

<<Java编程思想 (第4版)>>

图书基本信息

书名：<<Java编程思想 (第4版)>>

13位ISBN编号：9787111213826

10位ISBN编号：7111213823

出版时间：2007-6

出版时间：机械工业出版社

作者：[美] Bruce Eckel

页数：880

译者：陈昊鹏

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>

<<Java编程思想 (第4版)>>

内容概要

本书赢得了全球程序员的广泛赞誉，即使是最晦涩的概念，在Bruce Eckel的文字亲和力和小而直接的编程示例面前也会化解于无形。

从Java的基础语法到最高级特性（深入的面向对象概念、多线程、自动项目构建、单元测试和调试等），本书都能逐步指导你轻松掌握。

从本书获得的各项大奖以及来自世界各地的读者评论中，不难看出这是一本经典之作。

本书的作者拥有多年教学经验，对C、C++以及Java语言都有独到、深入的见解，以通俗易懂及小而直接的示例解释了一个个晦涩抽象的概念。

本书共22章，包括操作符、控制执行流程、访问权限控制、复用类、多态、接口、通过异常处理错误、字符串、泛型、数组、容器深入研究、Java

I/O系统、枚举类型、并发以及图形化用户界面等内容。

这些丰富的内容，包含了Java语言基础语法以及高级特性，适合各个层次的Java程序员阅读，同时也是高等院校讲授面向对象程序设计语言以及Java语言的绝佳教材和参考书。

第4版特点：

适合初学者与专业人员的经典的面向对象叙述方式，为更新的Java SE5/6增加了新的示例和章节。

? 测验框架显示程序输出。

?

设计模式贯穿于众多示例中：适配器、桥接器、职责链、命令、装饰器、外观、工厂方法、享元、点名、数据传输对象、空对象、代理、单例、状态、策略、模板方法以及访问者。

? 为数据传输引入了XML，为用户界面引入了SWT和Flash。

? 重新撰写了有关并发的章节，有助于读者掌握线程的相关知识。

? 专门为第4版以及Java SE5/6重写了700多个编译文件中的500多个程序。

? 支持网站包含了所有源代码、带注解的解决方案指南、网络日志以及多媒体学习资料。

? 覆盖了所有基础知识，同时论述了高级特性。

? 详细地阐述了面向对象原理。

? 在线可获得Java讲座CD，其中包含Bruce Eckel的全部多媒体讲座。

在www.MindView.net网站上可以观看现场讲座、咨询和评论。

专门为第4版以及Java SE5/6重写了700多个编译文件中的500多个程序。

支持网站包含了所有源代码、带注解的解决方案指南、网络日志以及多媒体学习资料。

覆盖了所有基础知识，同时论述了高级特性。

<<Java编程思想 (第4版)>>

详细地阐述了面向对象原理。

在线可获得Java讲座CD，其中包含Bruce Eckel的全部多媒体讲座。

在www.MindView.net网站上可以观看现场讲座、咨询和评论。

作者简介

Bruce

Eckel是MindView公司 (www.MindView.net) 的总裁, 该公司向客户提供软件咨询和培训。

他是C++标准委员会拥有表决权的成员之一, 拥有应用物理学学士和计算机工程硕士学位。

除本书外, 他还是《C++编程思想》的作者, 并与人合著了《C++编程思想第2卷》(这两本书的英文影

书籍目录

读者评论

前言

简介

第1章 对象导论

1.1 抽象过程

1.2 每个对象都有一个接口

1.3 每个对象都提供服务

1.4 被隐藏的具体实现

1.5 复用具体实现

1.6 继承

1.6.1 “是一个” (is-a) 与 “像是一个” (is-like-a) 关系

1.7 伴随多态的可互换对象

1.8 单根继承结构

1.9 容器

1.9.1 参数化类型 (范型)

1.10 对象的创建和生命期

1.11 异常处理: 处理错误

1.12 并发编程

1.13 Java与Internet

1.13.1 Web是什么

1.13.2 客户端编程

1.13.3 服务器端编程

1.22 总结

第2章 一切都是对象

2.1 用引用操纵对象

2.2 必须由你创建所有对象

2.2.1 存储到什么地方

2.2.2 特例: 基本类型

2.2.3 Java中的数组

2.3 永远不需要销毁对象

2.3.1 作用域

2.3.2 对象的作用域

2.4 创建新的数据类型: 类

2.4.1 域和方法

2.4.2 基本成员默认值

2.5 方法、参数和返回值

2.5.1 参数列表

2.6 构建一个Java程序

2.6.1 名字可见性

2.6.2 运用其他构件

2.6.3 static 关键字

2.7 你的第一个Java程序

编译和运行

2.8 注释和嵌入式文档

2.8.1 注释文档

<<Java编程思想 (第4版)>>

2.8.2 语法

2.8.3 嵌入式HTML

2.8.4 一些标签示例

2.8.5 文档示例

2.9 编码风格

2.10 总结

2.11 练习

第3章 操作符

3.1 更简单的打印语句

3.2 使用Java操作符

3.3 优先级

3.4 赋值

3.4.1 方法调用中的别名问题

3.5 算术操作符

3.5.1 一元加、减操作符

3.6 自动递增和递减

3.7 关系操作符

3.7.1 测试对象的等价性

3.8 逻辑操作符

3.8.1 短路

3.9 直接常量

3.9.1 指数记数法

3.10 按位操作符

3.11 移位操作符

3.12 三元操作符 if-else

3.13 字符串操作符 + 和 +=

3.14 使用操作符时常犯的错误

3.15 类型转换操作符

3.15.1 截尾和舍入

3.15.2 提升

3.16 Java没有“sizeof”

3.17 操作符小结

3.18 总结

第4章 控制执行流程

4.1 true和false

4.2 if-else

4.3 迭代

4.3.1 do-while

4.3.2 for

4.3.3 逗号操作符

4.4 Foreach语法

4.5 return

4.6 break和continue

4.7 臭名昭著的“goto”

4.8 switch

4.9 总结

第5章 初始化与清理

<<Java编程思想 (第4版)>>

5.1 用构造器确保初始化

5.2 方法重载

5.2.1 区分重载方法

5.2.2 涉及基本类型的重载

5.2.3 以返回值区分重载方法

5.3 缺省构造器

5.4 this关键字

5.4.1 在构造器中调用构造器

5.4.2 static的含义

5.5 清理：终结处理和垃圾回收

5.5.1 finalize()的用途何在

5.5.2 你必须实施清理

5.5.3 终结条件

5.5.4 垃圾回收器如何工作

5.6 成员初始化

5.6.1 指定初始化

5.7 构造器初始化

5.7.1 初始化顺序

5.7.2. 静态数据的初始化

5.7.3. 显式的静态初始化

5.7.4. 非静态实例初始化

5.8 数组初始化

5.8.1 可变参数列表

5.9 枚举类型

5.10 总结

第6章 访问权限控制

第7章 复用类

第8章 多态

第9章 接口

第10章 内部类

第11章 持有对象

第12章 通过异常处理错误

第13章 字符串

第14章 类型信息

第15章 泛型

第16章 数组

第17章 容器深入研究

第18章 Java I/O系统

第19章 枚举类型

第20章 注解

第21章 并发

第22章 图形化用户界面

附录A 补充材料

可下载的补充材料

Thinking in C：Java的基础

Java编程思想 研讨课

Hands-on Java研讨课CD

<<Java编程思想 (第4版)>>

Thinking in Objects研讨课

Thinking in Enterprise Java

Thinking in Patterns(with Java)

Thinking in Patterns研讨课

设计咨询与复审

附录B 资源

软件

编辑器与IDE

书籍

分析与设计

Python

我的著作列表

索引

章节摘录

1.10对象的创建和生命期 在使用对象时，最关键的问题之一便是它们的生成和销毁方式。每个对象为了生存都需要资源，尤其是内存。当我们不再需要一个对象时，它必须被清理掉，使其占有的资源可以被释放和重用。在相对简单的编程情况下，怎样清理对象看起来似乎不是什么挑战：你创建了对象，根据需要使用它，然后它应该被销毁。

然而，你很可能遇到相对复杂的情况。

例如，假设你正在为某个机场设计空中交通管理系统（同样的模型在仓库货柜管理系统、录像带出租系统或宠物寄宿店也适用）。

一开始问题似乎很简单：创建一个容器来保存所有的飞机，然后为每一架进入空中交通控制区域的飞机创建一个新的飞机对象，并将其置于容器中。

对于清理工作，只需在飞机离开此区域时删除相关的飞机对象即可。

但是，可能还有别的系统记录着有关飞机的数据，也许这些数据不需要像主要控制功能那样立即引人注目。

例如，它可能记录着所有飞离机场的小型飞机的飞行计划。

因此你需要有第二个容器来存放小型飞机，无论何时，只要创建的是小型飞机对象，那么它同时也应该置入第二个容器内。

然后某个后台进程在空闲时对第二个容器内的对象进行操作。

现在问题变得更困难了：怎样才能知道何时销毁这些对象呢？

当处理完某个对象之后，系统某个其他部分可能还在处理它。

在其他许多场合中也会遇到同样的问题，在必须明确删除对象的编程系统中（例如C++），此问题会变得十分复杂。

对象的数据位于何处？

怎样控制对象的生命周期？

C++认为效率控制是最重要的议题，所以给程序员提供了选择的权力。

为了追求最大的执行速度，对象的存储空间和生命周期可以在编写程序时确定，这可以通过将对象置于堆栈（它们有时被称为自动变量（automatic variable）或限域变量（scoped variable））或静态存储区内来实现。

这种方式将存储空间分配和释放置于优先考虑的位置，某些情况下这样控制非常有价值。

但是，也牺牲了灵活性，因为必须在编写程序时知道对象确切的数量、生命周期和类型。

如果试图解决更一般化的问题，例如计算机辅助设计、仓库管理或者空中交通控制，这种方式就显得过于受限了。

第二种方式是在被称为堆（heap）的内存池中动态地创建对象。

在这种方式中，直到运行时才知道需要多少对象，它们的生命周期如何，以及它们的具体类型是什么。

这些问题的答案只能在程序运行时相关代码被执行到的那一刻才能确定。

如果需要一个新对象，可以在需要的时刻直接在堆中创建。

因为存储空间是在运行时被动态管理的，所以需要大量的时间在堆中分配存储空间，这可能要远远大于在堆栈中创建存储空间的时间。

在堆栈中创建存储空间和释放存储空间通常各需要一条汇编指令即可，分别对应将栈顶指针向下移动和将栈顶指针向上移动。

创建堆存储空间的时间依赖于存储机制的设计。

动态方式有这样一个一般性的逻辑假设：对象趋向于变得复杂，所以查找和释放存储空间的开销不会对对象的创建造成重大冲击。

动态方式所带来的更大的灵活性正是解决一般化编程问题的要点所在。

Java完全采用了动态内存分配方式e。

每当想要创建新对象时，就要使用new关键字来构建此对象的动态实例。

.....

媒体关注与评论

时隔两年多,《Java编程思想(第4版)》的中文版又要和广大Java程序员和爱好者们见面了。这是Java语言本身不断发展和完善的必然要求,也是本书作者Bruce Eckel孜孜不倦的创作激情和灵感所结出的硕果。

《Java编程思想(第4版)》以Java最新的版本JDK5.0为基础,在第3版的基础上,添加了最新的语言特性,并且对第3版的结构进行了调整,使得所有章节的安排更加遵照循序渐进的特点,同时每一章的内容在分量上都更加均衡,这使读者能够更加容易地阅读本书并充分了解每章所讲述的内容。

在这里我们再次向Bruce Eckel致敬,他不但向我们展示了什么样的书籍才是经典书籍,而且还展示了经典书籍怎样才能精益求精,长盛不衰。

Java已经成为了编程语言的骄子。

我们可以看到,越来越多的大学在教授数据结构、程序设计和算法分析等课程时,选择以Java语言为载体。

这说明Java语言已经是人们构建软件系统时主要使用的一种编程语言。

但是,掌握好Java语言并不是一件可以轻松完成的任务,如何真正掌握Java语言,从而编写出健壮的、高效的以及灵活的程序是Java程序员们面临的重大挑战。

《Java编程思想(第4版)》就是一本能够让Java程序员轻松面对这一挑战,并最终取得胜利的的经典书籍。

本书深入浅出、循序渐进地把我们领入Java的世界,让我们在不知不觉中就学会了用Java的思想去考虑问题、解决问题。

本书不仅适合Java的初学者,更适合于有经验的Java程序员,这正是本书的魅力所在。

但是,书中并没有涵盖Java所有的类、接口和方法,因此,如果你希望将它当作Java的字典来使用,那么显然就要失望了。

我们在翻译本书的过程中力求忠于原著,为了保持连贯性,对原书第3版中仍然保持不变的部分,我们对译文除了个别地方之外,也没做修改。

对于本书中出现的大量的专业术语尽量遵循标准的译法,并在有可能引起歧义之处注有英文原文,以方便读者对照与理解。

全书由陈昊鹏翻译,郭嘉也参与了部分翻译工作。

由于水平有限,书中出现错误与不妥之处在所难免,恳请读者批评指正。

<<Java编程思想 (第4版)>>

编辑推荐

全球程序员必备图书《Java编程思想》的最新版本！

获奖历史：2003年《Software Development》杂志最佳书籍Jolt大奖 2003年《Java Developer 's Journal》读者选择最佳书籍奖 2001年《Java World》编辑选择最佳书籍奖 2000年《Java World》读者选择最佳书籍奖 1999年《Software Development》杂志最佳产品奖 1998年《Java Developer 's Journal》编辑选择最佳书籍奖海报：

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>