

<<Java第一步>>

图书基本信息

书名：<<Java第一步>>

13位ISBN编号：9787302186076

10位ISBN编号：7302186073

出版时间：2009-1

出版时间：清华大学出版社

作者：胡书敏，陈宝峰，程炜杰 著

页数：896

字数：1662000

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>

前言

Java编程是一门艺术，这虽然是智者的游戏，但是入门却非常简单。

在Java这个高雅的殿堂里，有着各种璀璨的明珠，有包容编程精粹的面向对象思想，有整合Web工作流的Struts组件，还有体现持久性的ORM思想，由此导致了Web开发领域发生了天翻地覆的变化。

虽然这些让人叹为观止的技术和思想能让我们发动头脑风暴，帮助我们开发出一个个Java项目，但是罗马不是在一天建成的，我们也有必要从一砖一瓦上，用脚踏实地的态度迈向Java殿堂的最高点。

<<Java第一步>>

内容概要

Java语言是复杂的，因为它承载着太多的Web开发需求点；Java又是简单的，只要用心，每个人都可以学会。

你手中的这本《Java第一步》，就是通过生动的语言和一个个引人入胜的案例，向你讲述Java开发的各个知识点。

本书作者已从事Java开发多年。

在本书中，作者以自己的经验向大家描述了一条学习Java的道路，其中不求向大家讲述最复杂最高深的知识点，但求通过讲述当前最实用的开发思想和方法，尽快地达到让大家掌握用Java语言开发中等规模项目的目的。

“知其然，知其所以然”是本书的创作动机，而通俗易懂更是贯穿本书写作的全程。

当你阅读本书时，将会发现作者在字里行间中，正用文字与你交流。

在本书的第九--N第九章里，将围绕“实用性”这个主题，以“面向对象”的思考方式，向大家逐渐展开Java开发的各项基础知识，比如异常处理、网络开发、数据库，以及Swing。

在这个部分里，大家将通过一种益智游戏—象棋，来理解Swing里的界面开发和事件处理方式。

在本书的第十到第十六章里，将围绕Java语言的精髓——设计模式和架构思想，通过贪吃蛇和生命游戏等案例，讲述Java项目设计过程中思考问题的方式。

通过了解本章的DAO、MVC等思想，大家能了解到Java设计思想的本质。

在前两部分的铺垫下，在本书的十七章及以后的部分里，讲述了Java的Web开发组件和Web开发模式。

这部分比较实用的Java技术将会使你能够很快地从事软件公司里的Web开发工作。

本书虽以“实用性”为宗旨，但大家在阅读本书时，不应过多地拘泥于“实用主义”，过分地关注其中的API，而应当通过语法关注设计模式和架构思想的本质，这样才能在Java艺术殿堂里得到更完美的享受。

本书不是一本技术说明手册，而是一本Java编程思想的指南。

你即将探索Java艺术的殿堂，这本书将会让你更快地获得Java的知识、智慧和成就感。

<<Java第一步>>

作者简介

胡书敏 大连理工大学计算机科学硕士。

现就职于上海威迅教育，有丰富的培训经验与教材开发经验，擅长架构及设计模式的综合运用。

5年Java方面的软件开发经验，开发过交通银行外汇交易满金宝系统等一系列大型银行和电子商务系列的Java程序，担任过项目经理和架构师。

曾著

书籍目录

第1章 Java与面向对象思想	1.1 初识Java语言	1.1.1 Java是什么	1.1.2 为什么要选用Java语言
1.1.3 Java语言的解释执行与跨平台特性	1.1.4 怎样学好Java	1.2 什么是面向对象思想	1.2.1 软件开发的方法学——面向对象思想提出的背景
1.2.2 面向对象的物质基础——类与封装	1.2.3 抽象与代码复用——抽象类与概念抽象	1.2.4 模块的开关——接口与功能抽象	1.2.5 同名但不同参——多态与重载
1.2.6 具有面向对象思想语言的优势	1.3 开发我们的第一个Java程序	1.3.1 如何配置Java的开发环境	1.3.1.1 下载和安装JDK
1.3.1.2 设置JDK的环境变量	1.3.3.3 安装Eclipse	1.3.3.4 添加Eclipse的中文插件	1.3.2 编写HelloWorld程序
1.3.3 运行HelloWorld程序	1.3.4 使用Eclipse集成开发环境运行HelloWorld程序	1.4 本章小结	第2章 开始学习Java语法
2.1 基本数据类型与操作符	2.1.1 基本数据类型	2.1.2 基本操作符	2.1.3 操作符优先级
2.1.4 对象型的数据类型	2.1.5 方法(函数)的参数和返回值	2.2 程序流程控制	2.2.1 顺序与分支
2.2.2 循环流程	2.2.3 有争议的goto语句	2.2.4 使用if语句判定是否是闰年	2.2.5 使用循环语句计算阶乘
2.2.6 使用switch语句控制输出	2.3 类与封装	2.3.1 类的构成要素	2.3.2 类的构造函数与构造次序
2.3.3 用package封装工具类和用import引入需要的类	2.3.4 修饰符与类内部对象的可见性	2.3.5 静态方法与抽象方法	2.3.6 类的封装性与模块的解耦合
2.3.7 垃圾收集机制与Finalize语句段	2.3.8 提供用电服务类的示例代码	2.4 继承与抽象类	2.4.1 继承的设计动机
2.4.2 抽象类与概念抽象	2.5 接口与实现	2.5.1 接口的语法	2.5.2 为什么要使用接口
2.5.3 接口与抽象类在本质上的区别——空调外的例子	2.5.4 通过实现接口的方式开发电视和收音机类	2.6 具有“分离”特性的多态	2.6.1 从重载中归纳多态的特性
2.6.2 多态与“分离”——实现小动物叫	2.7 RTTI与反射	2.7.1 君子无不可用之势——论后期编译的效果	2.7.2 在代码里实现RTTI的效果
2.7.3 类的照妖镜——反射	2.7.4 使用反射机制实现代码还原	2.8 通过约束降低出错可能性的泛型	2.8.1 什么是泛型
2.8.2 泛型的示例代码	2.9 本章小结	第3章 Java中的数据结构——集合类	3.1 集合类与数据容纳
3.1.1 在项目中自定义数据类型的难处	3.1.2 什么是数据结构	3.1.3 Java集中的两类数据结构	3.2 线性表型的集合
3.2.1 顺序访问的典范——数组类	3.2.2 数组的改进版本——Vbctor类	3.2.3 先进后出的Stack类	3.2.4 链表式的List接口以及LinkedList实现类
3.2.4.1 List接口里的方法	3.2.4.2 List实现类及其功能	3.2.4.3 List代码示例	3.2.5 不允许有重复元素的Set接口
3.2.5.1 Set接口里的方法	3.2.5.2 Set实现类及其功能	3.2.6 对第一类集合的归纳	3.2.7 使用Vector模拟Stack的动作
3.2.8 使用Vector模拟队列的动作	3.3 键值对型的集合	3.3.1 为什么要使用Hash类	3.3.2 键值对的典范——Hashtable类
3.3.3 通过“魔术代码”分析equals和hashCode方法	3.3.4 封装了散列表数据结构的Map接口	3.3.4.1 M印接口里的方法	3.3.4.2 实现Map接口的类
3.3.4.3 M印代码实例	3.3.4.4 采用Entry对象遍历Map	3.3.5 对第二类集合的归纳	3.4 枚举器与数据操作
3.4.1 访问集合类的“不确定性”难题	3.4.2 “大智若愚”的枚举器接口	3.4.3 枚举器“分离”思想的启示	3.4.4 使用枚举器定义统一的访问各集合的方法——一个打印的例子
3.5 深浅拷贝	3.5.1 “偷懒”的共享数据块的方法——浅拷贝	3.5.2 似是而非的浅拷贝——只拷贝ArrayList对象	3.5.3 “刨根掘底”的深拷贝——实现对ArrayList的整体克隆
3.6 Java集合类中的排序	3.6.1 让你的对象是可比较的	3.6.2 为你的对象定义比较器	3.7 本章小结
第4章 有异常怎么办	4.1 异常的基本概念	4.1.1 险恶的项目运行环境	4.1.2 步步为营的异常处理机制
4.1.3 Java虚拟机处理异常的流程	4.2 异常技术综述	4.2.1 错了不要紧——论运行期异常	4.2.2 自食其力——处理自己的异常
4.2.3 有问题就报告——异常的抛出声明	4.2.4 弥补漏洞的自定义异常	4.3 异常的陷阱	4.3.1 异常与继承
4.3.2 谁动了我的异常	4.3.3 论资排辈的异常级别	4.4 本章小结	第5章 IO流与流读写操作
5.1 输入流与输出流	5.1.1 什么是流对象	5.1.2 复杂的输入输出应用与简单的JavalO类库	5.1.3 Java里IO流对象的逻辑关系
5.2 IO对象使用说明	5.2.1 怎样读写文件——读写一个文本文件	5.2.2 怎样读写内存数据	5.2.3 怎样读写带格式的数据
5.2.4 标准设备与重定向	5.2.5 压缩和解压缩ZIP文件——创建一个压缩和解压缩的程序	5.2.6 用StreamTokenizer类输出不同格式的Token	5.3 本章小结
第6章 Java与多线程开发	6.1 线程到底是什么	6.1.1 线程的定义	6.1.2 线程的生命周期

6.1.3 操作系统与线程并发 6.2 定义自己的线程, 继承Thread类 6.2.1 Thread类与线程 6.2.2 使用Thread类实现计数器功能 6.3 解决继承局限性, 实现Runnable接口 6.3.1 Runnable接口与线程 6.3.2 使用Runnable接口实现计数器功能 6.4 避免冲突, 控制多线程的并发 6.4.1 劳逸结合——线程的睡眠 6.4.2 控制数据的有效性, 通过synchronized关键字声明同步——一个由并发引起的数据错误修改及解决例子 6.4.3 线程的等待与通知线程恢复工作 6.4.4 死锁的成因 6.4.5 特事特办, 线程的优先级 6.4.6 多线程经典案例——生产者消费者问题 6.5 本章小结第7章 Java与网络开发 7.1 网络开发的基本需求 7.1.1 网络通讯模型和网络通讯协议 7.1.2 通过TCP协议建立可靠的通讯信道 7.1.2.1 两军问题与建立可靠连接的难题 7.1.2.2 通过三次握手, 建立通讯信道 7.1.2.3 支持TCP协议的Java包 7.1.3 通过UDP协议高效地发送数据报文 7.1.3.1 UDP协议与数据报文 7.1.3.2 Java的UDP相关类说明 7.2 面向套接字编程 7.2.1 使用套接字实现基于TCP协议的服务器和客户机程序 7.2.1.1 开发服务器端代码 7.2.1.2 开发客户端代码 7.2.1.3 运行效果演示第8章 Java与数据库开发第9章 Java与图形界面开发第10章 生命游戏第11章 基于图形秘网络的Java聊天室第12章 学生信息管理系统第13章 MVC模式与项目开发第14章 设计模式与项目开发第15章 DAO模式与数据库访问第16章 设计模式的应用第17章 基于JSP+Servlet的Web架构第18章 基于JavaBean和EJB的Web构架 第19章 基于J2EE的在线购物系统第20章 Struts架构与Web应用第21章 Spring架构与Web应用第22章 ORM组件与数据库访问控制第23章 Struts与Spring集成开发第24章 基于Struts+Spring+Hibernate的在线购物系统第25章 Java的Web Service应用

章节摘录

第3章 Java中的数据结构——集合类 本章将通过讲解Java集合类的具体语法和使用方式，来向大家展示Java在数据结构方面编程的强大功能。

集合类的这些强大功能，源自于Java具有的面向对象思想特性。

所以，希望大家在学习这个章节的时候，把集合类的语法同面向对象思想中的“通过继承抽象基类功能”和“通过多态屏蔽逻辑动作细节”等原则结合起来，这样能更好地从模块层次的角度理解和运用Java集合类。

3.1 集合类与数据容纳 Java用集合类来容纳代码中的不同种类数据，这种容纳是建立在未知的基础上，即Java要用有限种类的集合类，来容纳代码中无限种类的数据对象。

用生活里的场景来对应的话，Java的集合类需要设计成“能容纳液态气态和固态等不同类型物质”的容器，要设计这样的“万能容器”确实很困难，但JDK的设计者们通过借鉴面向对象思想里的“多态”，实现了这个需求。

3.1.1 在项目中自定义数据类型的难处 如果我们在项目运行前，事先为项目代码准备好所有类型的数据对象，并根据这些数据对象的不同类型，为它们在内存里分配好空间，这是非常理想的，但事实上，我们做不到这点。

比如，在关于天气预报的项目里，我们需要一个对象来保存描述温度的信息，这个问题看上去很好解决，定义一个变量就行了，但没有那么简单。

如果描述温度信息的格式有多种，有用数字型的格式（比如多少度），也有用字符型的格式（比如“很冷”），并且程序员只有在接收到这个信息的时候才能明确地知道信息的格式，事情就比较麻烦。

为了保证能容纳所有类型的数据，我们不得不用Java语言所有对象的老祖宗类——Object类型数据来接收这个温度信息。

但是，如果问题变得再复杂点呢？

如果我们在代码里需要用（或一组）对象来描述一天里所有来访的客户信息，这样，除非这段代码被运行，否则我们甚至不知道当天会来多少客户和客户的种类，这给我们的编程带来了很大的难题。

因为我们不知道代码所需要的对象的数目和种类。

所以在代码里，要求能够在任何时候、任何地点创建任意数量的任何类型对象。

这对代码的要求似乎高了些，但是，很不幸的是，在编程中这是个“常规”要求。

不过，使用数据结构的思想能够较好地解决这一问题。

3.1.2 什么是数据结构 数据结构是在计算机编程领域里被广泛使用的术语，它用来研究数据对象的内部构成，即这个数据对象由哪些基本的数据元素构成，这个数据对象以什么方式构成，以及它呈什么结构。

数据结构主要研究数据的各种逻辑结构和存储结构，以及对不同种数据类型的各种操作。

因此，数据结构主要有两方面的内容：数据的逻辑存储结构和对数据的操作。

Java的集合类可以很好地用来描述基于Java语言的数据结构，在本章里，我们不会具体描述如何实现数据结构内部细节的内容，而是将通过学习Java的集合，在了解其数据组织方式和操作的基础上，熟悉这些集合类的用途和特长，为能够在项目中很好地使用这些集合类打下坚实的基础。

3.1.3 Java集合中的两类数据结构 归纳起来讲，Java的集合可以分为两类，第一类是以数组为代表，这类集合可以描述线性表类型的数据结构，第二类是以Map为代表，这类集合可以描述“键值对”类型的数据结构。

描述第一类数据结构的集合是以Collection为基类，在Collection类里，封装了线性表的插入、删除等基本操作；而数据结构的集合是以Map为基类。

两类集合分别通过继承和扩展这两个基类，可以逐渐细化，从而得到诸如堆栈和哈希表之类的数据集合对象。

这两类不同类型的集合，可以很好地在各种应用中存储各种类型的数据，所以说，熟悉了这两类

集合，就可以熟悉Java语言下的两类数据结构。

<<Java第一步>>

编辑推荐

《Java第一步：基础+设计模式+Servlet+EJB+Struts+Spring+Hibernate》让你用面向对象和设计模式的眼光审视Java。

作者以先行十年的经验，告诉你学习Java的道路。

采用理论结合案例的方式，学会Java从JDK到SSH的重要知识点。

生动活泼是《Java第一步——基础+设计模式+Servlet》的风格，在编程中体会快乐，这是我们的追求。

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>