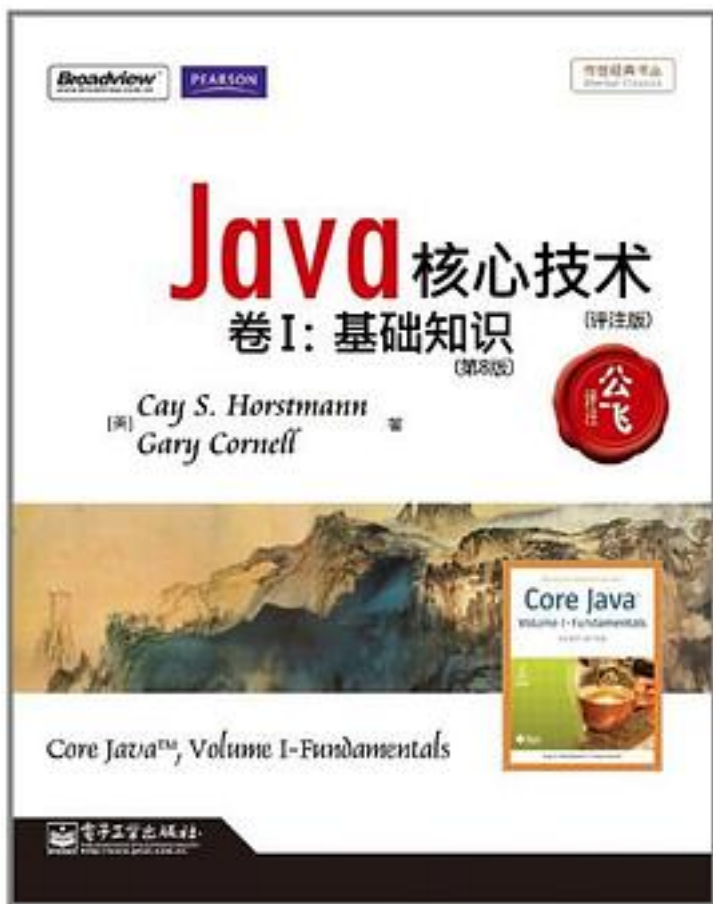


Java核心技术



[Java核心技术_下载链接1](#)

著者:Cay S. Horstmann

出版者:电子工业出版社

出版时间:2011-6

装帧:

isbn:9787121133626

Cay S.Horstmann等编著，公飞编译的《Java核心技术》(Core Java)

自第1版出版以来，一直备受广大Java程序设计人员的青睐，畅销不衰，是

Java经典书籍。第8版针对Java SE 6平台进行了全面更新，囊括了Java平台标准版(Java SE/J2SE)的全部基础知识，提供了大量完整且具有实际应用意义的实例，详细介绍了Java语言基础知识、面向对象程序设计、接口与内部类、事件监听器模型、swing图形用户界面程序设计、打包应用程序、异常处理、登录与调试、泛型程序设计、集合框架、多线程等内容。

本评注版力邀国内资深专家执笔，在英文原著基础上增加了点评和注释，并对原书内容进行了精简，删除了Java图形窗口编程的相关内容，即原书的第7-10章，同时对章节序号进行了相应的调整，从而使本书更集中于Java核心语言。

本评注版的目的在于以先行者的学研心得与实践感悟，对读者的阅读和学习加以点拨、指明捷径。

《Java核心技术》中示例程序经过精心设计，不但具有实用价值，而且易于阅读理解，可以作为初学者自己编写程序的良好开端，也能够帮助程序员快速地了解Java SE 6的新特性，或迅速从其他语言转向Java语言。

作者介绍:

Cay S. Horstmann是圣何塞州立大学计算机科学系教授、Java的倡导者。他经常在计算机会议上讲演。Cay Cornell已经教授程序设计专业课程20余年，并编写多本专著。他是Apress的创始人之一。他撰写的程序设计专业书籍十分畅销，是Jolt Award的最终获奖者之一，并在Visual Basic Magazine赢得了Readers Choice奖。

评注者简介:

公飞

曾工作于中创软件商用中间件有限公司、IBM全球信息科技服务部（GTS）和IBM全球业务咨询服务部（GBS）。目前供职于山东鲁能软件技术有限公司，负责该公司前瞻业务与技术研究、业务战略管理、解决方案发展、产品与技术规划、软件架构管理、项目管理、软件测试以及技术管理体系建设等方面的工作。

目录: preface xviii
acknowledgements xxiv
chapter 1 an introduction to java（新增批注共25条） 1
java as a programming platform 2
the java “white paper” buzzwords 3
simple 3
object oriented 4
network-savvy 5
robust 5

- secure 6
- architecture neutral 7
- portable 8
- interpreted 9
- high performance 9
- multithreaded 10
- dynamic 10
- java applets and the internet 11
- a short history of java 12
- common misconceptions about java 15
- .chapter 2 the java programming environment
(新增批注共20条) 21
- installing the java development kit 22
- downloading the jdk 22
- setting the execution path 26
- installing the library source and documentation 28
- installing the core java program examples 28
- navigating the java directories 29
- choosing a development environment 32
- using the command-line tools 33
- troubleshooting hints 43
- using an integrated development environment 45
- locating compilation errors 49
- running a graphical application 55
- building and running applets 58
- chapter 3 fundamental programming structures in java
(新增批注共44条) 63
- a simple java program 64
- comments 68
- data types 70
- integer types 70
- floating-point types 71
- the char type 72
- the boolean type 74
- variables 74
- initializing variables 76
- constants 76
- operators 77
- increment and decrement operators 78
- relational and boolean operators 79
- bitwise operators 81
- mathematical functions and constants 81
- conversions between numeric types 83
- casts 84
- parentheses and operator hierarchy 84
- enumerated types 85
- strings 86
- substrings 86
- concatenation 86
- strings are immutable 87
- testing strings for equality 88
- code points and code units 89
- the string api 90
- reading the on-line api documentation 92

- building strings 95
- input and output 96
- reading input 96
- formatting output 98
- file input and output 103
- control flow 105
- block scope 105
- conditional statements 106
- loops 109
 - determinate loops 113
 - multiple selections—the switch statement 117
 - statements that break control flow 119
- big numbers 122
- arrays 124
 - the “for each” loop 125
 - array initializers and anonymous arrays 126
 - array copying 126
 - command-line parameters 128
 - array sorting 129
 - multidimensional arrays 132
 - ragged arrays 135
- chapter 4 objects and classes（新增批注共55条） 139
- introduction to object-oriented programming 140
- classes 142
- objects 143
 - identifying classes 143
 - relationships between classes 144
 - using predefined classes 146
 - objects and object variables 146
 - the gregorianCalendar class of the java library 150
- mutator and accessor methods 152
- defining your own classes 158
 - an employee class 158
 - use of multiple source files 162
 - dissecting the employee class 163
 - first steps with constructors 164
 - implicit and explicit parameters 166
 - benefits of encapsulation 167
 - class-based access privileges 170
 - private methods 171
- final instance fields 171
- static fields and methods 172
- static fields 172
- static constants 173
- static methods 174
- factory methods 175
- the main method 175
- method parameters 178
- object construction 185
- overloading 185
- default field initialization 185
- default constructors 186
- explicit field initialization 187
- parameter names 188

- calling another constructor 188
- initialization blocks 189
- object destruction and the finalize method 193
- packages 194
- class importation 195
- static imports 196
- addition of a class into a package 197
- package scope 200
- the class path 201
- setting the class path 203
- documentation comments 204
- comment insertion 204
- class comments 205
- method comments 205
- field comments 206
- general comments 206
- package and overview comments 207
- comment extraction 207
- class design hints 208
- chapter 5 inheritance（新增批注共42条） 213
- classes, superclasses, and subclasses 214
- inheritance hierarchies 222
- polymorphism 222
- dynamic binding 224
- preventing inheritance: final classes and methods 226
- casting 227
- abstract classes 229
- protected access 234
- object: the cosmic superclass 235
- the equals method 236
- equality testing and inheritance 238
- the hashCode method 240
- the toString method 243
- generic array lists 248
- accessing array list elements 250
- compatibility between typed and raw array lists 254
- object wrappers and autoboxing 256
- methods with a variable number of parameters 259
- enumeration classes 260
- reflection 263
- the class class 263
- a primer on catching exceptions 266
- using reflection to analyze the capabilities of classes 268
- using reflection to analyze objects at runtime 273
- using reflection to write generic array code 277
- method pointers! 281
- design hints for inheritance 284
- chapter 6 interfaces and inner classes（新增批注共24条） 289
- interfaces 290
- properties of interfaces 296
- interfaces and abstract classes 297
- object cloning 298
- interfaces and callbacks 305
- inner classes 307

- use of an inner class to access object state 309
- special syntax rules for inner classes 312
- are inner classes useful? actually necessary? secure? 313
- local inner classes 316
- accessing final variables from outer methods 316
- anonymous inner classes 319
- static inner classes 322
- proxies 325
- properties of proxy classes 333
- chapter 7 exceptions, logging, assertions, and debugging
 - (新增批注共38条) 335
- dealing with errors 336
- the classification of exceptions 338
- declaring checked exceptions 340
- how to throw an exception 342
- creating exception classes 343
- catching exceptions 344
- catching multiple exceptions 346
- rethrowing and chaining exceptions 348
- the finally clause 349
- analyzing stack trace elements 352
- tips for using exceptions 357
- using assertions 361
- assertion enabling and disabling 361
- using assertions for parameter checking 362
- using assertions for documenting assumptions 363
- logging 364
 - basic logging 364
 - advanced logging 365
 - changing the log manager configuration 367
- localization 368
- handlers 369
- filters 373
- formatters 373
- a logging recipe 373
- debugging tips 381
 - using a console window 387
- tracing awt events 389
- letting the awt robot do the work 393
- using a debugger 396
- chapter 8 generic programming (新增批注共22条) 401
- why generic programming? 402
- who wants to be a generic programmer? 403
- definition of a simple generic class 404
- generic methods 406
- bounds for type variables 407
- generic code and the virtual machine 409
- translating generic expressions 411
- translating generic methods 411
- calling legacy code 413
- restrictions and limitations 414
- type parameters cannot be instantiated with primitive types 414
- runtime type inquiry only works with raw types 415
- you cannot throw or catch instances of a generic class 415

- arrays of parameterized types are not legal 416
- you cannot instantiate type variables 416
- type variables are not valid in static contexts of generic classes 418
- beware of clashes after erasure 418
- inheritance rules for generic types 419
- wildcard types 421
- supertype bounds for wildcards 423
- unbounded wildcards 424
- wildcard capture 425
- reflection and generics 430
- using class[t] parameters for type matching 431
- generic type information in the virtual machine 431
- chapter 9 collections（新增批注共55条） 437
- collection interfaces 438
- separating collection interfaces and implementation 439
- collection and iterator interfaces in the java library 441
- concrete collections 447
- linked lists 448
- array lists 459
- hash sets 459
- tree sets 463
- object comparison 464
- queues and dequeues 469
- priority queues 471
- maps 472
- specialized set and map classes 476
- the collections framework 481
- views and wrappers 487
- bulk operations 493
- converting between collections and arrays 494
- algorithms 494
- sorting and shuffling 496
- binary search 498
- simple algorithms 499
- writing your own algorithms 500
- legacy collections 502
- the hashtable class 502
- enumerations 502
- property maps 503
- stacks 504
- bit sets 504
- chapter 10 multithreading（新增批注共24条） 509
- what are threads? 511
- using threads to give other tasks a chance 517
- interrupting threads 524
- thread states 528
- new threads 529
- runnable threads 529
- blocked and waiting threads 530
- terminated threads 530
- thread properties 531
- thread priorities 531
- daemon threads 533
- handlers for uncaught exceptions 534

synchronization 535
an example of a race condition 536
the race condition explained 540
lock objects 541
condition objects 544
the synchronized keyword 549
synchronized blocks 553
the monitor concept 554
volatile fields 555
deadlocks 556
lock testing and timeouts 559
read/write locks 560
why the stop and suspend methods are deprecated 561
blocking queues 563
thread-safe collections 570
efficient maps, sets, and queues 570
copy on write arrays 572
older thread-safe collections 572
callables and futures 573
executors 577
thread pools 578
scheduled execution 582
controlling groups of tasks 583
synchronizers 584
semaphores 585
countdown latches 585
barriers 585
exchangers 586
synchronous queues 586
example: pausing and resuming an animation 586
threads and swing 592
running time-consuming tasks 594
using the swing worker 598
the single-thread rule 604
index 607
• • • • • ([收起](#))

[Java核心技术_下载链接1](#)

标签

Java

计算机

编程

java

程序设计

计算机_CS

技术

成长

评论

有很多和 c++ 的比较，Collection 那章 写的不错。评注还行。

仅有perl是不够的，Let's look at the object, Java!!!

[Java核心技术_下载链接1](#)

书评

写个大纲，希望有所帮助：

1-6章：概述和基本语法，为便于C++程序员迁移，附有大量C++对比注释，还有各种“注意”“警告”的tips，这编排很好。

7-9章：Java基于awt和swing包的界面开发由于缺少良好的IDE支持，界面设计和布局问题让习惯了VS的初学者很烦躁，789可跳读，核心...

学习Java有两部很好的书籍：《Thinking in Java》和《Core Java》。前者内容丰富，在讲述Java语言的同时把Java编程思想也娓娓道来，让人如沐春风；后者偏重于讲解Java技术，举例说明了大量API的使用。这两部书在讲解Java上殊途同归，都值得放在书桌上时常翻阅。不过看完后比...

每次我的朋友来问我，学java看什么书好，我总是说：《Thinking in Java》是你一定要看的，但不是现在，那本书不适合入门。要入门，来，借这本书去吧，然后递给他这本《Core Java》（卷一）。
Java十年，发展如此迅速，J2EE之火热，令我大感惊讶，我为我几年前选择学Java刚到庆...

卷1+卷2就是一套JavaSE的大百科全书了。看到《Java核心技术》新版更新已经出到第8版时，虽不狂喜，却也欣喜异常的。
作为一本力求讲解全部java基本知识的经典书籍，研读相关细节，所获心得颇多。从jdk的下载安装到部署应用程序；从数据类型控制流程到对象与类再到集合；从图形...

这本使JAVA里面最重要的一本书，是我开始JAVA之旅的第一本书，希望大家能够好好研读一下，可以学到很多东西，这本书所涵盖的内容比其他的要多地多，即包括基础知识，又包括高级的应用，不论你是自己编程还是做企业级的开发系统，我想这本书可以给你足够的参考。我那时看得是...

ActionScript和JAVA真的很像，很多类和常用方法是完全相同的。
第1章建议详看，因为这样可以对JAVA多一些了解；
第2章很短，可以详看，同时应该删除Flash
Builder独立版，改装插件版。因为独立版不支持JAVA开发，需要自己装一堆插件才行。在国内的网络环境下，去在线装哪些插件...

本书的内容很全面，但是不得不吐槽一下翻译，比如在P231，倒数第8行，“这个方法的调用者可以34使用……”，推测这里的34应该是没有用的，这种错误太低级了。另外翻译的表达不清晰，有效的降低了我对本书的理解，建议大家看原版

这本使JAVA里面最重要的一本书，是我开始JAVA之旅的第一本书，希望大家能够好好研读一下，可以学到很多东西，这本书所涵盖的内容比其他的要多地多，即包括基础知识，又包括高级的应用，不论你是自己编程还是做企业级的开发系统，我想这本书可以给你足够的参考。我那时看得是...

我从来没有看过一本这样轻松的计算机的书，我是有生以来第一次读到这么好的翻译的

书。在我对翻译的书绝对无望的时候，它力挽狂澜的告诉我，事情还没有坏到这个地步。我以前发誓此生不再看XXX人编著的书，不再看XXXX等人和著的书，我只看XXX著的书。但看过了cove java 之后...

该书第十版GUI仍在讲Swing
并发一章是该书缺陷的典型例子，内容包含了不同年代实现的大量并发编程的不同方法，却完全未指出当前大部分工程主要采用的措施，没有各类情况采取何种措施的指南，更没有不同方法间的性能对比，极其枯燥而缺乏实际价值。该章还出现了给出一大段程序然...

还要写题目，总感觉是在苦思冥想一个难以命名的变量，思考出来的结果又是一股高考作文的既视感。
作为一本入门参考书，绝对比国内现在的大部分教材要优秀得多。对于Java初学者来说，也是一本值得时常翻阅的技术手册。书中有较多地方进行java与c++的比较，这是很有...

这是一本讲的很详细的书，既可以用作入门书，也可以当手册，需要的时候查询。
700页的书，基本每个例子都敲了一遍，看完之后很有成就感。
当然，只看书是不够的，关键还是实践；基本功很重要，但是用不上的基本功就是废柴。加油！准备开...

很基础，目前看到图形界面350页，适合初学者的一本java教材。前面的基础讲解的不错。后面的图形界面的有说明也有应用的实例，不知道后面的异常和多线程内容怎么样，看前面的部分来推测 应该不错。建议学Java的看看。

《java核心技术(卷1)》书评
最近花了一个多星期的时间看了《Java核心技术》卷一，感觉这本书是一本介绍java内容比较全面的一本书，几乎所有Java的知识点都有涉及，而且讲解的也很详细，对java的初学者来说是一本很好的入门书籍。该书对java中jdk的下载和安装部署都进行了详细的...

core

java内容翔实，丰富，生动，就算对于java已经很熟悉的人手头备上一本虽是用来查阅也是十分方便的。在看完core java后再去品味TIJ会有令眼看世界的感觉。

我发现这本书对加强Java基本概念的理解非常有帮助。它是为有编程经验的人设计的，所以不会涉及令人痛苦的细节，但它确实帮我厘清我一直纠结的基础概念。阅读这本书花了我两天时间，我一定会买卷II的。感谢花时间撰写这种高质量图书的程序员。我用C++，所以C++注释部分对我非常...

《Java核心技术卷一》是唯一可以和《[Java编程思想]》媲美的一本 Java 入门书。单从技术的角度来看，前者更好一些。但上升到思想层面嘛，自然后者更好，两者的偏重点不同。思想不会变，所以《Java编程思想》还停留在第 4 版（2007 年）；而技术是要更新迭代的，所以《Java核心...

在上册的4.10 Class Design Hints小节，缺少第5部分。（P200）

断断续续快看完第九版第一卷英文版。刚开始时候，感觉这本书很贴近实际，开门见山便亮出正统完整的类似helloworld的例子，以及简单GUI的例子。在简洁地介绍java开发环境后，详细介绍一个完整的java程序涉及到的元素，充满了实用气息。然而随着深入章节，到了第6章，以及第9章...

[Java核心技术_下载链接1](#)