



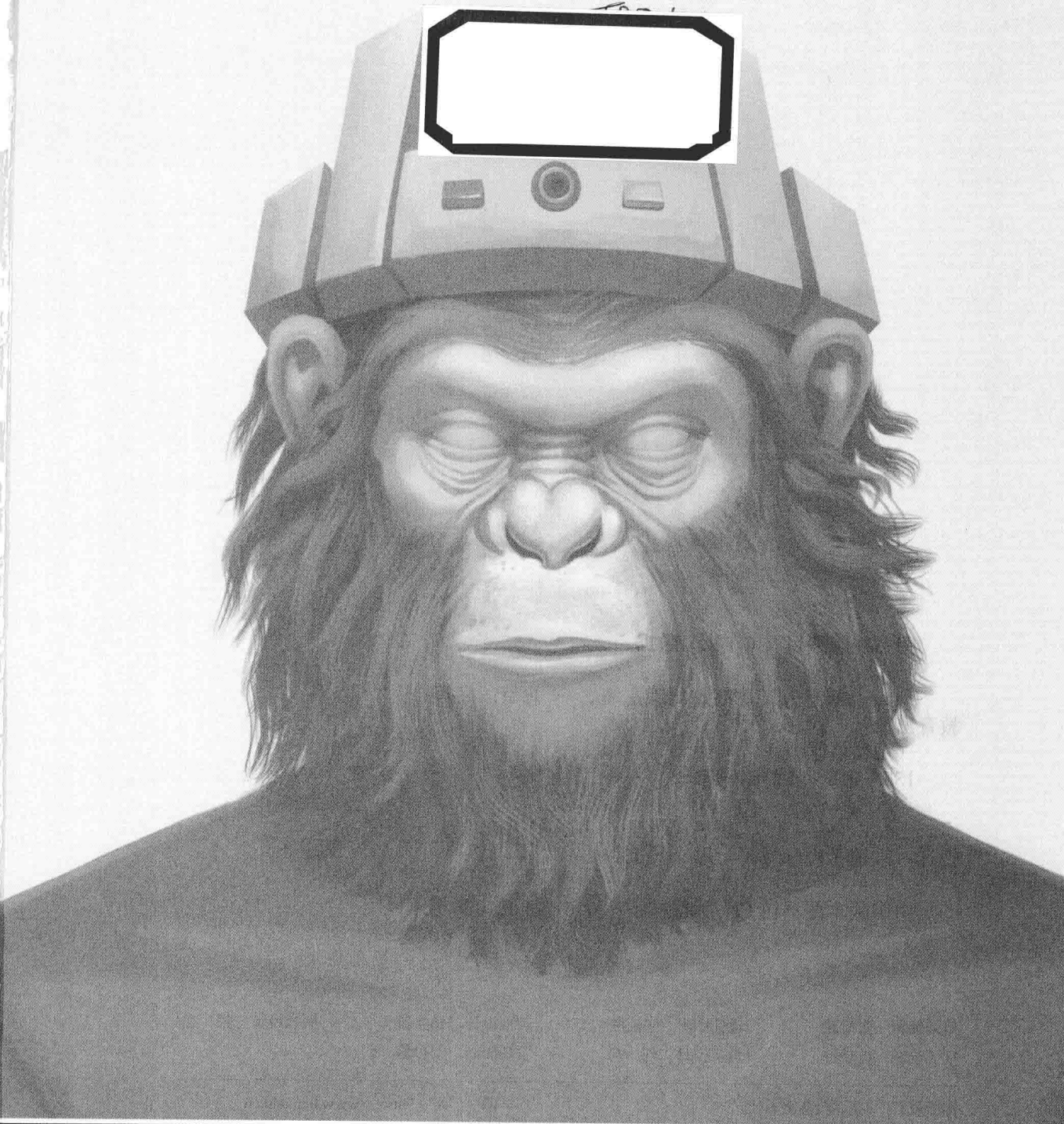
用微课学“程序猿”系列
学习 Java 的高阶开发

●智慧职教平台 ●微课 ●案例源码 ●教学课件 ●课程级资源

用微课学 Java 高级开发

主编 唐亮 纪威

高等教育出版社



YONG WEIKE XUE Java GAOJI KAIFA

用微课学

Java 高级开发

主编 唐亮 纪威

高等教育出版社·北京

图书在版编目 (C I P) 数据

Java 高级开发 / 唐亮, 纪威主编. — 北京: 高等教育出版社, 2016. 7

(用微课学)

ISBN 978-7-04-045071-2

I. ① J… II. ①唐… ②纪… III. ① Java 语言—程序设计—高等学校—教材 IV. ① TP312

中国版本图书馆 CIP 数据核字 (2016) 第 051066 号

策划编辑	张值胜	责任编辑	张值胜	封面设计	杨立新	版式设计	张杰
插图绘制	杜晓丹	责任校对	刘莉	责任印制	刘思涵		

出版发行	高等教育出版社	网 址	http://www.hep.edu.cn
社 址	北京市西城区德外大街 4 号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.hepmall.com.cn
印 刷	山东鸿君杰文化发展有限公司		http://www.hepmall.com
开 本	787mm × 1092mm 1/16		http://www.hepmall.cn
印 张	14.5	版 次	2016 年 7 月第 1 版
字 数	340 千字	印 次	2016 年 7 月第 1 次印刷
购书热线	010-58581118	定 价	35.00 元
咨询电话	400-810-0598		

本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换

版权所有 侵权必究

物料号 45071-00

III 内容简介

本书是高等教育出版社与达内时代科技集团（以下简称达内集团）联合出品的程序设计类新形态一体化教材，由达内集团诸多开发经验及授课经验丰富的一线讲师编写。本书以 IT 互联网企业实际用人的要求为导向，总结近几年国家应用型本科院校与示范性高职院校软件技术专业教学改革经验及达内集团在 IT 培训行业十多年的经验编写而成。

本书主要介绍 java 高级编程技术，全书共分为 7 章，包括：Java 桌面图形界面开发、Java I/O、Java 多线程、Java 网络编程、JDBC、XML 和 JSON、Java 反射和注释。每一章节都由基础知识、阶段案例、练习、总结等模块组成，通过不同模块间的互相辅助来搭建 Java 高级程序开发教学所需要掌握的知识结构。

与本书配套的数字课程已在“智慧职教”（www.icve.com.cn）网站上线，学习者可登录网站进行学习，也可通过扫描书中二维码观看教学视频，详见“郑重声明”页资源服务提示。

本书可作为应用型本科、高等职业院校软件技术专业的课程教材，也可作为 Java 开发者的学习和参考用书。

Java

III 丛书编委会

主任:

韩少云 达内教育集团

副主任: (以姓氏笔画为序)

王世来 浙江海洋学院东海科学技术学院

叶时平 树人大学

刘益和 内江师范学院

张平安 深圳信息职业技术学院

张志凌 牡丹江大学

赵宝江 佳木斯大学

秦 锋 安徽工业大学

凌 云 浙江工商大学

常中华 青岛职业技术学院

眭碧霞 常州信息职业技术学院

软件技术专业委员: (以姓氏笔画为序)

王 军 沈阳化工大学

王国贞 河北工业职业技术学院

王桂霞 内蒙古师范大学

王渝俊 四川科技职业技术学院

王 婷 湘潭大学

邓曦辉 晋中学院

甘诚志 宜春学院

吕振雷 黄河水利职业技术学院

朱 强 中原工学院信息商务学院

刘中华 辽宁机电职业学院

刘庆杰 防灾科技学院

刘彦宾 遵义师范学院

许海成 红河学院

李文斌 河北地质大学

李存华 淮海工学院

李庆年 南宁学院

李 林 重庆电子工程职业学院

李 忠 宜宾学院

杨 喜 吉首大学

何 淼 南京信息职业技术学院

张 伟 枣庄学院

张 婷 唐山学院

陈 哲 商丘职业技术学院

陈 琳 长江大学

邵雪航 黑龙江工商学院

范剑波 宁波工程学院

赵向军 江苏师范大学

赵 芳 新乡学院

赵竞雄 华北科技学院

姚 湘 无锡太湖学院

耿焕同 南京信息工程大学滨江学院

徐庆生 楚雄师范学院

黄先珍 菏泽学院

黄星寿 河池学院

商信华 信阳农林学院

梁 丰 浙江万里学院

程世辉 河南教育学院

程曾平 湖北汽车工业学院

焦 战 辽宁轻工职业学院

赖俊峰 内蒙古工业大学

谭 靖 华北理工大学轻工学院

魏海平 辽宁石油化工大学

III 总序

一直以来，我们都在探索信息技术（IT）教学的新方法，如何把看似复杂深奥的编程知识教给完全没有基础的学员，让编程不再难学，这是我们在过去的十年中一直追寻的目标。为此，我们尝试了一系列的方法去为零基础的学员讲解编程的技术，最初，我们以知识点为导向，将编程技术分解为若干个知识点，力图使我们的课程完整地覆盖所有的知识点，我们发现这样的培训方式收效甚微，大多数人能背下这些知识点，但并不能灵活地运用它们。我们知道，要编写一个完整的程序涉及方方面面的知识，也就是说知识其实是张网。后来，我们又希望能用案例和项目贯穿课程的始终，使用案例来讲解知识点，能很好地帮助学员理解和运用知识，但我们同时也发现，案例的设计和难度的把控，以及与知识点的契合程度很难把握。我们似乎找到了教学的法门，但却又不知如何去运用好这个法门。

最后，我们放弃一切已知的思维定势，从最根本的人本认知需求出发，我们认为，良好的教育应该是以人为本的教育，是发现知识和运用知识的过程，以人为本就是以人的认知曲线为依据，跟随人们发现知识的过程来设计案例。经过长期的实践和不断的摸索，我们整理了一套基于认知曲线的案例教学法。我们认为，一个初学编程的人的认知曲线分为：单词、语句、小段逻辑、完整程序等几个认知阶段，在不同的认知阶段，学员的关注点不同，如何将学员的关注点与要学习的知识有机地结合起来，我们想到按照学员的认知曲线去设计这些案例，这才是符合人的学习规律的，是以人为本的课程设计。

这套系列教材正是基于我们这一教学方法的具体产物，我们希望通过一系列按需设计的案例的贯穿来完整地完成一个有一定技术价值的程序，通过这一系列的程序来学习和理解知识的运用过程。因此，本套教材在编写初期就采用了人本认知的方式对所有的知识点进行了重构，所有章节内容的讲解方式一律遵循“知识介绍”“知识运用”“案例实战”“知识总结”这样一套学习方案。读者通过浏览目录会发现，在每个章节中都存在一个或多个以“【案例】”标识的节点，每一个案例都可延伸拓展为一个应用程序。而在全书的知识点讲解中还涉及大量的代码段的讲解，这其中又存在多个小案例，这些小案例的数量多达几十个。这样的小案例也方便读者对知识有更加全面的认识。每一个大大小小的案例都是精心设计的，我们坚信，通过这样一种案例式驱动的教学方式，既可以摆脱传统知识点学习的枯燥又可以很快地让读者掌握每一个知识点。

本系列教材的另一个突出特色是，作为“用微课学”系列教材，我们为每一个章节中的重要知识点录制了一段微课视频，这样的微课视频数量很多。这些视频都是我们在实际授课中的经验总结和教学方法呈现。通过这些微课视频，我们希望能帮助读者更进一步准确理解所学知识。同时，本系列教材还提供了整体设计、单元设计、电子课件、项目案例、源代码、工程资源等丰富的电子资源供读者使用。

III 前言

Java 是一种可以撰写跨平台应用程序的、面向对象的程序设计语言。Java 技术具有卓越的通用性、高效性、平台移植性和安全性，是一门真正做到“一次编译，到处运行”的高级语言。Java 语言多年来深受计算机开发者的喜爱。目前，在全球云计算、大数据、移动互联网日益繁荣的产业环境下，Java 更是具备了显著的优势和广阔前景。古有学好数理化走遍全天下，而在科技时代的今天，学好 Java 语言，会熟练运用 Java 语言，就可以在科技行业中任意畅游。

本书主要介绍 Java 高级编程技术，其针对有一定 Java 基础的人员所编写，注重 java 高级编程的应用。全书共分为 7 章，包括：Java 桌面图形界面开发、Java I/O、Java 多线程、Java 网络编程、JDBC、XML 和 JSON、Java 反射和注释。每一章节包含基础知识、阶段案例、练习、总结等内容并配套若干微课视频。

本书以提升 Java 高级技术为目标，注重 Java 开发高级技术的应用。通过对人的认知曲线的分析，我们总结了新的案例式教学法，进而设计了全新的学习方案。本书的设计打破了传统的教学方式，不再一味的说教，而是带着读者真正的“玩”起来。通过对本书的学习，可以强化读者分析问题和解决问题的能力，激发读者的创新实践能力，真正的做到学以致用。

本书凝聚了编者多年的教学和实践经验，但由于水平有限，疏漏之处在所难免，欢迎广大读者提出宝贵意见。

编者

2016 年 2 月 10 日

java

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任；构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人进行严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话 (010) 58581999 58582371 58582488

反盗版举报传真 (010) 82086060

反盗版举报邮箱 dd@hep.com.cn

通信地址 北京市西城区德外大街4号

高等教育出版社法律事务与版权管理部

邮政编码 100120

防伪查询说明

用户购书后刮开封底防伪涂层，利用手机微信等软件扫描二维码，会跳转至防伪查询网页，获得所购图书详细信息。用户也可将防伪二维码下的20位密码按从左到右、从上到下的顺序发送短信至106695881280，免费查询所购图书真伪。

反盗版短信举报

编辑短信“JB，图书名称，出版社，购买地点”发送至10669588128

防伪客服电话

(010) 58582300

资源服务提示

欢迎访问职业教育数字化学习中心——“智慧职教”（www.icve.com.cn），以前未在本网站注册的用户，请先注册。用户登录后，在首页或“课程”频道搜索本书对应课程“Java高级开发”进行在线学习。注册用户也可以在“智慧职教”首页或扫描本页右侧提供的二维码下载“智慧职教”移动客户端，通过该客户端选择本课程进行在线学习。

扫描下载官方APP



如果采用本教材的教师希望以本课程为基础，搭建面向本校学生的小规模专属在线课程（SPOC）开展翻转课堂式教学，可向高等教育出版社申请在“智慧职教云”平台（www.icve.com.cn）上开通专属云空间试用半年。申请事宜可用电子邮件联系本书责任编辑。

本书责任编辑邮箱：1548103297@qq.com

III 目录

第 1 章 Java 桌面图形界面开发	1	1.9.2 Graphics 类	31
1.1 GUI 基本组件	2	1.10 案例 满天星星	32
1.1.1 GUI 概念	2	1.10.1 案例描述	32
1.1.2 Swing 概述	2	1.10.2 案例分析	32
1.1.3 Swing 分类	2	1.10.3 案例代码	33
1.1.4 Swing 组件的层次结构	3	本章小结	37
1.1.5 Swing 组件	4	第 2 章 Java I/O	39
1.2 布局管理器	8	2.1 File 类	40
1.2.1 布局管理器概述	8	2.1.1 文件基本概念	40
1.2.2 FlowLayout 布局管理器	8	2.1.2 File 类中常用的方法	40
1.2.3 BorderLayout 布局管理器	10	2.1.3 文件过滤器	41
1.2.4 GridLayout 布局管理器	12	2.2 RandomAccessFile 类	42
1.3 案例 计算器的布局	14	2.3 Java I/O 流的分类	44
1.3.1 案例描述	14	2.3.1 流的分类	44
1.3.2 案例分析	15	2.3.2 流的概念	45
1.3.3 案例代码	15	2.4 字节流	45
1.4 事件处理	16	2.4.1 InputStream 和 OutputStream	45
1.4.1 事件和事件监听器	16	2.4.2 FileInputStream 和 FileOutputStream	47
1.4.2 事件适配器	17	2.4.3 字节缓冲流	49
1.4.3 监听器的实现方式	19	2.5 案例 文件加密器	52
1.4.4 使用内部类的方式实现	19	2.5.1 案例描述	52
1.4.5 类本身作为事件监听器	20	2.5.2 案例分析	52
1.4.6 匿名内部类实现方式	20	2.5.3 案例代码	53
1.4.7 计算器功能的实现	21	2.6 字符流	57
1.5 其他事件	23	2.6.1 文件和文本文件	58
1.6 案例 鼠标事件	24	2.6.2 Reader 和 Writer	58
1.6.1 案例描述	24	2.6.3 字符流的基本实现	59
1.6.2 案例分析	24	2.6.4 字符流的缓冲流	61
1.6.3 案例代码	24	2.6.5 PrintWriter 类	63
1.7 案例 下拉列表框的处理	26	2.7 文本文件读写	64
1.7.1 案例描述	26	2.8 案例 简易记事本	65
1.7.2 案例分析	26	2.8.1 案例描述	65
1.7.3 案例代码	26	2.8.2 案例分析	65
1.8 案例 动态调色板	28	2.8.3 案例代码	66
1.8.1 案例描述	28	本章小结	70
1.8.2 案例分析	28	第 3 章 Java 多线程	71
1.8.3 案例代码	29	3.1 线程	72
1.9 绘图的基本操作	31		
1.9.1 绘图的实现原理和基本概念	31		

3.1.1	线程概述	72	4.2.2	URLDecoder 和 URLEncoder	99
3.1.2	Java 中的线程	72	4.2.3	URL 和 URLConnection	100
3.1.3	使用线程的原因	72	4.3	基于 TCP 协议的网络编程	101
3.2	创建多线程的程序方式	73	4.3.1	TCP 协议基础	101
3.2.1	继承 java.lang.Thread 类	73	4.3.2	服务器端 ServerSocket	102
3.2.2	实现 java.lang.Runnable 接口	73	4.3.3	用 Socket 创建客户端	103
3.3	案例 流星的移动	74	4.3.4	用 Socket 进行通信	104
3.3.1	案例描述	74	4.3.5	多个客户端	106
3.3.2	案例分析	74	4.4	案例 广播聊天室	108
3.3.3	案例代码	74	4.4.1	案例描述	108
3.4	案例 抽奖器	78	4.4.2	案例分析	109
3.4.1	案例描述	78	4.4.3	案例代码	109
3.4.2	案例分析	79	4.5	基于 UDP 协议的网络编程	114
3.4.3	案例代码	79	4.5.1	UDP 协议基础	114
3.5	线程生命周期	83	4.5.2	DatagramSocket 发送接收数据	114
3.5.1	创建和就绪状态	83	本章小结		117
3.5.2	运行和阻塞状态	83	第 5 章 JDBC		119
3.5.3	线程死亡	84	5.1	JDBC 原理	120
3.6	线程操作	84	5.1.1	JDBC 概述	120
3.6.1	join 线程	84	5.1.2	JDBC 中常用的接口和类	120
3.6.2	后台线程	85	5.1.3	JDBC 驱动程序	122
3.6.3	线程睡眠	86	5.2	JDBC 编程步骤	123
3.6.4	线程让步	86	5.2.1	注册 Driver	123
3.6.5	线程优先级	87	5.2.2	建立连接	124
3.7	线程同步	88	5.2.3	获得 Statement 对象	124
3.7.1	线程安全问题	88	5.2.4	执行 SQL 语句	124
3.7.2	线程并发演示	89	5.2.5	处理结果集	124
3.7.3	线程同步方法	90	5.2.6	关闭数据库连接	125
3.8	案例 动态小球	92	5.3	案例 对用户的增删查改	126
3.8.1	案例描述	92	5.3.1	案例描述	126
3.8.2	案例分析	92	5.3.2	案例分析	127
3.8.3	案例代码	92	5.3.3	案例代码	127
本章小结		96	5.4	JDBC 中的异常和事务	133
第 4 章 Java 网络编程		97	5.4.1	JDBC 中的异常类型	133
4.1	网络基础	98	5.4.2	JDBC 中的事务编程	133
4.1.1	网络概述	98	5.5	JDBC 扩展	135
4.1.2	IP 地址和端口	98	5.5.1	批处理	135
4.2	Java 中的网络编程	98	5.5.2	分页	136
4.2.1	InetAddress 类	98	5.5.3	调用存储过程	139

本章小结	142	6.5.3 Java 对 JSON 的解析	175
第 6 章 XML 和 JSON	143	本章小结	179
6.1 XML 语法规则	144	第 7 章 Java 反射和注释	181
6.1.1 标记语言	144	7.1 类的加载、连接和初始化	182
6.1.2 创建一个简单的 XML	144	7.1.1 JVM 和类	182
6.1.3 XML 声明	146	7.1.2 类的加载	182
6.1.4 标记	147	7.1.3 类的连接	183
6.1.5 元素	147	7.1.4 类的初始化	183
6.1.6 实体引用 (转义字符)	148	7.1.5 类初始化的时机	184
6.1.7 属性	149	7.2 类加载器	185
6.1.8 CDATA 类型的数据	149	7.2.1 类加载器概述	185
6.1.9 注释	150	7.2.2 类加载机制	186
6.1.10 示例程序	151	7.2.3 创建并使用自定义类加载器	186
6.2 DTD 和 Schema	151	7.2.4 URLClassLoader	190
6.2.1 DTD/Schema 的由来	151	7.3 反射	190
6.2.2 文档类型定义 DTD	152	7.3.1 反射概述	190
6.2.3 Schema	155	7.3.2 获得 Class	191
6.3 Java 对 XML 的解析	156	7.4 反射获取对象信息	191
6.3.1 DOM 解析	156	7.5 创建对象和访问对象	195
6.3.2 SAX 解析	159	7.5.1 通过反射创建对象	195
6.3.3 JDOM 解析	162	7.5.2 通过反射访问对象的属性	196
6.3.4 DOM4J 解析	164	7.5.3 反射访问对象的方法	197
6.3.5 通过 DOM4J 生成 XML 文档	167	7.6 注解	198
6.4 XPath	169	7.6.1 基本注解	198
6.4.1 XPath 定义	169	7.6.2 自定义注解	200
6.4.2 路径表达式	170	7.7 案例 构造简易的 Hibernate	203
6.4.3 谓词	170	7.7.1 案例描述	203
6.4.4 通配符	171	7.7.2 案例分析	203
6.4.5 Java 对 XPath 的支持	171	7.7.3 案例代码	203
6.5 JSON	173	本章小结	216
6.5.1 JSON 概述	173		
6.5.2 JSON 数据格式	174		

第1章 Java 桌面图形界面开发



本章重点

本章介绍了Java中GUI组件AWT和Swing之间的关系,并且介绍了Swing分类、Swing组件的层次结构和常用的Swing组件。

紧接着通过代码描述了如何新建一个图形界面,以及容器和组件的用法,介绍常用的3种布局管理器,并且通过代码描述了每种布局的用法,最后通过一个计算器的案例把3种布局综合起来应用。

然后描述了Java AWT的事件机制,主要了解事件、事件源、事件监听器,需要重点掌握实现事件监听器的几种方式,最后结合计算器功能的实现对事件进行了应用。

最后描述了Java中的绘图机制,通过一个“满天星星”案例讲解了如何绘图,读者需要重点掌握。



本章资源

1. 文本 课程标准
2. 文本 整体设计
3. 文本 第1章 章节设计
4. PPT 第1章 Java 桌面图形界面开发
5. 微视频 001 GUI 基本组件_01
6. 微视频 002 GUI 基本组件_02
7. 微视频 003 布局管理器_01
8. 微视频 004 布局管理器_02
9. 微视频 005 计算器的布局
10. 微视频 006 事件和事件监听器
11. 微视频 007 事件适配器
12. 微视频 008 使用内部类的方式实现监听器
13. 微视频 009 类本身作为事件监听器
14. 微视频 010 匿名内部类实现监听器
15. 微视频 011 计算器功能的实现_01
16. 微视频 012 计算器功能的实现_02
17. 微视频 013 其他事件
18. 微视频 014 鼠标事件_01
19. 微视频 015 鼠标事件_02
20. 微视频 016 下拉列表框的处理
21. 微视频 017 动态调色板_01
22. 微视频 018 动态调色板_02
23. 微视频 019 绘图的基本操作
24. 微视频 020 满天星星
25. 案例源代码: chapter_01.
code_01 (计算器的布局)
26. 案例源代码: chapter_01.
code_02 (鼠标事件)
27. 案例源代码: chapter_01.
code_03 (下拉列表框的处理)
28. 案例源代码: chapter_01.
code_04 (动态调色板)
29. 案例源代码: chapter_01.
code_05 (满天星星)



文本 第 1 章
章节设计



PPT 第 1 章
Java 桌面图形界面
开发

PPT



微视频 001
GUI 基本组件_01



微视频 002
GUI 基本组件_02

1.1 GUI 基本组件

1.1.1 GUI 概念

图形用户界面 (Graphical User Interface, GUI) 是用图形的方式, 借助菜单、按钮等标准界面元素和鼠标操作, 帮助用户方便地向计算机系统发出指令、启动操作, 并将系统运行的结果以图形方式显示给用户的技术。图形用户界面与字符界面相比, 操作简单, 画面生动, 深受广大用户的喜爱, 已经成为目前几乎所有应用软件的既成标准。本章主要讲述 Java 的图形界面技术, 即使用 `javax.swing` 包来创建图形用户界面的方法, 包括 Swing 组件、窗口和菜单设计、布局管理器、绘图操作等。

1.1.2 Swing 概述

Java 使用 AWT 和 Swing 包下面的类来完成图形用户界面编程。其中, AWT 是 Sun 公司最早提供的 GUI 库, 这个 GUI 库提供了一些基本功能, 但是它提供的一些类在使用时比较繁琐, 提供的功能也比较局限, 后来又提供了一套类库, 即 Swing 包下的库。通过使用 AWT 和 Swing 提供的图形用户界面组件库可以完美地编写出图形用户界面, 在程序中只要依次创建所需要的图形组件, 并以合适的方式将这些组件组织在一起就可以开发出完美的用户界面。

使用 AWT 创建的图形用户界面应用和所在的运行平台有相同的界面风格, 比如在 Windows 操作系统上, 它表现出 Windows 风格; 而在 UNIX 操作系统上, 它则表现出 UNIX 风格。因为用 AWT 开发出的界面非常死板并且功能有限, Sun 公司随后提供的 Swing 替代了大部分 AWT 组件, 对 AWT 图形用户界面有了很好的补充和加强。但是要注意, Swing 并不是完全替代了 AWT 组件, 而是建立在 AWT 基础之上。

程序以一种拼凑的方式将这些图形用户界面的组件组织起来, 就是我们可以看到的用户界面, 但这些用户界面还不能与用户交互, 为了实现图形用户界面和用户交互操作, 还应该为程序提供事件处理。

1.1.3 Swing 分类

Swing 分为基本组件和容器。图 1-1 是大家所熟悉的 QQ 登录界面, 该界面中有输入账号和密码的输入框, 也有可以直接用鼠标单击的“登录”按钮, 这些输入框和按钮都是基本组件。

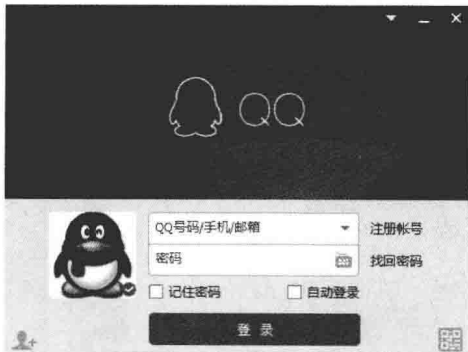


图 1-1
QQ 登录界面

基本组件是图形用户界面的最小单位，里面不再包含其他的成分，如按钮（JButton）、复选框（JCheckBox）、选择框（JcomboBox）、列表（JList）、文本域（JTextField）、多行文本域（JTextArea）、静态文本（JLabel）、滚动条（JScrollBar）等。

了解基本组件后下面介绍容器这个概念。细心观察 QQ 登录界面，发现界面中有输入框、登录按钮、多选框等基本组件，那这些基本组件是放在什么地方组合在一起展现给用户的？这个把多个基本组件组合在一起的就是容器。在 Java Swing 中，容器是一种特殊的组件，用来包含其他组件，如面板（JPanel）、窗口（Window）、对话框（JDialog）、文件对话框（FileDialog）、框架（JFrame）等。

1.1.4 Swing 组件的层次结构

Swing 组件都是 Sun 公司提供给开发者的类和接口，这些组件之间有继承关系。Component 组件是所有 AWT 组件的基类，它提供了基本的显示和事件处理特征，Container 类和其他 UI 组件都是从 Component 中派生出来的。图 1-2 所示为 Swing 组件的类层次结构图。

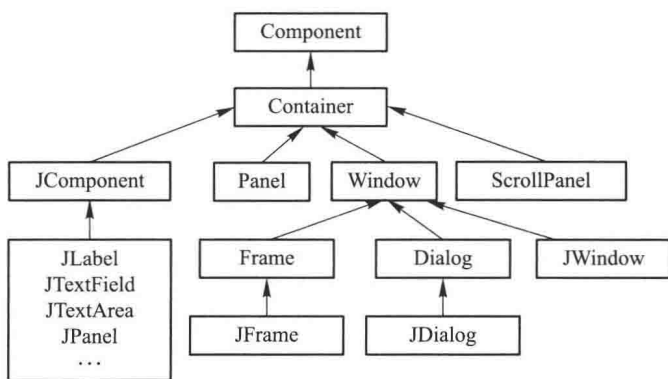


图 1-2
Java 图形用户界面开发组件继承关系

从图 1-2 可看出组件之间的继承关系，这么多组件，读者看起来可能比较凌乱。在 Swing 分类小结中简单描述了容器和组件，在图 1-2 中读者可以看到 Container 类下面有四个子类，这四个子类中到底哪些是容器哪些是组件？其实在这四个类中，除了 JComponent 类以外，其余三个都是 Java AWT 包中的类，而 JComponent 是 Java Swing 中所出现的类。Sun 公司在设计 Java Swing 时用 JComponent 把容器和组件由 Java AWT 过渡到 Java Swing，也就是说基本上大部分 Java Swing 中的容器和组件都继承 JComponent 类。

Container 类中除了 JComponent 类以外，其余三个都是 Java AWT 中的容器，这三个容器怎么使用呢？这里需要注意，在容器中又分为顶级容器和中间容器。顶级容器是指图形用户界面最外层的容器，如窗口（Window）。中间容器是指存放在顶级容器或者中间容器中的一种容器，如面板（Panel）。在实际编程中，要开发一个界面应该是这样的，顶级容器在最外面（JFrame），其次是中间容器，放在窗体上（JPanel），最后和基本组件，放在中间容器上。这里要注意，容器中除了可以放基本组件外，还可以存放容器，具体如图 1-3 所示。

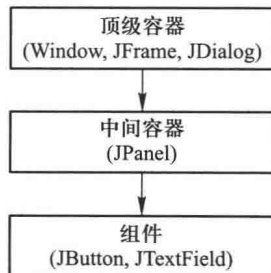


图 1-3
容器和组件应用关系

在图 1-2 中经常使用的顶级容器是窗体 (JFrame) 和对话框 (JDialog)，使用的中间容器是面板 (JPanel)。开发中经常使用的基本组件大部分用 JComponent 类下的子类，读者会发现 JComponent 类中包含了容器 JPanel 和基本组件，在这里不需花费太多精力去记住这些类之间的关系，后面会通过具体实例来讲解每个组件的用法，以及开发一个完整图形用户界面的流程。

Component 类的主要方法有：

- (1) void enable(): 使组件可用。
- (2) void disable(): 使组件不可用。
- (3) void show(): 显示组件。
- (4) void paint(): 绘制图形。
- (5) void repaint(): 刷新。

所有的 UI 组件都可继承或重载以上方法。

Container 类的主要方法有：

- (1) void add(Component c): 将指定组件加入到容器中。
- (2) void setLayout(): 设置布局管理器。

所有的包容器组件都可继承或重载以上方法。

1.1.5 Swing 组件

Java 提供了很多可以直接使用的基本组件，这里介绍一些比较常用的基本组件和容器。

1. JFrame 组件

JFrame 是独立于浏览器的可独立运行的主窗口，通常用于开发桌面应用程序。JFrame 具有边界和标题栏设置，其大小可以调整。JFrame 中需要加入组件，也可以加入菜单，或上面绘图。图 1-4 就是一个非常简单的窗体。

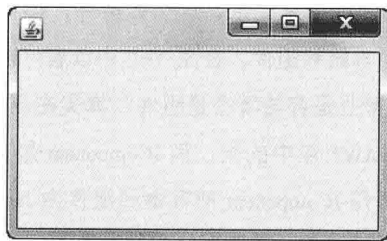


图 1-4 简单的窗体

看到这个简单的窗体，读者觉得写这样一个窗体复杂吗？下面是具体实现代码：

```
package com.tarena.javase.gui;
import javax.swing.JFrame;
public class JFrameDemo {
    public static void main(String[] args) {
        // 创建一个窗体对象
        JFrame f = new JFrame();
        // 设置窗体可见
```

```
f.setVisible(true);
}
}
```

上面程序的代码非常简单，主方法中的两行代码就可以创建一个窗体。运行程序看结果，这时你要细心看计算机屏幕的左上角会出现一个非常小的窗体，这个窗体可以用鼠标拉大或缩小，主方法中第一行代码是创建一个窗体对象，第二行代码是设置窗体可见。这里要注意默认创建的窗体是隐藏的，如果不调用第二行代码，则窗体是不可见的。

上面程序展示了如何创建一个简单的窗体，如果想要用代码设置窗体默认的大小或者窗体在屏幕上显示的位置，则可以调用 `JFrame` 中提供的对应的方法。

`JFrame` 组件中提供了如下两个常用的构造方法：

- (1) `JFrame()`：创建一个没有窗口标题的窗口框架。
- (2) `JFrame(String)`：创建一个指定窗口标题的窗口框架。

`JFrame` 的主要方法见表 1-1。

表 1-1 `JFrame` 的主要方法

方 法	功 能
<code>addNotify()</code>	创建一个对等的 <code>Frame</code>
<code>dispose()</code>	清除窗口对象
<code>getIconImage()</code>	获得窗口的图标
<code>getMenuBar()</code>	获得窗口的菜单栏
<code>getTitle()</code>	获得窗口的标题
<code> paramString()</code>	返回窗口的参数字符串
<code>isResizable()</code>	如果窗口大小可变，则返回真值
<code>remove(MenuComponent)</code>	从窗口中去掉指定的菜单栏
<code>setIconImage(Image)</code>	设置窗口的图标

2. `JLabel` 组件

`JLabel` 组件显示的是静态文本或者图片，在通常情况下是不能编辑的，只起到提示的作用。观察 QQ 登录界面输入框左侧的头像，它就可以用 `JLabel` 组件实现。

`JLabel` 组件中主要的构造方法如下：

- (1) `JLabel()`：创建空的标签。
- (2) `JLabel(String)`：创建一个带初始字符串的标签。
- (3) `JLabel(Icon)`：创建一个带有指定图像的标签。

`JLabel` 的主要方法见表 1-2。

表 1-2 JLabel 的主要方法

方 法	功 能
setText(String)	设置显示内容
getText()	获取显示内容
getAlignment()	获取对齐方式
setAlignment()	设置对齐方式

3. 文本域 (JTextField)

文本域一般用来让用户输入姓名、信用卡号这样的信息，它是一个能够接收用户的键盘输入的小块区域。

在创建文本域时，参数类型有四种：空的、空的并且具有指定长度、带有初始文本内容的和带有初始文本内容并具有指定长度的。

文本域的主要方法见表 1-3。

表 1-3 文本域的主要方法

方 法	功 能
getColumns()	返回文本域的列数
getEchoChar()	返回用于回显的字符
setEchoCharacter(Char)	设置文本域的回显字符
setEditable(Boolean)	设置文本域是否为只读状态
setText(String)	设置文本域中的文本值
getText()	返回文本域中的输入文本值

4. 按钮 (JButton)

几乎所有的交互式程序中都会有按钮，它是一种交互能力强且交互方便的控件，这个组件提供了“按下并动作”的基本用户界面。

构造方法如下：

- (1) JButton(): 建立一个没有标示字符串的新按钮类对象。
- (2) JButton(String label): 建立一个标示字符串为 label 的新按钮类对象。

下面通过一个简单的程序对刚才所描述的组件加以应用，具体代码如下：

```
package com.tarena.javase.gui;

import javax.swing.JFrame;

public class JFrameDemo {
    public static void main(String[] args) {
        // 创建一个窗体对象
        JFrame f = new JFrame();
        // 设置窗体大小为 200 宽 200 高
    }
}
```