

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!

Java

JDK 8

学习笔记

林信良 著

- 分享作者学习Java心得
- 涵盖OCP/JP(原SCJP)考试范围
- Lambda、新时间日期等Java SE 8新功能介绍
- JDK基础与IDE操作交相对应
- 提供Lab文档与操作教学视频



碁峯

www.gotop.com.tw

清华大学出版社

累计销量

10万册

JWor

Java权威

全新改版!

Java

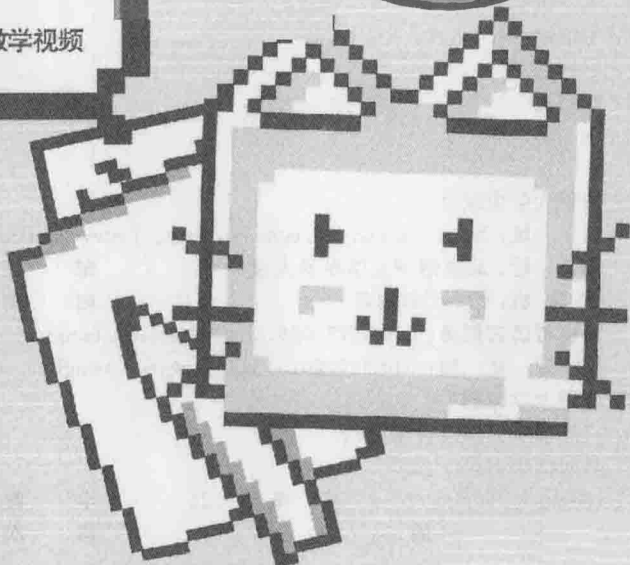
JDK 8

学习笔记

- 分享作者学习Java心得
- 涵盖OCP/JP(原SCJP)考试范围
- Java JDK 8新功能介绍
- JDK基础与IDE操作交相对应
- 提供Lab文档与操作教学视频



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,汇集了学生在学习 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 8 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 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,提供的教学视频可以让读者更清楚地掌握操作步骤。

本书适合 Java 的初、中级读者以及广大 Java 应用开发人员。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java JDK 8 学习笔记/林信良 著. —北京:清华大学出版社,2015

ISBN 978-7-302-38898-2

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

中国版本图书馆 CIP 数据核字(2015)第 004853 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:邱晓玉

责任印制:沈 露

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 刷 者:清华大学印刷厂

装 订 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:40.25

字 数:980 千字

版 次:2015 年 3 月第 1 版

印 次:2015 年 3 月第 1 次印刷

印 数:1~5000

定 价:68.00 元

序

你拿起了这本书，翻开这篇序，我有了机会问你一个问题：“为什么想翻开这本书？”

“当然是想学 Java 啊！笨蛋(作者)！”

翻开一本书，无非是想从书中得到知识，只是为何你要得到书中的知识，才是我想知道的答案，而这个答案决定了你在取得知识的过程中是否快乐！

多数人在取得知识的过程中并不快乐，因而只能幻想着取得知识之后，就能拥有快乐，我们的社会也不断塑造着这样的幻想……学会××之后就可以“找到工作”“年薪百万”“进外企”……不少人在完成买书或报名课程的那一瞬间，就仿佛看到童话故事结尾幸福又快乐的日子，甚至取得知识时花费越高，就越成为一种支持这自我满足的依据。

取得知识的过程中快乐很重要，可惜的是，多数教育并不将取得知识过程中是否快乐这件事摆在优先，甚至强调为了能拥有幸福又快乐的结局，你必须忍耐学习过程中种种不快乐的事情，有的人可能从没了解到取得知识的过程中也能够快乐这件事，也许你也早就忘了……

“不就是学个 Java，跟快乐有什么关系！”

如果你学的过程中不快乐，很快地，你就会对学习的对象感到厌烦，无法体验到逐步成长的喜悦，享受不到解决问题的成就感，失去探索更高级知识的动力，就算勉为其难地完成了学习过程，开始用着似懂非懂、半生不熟的知识闯荡江湖，紧接而来的是害怕着你当初不懂现在也不想搞懂的知识，也畏惧着别人带进来的新知识，只要有你参与的东西，多半掺杂了一团浆糊，造成了伙伴的困扰也伤害了自己，幸福与快乐的日子永远不会到来，你在学习的过程也没有过快乐，真的是亏大了！

只是想着学习的过程中是否快乐，结局难道不重要吗？网络上对程序设计这块有句名言“程序是照你写的跑，不是照你想的跑”，事实上确实是如此，不过“就人生来说的话，不会照你想的跑，也不会照你规划的进行”，万一结局不是我所想象的，至少学习过程我乐在其中，以后有没有用那就再说了！

“就人生来说的话，不会照你想的跑，也不会照你规划的进行。”简而言之就是世事难料，现在当红的技术难保日后不会没落，想当预言家，幻想能够选对一项知识，在苦痛学习过程之后得到美满结局，这是很没有保障的，现在冷门的知识也有可能咸鱼翻身，到时是不是感觉赚很大是一回事，不过届时你也许只会想着“其实我当时只是觉得好玩”！

林信良

2015 年 1 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 16 个章节，新一版为 18 个章节，第 12 章 Lambda 无疑是新的章节，也是 JDK8 最重要的新增功能。第 13 章“时间与日期”，一开始先谈了对时间与日期应有的基本知识，然后将旧版中的 Date 与 Calendar 做了更详细介绍，因为有许多现存 API 仍在使用它们，紧接着该章介绍了 JDK8 新时间与日期 API。

第 14 章“NIO 与 NIO2”一开始谈了 NIO 的基础，接着将旧版的 NIO2 也放进该章。第 15 章“通用 API”为旧版本“通用 API”，该章删除了 Date 与 Calendar、NIO2，并将一些 JDK8 的新增功能放了进去。

当然，JDK8 中还有不少新增的小功能，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 JDK8 新功能，本书亦提供有 JDK8 新功能快速查询目录。

全书的程序代码都做了重新审视与修改，主要着重在增加可读性，每个方法片段尽量控制不超过 15 行，在 9.1.6 节简介过 Lambda 之后，在可能且有助于可读性的情况下，会使用 Lambda 相关语法或 API 来实作程序范例。

旧版中有个“窗口程序设计”章节，在新版中没有消失，只不过被移至附录 B，这多半表示了 Java 在窗口程序这块的地位，当然，Java 有 JavaFX 这项技术，能否扩展 Java 在窗口程序的市场仍有待观察。“窗口程序设计”章节移至附录，主要是保留给对窗口程序仍有兴趣的读者。

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 为程序代码相关文字，使用了固定字体以区分。

程序范例

你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/download/Acl042200>

本书许多的范例都使用完整程序操作来展现，当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，表示可以在项目找到 Guess.java 文件。如果程序代码中出现标号与提示文字，表示后续的正文中，会有对应于标号及提示的更详细说明。

原则上，建议每个项目范例都亲自动作撰写，但如果由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  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 文件夹中会有练习项目的基础，可以打开项目后，完成项目中遗漏或必须补齐的程序代码或设定。

如果使用以下的程序代码呈现，表示它是一个完整的程序内容，但不是项目的一部分，主要用来展现一个完整文档如何撰写：

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

如果使用以下的程序代码，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());
```

操作步骤

本书将 IDE 进行设定的相关操作步骤也作为练习的一部分，你会看到如下的操作步骤说明：

(1) 选择“文件”|“新建项目”命令，在弹出的“新建项目”对话框的“类别”列表中选择 Java，在“项目”列表中选择“Java 应用程序”，接着单击“下一步”按钮。

(2) 在“项目名称”文本框中输入项目名称 Hello2，在“项目位置”文本框中输入 C:\workspace。注意，“项目文件夹”会储存至 C:\workspace\Hello2。

(3) 在“创建主类”文本框中输入 cc.openhome.Main，这表示会有个 Main 类放在 cc.openhome 包，当中会自动建立 main() 程序进入点的方法，接着单击“完成”按钮建立项目。

如果操作步骤旁有个  图标，表示范例文件的 videos 文件夹中对应的章节文件夹有操作步骤的视频，可观看它以更了解实际操作过程。

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，提供一些额外的资源或思考方向，暂时忽略这些提示对课程进行并没有影响，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集中精神阅读。

附录

范例文件包括本书中所有范例，提供 NetBeans 范例项目，附录 A 说明如何使用这些范例项目，附录 B 则收录了上一版的“窗口程序设计”一章的内容。

联系作者

若有本书堪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

Chapter 1	Java 平台概论	1
1.1	Java 不只是语言	2
1.1.1	前世今生	2
1.1.2	三大平台	5
1.1.3	JCP 与 JSR	6
1.1.4	Oracle JDK 与 OpenJDK	7
1.1.5	建议的学习路径	9
1.2	JVM/JRE/JDK	12
1.2.1	什么是 JVM	13
1.2.2	区分 JRE 与 JDK	15
1.2.3	下载、安装 JDK	16
1.2.4	认识 JDK 安装内容	19
1.3	重点复习	20
1.4	课后练习	21
Chapter 2	从 JDK 到 IDE	22
2.1	从 Hello World 开始	23
2.1.1	撰写 Java 原始码	23
2.1.2	PATH 是什么	25
2.1.3	JVM(java)与 CLASSPATH	28
2.1.4	编译程序(javac)与 CLASSPATH	31
2.2	管理原始码与位码文档	33
2.2.1	编译程序(javac)与 SOURCEPATH	33
2.2.2	使用 package 管理类	35
2.2.3	使用 import 偷懒	37
2.3	使用 IDE	39
2.3.1	IDE 项目管理基础	40
2.3.2	使用了哪个 JRE	44
2.3.3	类文档版本	45
2.4	重点复习	48
2.5	课后练习	49
Chapter 3	基础语法	53
3.1	类型、变量与运算符	54
3.1.1	类型	54
3.1.2	变量	57
3.1.3	运算符	60
3.1.4	类型转换	66
3.2	流程控制	69
3.2.1	if...else 条件式	69
3.2.2	switch 条件式	72
3.2.3	for 循环	74
3.2.4	while 循环	75
3.2.5	break、continue	77
3.3	重点复习	78
3.4	课后练习	79
3.4.1	选择题	79
3.4.2	操作题	81
Chapter 4	认识对象	82
4.1	类与对象	83
4.1.1	定义类	83
4.1.2	使用标准类	86
4.1.3	对象指定与相等性	89
4.2	基本类型打包器	90
4.2.1	打包基本类型	90
4.2.2	自动装箱、拆箱	91
4.2.3	自动装箱、拆箱的内幕	92
4.3	数组对象	95
4.3.1	数组基础	95
4.3.2	操作数组对象	98
4.3.3	数组复制	104
4.4	字符串对象	107
4.4.1	字符串基础	107
4.4.2	字符串特性	110
4.4.3	字符串编码	113
4.5	查询 Java API 文件	115
4.6	重点复习	118
4.7	课后练习	119
4.7.1	选择题	119
4.7.2	操作题	121

Chapter 5 对象封装 122

5.1 何谓封装 123

5.1.1 封装对象初始流程 123

5.1.2 封装对象操作流程 125

5.1.3 封装对象内部数据 128

5.2 类语法细节 131

5.2.1 public 权限修饰 131

5.2.2 关于构造函数 133

5.2.3 构造函数与方法重载 134

5.2.4 使用 this 136

5.2.5 static 类成员 139

5.2.6 不定长度自变量 145

5.2.7 内部类 146

5.2.8 传值调用 148

5.3 重点复习 151

5.4 课后练习 152

5.4.1 选择题 152

5.4.2 操作题 155

Chapter 6 继承与多态 157

6.1 何谓继承 158

6.1.1 继承共同行为 158

6.1.2 多态与 is-a 162

6.1.3 重新定义行为 166

6.1.4 抽象方法、抽象类 169

6.2 继承语法细节 170

6.2.1 protected 成员 170

6.2.2 重新定义的细节 172

6.2.3 再看构造函数 174

6.2.4 再看 final 关键字 176

6.2.5 java.lang.Object 178

6.2.6 关于垃圾收集 182

6.2.7 再看抽象类 185

6.3 重点复习 188

6.4 课后练习 189

6.4.1 选择题 189

6.4.2 操作题 193

Chapter 7 接口与多态 194

7.1 何谓接口 195

7.1.1 接口定义行为 195

7.1.2 行为的多态 199

7.1.3 解决需求变化 202

7.2 接口语法细节 208

7.2.1 接口的默认 208

7.2.2 匿名内部类 212

7.2.3 使用 enum 枚举常数 217

7.3 重点复习 219

7.4 课后练习 220

7.4.1 选择题 220

7.4.2 操作题 224

Chapter 8 异常处理 226

8.1 语法与继承架构 227

8.1.1 使用 try、catch 227

8.1.2 异常继承架构 230

8.1.3 要抓还是要抛 235

8.1.4 贴心还是造成麻烦 238

8.1.5 认识堆栈追踪 240

8.1.6 关于 assert 244

8.2 异常与资源管理 247

8.2.1 使用 finally 247

8.2.2 自动尝试关闭资源 249

8.2.3 java.lang.AutoCloseable 接口 251

8.3 重点复习 256

8.4 课后练习 257

8.4.1 选择题 257

8.4.2 操作题 261

Chapter 9 Collection 与 Map 262

9.1 使用 Collection 收集对象 263

9.1.1 认识 Collection 架构 263

9.1.2 具有索引的 List 264

9.1.3 内容不重复的 Set 268

9.1.4 支持队列操作的 Queue 272

9.1.5 使用泛型 275

9.1.6 简介 Lambda 表达式 279

9.1.7 Iterable 与 Iterator 281

9.1.8 Comparable 与 Comparator 284

9.2 键值对应的 Map	290	Chapter 12 Lambda	382
9.2.1 常用 Map 操作类	291	12.1 认识 Lambda 语法	383
9.2.2 访问 Map 键值	295	12.1.1 Lambda 语法概览	383
9.3 重点复习	298	12.1.2 Lambda 表达式与函数 接口	386
9.4 课后练习	299	12.1.3 Lambda 遇上 this 与 final	388
9.4.1 选择题	299	12.1.4 方法与构造函数 参考	391
9.4.2 操作题	303	12.1.5 接口默认方法	394
Chapter 10 输入/输出	304	12.2 Functional 与 Stream API	399
10.1 InputStream 与 OutputStream	305	12.2.1 使用 Optional 取代 null	399
10.1.1 串流设计的概念	305	12.2.2 标准 API 的函数接口	401
10.1.2 串流继承架构	308	12.2.3 使用 Stream 进行管道 操作	404
10.1.3 串流处理装饰器	311	12.2.4 进行 Stream 的 reduce 与 collect	408
10.2 字符处理类	316	12.2.5 关于 flatMap() 方法	413
10.2.1 Reader 与 Writer 继承 架构	316	12.3 Lambda 与并行处理	416
10.2.2 字符处理装饰器	318	12.3.1 Stream 与平行化	416
10.3 重点复习	320	12.3.2 使用 Completable- Future	420
10.4 课后练习	321	12.4 重点复习	423
10.4.1 选择题	321	12.5 课后练习	424
10.4.2 操作题	322	Chapter 13 时间与日期	425
Chapter 11 线程与并行 API	324	13.1 认识时间与日期	426
11.1 线程	325	13.1.1 时间的度量	426
11.1.1 线程简介	325	13.1.2 年历简介	427
11.1.2 Thread 与 Runnable	328	13.1.3 认识时区	428
11.1.3 线程生命周期	329	13.2 认识 Date 与 Calendar	429
11.1.4 关于 ThreadGroup	336	13.2.1 时间轴上瞬间的 Date	429
11.1.5 synchronized 与 volatile	339	13.2.2 格式化时间日期的 DateFormat	430
11.1.6 等待与通知	349	13.2.3 处理时间日期的 Calendar	433
11.2 并行 API	353	13.2.4 设定 TimeZone	436
11.2.1 Lock、ReadWriteLock 与 Condition	354	13.3 JDK8 新时间日期 API	437
11.2.2 使用 Executor	364		
11.2.3 并行 Collection 简介	375		
11.3 重点复习	379		
11.4 课后练习	380		
11.4.1 选择题	380		
11.4.2 操作题	381		

13.3.1	机器时间观点的 API	437	15.5	重点复习	496
13.3.2	人类时间观点的 API	439	15.6	课后练习	497
13.3.3	对时间的运算	441	15.6.1	选择题	497
13.3.4	年历系统设计	444	15.6.2	操作题	497
13.4	重点复习	445	Chapter 16	整合数据库	498
13.5	课后练习	446	16.1	JDBC 入门	499
Chapter 14	NIO 与 NIO2	447	16.1.1	JDBC 简介	499
14.1	认识 NIO	448	16.1.2	连接数据库	503
14.1.1	NIO 概述	448	16.1.3	使用 Statement、 ResultSet	509
14.1.2	Channel 架构与操作	449	16.1.4	使用 PreparedStatement、 CallableStatement	514
14.1.3	Buffer 架构与操作	450	16.2	JDBC 进阶	518
14.2	NIO2 文件系统	452	16.2.1	使用 DataSource 取得 联机	518
14.2.1	NIO2 架构	453	16.2.2	使用 ResultSet 卷动、 更新数据	522
14.2.2	操作路径	454	16.2.3	批次更新	524
14.2.3	属性读取与设定	456	16.2.4	Blob 与 Clob	526
14.2.4	操作文档与目录	459	16.2.5	交易简介	526
14.2.5	读取、访问目录	461	16.2.6	metadata 简介	534
14.2.6	过滤、搜索文档	466	16.2.7	RowSet 简介	537
14.3	重点复习	468	16.3	重点复习	542
14.4	课后练习	469	16.4	课后练习	543
Chapter 15	通用 API	470	16.4.1	选择题	543
15.1	日志	471	16.4.2	操作题	544
15.1.1	日志 API 简介	471	Chapter 17	反射与类加载器	545
15.1.2	指定日志层级	473	17.1	运用反射	546
15.1.3	使用 Handler 与 Formatter	475	17.1.1	Class 与 class 文档	546
15.1.4	自定义 Handler、 Formatter 与 Filter	476	17.1.2	使用 Class.forName()	548
15.1.5	使用 logging.properties	478	17.1.3	从 Class 获得信息	550
15.2	国际化基础	480	17.1.4	从 Class 建立对象	553
15.2.1	使用 ResourceBundle	480	17.1.5	操作对象方法与成员	556
15.2.2	使用 Locale	481	17.1.6	动态代理	558
15.3	规则表示式	483	17.2	了解类加载器	562
15.3.1	规则表示式简介	483	17.2.1	类加载器层级架构	562
15.3.2	Pattern 与 Matcher	491	17.2.2	建立 ClassLoader 实例	565
15.4	JDK8 API 增强功能	494			
15.4.1	StringJoiner、Arrays 新增 API	494			
15.4.2	Stream 相关 API	495			

17.3	重点复习	567	18.4	重点复习	599
17.4	课后练习	568	18.5	课后练习	600
17.4.1	选择题	568	Appendix A	如何使用本书项目	601
17.4.2	操作题	568	A.1	项目环境配置	602
Chapter 18	自定义泛型、枚举与 注释	569	A.2	打开案例	602
18.1	自定义泛型	570	Appendix B	窗口程序设计	603
18.1.1	使用 extends 与?	570	B.1	Swing 入门	604
18.1.2	使用 super 与?	575	B.1.1	简易需求分析	604
18.2	自定义枚举	578	B.1.2	Swing 组件简介	605
18.2.1	了解 java.lang. Enum 类	578	B.1.3	设计主窗口与菜单列	607
18.2.2	enum 高级运用	581	B.1.4	关于版面管理	612
18.3	关于注释	587	B.1.5	事件处理	615
18.3.1	常用标准注释	587	B.2	文档打开、存储与编辑	620
18.3.2	自定义注释类型	590	B.2.1	操作打开文档	620
18.3.3	JDK8 标注增强功能	594	B.2.2	制作存储、关闭文档	623
18.3.4	执行时期读取注释 信息	596	B.2.3	文字区编辑、剪切、 复制、粘贴	626

Java SE 8 新功能

Unicode 6.2.0	3-3
匿名类捕捉了等效 <code>final</code> 的局部变量, <code>final</code> 可省略	7-26
简介 Lambda 表示式	9-21
Iterable 的 <code>forEach()</code>	9-27
高级排序 API	9-33
Map 的 <code>forEach()</code> 方法	9-42
UncheckedIOException	10-3
StampedLock	11-41
Lambda 语法	12-2
@FunctionalInterface	12-8
方法与构造函数参考	12-12
接口默认方法	12-15
Functional 与 Stream API	12-21
CompletableFuture	12-46
新时间日期 API	13-15
Files 的 <code>lines()</code>	14-19
Files 的 <code>list()</code> 与 <code>walk()</code>	14-24
Logger 增加接受 <code>Supplier</code> 的重载方法	15-6
Pattern 的 <code>splitAsStream()</code>	15-29
StringJoiner、String.join()	15-29
Arrays 的 <code>parallelPrefix()</code> 、 <code>parallelSetAll()</code> 、 <code>parallelSort()</code>	15-30
Stream 相关 API	15-31
TimeStamp 的 <code>toInstant()</code> 与 <code>from()</code>	16-25
标注增强	18-29
AnnotatedElement 的 <code>getDeclaredAnnotation()</code> 、 <code>getDeclaredAnnotationsByType()</code> 、 <code>getAnnotationsByType()</code>	18-34

Java 平台概论

Chapter

1

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 20 个年头, 经过这些年的演进, 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语言的主要特性。探索 Java 的历史演进, 对于掌握 Java 特性与各式可用资源, 着实有其帮助。

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, 当时名称不是 Java, 而是取名为 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、手机等消费性电子产品)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、5inLCD 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), 就顺手取了这个名称。但后来发现 Oak 名称已经被注册了, 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 本身会见到许多为了节省资源而作的设计, 像是动态加载类别文档、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 认为因特网与 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, 如图 1.1 所示。

1995 年 5 月 23 日(这一天被公认为 Java 的誕生日), 正式将 Oak 改名为 Java, Java Development Kits(当时 JDK 全名)1.0a2 版本正式对外发表, 而在 1996 年 Netscape Navigator 2.0 也正式支持 Java, Microsoft Internet Explorer 也开始支持 Java, 从此 Java 在因特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。