



计 算 机 类 本 科 规 划 教 材

Java 程序设计及应用

——增量式项目驱动一体化教程

◆ 苑俊英 陈海山 主编



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

计算机类本科规划教材

Java 程序设计及应用

——增量式项目驱动一体化教程

苑俊英 陈海山 主编

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书共4部分:第一部分和第三部分为Java程序设计基本技能,第二部分和第四部分为Java语言实训内容。在教学安排上,教师可以打乱基本技能的编写顺序,通过项目驱动的形式进行基本技能的讲解,即在第二部分简易计算器项目基础上,进行第一部分内容的融合与展开;在第四部分局域网聊天工具项目基础上,进行第三部分内容的融合与展开。

本书既可作为计算机及相关专业的Java语言课程的教材,也可作为工程人员和科技工作者的自学参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Java 程序设计及应用:增量式项目驱动一体化教程/苑俊英,陈海山主编. —北京:电子工业出版社,2013.11
计算机类本科规划教材

ISBN 978-7-121-20764-8

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

中国版本图书馆CIP数据核字(2013)第137095号

策划编辑:章海涛

责任编辑:陈晓莉

印 刷:三河市鑫金马印装有限公司

装 订:三河市鑫金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

开 本:787×1 092 1/16 印张:22.25 字数:560千字

印 次:2013年11月第1次印刷

定 价:45.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前言

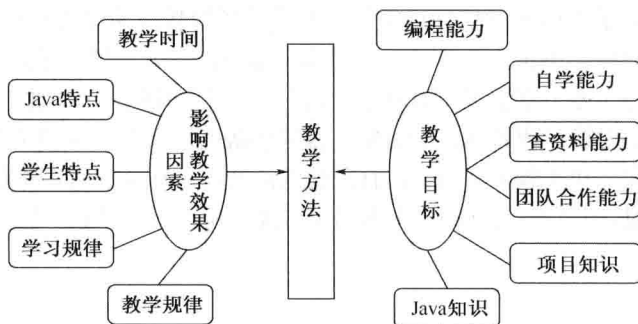
Java 程序设计语言是一种可以编写跨平台应用程序的面向对象的程序设计语言，具有超强的通用性、高效性、平台移植性和安全性，广泛应用于个人计算机、数据中心、游戏控制台、科学超级计算机、移动电话和互联网，同时拥有全球最大的开发者专业社群。自 Sun Microsystems 公司于 1995 年 5 月推出以来，Java 程序设计语言是当今最具代表性的面向对象编程语言之一，也是实际软件项目开发中所使用的主流编程语言之一。在全球云计算和移动互联网产业蓬勃发展的环境下，Java 更具备了显著优势和广阔前景。

在本科教学计划中，Java 一般都安排在编程设计基础课程（一般为 C 语言）之后。Java 与 C 语言相比，其基本语法十分相似，但 Java 的应用性更强，入手也相对容易。如果说学习 C 语言是给打好基础的话，那么学习 Java 语言则更应该侧重提高学生的编程能力和查阅资料能力。我们应该利用这些知识的内在联系，以及教学侧重点的不同，在教学过程中予以体现。然而，传统的 Java 教学模式都是把相当多的课时花费在 Java 的基本语法上，既没有发挥 Java 应用性强的特点来调动学生的学习积极性，也没有把 Java 的优势展示给学生，导致本科生的 Java 语言应用水平往往还比不上大专的学生。

编者认为，Java 程序设计知识的结构并不决定教学的顺序和组织方式，所以，在本科 Java 语言的教学过程中，应该利用学生已经具备的语法基础和 Java 语言应用性强的特点，引导学生重点学习 Java 的高级应用特性，通过在实际项目应用中展现 Java 语言的高级特性，来调动学生的积极性，提高学生的动手编程能力和独立查阅资料文献的能力。由于编程的核心是应用数据结构和算法去解决实际解决事物间逻辑关系，教学的过程也必须以解决实际问题为核心，通过实际问题让学生理解和掌握数据结构与算法在实际中的应用，最终提高学生的编程能力。

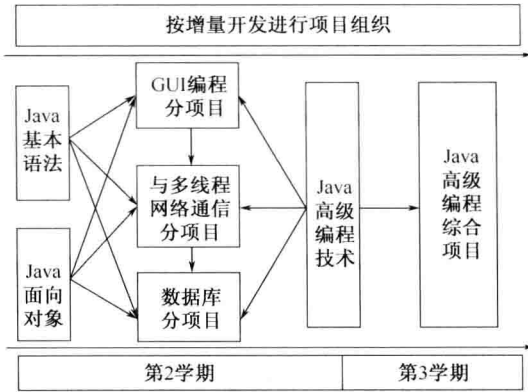
编者认为，Java 教学过程应该有几个侧重：通过高级编程技术调动学生学习积极性，通过由简单到复杂的实验使学生掌握 Java 语法的应用，通过实际项目的完成提高学生编程能力。然而，目前市场上流行的 Java 参考书大部分按传统的语法方式讲解 Java，有些则直接给出了若干项目的实现，也有少数参考书提出了利用游戏或者多媒体的方法来教学，但没有一本参考书综合考虑了 Java 的特点、学生的学习情况和学习规律。

下图中展示了影响 Java 课程教学效果的若干因素，以及其与教学方法和教学目标之间的关系，其中影响教学效果的若干因素和教学目标共同决定了应该采用什么样的教学方法。



在这样的背景下，编者结合多年的教学经验和工程经验，综合已有的参考书各自的优点，在糅合 Java 的理论、实验和实训教学的基础上，编写了这本以项目为主线的 Java 综合教程，旨在通过由浅入深、由小到大的项目教学和实践过程，有机地组织教学顺序、引导学生把 Java 的知识点融入程序设计中，调动学生的积极性，使得学生在快乐学习的过程中，最终提高学生的编程能力和查阅资料的能力。

此教学方法从 Java 理论学习与实验操作、Java 知识的掌握与编程能力的提高、教学要求与学生多方面能力提高等方面进行了统一安排，达到了多个课程目标的一体化实现（见下图）。



在采用项目教学的过程中，必须设置相应的课程设计：能够体现 Java 高级编程应用的实例，通过模仿编程巩固知识的掌握、提高学生编程能力。该方法已经在中山大学南方学院的 Java 程序设计课程中采用，经过 2010 级和 2011 级计算机专业两届学生的实践证明：采用一体化增量式项目教学可以在很大程度上提高学生的实践能力。

本书共 4 部分：第一部分和第三部分为 Java 程序设计基本技能，第二部分和第四部分为 Java 实训内容。其中第二部分和第四部分实训内容，是在第一部分和第三部分 Java 基本技能基础上的提升。在教学安排上，可以打乱基本技能的编写顺序，通过项目驱动的形式进行基本技能的讲解。即在第二部分简易计算器项目基础上，进行第一部分内容的融合与展开；在第四部分局域网聊天工具项目基础上，进行第三部分内容的融合与展开。本书编写体例：为方便读者阅读，本书第一、三部分的程序以“示例”的形式描述；第二、四部分中的项目框架和实现以“代码”的形式描述，某些知识点的例子程序用“例程”的形式描述。

本书作者苑俊英具有多年的教学经验，陈海山具有多年在国内外知名企业的工程实践经验。本书在编写过程中得到了中山大学信息科学与技术学院杨智教授的支持与帮助，在此表示诚挚的谢意。在本书完成过程中，中山大学南方学院的蔡力能、陈天富、陈振杰、李满铨、王添、姚骏琳、朱彦瑾等同学参与了本书实验及实训项目，陈泽升、陈思远、丁敏如、洪笃辉、李聃、林南鹏、倪坪雄、舒挺、谢思妙、谢智锋、张海威、张清伟等同学参与了本书的校对工作。

本书还配有教学课件和实验实训例程，有需要的读者可与作者联系，邮箱为 cihisa@126.com。由于作者水平有限，编写时间仓促，在本书中可能会出现一些错误，恳请读者提出宝贵意见。

作者
2013 年 8 月

目 录

第一部分 Java 程序设计基本技能 (一)

第 1 章 Java 的特点、基本内容和编程环境.....2

1.1 Java 语言的历史和特点.....2	
1.1.1 Java 语言的发展历史.....2	
1.1.2 Java 语言的特点.....2	
1.2 本书基本内容与教学思路.....4	
1.3 Java 开发环境介绍.....4	
1.3.1 JDK 的下载、安装及环境变量的设置.....4	
1.3.2 Eclipse 的安装及使用.....7	
1.3.3 MyEclipse 8.5.0 安装、界面介绍.....9	
1.3.4 MySQL 的安装及功能介绍.....10	
1.3.2 程序调试技术.....14	
1.4 项目开发模板.....14	
1.4.1 增量开发简介.....14	
1.4.2 本书项目开发采用的模板.....15	
实验 01——开发环境搭建与使用.....16	
习题.....17	

第 2 章 Java 应用基础.....18

2.1 标识符和关键字.....18	
2.2 数据类型.....19	
2.2.1 基本数据类型.....19	
2.2.2 复合数据类型.....20	
2.2.3 枚举类型.....20	
2.2.4 类型转换.....21	
2.3 变量和常量.....22	
2.3.1 常量.....22	
2.3.2 变量.....23	
2.4 运算符和表达式.....26	
2.4.1 赋值运算符.....26	
2.4.2 算术运算符.....26	
2.4.3 关系运算符.....27	
2.4.4 条件运算符.....27	
2.4.5 逻辑运算符.....27	

2.4.6 位运算符.....28	
2.4.7 其他运算符.....28	
2.4.8 运算的优先级和结合性.....29	
2.4.9 表达式.....29	
2.5 程序控制结构.....32	
2.5.1 选择结构.....32	
2.5.2 循环结构.....36	
2.5.3 跳转语句.....40	
2.6 本章小结.....42	
习题.....42	

第 3 章 面向对象编程.....44

3.1 对象和类.....44	
3.2 句柄.....48	
3.3 方法的调用和参数的传递.....48	
3.4 类的继承和多态.....55	
3.4.1 继承.....55	
3.4.2 多态.....60	
3.5 包和接口.....62	
3.6 内部类与匿名的内部类.....67	
3.7 本章小结.....68	
习题.....68	

第 4 章 数组、字符串、向量和泛型.....69

4.1 数组.....69	
4.1.1 一维数组.....69	
4.1.2 多维数组.....73	
4.2 增强的 for 循环.....75	
4.3 字符串和字符串缓冲区.....76	
4.3.1 String.....77	
4.3.2 StringBuffer.....79	
4.4 向量.....81	
4.5 集合-数组链表 ArrayList.....83	
4.6 泛型.....85	
4.6.1 使用泛型的原因.....85	
4.6.2 泛型在集合类中的使用.....86	

4.6.3 定义泛型类和泛型接口	86
4.7 枚举	88
4.8 本章小结	89
习题	89
第5章 图形用户界面	91
5.1 Swing 概述	91
5.1.1 Swing 组件	91
5.1.2 Swing 容器	93
5.1.3 基于Swing的JavaGUI设计思路	97
5.2 布局管理器	97
5.2.1 FlowLayout 流式布局管理器	97
5.2.2 GridLayout 网格布局管理器	99
5.2.3 BorderLayout 边界布局管理器	100
5.2.4 CardLayout 卡片布局管理器	101
5.2.5 GridBagLayout 网格袋布局管理器	102
5.2.6 BoxLayout 盒式布局管理器	103
5.3 事件处理	104
5.3.1 Java 事件处理机制	105
5.3.2 鼠标事件处理	106
5.3.3 按钮动作事件处理	108
5.4 高级图形用户界面	111
5.4.1 菜单	111
5.4.2 表格	122
5.4.3 树	123
5.5 本章小结	125
习题	125

第二部分 JavaGUI 实训——简易计算器

第6章 标准型计算器	128
6.1 需求分析与项目目标	128
6.1.1 需求分析	128
6.1.2 项目目标	132
6.2 功能分析与软件设计	132
6.2.1 功能分析	132
6.2.2 软件设计	133
6.2.3 四则运算过程	134
6.3 标准型计算机增量项目开发计划	135

6.4 增量 6-1: 显示界面	136
6.4.1 Java GUI 程序设计过程	136
6.4.2 增量 6-1 的编程实现	137
6.4.3 增量 6-1 的程序分析——GUI 界面编程 模板分析	139
6.4.4 Java 图形界面的程序编码流程	140
6.4.5 图形设计: Swing 容器	140
6.4.6 图形设计: Swing 基本组件——文本框 JTextField	142
6.4.7 图形设计: Swing 基本组件——按钮 JButton	144
6.5 增量 6-2: 处理数字和运算符输入	146
6.5.1 按钮的事件响应	146
6.5.2 使用字符串数组链表记录输入内容	147
6.5.3 增量 6-2 的编程实现	148
6.6 增量 6-3: 四则运算 (1)	150
6.6.1 用 startCalculation 方法实现四则 运算过程	150
6.6.2 增量 6-3 的编程实现	150
6.7 增量 6-4: 四则运算 (2)	151
6.8 增量 6-5: 处理回退键输入和清空键输入	152
6.9 四则运算的另一种算法	153
6.10 总结回顾与知识扩展	154
实验 02——(标准型) 计算器: 实现增量 6-1	154
实验 03——(标准型) 计算器: 实现增量 6-2	155
实验 04——(标准型) 计算器: 实现增量 6-3	156
实验 05——(标准型) 计算器: 实现增量 6-4	156
实验 06——(标准型) 计算器: 实现增量 6-5 和 界面优化	157
实验 07——(标准型) 计算器: 程序优化	158
习题	159

第7章 科学型计算器

7.1 需求分析与项目目标	160
7.1.1 需求分析	160
7.1.2 项目目标	162
7.2 功能分析与软件设计	162
7.3 增量项目开发	163
7.3.1 增量 7-1: 直接实现科学计算器	163

7.3.2 增量 7-2: 通过继承实现科学计算器	164
实验 08——直接实现科学计算器	167
实验 09——通过继承实现科学计算器	168
7.4 总结回顾与知识扩展	168
第 8 章 复合型计算器	170
8.1 需求分析与项目目标	170
8.1.1 需求分析	171
8.1.2 项目目标	174
8.2 功能分析与软件设计	174
8.3 技术准备与增量项目开发	174
8.4 复合型计算器增量开发计划	174
8.5 增量 8-1: 复合计算器 I 型和 II 型	175
8.6 增量 8-2: 复合计算器 III 型	177
8.7 增量 8-3: 复合计算器 IV 型	177
实验 10——实现复合型计算器 (三选一)	177
8.9 总结回顾	178

第三部分 Java 程序设计基本技能 (二)

第 9 章 异常处理	180
9.1 异常与错误	180
9.2 异常的处理方式	181
9.2.1 捕获异常	181
9.2.2 声明抛出异常	183
9.3 异常的使用原则	183
9.3.1 在可能出现异常的情况下使用异常	184
9.3.2 不要不处理捕获的异常	184
9.3.3 不要不指定具体的异常代码	184
9.3.4 不要把每一条语句都用 try...catch 封装	184
9.3.5 异常声明的数量	185
9.4 断言	185
9.5 本章小结	187
习题	187

第 10 章 文件与数据流	188
10.1 流的概念	188
10.2 字节流	188
10.2.1 InputStream 和 OutputStream	188

10.2.2 文件字节流	190
10.2.3 字节数组流	192
10.2.4 数据流	193
10.2.5 缓冲字节流	195
10.2.6 标准输入输出流	197
10.3 字符流	198
10.3.1 Reader 和 Writer	198
10.3.2 文件字符流	199
10.3.3 字符数组流	201
10.3.4 缓冲字符流	202
10.3.5 PrintWrite	203
10.3.6 字符与字节相互转换流	204
10.4 随机访问文件	205
10.5 对象序列化	206
10.6 文件	208
10.7 本章小结	211
习题	211

第 11 章 网络编程

11.1 计算机网络基本知识	212
11.1.1 TCP/IP	212
11.1.2 网络地址	212
11.1.3 端口	213
11.1.4 套接字	213
11.1.5 URL	213
11.1.6 UDP	213
11.2 URL 类	213
11.2.1 创建 URL 对象	214
11.2.2 读取页面信息	214
11.2.3 创建一个到 URL 的连接	215
11.3 InetAddress 类	217
11.3.1 获取本地主机的 IP 地址	217
11.3.2 获取 Internet 给定的主机和 IP 地址	217
11.4 TCP Socket 通信	218
11.4.1 实现客户端的 Socket 类	218
11.4.2 实现服务器端的 ServerSocket 类	219
11.4.3 TCP Socket 通信示例	219
11.5 UDP 通信	221
11.5.1 数据报包 DatagramPacket 类	222

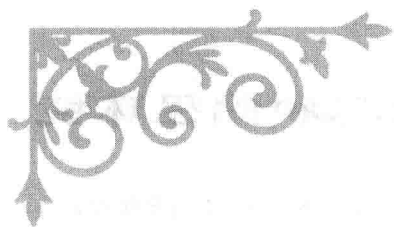
11.5.2 收发数据报 DatagramSocket 类	223
11.5.3 基于 UDP 通信举例	224
11.6 本章小结	225
习题	225
第 12 章 多线程	226
12.1 线程的基本概念	226
12.2 线程的创建	226
12.2.1 继承 Thread 类	226
12.2.2 实现 Runnable 接口	228
12.3 线程的状态与线程调度	229
12.4 线程同步	233
12.4.1 线程同步的实例	234
12.4.2 线程同步方法 1: 使用 synchronized 方法	235
12.4.3 线程同步方法 2: 使用 synchronized 数据	237
12.4.4 线程同步方法 3: 使用 class 同步	237
12.5 线程间通信	238
12.6 多线程中的死锁问题	240
12.6.1 死锁问题的产生的例子	240
12.6.2 死锁问题常用解决方法	242
12.7 本章小结	243
习题	243
第 13 章 Java 数据库编程技术	244
13.1 SQL 基本知识	244
13.1.1 SQL 基本语句	244
13.1.2 SQL 中基本数据类型	245
13.1.3 SQL 数据库创建过程	246
13.2 JDBC 基本知识	246
13.2.1 java.sql 软件包	246
13.2.2 JDBC 数据库访问过程	247
13.3 MySQL 与 Java 数据库开发	248
13.3.1 使用 JDBC 实现与 MySQL5.0 数据库的开发	248
13.3.2 通过 JDBC 实现 MySQL 数据库开发的一般过程	252
13.4 本章小结	253
习题	253

第四部分 网络通信与数据库实训——局域网聊天工具

第 14 章 点对点 (P2P) 聊天工具的例子: 网络通信	257
14.1 需求分析与项目目标	257
14.1.1 需求分析	257
14.1.2 用例分析	258
14.1.3 需求列表	260
14.1.4 项目目标	260
14.2 功能分析与软件设计	260
14.2.1 界面设计	261
14.2.2 文本聊天功能	262
14.2.3 文件传输功能	262
14.2.4 网络通信功能	262
14.2.5 增量开发计划	265
14.3 增量项目开发	266
14.3.1 增量 14-1: 实现聊天器界面	266
14.3.2 增量 14-2: 实现文本聊天	266
14.3.3 增量 14-3: 实现文件传输过程	271
实验 11——增量 14-1: 实现聊天器界面	273
实验 12——增量 14-2: 实现文本聊天	274
实验 13——增量 14-3: 实现文件传输	275
14.4 总结回顾与知识扩展	277
习题	277

第 15 章 基于 C/S 结构的聊天工具 I: 多线程技术	278
15.1 C/S 模式与多线程技术的知识准备	278
15.1.1 服务器/客户端 (C/S) 模式	278
15.1.2 多线程技术	285
15.2 需求分析与项目目标	287
15.2.1 需求分析	287
15.2.2 项目目标	292
15.3 功能分析与软件设计	293
15.3.1 C/S 模式中客户端软件设计	293
15.3.2 C/S 模式中服务器软件设计	295
15.3.3 C/S 模式中的通信协议	297
15.3.4 增量开发计划	298

15.4 增量 15-1: 采用多线程技术分离服务器与 客户端代码	299	16.1.2 需求列表	322
15.5 增量 15-2: 采用 JTable 实现好友列表、 用户列表	300	16.2 功能分析与软件设计	323
15.6 增量 15-3: 实现服务器端的用户管理功能	304	16.2.1 登录验证功能	323
15.7 增量 15-4: 实现 C/S 结构中客户端之间的 聊天功能	307	16.2.2 用户数据下载功能	324
15.8 增量 15-5: 实现 C/S 结构中的群聊功能	309	16.2.3 用户信息数据表的设计	326
15.9 总结回顾与知识扩展	310	16.2.4 服务器与数据库中用户信息数据表的 接口设计	329
实验 14——增量 15-1: 使用多线程技术分离服务器 与客户端代码	310	16.2.5 增量计划	331
实验 15——增量 15-2: 采用 JTable 实现好友列表	314	16.3 增量 16-1: 采用数据库保存用户数据	331
实验 16——增量 15-3: 实现 C/S 结构中端的 用户管理	316	16.4 增量 16-2: 实现服务器数据初始化与用户 注册/登录	334
实验 17——增量 15-4: 实现 C/S 结构中客户端 之间的聊天功能	317	16.4.1 建立服务器与数据库的连接	335
实验 18——增量 15-5: 实现 C/S 结构中的群聊功能	319	16.4.2 服务器数据初始化	335
习题	321	16.4.3 用户注册/登录功能	337
第 16 章 基于的 C/S 结构聊天工具 II: 数据库技术	322	16.5 增量 16-3: 实现用户数据下载功能	339
16.1 需求分析与项目目标	322	16.6 总结回顾与知识扩展	341
16.1.1 需求分析	322	实验 19——增量 16-1: 采用数据库保存用户数据	341
		实验 20——增量 16-2: 实现数据库初始化与用户 注册/登录	342
		实验 21——增量 16-3: 实现用户数据下载功能	344
		习题	345
		参考文献	346



第 一 部 分

Java 程序设计基本技能（一）

本教材的第一部分主要介绍 Java 语言基础，包括编程环境的搭建、Java 基本语法、面向对象的概念、类与对象、数组字符串、图形用户界面等基本技能。该部分在第二部分计算器的实训项目驱动下，讲解 Java 的基本知识。在教学内容安排上，将第一部分和第二部分相结合进行教学。



第1章 Java 的特点、基本内容和编程环境

Java 作为目前的主流面向对象程序设计语言之一，因其面向对象、跨平台、支持多线程和分布式等特点，在 Web 应用程序开发、网络编程、手机游戏等各个方面都得到了广泛的应用，并且受到越来越多程序设计者的青睐。本章主要介绍 Java 语言的历史及特点、Java 开发环境，以及本书中采用的项目开发模板。

1.1 Java 语言的历史和特点

Java 是由 Sun 公司于 1995 年首次推出的一种面向对象的程序设计语言，到现在已经发展了 10 余年。本节重点介绍 Java 语言的历史和特点，使初学者对 Java 有个大概的了解。

1.1.1 Java 语言的发展历史

Java 是由 Sun 公司开发而成的新一代编程语言。Sun 公司的 Java 语言开发小组成立于 1992 年，该小组的领导人 James Gosling 是一位非常杰出的程序员。开发小组在开始之前主要的开发方向是诸如开拓交互式电视、面包烤箱等消费类电子产品市场。由于通过改写 C 编译器无法满足与系统的平台无关性，所以于 1991 年 6 月决定准备开发一种新的语言，并起名为 Oak (Java 语言的前身)。由于当时 Oak 已经被注册，所以改名为 Java，其为太平洋上一个盛产咖啡的岛屿的名字。1995 年 3 月，Sun 公司发布了利用 Java 编写的第一个交互式浏览器 HotJava，引起了世界的关注。

1995 年 5 月 Sun 公司正式推出 Java 程序设计语言和 Java 平台的名称。Java 平台由 Java 虚拟机 (Java Virtual Machine, 简称 JVM) 和 Java 应用程序编程接口 (Application Programming Interface, 简称 API) 构成。经过 10 余年的发展，目前 Java 语言已经是一门被广泛使用的编程语言，而且学习和使用 Java 语言的软件开发人员也在不断增加。J2SE 平台 (Java 的标准版) 的版本也不断地发展，从 1996 年的 JDK 1.0 发展到目前的最新版本 JDK 7.0。

1.1.2 Java 语言的特点

Java 作为一种高级程序设计语言，与其他高级语言相比，最重要的特点是它的与平台无关性，也就是常说的 “Write once run anywhere”。另外，Java 在安全性、健壮性、分布式、面向对象等方面也有其自身的特点。

1. 简单

与高级语言 C++ 相比，Java 语言简单。Java 语言丢弃了 C++ 中许多复杂的概念和容易疑惑的内容。例如：Java 语言中没有 C++ 语言中的多继承和指针、去掉了 #include 和 #define 等预处理功能、没有运算符的重载、不支持 C++ 语言的强制自动类型转换等。

Java 语言还增加了一些比较实用的功能。如可以自动对内存进行管理、自动进行垃圾收集、显示地进行强制类型转换等。由于 Java 语言在诸如变量声明、流程控制、参数传递等方面与 C 或者 C++ 语言类似，所以对于已经学过 C 或者 C++ 语言的人来说，Java 语言更容易学习和掌握。

2. 面向对象

Java 语言是一种面向对象的语言,它通过把现实世界的事物抽象成对象的概念并使用方法实现对对象的操作,将复杂问题抽象化,从而更好地解决实际问题。

在 Java 程序设计中,抽象层可以通过使用接口和抽象类来实现。

Java 程序将数据和操作数据的方法封装在一起形成类,通过对数据以及对数据进行操作的方法进行封装实现了信息的隐藏。在 Java 程序中使用 `public`、`protected` 和 `private` 等访问控制符控制类对数据成员的访问权限。封装可以将那些不想被其他成员访问的数据封起来,从而使得程序更易于维护。

通过类的继承,不但使子类可以继承父类的属性和方法,还可以扩展子类自己的功能,达到代码的重用。在 Java 语言中,类的继承是通过 `extends` 关键字来实现的。例如:有一个交通工具类 `Vechile`,可以把汽车、飞机、轮船等交通工具的公共属性定义在该类中;而表示汽车、飞机、轮船等各个交通工具的类作为 `Vechile` 的子类,它们除了可以拥有 `Vechile` 类的成员属性和方法外,还可以拥有自己特定的成员属性和方法。但是,Java 语言的类只支持单一继承,即一个类只能有一个直接父类。如果想实现多继承,可以使用 Java 语言中的接口来实现。

通过对同一个方法调用不同类型的对象而产生不同行为称为多态性。多态增强了程序的扩展性,提高了项目的可阅读性。多态性是基于继承的,也就是说,对于父类中的一个属性或方法,在被多个子类继承之后,可以表现出不同的行为。

3. 与平台无关性和可移植性

与其他语言相比,Java 语言的与平台无关性是它最大的特点,这一特性主要体现在 Java 虚拟机(JVM)上,Java 虚拟机机制是 Java 语言实现的核心机制之一。和许多语言直接把代码编译成机器可识别的指令不同,Java 语言在编译时会把源代码编译成字节码,也就是 `.class` 文件。而 Java 虚拟机可以通过 `ClassLoader` 来加载字节码,并最后将其转换成机器可识别的指令进行执行。

由于 Java 语言具有与平台无关性的特点,所以可以很容易地将 Java 程序的可执行代码从一台机器上移植到另一台机器上,只要机器上配有可以将字节码转换成机器可识别指令的 Java 虚拟机就可以。因此说,Java 程序是可移植的。

4. 可靠性和安全性

由于 Java 语言主要应用于网络程序的开发,所以安全性是很多人担心的问题。事实上,作为 Java 语言核心机制之一的代码安全监测机制通过对字节码的校验,可使不合法的字节码无法被解释执行。同时,Java 语言在编译期间需要对表达式和参数进行类型检查,如果在编译期间出现类型不兼容的情况,编译器就会报错。除此以外,还可以对数组边界是否越界进行检查,对不同类型的对象要求进行强制类型转换等措施,都提高了 Java 程序的可靠性和安全性。

5. 多线程并发机制

Java 语言提供了对多线程的支持。多线程是指在同一时间内有多个线程在执行,通过多线程可以在同一时间内完成多个任务,提高程序的执行效率。Java 语言通过继承 `Thread` 类或实现 `Runnable` 接口实现多线程。另外,Java 语言还提供了 `synchronized` 关键字保证线程之间的同步,使得线程之间共享的数据能够得到正确的操作。多线程的使用可以简化网络实时交互行为。C++ 语言不支持多线程。

6. 分布式

Java 支持网络应用程序, 支持 C/S 模式, 在 API 中提供了一个有关网络的类库, 例如: net 类、socket 类、URL 类等, 开发人员可以方便地使用类库中的方法实现分布式的操作。Java 程序通过使用 URL 对象, 可以对 Internet 中的网络资源进行访问。同时, Java 还提供了一种 RMI (Remote Method Invocation, 远程方法调用) 机制, 使得 Java 在分布式应用方面的能力得到了增强, 这也是开发分布式应用的一个很重要的解决方案。

7. 动态的内存管理机制

Java 中动态内存管理是通过垃圾回收机制实现的, Java 的垃圾回收机制也是 Java 的核心机制之一。在系统空闲时, 对程序运行过程中没有用的对象自动进行回收; 对不再使用的对象, 会自动释放它们所占用的内存空间; 也可以将一个对象的句柄设置为 null, 来通知垃圾收集器对其进行回收。

1.2 本书基本内容与教学思路

本书以项目驱动, 采取增量式的开发方式。全书共分 4 部分, 第一部分和第三部分为 Java 程序设计基本技能, 第二部分和第四部分为 Java 实训内容。

在教学安排上, 可以打乱 Java 基本技能的编写顺序, 通过项目驱动的形式进行基本技能的讲解。即在第二部分简易计算器项目基础上, 进行第一部分内容的融合与展开; 在第四部分局域网聊天工具项目基础上进行第三部分内容的融合与展开。建议在 Java 课程内容的安排上, 通过两个实训项目的增量完成, 将第一部分和第三部分的 Java 技术融入到项目内讲解, 教学思路采用: 根据实训内容的安排设置理论课程内容的学时分配与进度。

1.3 Java 开发环境介绍

要在一台计算机上编写和运行 Java 程序的首要工作是建立起 Java 开发环境。建立 Java 开发环境就是在计算机上安装 Java 开发工具包, 并在计算机中设置相应的参数, 使开发工具包可以在计算机中顺利地正确运行。

Sun 公司免费提供的开发工具包的早期版本简称为 JDK (Java Developer's Kit)。现在推出的开发工具包分化成为 3 个版本: J2SE (Java2 Platform, Standard Edition)、J2EE (Java2 Platform, Enterprise Edition) 和 J2ME (Java2 Platform, Micro Edition)。J2SE 是用于工作站和个人计算机的标准开发工具包, J2EE 是用于企业级开发的工具包, J2ME 是用于开发智能消费型电子产品。

Java 开发环境包括学习型 (JDK+文本编辑器) 和开发型环境 (Eclipse、MyEclipse 等)。下面将分别进行介绍。

1.3.1 JDK 的下载、安装及环境变量的设置

1. JDK 的下载与安装

JDK 是一切 Java 应用程序的基础, 所有的 Java 应用程序都是构建在这个基础之上的。JDK 包含了 Java 开发所必需的常用库。JDK 可以到 Sun 公司的网站 (<http://java.sun.com>) 下载。下载时要注意自己计算机的操作系统类型, 下载的安装程序应当与自己计算机的操作系统相匹配, 而且版本一般选择最新的。本书使用的 JDK 版本为 jdk-7u1-windows-i586。

下载完成后，只要遵循安装程序的指示一步步地进行安装即可。具体安装步骤如下。

(1) 双击 JDK 安装文件，进入 JDK 安装向导界面，如图 1-1 所示。

(2) 单击“下一步”按钮进入自定义安装界面，在该界面可以修改安装路径，如图 1-2 所示。

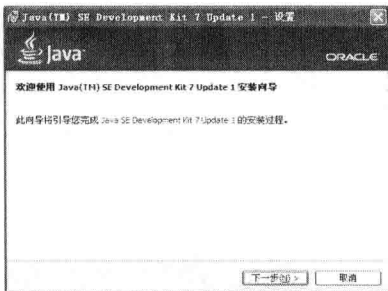


图 1-1 进入安装向导界面

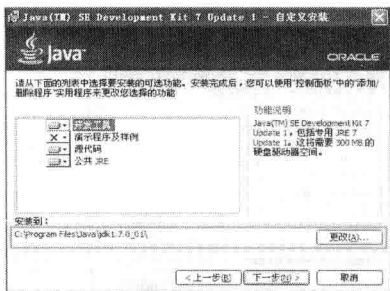


图 1-2 自定义安装

(3) 进行设置后，单击“下一步”按钮即可安装，如图 1-3 所示为安装过程。

(4) 修改 JRE 安装路径，如图 1-4 所示，可单击“更改”按钮进行更改。

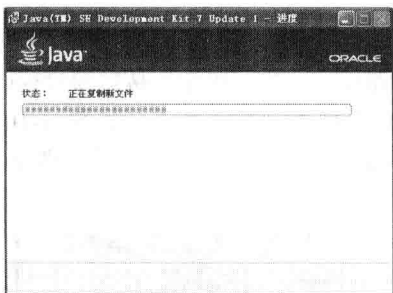


图 1-3 安装进度

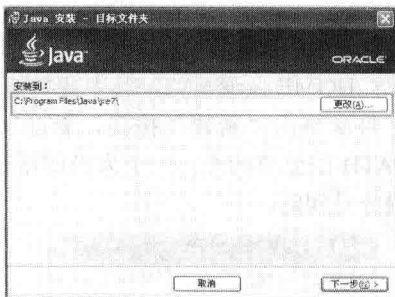


图 1-4 安装 JRE

(5) 修改完成后，单击“下一步”按钮进入 JRE 安装（见图 1-5），安装完成（见图 1-6）。



图 1-5 正在安装 JRE

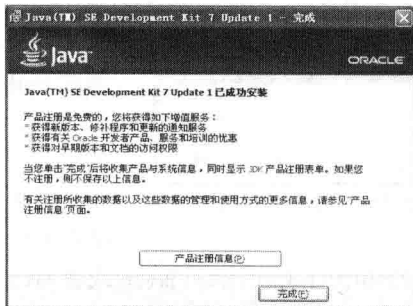


图 1-6 安装完成

2. 环境变量的配置

成功安装 JDK 后，需要手动配置 JDK 环境变量。一共需要设置 3 个环境变量：JAVA_HOME、CLASSPATH、Path。值得注意的是，不同操作系统下配置过程稍有不同，本教材以 Windows XP 为例进行介绍，具体配置过程如下：

(1) 右击桌面上的“我的电脑”，在弹出的快捷菜单中选择“属性”命令，打开“系统属性”对话框，如图 1-7 所示。

(2) 切换到“高级”选项卡，单击“环境变量”按钮，打开“环境变量”对话框，如图 1-8 所示。

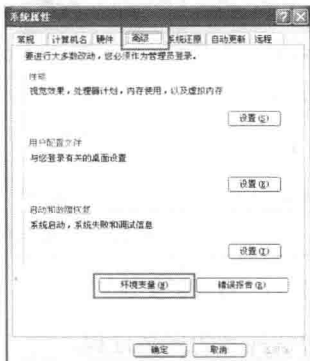


图 1-7 “系统属性”对话框



图 1-8 “环境变量”对话框

(3) 单击“新建”按钮，弹出“新建系统变量”对话框，在对话框中填入如图 1-9 所示新建变量 JAVA_HOME 及变量值，其中变量值为 JDK 安装路径。单击“确定”按钮即可返回。

(4) 再次单击“新建”按钮，新建 CLASSPATH 变量并设置变量值，如图 1-10 所示。CLASSPATH 的值有两个，一个为当前路径（用“.”表示），一个为 J2SE 类库所在路径，两个路径之间用分号间隔。



图 1-9 新建 JAVA_HOME

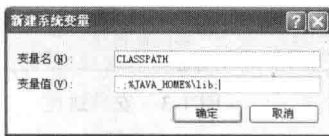


图 1-10 新建 CLASSPATH

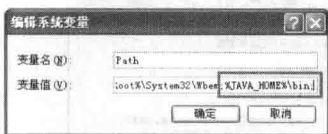


图 1-11 修改 Path 路径

(5) 单击图 1-8 中的“编辑”按钮，修改 Path 路径，向已有的 Path 路径中添加 bin 文件路径，如图 1-11 所示。

(6) 设置完成后，连续单击“确定”按钮，完成环境变量的设置。

(7) 选择“开始”→“运行”命令，输入 cmd 后，进入 MS-DOS 窗口，在 DOS 窗口中输入 javac 或者 java 命令，出现如图 1-12 和图 1-13 所示的 javac 或者 java 命令的使用方法时，表示环境变量已经配置成功。



图 1-12 javac 命令使用方法



图 1-13 java 命令使用方法

1.3.2 Eclipse 的安装及使用

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。Eclipse 附带了一个标准的插件集，包括 Java 开发工具。Eclipse 还包括插件开发环境，这个组件主要针对希望扩展 Eclipse 的软件开发人员，以允许开发人员构建与 Eclipse 环境无缝集成的工具。

1. 下载并安装 Eclipse

Eclipse 软件包可以到官方网站 <http://www.eclipse.org/downloads/> 下载，它可以安装在各种操作系统上，在 Windows 下安装 Eclipse，除了需要 Eclipse 软件包以外，还需要 Java 的 JDK 来支持 Eclipse 运行，并需要设置相关的环境变量，关于 JDK 的安装及环境变量设置可参考 1.3.1 节。

Eclipse 属于绿色软件，不需要运行安装程序，不需要向 Windows 注册表填写信息，只需要将 Eclipse 压缩包解压就可以运行。

(1) 将下载的 eclipse-SDK-4.2.1-win32.zip 解压，压缩到一个本地目录（例如 D:\eclipse）中，双击该目录中的 eclipse.exe 文件即可打开 Eclipse，图 1-14 所示为 Eclipse 的启动画面。

(2) 之后弹出 Workspace Launcher 对话框，如图 1-15 所示，选择或新建一个文件用来保存创建的项目，可以勾选复选框。

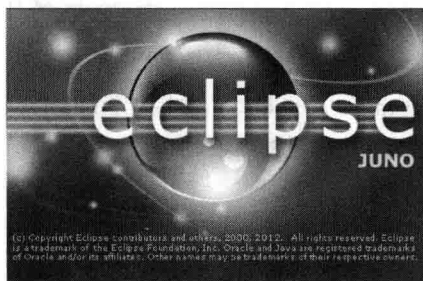


图 1-14 Eclipse 启动画面

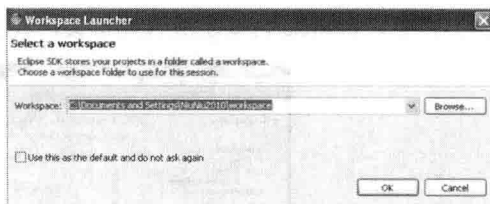


图 1-15 Workspace Launcher 对话框

(3) 设置好后，单击“OK”按钮，打开 Eclipse 工作界面，如图 1-16 所示。

2. Eclipse 的使用

下面以一个简单的 Hello Friends! 为例介绍 Eclipse 的基本使用步骤。

(1) 单击图 1-16 右上角的 Workbench 进入 Eclipse 开发界面，如图 1-17 所示。

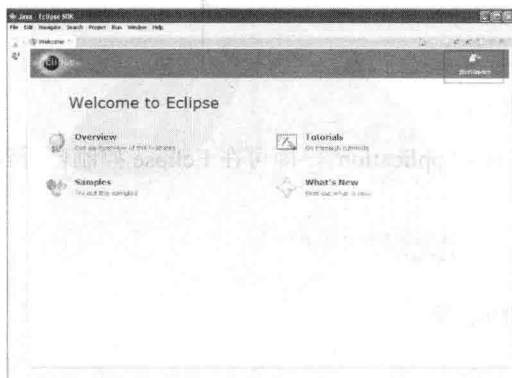


图 1-16 Eclipse 工作界面



图 1-17 Java 工作界面