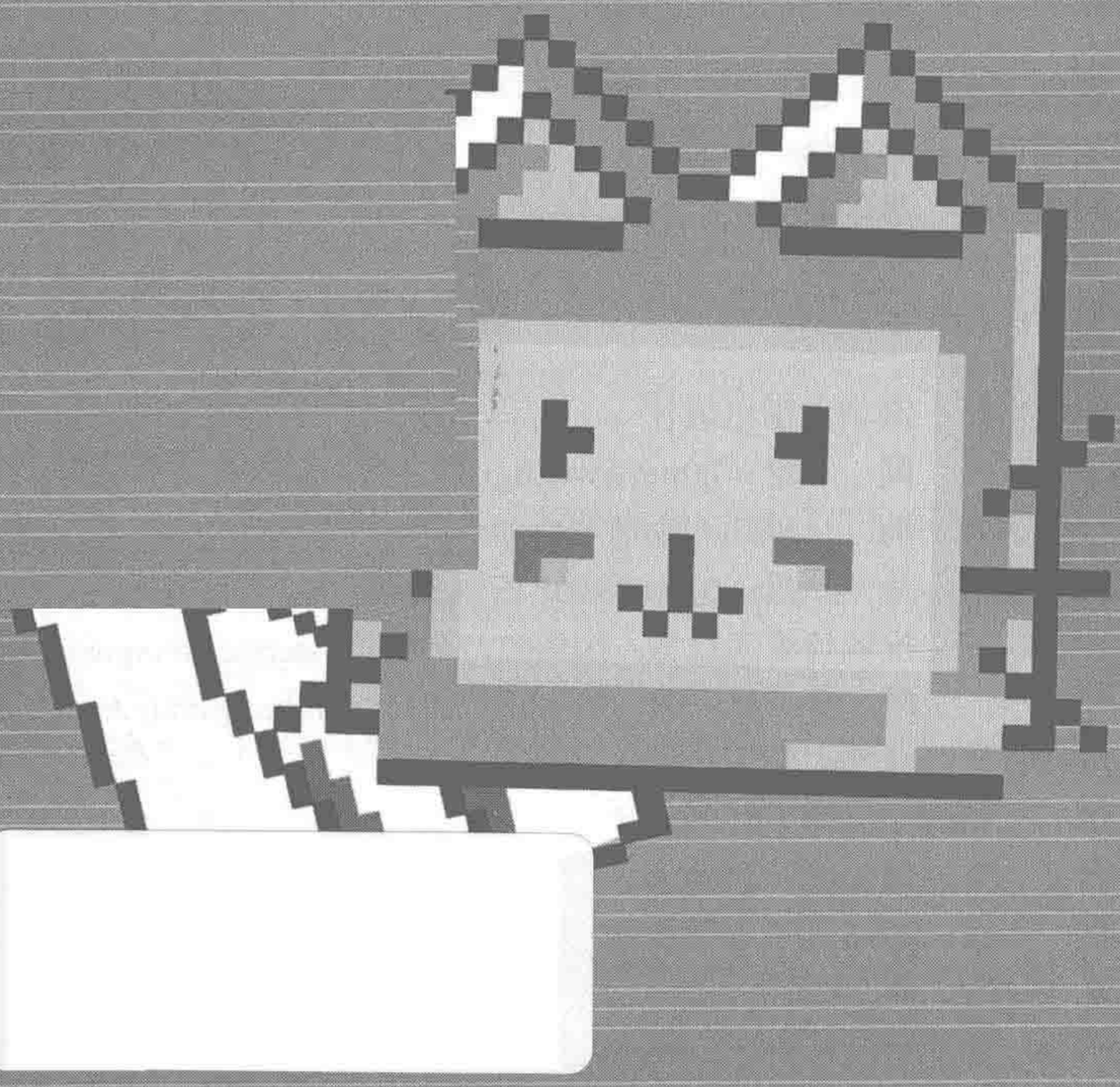


碁峰
www.gotop.com.tw

清华大学出版社

JDK 9 Java 学习笔记

林信良 编著



清华大学出版社

北京

内 容 简 介

本书是作者多年来教学实践经验的总结，汇集了学员在学习 Java 或认证考试时遇到的概念、操作、应用等问题及解决方案。

本书针对 Java SE 9 新功能全面改版，无论是章节架构或范例程序代码，都做了重新编写与全面翻新，并详细介绍了 Java 9 的模块化，JVM、JRE、Java SE API、JDK 与 IDE 之间的对照关系。必要时可从 Java SE API 的源代码分析，了解各种语法在 Java SE API 中如何应用。对于建议练习的范例提供了 Lab 文档，以突出练习重点。此外，本书还将 IDE 操作纳为教学内容之一，让读者能与实践相结合，轻松快速掌握 Java 编程技巧。

本书适合 Java 的初、中级读者以及广大 Java 应用开发人员阅读。

本书资料可通过 http://www.tupwk.com.cn/downpage 免费下载。

北京市版权局著作权合同登记号　图字：01-2018-3785

图书在版编目(CIP)数据

Java JDK 9 学习笔记 / 林信良编著. —北京：清华大学出版社，2018
ISBN 978-7-302-50118-3

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

中国版本图书馆 CIP 数据核字(2018)第 100148 号

责任编辑：王　定
封面设计：牛艳敏
版式设计：思创景点
责任校对：孔祥峰
责任印制：李红英

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　印　　张：36.75　　字　　数：941 千字
版　　次：2018 年 6 月第 1 版　　印　　次：2018 年 6 月第 1 次印刷
定　　价：98.00 元

产品编号：079536-01

序

当你拿起这本书，翻开这篇序，我便有了机会问你一个问题：“为什么想翻开这本书？”

“这个梗上一版的序用过啦！”

好吧！那学 Java 目的是什么呢？从事程序设计？那么我就有个新梗了：“什么是‘设计’呢？”

唔！好难回答的问题，来拜一下 Google 大神吧！搜索“什么是设计”，看完答案之后就更不懂了，在没有一个具体目标之前，得到的可能都是近乎空洞的答案吧！

试着在身边找个东西，比如键盘，你觉得设计得好吗？不好？哪里觉得不好？触感！什么样的是不好？反馈的力道！反馈是来自哪里？键轴！键轴是由哪些成分组成的呢？底座、弹簧、轴心、轴帽！造成反馈差异性的主要来源是什么？弹簧和轴心！喔？弹簧啊？它的圈数是多少呢？……

当你逐一挖掘出其中的元素之后，面对一个觉得设计不错的键盘，或许你就能知道其中有哪些“设计”了。

很多时候，当谈到一件东西设计得好或不好时，并不会明确地知道自己在讲什么，只是综合了各种感觉而得到的模糊结论。当然，在平时生活中，并不用每件事物都得探究到底，只是当某个事物是喜爱的、想赖以维生的，或者是两者综合，以为喜欢某个事物，因而想要进一步赖以维生，这个时候就不能只靠个模糊结论来搪塞下去了。

当你拿起这本书，表示选择了 Java 这门程序设计语言，为什么呢？因为 Java 可以写程序。可以写程序的语言很多啊！因为可以写手机 APP？那为什么不选 Objective-C 或 Swift 呢？因为听说业界很缺？喔！缺的是哪个工作性质的职位？手机……好吧！再问下去，可能有人只是被说服参加了三个月的 APP 补习课程，只好硬着头皮继续学下去了……

Java 本身是门程序设计语言，本身就有设计的成分在里头，基于面向对象典范，后来有了一些函数式典范的影子。Java 不是门简洁的语言，然而为了解决这方面问题，近来在语言设计上有了不少简化语法，把 Java 用在许多层面，Java 面对了为模块化制订标准的需求……

然而解决问题并不能只靠程序语言，面对不同的问题，需要设计各种流程，也有不同的数据结构设计。对于更复杂的问题，得靠更有效率的算法。当项目有一定规模，有弹性的架构设计是必要的。为了掌握程序的行为，就得设计可测试的程序；想要避免被入侵，必须将安全上的设计纳入考虑；当庞大的数据迎面而来，平行化方案的设计可能就得出现了……

设计是用来解决问题的！你喜欢或讨厌一种设计，代表它在解决问题上是否顺你的心、得你的意，从而认定一个设计是否优雅。

良葛格的 Java 学习笔记

既然已经选择了 Java，那表示你得接受它的设计了。那么接下来的问题，就在于怎么使用 Java 来表达你的设计了。不过，这也得真的有能够表达的设计，你有能表达的算法设计吗？数据结构设计？测试上的设计？架构上的设计？安全上的设计？平行处理的设计？……

过去、现在或未来，你写的程序中，真的有“设计”的成分在吗？

林信良

2017 年 12 月

导 读

这份导读让你可以更了解如何使用本书。

新旧版差异

就目录上来说，你可以看出的差异是，上一版为 18 个章节，新版为 19 个章节，第 19 章“深入模块化”是新的章节，也是 JDK9 最重要的新增功能。然而，认识模块化最好的方式是从实际的例子着手，因此第 1~18 章，全部的范例都是采用模块项目构建。而在各章说明时若有需要，也适时地带入了常用的模块化概念。第 19 章一开始则是整理前 18 个章节遇到过的模块化介绍，然后紧接着深入探讨模块化。

当然，照例要谈一些 JDK9 的其他新增功能，散落在各章节中适当的地方介绍。如果发现页侧有 图标，表示提及 JDK9 新功能，本书还提供了 JDK9 新功能快速查询目录。

虽然程序常用来处理计算，然而许多开发者对数字处理其实认识不多，因而第 15 章通用 API 增加了数字处理的内容；为了认识 JDK9 Stack-Walking API，读者有必要先认识如何取得并运用 StackTraceElement 来进行堆栈追踪，因而在第 15 章还增加了一个小节来介绍堆栈追踪，对于读者了解应用程序行为，或者是处理 Bug，应该会有所帮助。

在第 2 章介绍了模块化之后，范例项目便基于模块化设计了，并使用了 JDK9 的语法增强或改进部分程序代码；在 9.1.6 节介绍 Lambda 之后，为了提高可读性，使用 Lambda 相关语法或 API 来操作程序范例。

旧版有个附录 B“窗口程序设计”，在新版中删掉了，这表示了 Java 在窗口程序这块的地位。当然，Java 有 Java FX 这项技术，如果仍希望使用 Java 进行窗口程序设计，可以寻找 Java FX 的专书。附录 B 虽然不在了，不过范例项目的程序代码留着作为第 19 章的练习，读者可以自行研读原始码，并试着将之模块化。如果一定需要点说明，可以参考旧版《Java SE 6 技术手册》第 19 章说明：

- github.com/JustinSDK/JavaSE6Tutorial/blob/master/docs/CH19.md

字型

本书正文中与程序代码相关的文字，都用固定字体来加以呈现，以与一般名词相区别。例如，JDK 是一般名词，而 `String` 则是程序代码相关文字，所以使用了固定字体以区分。

程序范例

读者可以在以下网址下载本书的范例：

- http://www.tupwk.com.cn/downpage
- http://books.gotop.com.tw/v_ACL052100

本书许多范例都使用完整程序操作来展现，如看到以下程序代码示范时：

ClassObject Guess.java

```
package cc.openhome;

import java.util.Scanner; ←——❶ 告诉编译程序接下来想偷懒
import static java.lang.System.out;

public class Guess {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); ←——❷ 建立 Scanner 实例
        int number = (int) (Math.random() * 10);
        int guess;

        do {
            System.out.print("猜数字（0 ~ 9）:");
            guess = scanner.nextInt(); ←——❸ 取得下一个整数
        } while(guess != number);

        out.println("猜中了...XD");
    }
}
```

范例开始的左边名称为 ClassObject 表示可以在范例文件的 samples 文件夹的各章节文件夹中找到对应的 ClassObject 项目；而右边名称为 Guess.java 表示可以在项目中找到 Guess.java 文件。如果程序代码中出现标号与提示文字，表示在后续的正文中，会有对应于标号及提示的更详细说明。

原则上，建议读者每个项目范例都能亲手动作撰写，但出于教学时间或操作时间上的考虑，本书有建议进行的练习。如果在范例开始前有个Lab图标，例如：

Game1 SwordsMan.java

```
package cc.openhome;

public class SwordsMan extends Role {
    public void fight() {
        System.out.println("挥剑攻击");
    }
}
```

表示建议范例动手操作，而且在范例文件的 labs 文件夹中提供了练习项目的基础，打开项目后，完成项目中遗漏或必须补齐的程序代码或设定即可。

如果使用以下程序代码呈现，则表示它是一个完整的程序内容，而不是项目的一部分，这主要用来展现一个完整文档的撰写方法。

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

如果使用以下程序代码，则表示它是个代码段，主要展现程序撰写时需要特别注意的片段：

```
SwordsMan swordsMan = new SwordsMan();
...
System.out.printf("剑士 (%s, %d, %d)%n", swordsMan.getName(),
        swordsMan.getLevel(), swordsMan.getBlood());
Magician magician = new Magician();
...
System.out.printf("魔法师 (%s, %d, %d)%n", magician.getName(),
        magician.getLevel(), magician.getBlood());
```

提示框

在本书中会出现以下提示框：

提示 »» 针对课程中所提到的观点，提供了一些额外的资源或思考方向，暂时忽略这些提示对课程的影响，但有时间的话，针对这些提示做阅读、思考或讨论是有帮助的。

注意 »» 针对课程中所提到的观点，以提示框方式特别呈现出必须注意的一些使用方式、陷阱或避开问题的方法，看到这个提示框时请读者集中精神阅读。

附录

范例文件包括本书中的所有范例，提供 NetBeans 范例项目，附录 A 用于说明如何使用这些范例项目。

联系作者

若有本书勘误反馈等相关书籍问题，可通过网站与作者联系。网址如下：

http://openhome.cc

目　录

Java SE 9 新功能索引

良高格的Java学习笔记

Chapter 1

Java 平台概论

学习目标

- Java 版本迁移简介
- 认识 Java SE、Java EE、Java ME
- 认识 JDK 规范与操作
- 了解 JVM、JRE 与 JDK
- 下载与安装 JDK

1.1 Java不只是语言

从1995年至今，Java已经过了20多个年头，经过这些年的改进，正如本节标题所示，Java已不仅是个程序语言，也代表了解决问题的平台(Platform)，更代表了原厂、各个厂商、社群、开发者与用户沟通的成果。若仅以程序语言的角度来看待Java，正如冰山一角，你仅看到Java身为程序语言的一部分，而没看到Java身为程序语言之外，更可贵也更为庞大的资源。

1.1.1 前世今生

一个语言的诞生有其目的，因为这个目的而成就了该语言的主要特性。探索Java的历史演变，对于掌握Java特性与各种可用资源，有很大帮助。

1. Java诞生

Java最早是Sun公司绿色项目(Green Project)中撰写Star7应用程序的程序语言，当时的名称并不是Java，而是Oak。

绿色项目始于1990年12月，由Patrick Naughton、Mike Sheridan与James Gosling(James Gosling被尊称为Java之父)主持，目的是希望构筑出下一波计算机应用趋势并加以掌握，他们认为下一波计算机应用趋势会集中在消费性数字产品(就像现在的PDA、手机等消费性电子商品)的使用上。1992年9月3日，Green Team项目小组展示了Star7手持设备，这个设备具备无线网络连接、5寸LCD彩色屏幕、PCMCIA接口等功能，而Oak在绿色项目中的目的，是用来撰写Star7上应用程序的程序语言。

Oak名称的由来，是因为James Gosling的办公室窗外有一棵橡树(Oak)，就取了这个名称。但后来发现Oak已经被注册了，工程师们边喝咖啡边讨论着新名称，最后灵机一动而改名为Java。

Java本身有许多为了节省资源而作的设计，如动态加载类别文档、字符串池(String Pool)等特性，这是因为Java一开始就是为了消费性数字产品而设计，而这类小型装置通常有着有限内存与运算资源。

全球信息网(World Wide Web)兴起，Java Applet成为网页互动技术的代表。1993年第一个全球信息网浏览器Mosaic诞生，James Gosling认为互联网与Java的一些特性不谋而合，利用Java Applet在浏览器上展现互动性媒体，在当时而言，对视觉感官是一种革命性的颠覆。Green Team仿照Mosaic开发出以Java技术为基础的浏览器WebRunner(原名为BladeRunner)，后来改名为HotJava。虽然HotJava只是一个展示性产品，但它使用Java Applet展现的多媒体效果立即吸引了许多人的注意，图1.1所示即为JDK所附的Java Applet范例。

1995年5月23日(这一天被公认为Java的诞生日)，正式将Oak改名为Java，Java Development Kits(当时JDK的全名)1.0a2版本正式对外发表。到1996年，Netscape Navigator 2.0也正式支持Java，Microsoft Internet Explorer也开始支持Java。从此，Java在互联网的世界中逐渐风行起来。虽然Star7产品并不被当时消费性市场接受，绿色项目面临被裁撤的命运，然而全球信息网的兴起却给了Java新的生命与舞台。

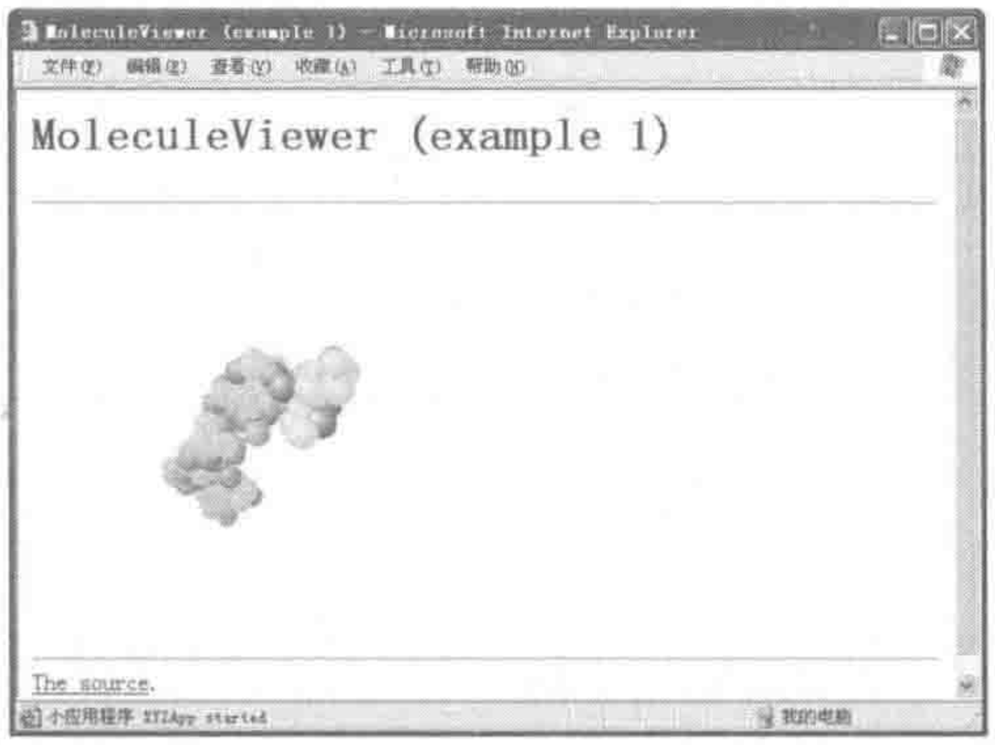


图 1.1　JDK 所附的 Java Applet 范例(JDK 文件夹\demo\applets\MoleculeViewer\example1.html)

2. 版本演进

随着 Java 越来越受到瞩目，Sun 在 1998 年 12 月 4 日发布 Java 2 Platform，简称 J2SE 1.2。Java 开发者版本一开始是以 Java Development Kit 名称发表，简称 JDK，而 J2SE 则是平台名称，包含了 JDK 与 Java 程序语言。

Java 平台标准版约以两年为周期推出重大版本更新，1998 年 12 月 4 日发表 J2SE 1.2，2000 年 5 月 8 日发表 J2SE 1.3，2002 年 2 月 13 日发表 J2SE 1.4，Java 2 这个名称也从 J2SE 1.2 开始沿用至各个版本。

2004 年 9 月 29 日发表的 Java 平台标准版的版号不是 1.5，而直接跳到 5.0，称为 J2SE 5.0，这是为了彰显这个版本与之前版本有极大不同，如语法上的简化、增加泛型(Generics)、枚举(Enum)、注释(Annotation)等重大功能。

2006 年 12 月 11 日发表的 Java 平台标准版，除了版本号之外，名称也有了变化，称为 Java Platform, Standard Edition 6，简称 Java SE 6。JDK6 全名则称为 Java SE Development Kit 6，也就是不再像以前 Java 2 带有 2 这个号码，版本号 6 或 1.6.0 都使用，6 是产品版本(Product Version)，而 1.6.0 是开发者版本(Developer Version)。

大部分的 Java 标准版平台都会取个代码名称(Code Name)，例如 J2SE 5.0 的代码名称为 Tiger(老虎)，为了引人注目，在发表会上还真的抱了一只小白老虎出来作为噱头，而许多书的封面也相应地放上老虎的图片。有关 JDK 代码名称与发布日期，如表 1.1 所示。

表 1.1　Java 版本、代码名称与发布日期

版　本	代 码 名 称	发 布 日 期
JDK 1.1.4	Sparkler(烟火)	1997/9/12
JDK 1.1.5	Pumpkin(南瓜)	1997/12/3
JDK 1.1.6	Abigail(圣经故事人物名称)	1998/4/24
JDK 1.1.7	Brutus(罗马政治家名称)	1998/9/28
JDK 1.1.8	Chelsea(足球俱乐部名称)	1999/4/8
J2SE 1.2	**Playground(游乐场)**	**1998/12/4**
J2SE 1.2.1	无	1999/3/30
J2SE 1.2.2	Cricket(蟋蟀)	1999/7/8
J2SE 1.3	**Kestrel(红隼)**	**2000/5/8**