

《Java 7入门经典》

图书基本信息

书名：《Java 7入门经典》

13位ISBN编号：9787302289593

10位ISBN编号：730228959X

出版时间：2012-7-1

出版社：清华大学出版社

作者：(美) 霍尔顿(Horton, I.)

页数：1124

译者：梁峰

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

前言

欢迎阅读《Java 7 入门经典》，本书通俗易懂，是学习Java语言以及Java平台API的综合性入门指南。本书提供了在Java JDK 7或更高版本环境下进行程序开发的一些基础知识。本书除了教授Java知识外，还会介绍与Java程序员有关的各种主题。本书的组织结构经过了仔细设计，符合思维逻辑，让Java编程学习的各个阶段都环环相扣。本书读者对象 Java编程应用范围广泛，而且随着语言本身以及关联的函数库的增长，Java的应用领域仍然在不断扩展。自从发布以来，Java作为一门面向对象的语言已成为Internet编程、跨平台应用以及教学中的首选语言。在笔者看来，这其中有三个原因：Java与生俱来的系统无关的特性、Java语言的简单性和强大，以及Java作为工具可以有效并轻松实现的编程任务的广泛性。对于主要的应用程序开发，Java是许多程序员的选择。与其他语言相比，Java能提供便捷的开发和维护优势。此外，还有在各种计算机和操作系统中不用修改代码即可运行的内在能力。使用Java可以做到更多、更快、更容易。本书旨在提供对于Java语言的全面理解以及在一些Java应用程序上下文环境中进行编程的经验，希望读者能在其中的每个核心领域都打下坚实的基础。书中介绍的Java各方面知识都通过实际的例子进行解释和说明，这些例子也可以自己创建并运行，而且也应该自己进行试验。每章末尾的练习有助于对所学知识进行尝试。在理解本书介绍的主题内容之后，就能开始编写有特色而且有效果的Java程序了。书名中的“入门”更多是指本书的编写风格而不是指读者的能力水平，所以也可以称为“直通Java”，这是因为本书的结构设计合理，不论是已有其他语言编程经验的程序员还是刚入门的新手，本书都适合阅读。本书假定您至少有一点编程的基础，比如应该至少理解程序运行的基本概念。但是在阅读本书之前并不需要太多的预备知识。本书讲解的进度虽然有些快，但却覆盖了对Java运行工作机制的所有必要解释。

《Java 7入门经典》

内容概要

无论学习Java是为了编写嵌入网页的定制applet，还是为了编写大型应用程序，本书都十分适合阅读。本书是关于Java语言讲解最为细致、全面的入门书籍，介绍了使用最新的Java JDK 7开发程序所需要的所有基础知识。书中包含大量的示例，清晰地解释了涉及的关键概念，演示了Java开发的复杂细节。在阅读各个章节的过程中，您还能获得无价的编程经验，并逐步开始编写功能全面的Java程序。Ivor Horton在编写入门类编程图书方面独具心得，著作深受好评。Java SE 7相对之前发布版本更新了超过30%的语言特性，本书从基础知识入门，介绍了使用这一最新发布版本编写Java程序的方方面面，涵盖了Java的所有语言新特性，可为读者打下坚实的基础。

《Java 7入门经典》

作者简介

Ivor Horton是Java、C和C++编程语言方面的杰出作者，由他执笔的图书都非常有名，如《Visual C++ 2010入门经典(第5版)》、《C语言入门经典(第4版)》、《C++入门经典(第3版)》等。他编写的图书十分适合初学者学习，讲解细腻、全面，示例丰富，深受读者好评。Ivor Horton还是私人实践方面的系统顾问。

书籍目录

第1章 Java简介

1

1.1 Java概览

1

1.2 Java语言的特性

2

1.3 学习Java

2

1.3.1 Java程序

3

1.3.2 Java学习路线图

3

1.4 Java环境

4

1.5 Java中的面向对象编程

11

1.5.1 什么是对象

12

1.5.2 如何定义对象类

13

1.5.3 对象操作

15

1.5.4 Java程序语句

17

1.5.5 封装

17

1.5.6 类与数据类型

18

1.5.7 类与子类

18

1.5.8 使用对象的优势

18

1.5.9 标记

18

1.5.10 泛型类

19

1.6 Java程序结构

19

1.6.1 Java的类库

19

1.6.2 Java应用程序

21

1.7 Java和Unicode

23

1.8 小结

24

1.9 资源

25	
第2章 程序、数据、变量和计算	
27	
2.1 数据和变量	
27	
2.1.1 变量的命名	
28	
2.1.2 变量名与Unicode	
28	
2.1.3 变量与类型	
29	
2.2 整数数据类型	
29	
2.3 浮点数数据类型	
33	
2.3.1 浮点数字面量	
33	
2.3.2 声明浮点型变量	
34	
2.4 固定变量的值	
34	
2.5 算术运算	
35	
2.5.1 整数计算	
35	
2.5.2 整数除法和余数	
40	
2.5.3 增量与减量运算符	
40	
2.5.4 短整数类型的计算	
42	
2.5.5 整数算术中的错误	
43	
2.5.6 浮点计算	
43	
2.5.7 其他浮点算术运算符	
44	
2.5.8 浮点算术中的错误情况	
44	
2.5.9 混合算术表达式	
45	
2.5.10 显式转换	
45	
2.5.11 赋值中的自动类型转换	
46	
2.6 op=运算符	
46	
2.7 数学函数和常量	
47	

2.8 存储字符	51
2.8.1 字符转义序列	51
2.8.2 字符算术	52
2.9 位运算	54
2.9.1 使用AND和OR运算符	55
2.9.2 使用异或运算符	58
2.9.3 位移操作	59
2.9.4 位操作方法	62
2.10 取值范围为固定整数值集合的变量	64
2.11 布尔变量	66
2.12 运算符的优先级	67
2.13 程序注释	68
2.14 小结	70
第3章 循环与逻辑	73
3.1 决策	73
3.1.1 比较	74
3.1.2 if语句	74
3.1.3 嵌套的if语句	78
3.1.4 比较枚举值	81
3.2 逻辑运算符	81
3.2.1 逻辑与操作	82
3.2.2 逻辑或操作	84
3.2.3 异或操作	84
3.2.4 布尔非操作	84

3.2.5 使用标准库方法测试字符	85
3.3 条件运算符	86
3.4 switch语句	88
3.5 变量的作用域	94
3.6 循环	96
3.6.1 循环的种类	97
3.6.2 使用浮点值计数	103
3.6.3 嵌套循环	103
3.6.4 continue语句	105
3.6.5 带标签的continue语句	105
3.6.6 在循环中使用break语句	107
3.7 断言	112
3.8 小结	114
第4章 数组与字符串	117
4.1 数组	117
4.1.1 数组变量	118
4.1.2 定义数组	118
4.1.3 数组长度	119
4.1.4 访问数组元素	119
4.1.5 重用数组变量	119
4.1.6 初始化数组	120
4.1.7 使用数组	123
4.1.8 二维数组	126
4.1.9 字符数组	130
4.2 字符串	

131	
4.2.1	字符串字面量
131	
4.2.2	创建String对象
132	
4.2.3	字符串数组
134	
4.3	字符串操作
135	
4.3.1	连接字符串
135	
4.3.2	比较字符串
139	
4.3.3	字符串排序
144	
4.3.4	访问字符串中的字符
146	
4.3.5	在字符串中查找字符
148	
4.3.6	查找子字符串
149	
4.3.7	提取子字符串
152	
4.3.8	String对象的修改版本
156	
4.3.9	从String对象创建字符 数组
157	
4.3.10	使用字符串执行基于集合 的for循环
157	
4.3.11	在字符串中获取字节数组 形式的字符
158	
4.3.12	从字符数组中创建String 对象
158	
4.4	可变字符串
159	
4.4.1	创建StringBuffer对象
160	
4.4.2	StringBuffer对象的容量
160	
4.4.3	为StringBuffer对象修改字 符串的长度
162	
4.4.4	增长StringBuffer对象
162	
4.4.5	寻找子字符串的位置

164	
4.4.6	替换缓冲区中的子字符串
165	
4.4.7	插入字符串
165	
4.4.8	从可变字符串中提取字符
166	
4.4.9	可变字符串的其他操作
167	
4.4.10	从StringBuffer对象创建String对象
167	
4.5	小结
169	
第5章	定义类
173	
5.1	类的定义
173	
5.1.1	类定义中的域
174	
5.1.2	类定义中的方法
175	
5.1.3	访问变量和方法
176	
5.1.4	Final域
177	
5.2	定义类
177	
5.3	定义方法
178	
5.3.1	方法的返回值
179	
5.3.2	参数列表
179	
5.3.3	定义类方法
181	
5.3.4	访问方法中的类数据成员
182	
5.3.5	变量this
182	
5.3.6	初始化数据成员
183	
5.4	构造函数
186	
5.4.1	默认构造函数
187	
5.4.2	创建类的对象
188	
5.5	定义和使用类

190
5.6 方法重载
192
5.6.1 多个构造函数
193
5.6.2 使用构造函数复制对象
196
5.7 使用对象
196
5.8 递归
202
5.9 理解包
204
5.9.1 对类打包
204
5.9.2 将类从包添加到程序中
207
5.9.3 程序中的包和名称
208
5.9.4 导入静态类成员
208
5.9.5 标准包
209
5.10 类成员的访问控制
212
5.10.1 使用访问属性
212
5.10.2 设定访问属性
214
5.10.3 选择访问属性
215
5.11 嵌套类
220
5.11.1 静态嵌套类
221
5.11.2 使用非静态嵌套类
225
5.11.3 使用非顶级类的嵌套类
227
5.11.4 本地嵌套类
228
5.12 小结
228
第6章 扩展类与继承
231
6.1 使用已有的类
231
6.2 类继承
233

6.2.1 继承数据成员	233
6.2.2 继承方法	234
6.2.3 覆盖基类方法	237
6.3 @Override标记	239
6.4 选择基类访问属性	239
6.5 多态	240
6.6 多级继承	246
6.7 抽象类	246
6.8 通用超类	247
6.8.1 toString()方法	249
6.8.2 判定对象的类型	249
6.8.3 复制对象	250
6.9 接受可变数目参数的方法	255
6.10 转换对象	256
6.10.1 转换对象的时机	258
6.10.2 识别对象	259
6.11 枚举进阶	260
6.12 设计类	264
6.13 使用final修饰符	274
6.14 接口	275
6.14.1 在程序中封装常量	275
6.14.2 用接口声明方法	278
6.14.3 扩展接口	281
6.14.4 使用接口	282
6.14.5 将接口类型作为方法的	

参数使用

288

6.14.6 在接口定义中嵌套类

289

6.14.7 接口与真实环境

289

6.15 匿名类

289

6.16 小结

290

第7章 异常

293

7.1 异常的基本思想

293

7.2 异常类型

294

7.2.1 Error类型的异常

295

7.2.2 RuntimeException类型的异常

295

7.2.3 Exception类的其他子类

295

7.3 处理异常

295

7.3.1 设定方法能够抛出的异常

296

7.3.2 处理异常

296

7.3.3 try代码块

296

7.3.4 catch代码块

297

7.3.5 在一个代码块中捕获多种异常类型

301

7.3.6 finally代码块

301

7.3.7 构造方法

302

7.3.8 执行顺序

303

7.3.9 嵌套的try代码块

309

7.3.10 重新抛出异常

309

7.4 异常对象

310

7.4.1 Throwable类

310	
7.4.2	标准异常
313	
7.5	定义自己的异常
313	
7.5.1	定义异常类
313	
7.5.2	抛出自己的异常
315	
7.5.3	异常的抛出策略
315	
7.6	小结
320	
第8章	理解流
323	
8.1	流与输入输出操作
323	
8.2	流的概念
324	
8.2.1	输入流与输出流
324	
8.2.2	二进制流与字符流
325	
8.3	输入输出类
326	
8.3.1	基本的输入流操作
327	
8.3.2	缓冲输入流
327	
8.3.3	基本的输出流操作
328	
8.3.4	流读取器和编写器
329	
8.4	标准流
332	
8.4.1	从键盘读入数据
333	
8.4.2	写到命令行中
340	
8.4.3	printf()方法
340	
8.4.4	将数据格式化为字符串
345	
8.5	小结
346	
第9章	访问文件与目录
349	
9.1	访问文件系统
349	

9.2 使用Path对象	351
9.2.1 访问系统属性	353
9.2.2 设置系统属性	355
9.2.3 测试和检查Path对象	355
9.2.4 查询文件和目录	356
9.2.5 获取文件属性	357
9.2.6 其他的路径操作	360
9.3 创建与删除目录和文件	361
9.3.1 创建目录	361
9.3.2 创建文件	363
9.3.3 删除文件和目录	364
9.4 获取目录内容	365
9.5 关闭流	366
9.6 移动与复制文件和目录	368
9.6.1 对文件或目录重命名	369
9.6.2 复制文件和目录	370
9.6.3 遍历文件树	375
9.7 小结	378
第10章 写文件	381
10.1 文件I/O基础	381
10.2 文件输出	382
10.3 通过输出流写文件	383
10.4 使用Writer写文件	386
10.5 缓冲区	389
10.5.1 缓冲区的容量	

389	
10.5.2	缓冲区的位置和限制
390	
10.5.3	设置位置和限制
391	
10.5.4	创建缓冲区
392	
10.5.5	标记缓冲区
396	
10.5.6	缓冲区数据传输
397	
10.5.7	使用视图缓冲区
399	
10.5.8	准备缓冲区以输出到文件中
400	
10.6	使用通道写文件
401	
10.6.1	通道接口
402	
10.6.2	通道操作
403	
10.6.3	为文件获取通道
403	
10.6.4	通道写操作
404	
10.7	文件写操作
406	
10.7.1	将缓冲区的一部分写入到文件中
406	
10.7.2	文件的位置
409	
10.7.3	使用视图缓冲区加载字节缓冲区
410	
10.7.4	将变长字符串写入到文件中
410	
10.7.5	直接和间接缓冲区
416	
10.7.6	使用通道写入数值数据
417	
10.7.7	将混合数据写入到文件中
421	
10.7.8	从多个缓冲区写入
427	

10.8 强制将数据写入到设备中	430
10.9 小结	431
第11章 读文件	433
11.1 文件读取操作	433
11.1.1 使用输入流读取文件	434
11.1.2 使用缓冲读取器读取文件	437
11.2 使用通道读取文件	439
11.2.1 读取文本文件	441
11.2.2 从缓冲区获取数据	442
11.2.3 读取二进制数据	444
11.2.4 读取混合数据	446
11.2.5 压缩缓冲区	449
11.3 复制文件	452
11.4 随机访问文件	455
11.5 内存映射文件	460
11.5.1 锁住文件	463
11.5.2 锁住文件的特定部分	465
11.5.3 实用的文件加锁考量	465
11.6 小结	468
第12章 序列化对象	471
12.1 在文件中存储对象	471
12.2 将对象写入到文件中	472
12.2.1 将基本类型数据写入到对象流中	474
12.2.2 实现Serializable接口	

475	
12.3	从文件读取对象
478	
12.3.1	判定反序列化对象所属的类
481	
12.3.2	使用对象序列化
482	
12.3.3	手动序列化类
485	
12.3.4	序列化问题和困难
486	
12.4	小结
488	
	第13章 泛型
491	
13.1	什么是泛型
491	
13.2	定义泛型类
492	
13.2.1	实现泛型类
493	
13.2.2	实例化泛型类
495	
13.2.3	泛型类实例在运行时的类型
500	
13.2.4	泛型类实例之间的关系
502	
13.2.5	多个类型参数
502	
13.2.6	类型参数的作用域
503	
13.2.7	泛型类的静态域
504	
13.2.8	类型参数的边界
504	
13.3	泛型类和泛型接口
508	
13.3.1	实现针对容器类的基于集合的循环
508	
13.3.2	实现迭代器的功能
510	
13.3.3	二叉树的参数化类型
512	
13.4	原生类型的变量
523	

13.5 使用通配符类型参数	524
13.5.1 针对通配符的约束	526
13.5.2 深入Class类	529
13.6 数组与参数化类型	530
13.7 参数化方法	533
13.8 参数化类型及继承	539
13.9 总结	540
第14章 集合框架	543
14.1 理解集合框架	543
14.2 对象集合	544
14.2.1 集	544
14.2.2 序列	545
14.2.3 地图	546
14.3 迭代器	547
14.4 集合类	549
14.5 使用EnumSet	552
14.6 数组集合类	554
14.6.1 创建矢量	554
14.6.2 存储对象	557
14.6.3 检索对象	558
14.6.4 移除对象	560
14.6.5 搜索对象	561
14.6.6 应用矢量	562
14.6.8 对集合中的元素进行 排序	566

14.6.9 堆栈存储器	567
14.7 链表	573
14.8 使用地图	575
14.8.1 散列过程	575
14.8.2 把自己的类用作键	576
14.8.3 创建HashMap容器	577
14.8.4 存储、检索和移除对象	578
14.8.5 处理地图中的所有元素	579
14.9 小结	589
14.10 练习	589
第15章 一组有用的类	591
15.1 数组的实用方法	591
15.1.1 填充数组	591
15.1.2 复制数组	593
15.1.3 比较数组	593
15.1.4 数组的排序	594
15.1.5 搜索数组	598
15.1.6 将数组的内容用作字符串	600
15.2 Observable和Observer对象	601
15.2.1 定义Observable对象的类	602
15.2.2 Observable类的方法	602
15.3 生成随机数	605

15.4 日期和时间	608
15.4.1 Date类	608
15.4.2 解释Date对象	609
15.4.3 公历	612
15.5 正则表达式	618
15.6 使用Scanner对象	637
15.6.1 创建Scanner对象	637
15.6.2 从Scanner对象获取输入	638
15.6.3 测试标记	640
15.6.4 为标记定义自己的模式	641
15.7 小结	642
15.8 练习	642
第16章 线程	645
16.1 理解线程	645
16.1.1 创建线程	647
16.1.2 停止线程	652
16.1.3 连接线程	653
16.1.4 线程的调度	654
16.1.5 实现Runnable接口	654
16.2 管理线程	656
16.2.1 同步	657
16.2.2 死锁	672
16.3 使用执行器	674
16.3.1 使用执行器	

674	
16.3.2	执行Callable<V>任务
675	
16.3.3	Future<V>对象方法
675	
16.3.4	关闭线程池
676	
16.4	线程优先级
685	
16.5	小结
687	
16.6	练习
687	
	第17章 创建窗口
689	
17.1	Java中的图形用户界面
689	
17.2	创建窗口
691	
17.2.1	设计窗口
692	
17.2.2	在GUI代码中禁止死锁
692	
17.3	组件和容器
696	
17.3.1	Window和Frame组件
697	
17.3.2	Window面板
698	
17.4	组件的基础知识
699	
17.4.1	组件的属性
699	
17.4.2	组件的大小和位置
700	
17.4.2	点和矩形
703	
17.4.3	组件的可视化特性
706	
17.4.4	Swing 组件
714	
17.5	使用Swing容器
716	
17.6	容器的布局管理器
718	
17.6.1	流布局管理器
719	
17.6.2	使用边界布局管理器

724	
17.6.3	使用卡片布局管理器
726	
17.6.4	使用网格布局管理器
728	
17.6.5	使用BoxLayout管理器
730	
17.6.7	使用GridBagLayout 管理器
736	
17.6.8	使用SpringLayout 管理器
743	
17.7	向窗口中添加菜单
750	
17.7.1	创建JMenu和 JmenuItem对象
750	
17.7.2	创建菜单
751	
17.7.3	为菜单添加菜单项
753	
17.7.4	添加菜单快捷键和 加速器
757	
17.8	小结
759	
17.9	练习
759	
第18章	处理事件
761	
18.1	交互式Java程序
761	
18.2	事件处理过程
763	
18.3	事件类
764	
18.3.1	低级事件类
764	
18.3.2	使窗口处理自己的 事件
766	
18.3.3	允许其他低级事件
768	
18.3.4	低级事件监听器
769	
18.3.5	使用适配器类
774	
18.3.6	语义事件

776	
18.3.7	语义事件监听器
776	
18.4	applet中语义事件的处理
777	
18.4.1	其他事件处理方式
785	
18.4.2	处理低级事件和语义事件
787	
18.5	应用程序中的语义事件监听器
788	
18.6	使用动作
794	
18.6.1	Action接口
794	
18.6.2	Action方法
795	
18.6.3	把动作作用作菜单项
796	
18.6.4	定义Action类
797	
18.7	添加工具栏
803	
18.8	添加菜单图标
812	
18.9	添加工具提示
813	
18.10	禁用动作
815	
18.11	小结
815	
18.12	练习
815	
第19章	在窗口中绘图
817	
19.1	使用模型/视图体系结构
817	
19.2	组件坐标系统
821	
19.3	在组件上绘图
822	
19.3.1	图形环境
822	
19.3.2	绘图过程
825	
19.3.3	渲染操作
826	

19.4 形状	826
19.4.1 定义点的类	826
19.4.2 直线和矩形	827
19.4.3 圆弧和椭圆	832
19.4.4 曲线	835
19.4.5 复杂路径	843
19.5 填充图形	848
19.6 管理图形	853
19.6.1 存储模型中的图形	854
16.6.2 绘制图形	856
19.7 用鼠标绘图	857
19.7.1 处理鼠标事件	857
19.8 定义自己的图形类	864
19.8.1 定义直线	865
19.8.2 定义矩形	866
19.8.3 定义圆	868
19.8.4 绘制曲线	871
19.9 修改光标	873
19.10 小结	874
19.11 练习	874
第20章 扩展GUI	875
20.1 创建状态栏	875
20.1.1 用于Sketcher的 状态栏类	876
20.1.2 更新面板	878

20.2 使用对话框	880
20.2.1 模态对话框和非模态对话框	881
20.2.2 一个简单的模态对话框	882
20.2.3 即时消息对话框	885
20.2.4 即时输入对话框	887
20.3 使用对话框创建文本元素	889
20.3.1 为文本定义菜单项和工具栏按钮	889
20.3.2 定义文本类	891
20.3.3 创建文本元素	892
20.4 字体选择对话框	895
20.4.1 FontDialog类	896
20.4.2 创建字体对话框按钮	896
20.4.3 添加数据面板	898
20.4.4 实现字体列表	899
20.4.5 显示所选的字体	901
20.4.6 使用分隔面板	902
20.4.7 使用微调按钮	903
20.4.8 使用单选按钮来选择字体样式	905
20.4.9 监听单选按钮	906
20.5 弹出式菜单	908
20.5.1 显示弹出式菜单	910
20.5.2 实现上下文菜单	913

20.7 变换用户坐标系统	921
20.8 选择自定义颜色	937
20.8.1 使用缓存的图像	938
20.8.2 使用JColorChooser对话框	938
20.8.3 添加自定义颜色GUI	939
20.8.4 显示颜色选择对话框	941
20.9 小结	942
20.10 练习	943
第21章 填充和打印文档	945
21.1 串行化草图	945
21.1.1 实现串行化	946
21.1.2 串行化元素列表	946
21.2 用来保存草图的基本架构	946
21.2.1 指定文件名	946
21.2.1 验证草图的目录	947
21.2.3 记录草图是否修改	948
21.2.4 处理File菜单事件	949
21.3 使用文件选择器	950
21.3.1 显示文件保存对话框	950
21.3.2 显示文件打开对话框	951
21.3.3 定制文件选择对话框	951
21.4 实现文件操作	953
21.4.1 创建定制的文件对话框	954
21.4.2 实现保存操作	

955	
21.4.3	实现Save As操作
959	
21.4.4	实现文件打开操作
961	
21.4.5	启动新草图
963	
21.4.6	禁止在关闭时丢失数据
964	
21.5	在Java中打印
966	
21.5.1	创建和使用PrinterJob对象
967	
21.5.2	打印页面
970	
21.5.3	打印整个草图
974	
21.5.4	横向打印
978	
21.5.5	提高打印性能
980	
21.2.6	实现页面设置
981	
21.5.7	使用Java打印对话框
984	
21.5.8	多页面文档的打印
987	
21.5.9	使用Book对象进行打印
995	
21.5.10	打印Swing组件
997	
21.6	小结
999	
21.7	练习
1000	
第22章	Java和XML
1001	
22.1	XML
1001	
22.1.1	XML的作用
1002	
22.1.2	在Java中处理XML
1002	
22.2	XML文档结构
1002	
22.2.1	结构良好的XML

文档	1003
22.2.2 有效的XML文档	1004
22.2.3 XML文档中的元素	1004
22.2.4 元素的属性	1008
22.3 XML中的数据结构	1011
22.4 文档类型定义	1011
22.4.1 声明DTD	1012
22.4.2 定义DTD	1013
22.4.3 Sketcher的DTD	1019
22.5 结构良好的文档的规则	1023
22.6 XML名称空间	1024
22.6.1 名称空间声明	1024
22.6.2 XML名称空间和 DTD	1026
22.7 XML 模式	1026
22.7.1 定义模式	1027
22.7.2 定义模式元素	1028
22.7.3 指定数据类型	1029
22.7.4 定义复杂元素的属性	1029
22.7.5 值的限制	1030
22.7.6 定义属性组	1031
22.7.7 指定一组元素选项	1031
22.8 Sketcher模式	1032
22.8.1 定义直线元素	1033
22.8.2 定义矩形元素类型	1035

22.8.3 定义圆元素类型	1035
22.8.4 定义曲线元素类型	1036
22.8.5 定义文本元素类型	1037
22.8.6 完整的Sketcher模式	1037
22.8.7 使用模式的文档	1041
22.9 用XML文档编程	1041
22.9.1 SAX 处理	1042
22.9.2 DOM 处理	1043
22.10 访问解析器	1043
22.11 使用SAX	1044
22.11.1 解析器的属性和特征	1046
22.11.2 用SAX解析文档	1048
22.11.3 实现SAX处理程序	1049
22.11.4 处理其他解析事件	1058
22.11.5 解析模式实例文档	1059
22.12 小结	1064
22.13 练习	1064
第23章 创建和修改XML文档	1067
23.1 文档对象模型	1067
23.2 设置DOM解析器的特征	1069
23.3 解析文档	1070
23.4 导航Document对象树	1071
23.4.1 节点类型	1072

23.4.2 可忽略的空白和元素内容	1077
23.4.3 访问属性	1078
23.5 转换XML	1081
23.6 创建Document对象	1085
23.7 把草图存储为XML文档	1089
23.7.1 添加元素节点	1090
23.7.2 为完成的草图创建Document对象	1097
23.7.3 把草图保存为XML文件	1099
23.8 读取草图的XML表示	1103
23.8.1 从XML中创建基类对象	1103
23.8.2 从XML节点中创建元素	1105
23.8.3 处理Import XML事件	1109
23.8.4 读取XML文件	1110
23.8.5 创建模型	1111
23.9 小结	1114
23.10 练习	1114
附录A 关键字	1117
附录B 计算机的算术	1119

章节摘录

版权页：插图：1) 十六进制字面量 Java中的十六进制字面量在开头都有0)【或0X，而且遵循使用字母A到F（或者a到f）分别表示数字10到15的约定。如果对十六进制不是很熟悉，这里有一些例子：如果不熟悉十六进制数，可以在附录8中找到有关它们工作方式的说明。上面所有的十六进制字面量都是int类型。如果想要设定一个long类型的十六进制字面量，就必须在字面量后面添加一个L，就像对十进制字面量一样。例如，0xFL是一个与十进制值15等同的十六进制字面量。当然，可以将一个字面量，例如0xAABBCCD9L写成0xAABB_CCD9L。这里的下划线字符将十六进制数字每4个分成一组。每组的4个十六进制数对应内存中的两个字节。与十进制整数字面量一样，下划线只能在十六进制字面量的数字之间出现，因此0x_3ABC和0x3ABC_都不对。2) 二进制字面量 有时，将整数字面量设置为二进制值更加方便。在一个字面量的前面加上0b或0B就能将其标识为一个二进制数。在这种情况下，数字只能是0或1。例如，0b110010101011或0B110010101011和0xCAB以及十进制值3243一样。也可以在二进制字面量中使用下划线字符，所以可以将值写为0b1100 1010 1011，那样更容易阅读。每组的4个二进制数对应一个十六进制数。当然，二进制字面量也可以是long类型；只需要在数字后面附加一个L。0b_1000和0b1000_都不正确，因为下划线只能在数字之间出现。3) 八进制字面量 写八进制数字字面量时要以零开头，所以035和067都是int类型的八进制数，而0777777L是long类型的八进制字面量，后者也可以写成0777_777L。八进制数只能使用数字0~7，而且每个八进制数都定义为3比特。在以前机器都用3比特倍数的长字来存储数字时，经常使用八进制数。现在很少有必要使用八进制数，但是必须注意不要意外使用它们。如果在一个整数字面量的前面添加一个0，Java编译器就会认为在指定一个八进制值。除非其中一个数字比7大，导致编译器将其标记为错误，否则将无法知道该错误，而这个数字也不会是想象中的值。

2.2.1 声明整型变量 如前所述，可以使用如下语句声明long类型的变量：
long bigOne；该语句是对变量bigOne的声明，指定变量bigOne存储一个long类型的值。编译该语句时，会为变量bigOne分配8字节的内存。Java不会自动初始化一个这样的变量。如果想要变量有一个初始值，而不是使用内存上次使用后留下来的无效值，就必须在声明中指定自己的值。为了声明变量bigOne并初始化为2, 999, 999, 999，只需要这样写：long bigOne=2_999_999_999L；这会将该变量设置为等号后面的值。在声明变量时总是初始化是一个很好的实践。在字面量中插入下划线是为了使它易读。注意如果在计算中使用没有赋值的变量，程序将无法编译。有时候如果不在声明变量时对其初始化，编译器就不能在使用该变量之前判断出它是否已经初始化，即使看起来很明显已经初始化。这也会被标记为一个错误，但是如果能在声明这些变量时总是初始化它们，就可以避免这些问题。

《Java 7入门经典》

编辑推荐

《Java 7入门经典》为编程导师Ivor Horton最新力作。

精彩短评

- 1、 很好的书，准备开始学习盛宴
- 2、 编程导师Ivor Horton最新著作
- 3、 找了半天还是在东上买了
- 4、 java入门编程的书籍满天飞，《Java 7入门经典》正如书名那样，堪称所有书籍中的经典。后悔未能早遇此书，所幸运的现在读此书。整本书对于入门的人十分合适，刚开始你不用担心自己没有一点的java基础。书的前几章，由浅入深的介绍了java是一门什么样的语言，又通过何种方法掌握她。书中有插图，有实例源码，掌握起来效率高。你不会因为是新手看着着，因困惑放弃继续学习的兴趣。边看书边模仿写代码，每当一个程序跑通了，那种兴奋的滋味油然而生。这本书对于有一定经验的工程师，精读起来或许有点浪费时间，但可以跳过前几章基础的内容，选择阅读书中一些介绍java编程的技巧和有用的工具的章节。
如果，书中能把java 7的内容单独介绍一下或许更能体现她的价值。个人意见仅供参考。
- 5、 买回来这么久了，都没怎么看过，就知道作者写的C语言很出名
- 6、 经典的老式教材，看着比较枯燥，但口碑和共用都在那里！
- 7、 经典著作，值得一读，快速浏览一遍，对Java有了比较全面的了解。
- 8、 书中的内容还是不错的。讲的挺详细的。
- 9、 大师之作，当然很喜欢的，就是有点贵。多点活动打折就好了。
- 10、 如果有其他的OOP编程的基础,建议不买此书.讲解的很到位,不过都是很基础的东西,像HHS,根本都没有提到。
- 11、 这本书对于JAVA程序员有一定的帮助，建议比较系统的学习。
- 12、 翻译太烂了，啰嗦的要死。例子很多，不错。
- 13、 有残页，后来换了，很不出
- 14、 我买的是z实惠的商品除了封面上有些瑕疵30左右的价格真的是非常满意啊 呵呵真是 碰到好的机会了呢 笑(^-^)当然作为商品的评价 我会按照标准的售价来评价的作为 80多的书籍 内容当然是没有说的只是觉得 人民邮电的书籍似乎更好些呢例如 c primer plus
- 15、 很详细
- 16、 看了其中几章,都是从很基础的知识开始讲起,很扎实的书,适合没有经验的从头学起;
- 17、 推荐这个数据，是入门学习的好教程 可以学习的哈
- 18、 JAVA入门必备，好书，买值了，虽然贵了点
- 19、 不知道是排版的原因，还是译文的原因，还是原著的原因，有些概念描述得不清楚
- 20、 这本书感觉文字说的太多，没有与例子结合，例子没有中文注解，例子又很少，初学者很不好懂
- 21、 内容强大，很实用，真在看。
- 22、 网上看了样章才出手的，经典

精彩书评

1、java入门编程的书籍满天飞，《Java 7 入门经典》正如书名那样，堪称所有书籍中的经典。后悔未能早遇此书，所幸运的现在读此书。整本书对于入门的人十分合适，刚开始你不用担心自己没有一点的java基础。书的前几章，由浅入深的介绍了java 是一门什么样的语言，又通过何种方法掌握她。书中有插图，有实例源码，掌握起来效率高。你不会因为是新手看着着，因困惑放弃继续学习的兴趣。边看书边模仿写代码，每当一个程序跑通了，那种兴奋的滋味油然而生。这本书对于有一定经验的工程师，精读起来或许有点浪费时间，但可以跳过前几章基础的内容，选择阅读书中一些介绍java编程的技巧和有用的工具的章节。如果，书中能把java 7 的内容单独介绍一下或许更能体现她的价值。个人意见仅供参考。

章节试读

1、《Java 7入门经典》的笔记-第173页

本来类的概念就抽象，觉得此章介绍过于详细哆嗦，看得我云里雾里的。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com