

<<疯狂Java讲义精粹>>

图书基本信息

书名：<<疯狂Java讲义精粹>>

13位ISBN编号：9787121155796

10位ISBN编号：7121155796

出版时间：2012-1

出版时间：电子工业出版社

作者：李刚

页数：577

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：<http://www.tushu007.com>

<<疯狂Java讲义精粹>>

前言

前言 2011年7月, Oracle如约发布了Java 7正式版。

接下来笔者对《疯狂Java讲义》进行了升级:使用Java 7改写了全书所有程序,全面介绍了Java 7的各种新特性。

《疯狂Java讲义》面市的近3年时间内,得到了广大读者的一致好评。

笔者也收到了一些关于《疯狂Java讲义》的建议:有人觉得这本书实在太太、太厚了;也有读者建议把《疯狂Java讲义》拆分成上、下册来出版;针对这些建议,笔者慎重地斟酌了很久:Java经过10多年的发展,已经成为世界上应用最广泛的编程语言,而且Java语言的强大是有目共睹的。

功能如此丰富、强大,历史如此悠久的一门编程语言,如果希望通过一本三、四百页的书能讲透,笔者觉得不太现实。

虽然笔者也很希望对《疯狂Java讲义》进行精简,但实在难以取舍。

如果把《疯狂Java讲义》拆分成上、下册来出版,这种做法会增加读者的经济负担(1本书变成了2本书,总价会高很多)。

后来出版社的朋友启发说:可以尝试把《疯狂Java讲义》部分内容抽取成“精粹本”;又想到有高校老师反应:学校在安排Java课程之前,已经安排了数据库开发相关课程,《疯狂Java讲义》中关于SQL、JDBC编程的内容有些重复。

出于上面两种需求,笔者对《疯狂Java讲义》内容进行了调整,删除了部分内容,得到这本《疯狂Java讲义精粹》,本书更适合作为高校教材使用。

在以“疯狂Java体系”图书为教材的疯狂软件教育中心(www.fkjava.org)里,经常有学生询问:为什么叫疯狂Java这个名字?

也有一些读者通过网络、邮件来询问这个问题。

其实这个问题答案可以在本书第1版的前言中找到。

疯狂的本质是一种“享受编程”的状态。

在一些不了解编程的人看来:编程的人总面对着电脑、在键盘上敲打,这种生活实在太枯燥了,但实际上是因为他们并未真正了解编程,并未真正走进编程。

在外人眼中:程序员不过是在敲打键盘;但在程序员心中:程序员敲出的每个字符,都是程序的一部分。

程序是什么呢?

程序是对现实世界的数字化模拟。

开发一个程序,实际是创造一个或大或小的“模拟世界”。

在这个过程中,程序员享受着“创造”的乐趣,程序员沉醉在他所创造“模拟世界”里:疯狂地设计、疯狂地编码实现。

实现过程不断地遇到问题,然后解决它;不断地发现程序的缺陷,然后重新设计、修复它——这个过程本身就是一种享受。

一旦完全沉浸到编程世界里,程序员是“物我两忘”的,眼中看到的、心中想到的,只有他正在创造的“模拟世界”。

在学会享受编程之前,编程学习者都应该采用“案例驱动”的方式,学习者需要明白程序的作用是:解决问题——如果你的程序不能解决你自己的问题,如何期望你的程序去解决别人的问题呢?

那你的程序的价值何在?

——知道一个知识点能解决什么问题,才去学这个知识点,而不是盲目学习!

因此本书强调编程实战,强调以项目激发编程兴趣。

仅仅只是看完这本书,你不会成为高手!

在编程领域里,没有所谓的“武林秘笈”,再好的书一定要配合大量练习,否则书里的知识依然属于作者,而读者则仿佛身入宝山而一无所获的笨汉。

本书配合了大量高强度的练习,希望读者强迫自己去完成这些项目。

<<疯狂Java讲义精粹>>

这些习题的答案可以参考本书所附光盘中《疯狂Java实战演义》的配套代码。

如果需要获得编程思路和交流，可以登录<http://www.crazyit.org>与广大读者和笔者交流。

本书有什么特点 本书是《疯狂Java讲义》的精粹版，本书并不是一本简单的Java入门教材。本书的内容完全取自《疯狂Java讲义》，只是删减了原书中关于SQL、JDBC编程，以及Java图形界面编程的相关内容。

因此，本书具有如下三个特点。

1. 阐述透彻、原理清晰 本书并不是简单地罗列Java语法规则，而是尽量从语法设计者的角度向读者解释每个语法规则的作用、缘由；本书力求从运行机制来解释代码的运行过程，从内存分配的细节上剖析程序的运行细节。

阅读本书不仅要求读者知道怎么做，而且要求读者能理解“为什么这么做”。

2. 再现李刚老师课堂氛围 本书的内容是笔者6年多授课经历的总结，知识体系取自疯狂Java实战的课程体系。

本书力求再现笔者的课堂氛围：以浅显比喻代替乏味的讲解，以疯狂实战代替空洞的理论。

书中包含了大量“注意”、“学生提问”部分，这些正是上千个Java学员所犯错误的汇总。

3. 注释详细，轻松上手 为了降低读者阅读的难度，书中代码的注释非常详细，几乎每两行代码就有一行注释。

不仅如此，本书甚至还把一些简单理论作为注释穿插到代码中，力求让读者能轻松上手。

本书所有程序中关键代码以粗体字标出，也是为了帮助读者能迅速找到这些程序的关键点。

本书写给谁看 如果你仅仅想对Java有所涉猎，那么本书并不适合你；如果你想全面掌握Java语言，并使用Java来解决问题、开发项目，或者希望以Java编程作为你的职业，那么本书将非常适合你。希望本书能引爆你内心潜在的编程激情，如果本书能让你产生废寝忘食的感觉，那笔者就非常欣慰了。

2011-12-17

<<疯狂Java讲义精粹>>

内容概要

《疯狂Java讲义精粹（含CD光盘1张）》是《疯狂Java讲义》的精粹版，本书同样保留了《疯狂Java讲义》的特性：

本书基于Java 7完成，全面介绍了Java 7的新特性。

本书删除了《疯狂Java讲义》中关于图形界面编程、数据库编程的内容，保留了Java的基本语法结构、Java的面向对象特征、Java集合框架体系、Java泛型、异常处理、Java注释、Java的IO流体系、Java多线程编程、Java网络通信编程和Java反射机制的内容。

本书的这种调整一方面是为了对《疯狂Java讲义》进行精简，另一方面也是应广大高校老师的反馈：高校课程前期已经讲授过数据库相关内容，因此本书删除了相关内容。

本书的写作是基于Java 7完成的，全面介绍了Java 7的二进制整数、菱形语法、增强switch语句、多异常捕获、自动关闭资源的try语句、JDBC 4.1新特性、NIO.2、AIO等新特性。

本书光盘里包含书中所有示例的代码和《疯狂Java实战演义》的所有项目代码。但由于

本书并未包含Java图形界面编程的相关知识，因此读者在阅读带图形界面的项目代码时可能有一定困难，此时读者可以参考《疯狂Java讲义》的相关内容，也可以登录<http://www.crazyit.org>站点与笔者及本书庞大的读者群相互交流。

<<疯狂Java讲义精粹>>

作者简介

李刚，从事10多年的JavaEE应用开发，现任疯狂软件教育中心教学总监，疯狂Java实训营创始人，疯狂Java体系图书作者，曾任LITEON公司的j2EE技术主管，负责该公司的企业信息化平台的架构设计，曾任广州电信、广东龙泉科技等公司的技术培训导师，曾兼任广东技术师范学院计算机科学系的兼职副教授，2007年3月26日的{电脑报}专访人物，培训的学生已在华为、立信、普信、网易、电信盈科等公司就职，国内著名的高端IT技术作家，已出版《疯狂Java讲义》、《疯狂Android讲义》、《轻量级JavaEE企业应用实战》、《疯狂Jjax讲义》、《疯狂XML讲义》、《经典JavaEE企业应用实战》、《Struts2.1权威指南》、《Ruby On Rails敏捷开发最佳实践》等着作。

其中疯狂Java体系图书都是已经得到广泛的市场认同，并多次重印的畅销图书，并被多家高校选作教材，部分图书已被翻译成繁体中文版、输出到台湾地区。

<<疯狂Java讲义精粹>>

书籍目录

第1章 Java语言概述

1.1 Java语言的发展简史

1.2 Java的竞争对手及各自优势

1.2.1 C#简介和优势

1.2.2 Ruby简介和优势

1.2.3 Python简介和优势

1.3 Java程序运行机制

1.3.1 高级语言的运行机制

1.3.2 Java程序的运行机制和JVM

1.4 开发Java的准备

1.4.1 下载和安装Java的JDK

不是说JVM是运行Java程序的虚拟机吗？

那JRE和JVM的关系是怎样的呢？

为什么不安装公共JRE呢？

1.4.2 设置PATH环境变量

为什么选择用户变量？

用户变量与系统变量有什么区别？

1.5 第一个Java程序

1.5.1 编辑Java源代码

1.5.2 编译Java程序

当我们编译C程序时，不仅需要指定存放目标文件的位置，也需要指定目标文件的文件名，这里使用javac编译Java程序时怎么不需要指定目标文件的文件名呢？

1.5.3 运行Java程序

1.5.4 根据CLASSPATH环境变量

定位类

1.6 Java程序的基本规则

1.6.1 Java程序的组织形式

1.6.2 Java源文件的命名规则

1.6.3 初学者容易犯的错误

1.7 垃圾回收机制

1.8 何时开始使用IDE工具

我想学习Java编程，到底是学习Eclipse好呢，还是学习NetBeans好呢？

1.9 本章小结

本章练习

第2章 数据类型和运算符

2.1 注释

2.1.1 单行注释和多行注释

2.1.2 文档注释

API文档是什么？

为什么要掌握查看API文档的方法？

<<疯狂Java讲义精粹>>

2.2 标识符和关键字

2.2.1 分隔符

2.2.2 标识符规则

2.2.3 Java关键字

2.3 数据类型分类

2.4 基本数据类型

2.4.1 整型

2.4.2 Java 新增的二进制整数

2.4.3 字符型

2.4.4 浮点型

2.4.5 Java 新增的数值中使用

下划线分隔

2.4.6 布尔型

2.5 基本类型的类型转换

2.5.1 自动类型转换

2.5.2 强制类型转换

2.5.3 表达式类型的自动提升

2.6 直接量

2.6.1 直接量的类型

2.6.2 直接量的赋值

2.7 运算符

2.7.1 算术运算符

2.7.2 赋值运算符

2.7.3 位运算符 3

2.7.4 扩展后的赋值运算符

2.7.5 比较运算符

2.7.6 逻辑运算符

2.7.7 三目运算符

2.7.8 运算符的结合性和优先级

2.8 本章小结

本章练习

第3章 流程控制与数组

3.1 顺序结构

3.2 分支结构

3.2.1 if条件语句

3.2.2 Java 的switch分支语句

3.3 循环结构

3.3.1 while循环语句

3.3.2 do while循环语句

3.3.3 for循环

3.3.4 嵌套循环

3.4 控制循环结构

3.4.1 使用break结束循环

3.4.2 使用continue结束本次循环

3.4.3 使用return结束方法

3.5 数组类型

<<疯狂Java讲义精粹>>

3.5.1 理解数组：数组也是一种类型

int[]是一种类型吗？

怎么使用这种类型呢？

3.5.2 定义数组

3.5.3 数组的初始化

能不能只分配内存空间，不赋初始值呢？

3.5.4 使用数组

为什么要我记住这些异常信息？

3.5.5 foreach循环

3.6 深入数组

3.6.1 内存中的数组

为什么有栈内存和堆内存之分？

3.6.2 基本类型数组的初始化

3.6.3 引用类型数组的初始化

3.6.4 没有多维数组

我是否可以让图3.13中灰色覆盖的数组元素再次指向另一个数组？

这样不就可以扩展成三维数组吗？

甚至扩展成更多维的数组？

3.6.5 操作数组的工具类

3.6.6 数组的应用举例

3.7 本章小结

本章练习

第4章 面向对象（上）

4.1 类和对象

4.1.1 定义类

构造器不是没有返回值吗？

为什么不能用void修饰呢？

4.1.2 对象的产生和使用

4.1.3 对象、引用和指针

4.1.4 对象的this引用

4.2 方法详解

4.2.1 方法的所属性

4.2.2 方法的参数传递机制

4.2.3 形参个数可变的方法

4.2.4 递归方法

4.2.5 方法重载

为什么方法的返回值类型不能用于区分重载的方法？

4.3 成员变量和局部变量

4.3.1 成员变量和局部变量

4.3.2 成员变量的初始化和内存中的运行机制

<<疯狂Java讲义精粹>>

4.3.3 局部变量的初始化和内存中的运行机制

4.3.4 变量的使用规则

4.4 隐藏和封装

4.4.1 理解封装

4.4.2 使用访问控制符

4.4.3 package、import和import static

4.4.4 Java的常用包

4.5 深入构造器

4.5.1 使用构造器执行初始化

构造器是创建Java对象的途径，是不是说构造器完全负责创建Java对象？

4.5.2 构造器重载

为什么要用this来调用另一个重载的构造器？

我把另一个构造器里的代码复制、粘贴到这个构造器里不就可以了吗？

4.6 类的继承

4.6.1 继承的特点

4.6.2 重写父类的方法

4.6.3 super限定

4.6.4 调用父类构造器

为什么我创建Java对象时从未感觉到java.lang. Object类的构造器被调用过？

4.7 多态

4.7.1 多态性

4.7.2 引用变量的强制类型转换

4.7.3 instanceof运算符

4.8 继承与组合

4.8.1 使用继承的注意点

4.8.2 利用组合实现复用

使用组合关系来实现复用时，需要创建两个Animal对象，是不是意味着使用组合关系时系统开销更大？

4.9 初始化块

4.9.1 使用初始化块

4.9.2 初始化块和构造器

4.9.3 静态初始化块

4.10 本章小结

本章练习

第5章 面向对象（下）

5.1 Java 增强的包装类

Java为什么要对这些数据进行缓存呢？

5.2 处理对象

5.2.1 打印对象和toString

方法

5.2.2 ==和equals方法

判断obj是否为Person类的实例时，为何不用obj instanceof Person来判断呢？

<<疯狂Java讲义精粹>>

5.3 类成员

5.3.1 理解类成员

5.3.2 单例 (Singleton) 类

5.4 final修饰符

5.4.1 final成员变量

5.4.2 final局部变量

5.4.3 final修饰基本类型变量和引用类型变量的区别

5.4.4 可执行“宏替换”的final变量

5.4.5 final方法

5.4.6 final类

5.4.7 不可变类

5.4.8 缓存实例的不可变类

5.5 抽象类

5.5.1 抽象方法和抽象类

5.5.2 抽象类的作用

5.6 更彻底的抽象：接口

5.6.1 接口的概念

5.6.2 接口的定义

5.6.3 接口的继承

5.6.4 使用接口

5.6.5 接口和抽象类

5.6.6 面向接口编程

5.7 内部类

5.7.1 非静态内部类

非静态内部类对象和外部类对象的关系是怎样的？

5.7.2 静态内部类

为什么静态内部类的实例方法也不能访问外部类的实例属性呢？

接口里是否能定义内部接口？

5.7.3 使用内部类

既然内部类是外部类的成员，那么是否可以为外部类定义子类，在子类中再定义一个内部类来重写其父类中的内部类？

5.7.4 局部内部类

5.7.5 匿名内部类 1

5.7.6 闭包 (Closure) 和回调

5.8 枚举类

5.8.1 手动实现枚举类

5.8.2 枚举类入门

5.8.3 枚举类的Field、方法和构造器

5.8.4 实现接口的枚举类

枚举类不是用final修饰了吗？

怎么还能派生子类呢？

5.8.5 包含抽象方法的枚举类

5.9 对象与垃圾回收

5.9.1 对象在内存中的状态

5.9.2 强制垃圾回收

章节摘录

版权页：插图：方法重载和方法重写在英语中分别是overload和override，经常看到有些初学者或一些低水平的公司喜欢询问重载和重写的区别？

其实把重载和重写放在一起比较本身没有太大的意义，因为重载主要发生在同一个类的多个同名方法之间，而重写发生在子类和父类的同名方法之间。

它们之间的联系很少，除了二者都是发生在方法之间，并要求方法名相同之外，没有太大的相似之处。

当然，父类方法和子类方法之间也可能发生重载，因为子类会获得父类方法，如果子类定义了一个与父类方法有相同的方法名，但参数列表不同的方法，就会形成父类方法和子类方法的重载。

<<疯狂Java讲义精粹>>

编辑推荐

《疯狂Java讲义精粹》来自作者6年的Java培训经历，凝结了作者近6000小时的授课经验，总结了上千个Java学员学习过程中的典型错误，1.案例驱动，引爆编程激情《疯狂Java讲义精粹》不再是知识点的铺陈，而是致力于将知识点融入实际项目的开发，所以书中涉及了大量Java案例：仿QQ的游戏大厅、MySQL企业管理器、仿EditPlus的文本编辑器、多线程、断点下载工具……希望读者通过编写这些程序找到编程的乐趣。

2.再现李刚老师课堂氛围《疯狂Java讲义精粹》的内容是笔者近6年授课经验的总结，知识体系取自李刚疯狂Java实战课程体系，《疯狂Java讲义精粹》力求再现笔者的课堂氛围：以浅显比喻代替乏味的讲解，以疯狂实战代替空洞的理论，3.注释详细，轻松上手为了降低读者阅读的难度，书中代码的注释非常详细，几乎每两行代码就有一行注释。

不仅如此，《疯狂Java讲义精粹》甚至还把一些简单理论作为注释穿插到代码中，力求让读者能轻松上手。

<<疯狂Java讲义精粹>>

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:<http://www.tushu007.com>