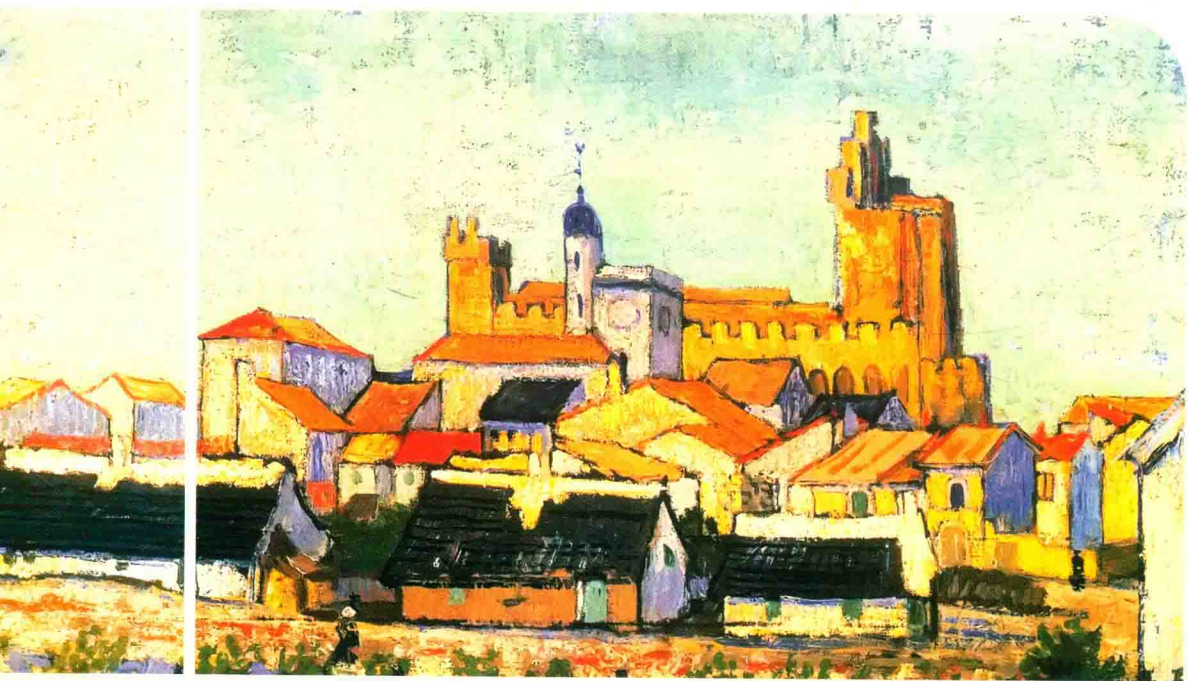




21世纪高等学校计算机
专业实用规划教材

Java程序设计基础与实践 (题库版)

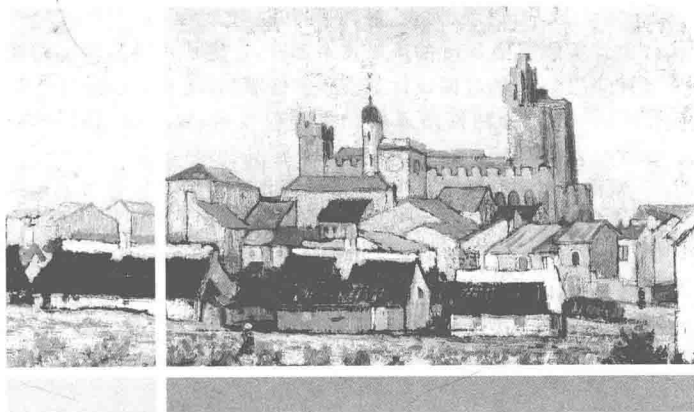
◎ 周洪利 许宏丽 周围 编著



清华大学出版社



21世纪高等学校计算机
专业实用规划教材



Java程序设计基础与实践 (题库版)

◎ 周洪利 许宏丽 周围 编著

清华大学出版社
北京

内 容 简 介

本书将 Java 编程技术分为三部分,分别是基础篇、核心篇和实践篇。

基础篇介绍 Java 的入门基础部分,主要包括 Java 技术概括、基础数据类型、表达式、流程控制语句,以及类的初步概念和使用方法。

核心篇介绍 Java 的面向对象技术,以及异常处理和数据容器技术,包括继承、接口、多态的概念和实现,在数据容器部分介绍 Java 的枚举、泛型、列表、集、队列和映射。

实践篇通过四个实际设计案例贯穿 Java 的高级技术部分,在完成设计案例的同时介绍其所应用到的相关技术。其中计算器设计介绍 Java 的界面设计基础和事件驱动,文本编辑器设计介绍 Java 的界面布局技术、流和文件等。网络聊天室设计介绍网络通信和多线程技术,Web 应用设计介绍数据库和 JavaFX 技术。

本书是全面介绍 Java 编程技术的高校本科生教材,也可作为工程开发参考书使用。本书的全部案例都基于 JDK 8,开发平台为 NetBeans 8.0。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java 程序设计基础与实践(题库版)/周洪利,许宏丽,周围编著. —北京:清华大学出版社,2018

(21 世纪高等学校计算机专业实用规划教材)

ISBN 978-7-302-49963-3

I. ①J… II. ①周… ②许… ③周… III. ①JAVA 语言—程序设计—习题集 IV. ①TP312.8-44

中国版本图书馆 CIP 数据核字(2018)第 067753 号

责任编辑:黄 芝

封面设计:刘 键

责任校对:梁 毅

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:20 字 数:489 千字

版 次:2018 年 9 月第 1 版 印 次:2018 年 9 月第 1 次印刷

印 数:1~1000

定 价:59.00 元

产品编号:078110-01

前 言

Java 程序设计技术是当前计算机程序设计中用最广泛的程序设计语言,本书作为面向高校的本科生的教材和面向工程开发人员的参考书,全面介绍 Java 的程序设计技术。

本书的主要特点在于其内容组织方式,在本书的 12 章中,将 Java 的全部内容分为三部分:基础篇、核心篇和实践篇。其中前两部分按照内容组织,第三部分通过实际的开发案例组织。

1) 基础篇

基础篇内容包括 Java 的技术概括、开发平台、表达式、流程控制和基础类等内容。本书采用集成开发平台 NetBeans,这相比命令行方式具有更高的开发效率。本部分的重要内容是基础数据类型、表达式和流程控制语句,在基础类一章主要介绍 String 类、Math 类,以及 JDK 中的控制台输入和输出方法。

2) 核心篇

本部分介绍面向对象的核心技术,以及异常处理和数据容器技术。作为一门纯面向对象的程序设计语言,学习 Java 的类和对象、继承与接口的实现是掌握面向对象程序设计的关键,同时也是熟练使用 Java 类库的基础。本书强调使用 Java 类库,细致了解 Java 类库结构是实现一个工程开发必不可少的学习环节。在数据容器一章介绍了 Java 的各种高级数据组织方式,包括枚举、泛型、列表、集、队列和映射等内容,Java 类库在数据容器方面提供了丰富和多样化的支持。

3) 实践篇

本部分是本书的重要特色之一,通过四个实际开发案例贯穿了 Java 的主要实用化内容。案例的组织方式更接近工程开发,同时相比于传统的按内容组织方式,通过案例教学也更容易理解和掌握这些内容。在 Java 的界面设计方面,在计算器设计、文本编辑器设计和 Web 应用设计中分别进行了不同的介绍,在计算器设计中介绍基本的组件和事件驱动方法,在文本编辑器中介绍了界面布局技术,在 Web 应用设计中介绍了 JavaFX 界面设计技术。在文本编辑器设计一章中重点介绍了 Java 的流和文件技术,并最终设计实现了一个支持多种编码标准的实用化的文本编辑器。在网络聊天室设计一章重点介绍网络通信和多线程技术,并最终设计实现了一个简易的文本信息网络聊天室软件。在 Web 应用系统设计中重点介绍 Java 的数据库技术和 JavaFX,并介绍了通过工具软件 JavaFX Scene Builder 实现界面设计的方法,实现了在完全不需要了解 JavaFX 的 FXML 脚本语言的情况下实现界面设计,最终设计实现了一个网络课程作业提交系统软件。

本书作为高校教材,如果讲授全部内容,推荐学时为 64 学时。如果采用 48 学时,推荐讲授内容为前 7 章和第 9 章。在 48 学时教学过程中建议实验 6~7 个,全部实验在

NetBeans 下完成,分别是表达式实验、类库实验、流程控制实验、类和对象实验、继承和接口实验、界面设计实验。

本书作者多年从事 Java 课程教学,设计有教学测试题库和实验题库,可以实现随机抽题作业和单元测试。

本书由周洪利、许宏丽、周围编写,其中周洪利编写第 1、2、8、10、11、12 章,许宏丽编写第 3、4、9 章,周围编写第 5、6、7 章。

由于时间仓促,作者水平有限,可能存在疏漏之处。在此,诚恳期望得到各领域的专家和广大读者的批评指正。

题库课程获取步骤

- (1) 打开手机微信“扫一扫”,扫描书后刮刮卡的权限二维码,即可获得权限。
- (2) 再扫描每一章末的题库二维码,即可在线做题并提交查看结果。

作 者

2018 年 5 月

目 录

第 1 章 Java 概述	1
1.1 Java 的发展过程	1
1.1.1 Java 技术的形成	1
1.1.2 Java 的成熟和发展	2
1.2 Java 技术	3
1.2.1 Java 的技术特点	3
1.2.2 Java 的相关技术	4
1.3 Java 开发平台	6
1.3.1 JDK 和 IDE 下载安装	7
1.3.2 NetBeans 简介	7
1.3.3 创建 Java 项目	8
1.4 Java 源程序	10
1.4.1 Java 程序结构	10
1.4.2 Java 语句排列	11
1.5 Java 类库	12
1.5.1 类库引用	12
1.5.2 Java 类库简介	14
1.5.3 类文档	16
第 2 章 Java 表达式	17
2.1 变量	17
2.1.1 标识符	17
2.1.2 基础数据类型	18
2.1.3 变量声明	19
2.1.4 常量	19
2.2 数组	20
2.2.1 数组和数组元素	20
2.2.2 数组的声明	21
2.2.3 多维数组	21
2.3 表达式	23

2.3.1	赋值语句	23
2.3.2	运算符分类和运算形式	24
2.3.3	运算优先级	25
2.3.4	运算数据类型	25
2.4	运算符	28
2.4.1	算术运算	28
2.4.2	关系运算	30
2.4.3	逻辑运算	31
2.4.4	位运算	33
第3章	常用类	37
3.1	类的引用	37
3.1.1	类和对象	37
3.1.2	引用类型	38
3.1.3	类库的引用	40
3.2	Math类	41
3.2.1	三角函数方法	42
3.2.2	指数函数方法	42
3.2.3	其他函数方法	43
3.2.4	random方法	43
3.3	String类	44
3.3.1	构造字符串	44
3.3.2	获取字符串信息	46
3.3.3	字符串连接与比较	47
3.3.4	字符串分割与转换	51
3.4	控制台输入输出	51
3.4.1	控制台输入输出类	52
3.4.2	Scanner类	52
3.4.3	Console类	55
3.4.4	输出方法	56
第4章	流程控制	62
4.1	顺序结构	62
4.1.1	程序结构	62
4.1.2	顺序结构	63
4.1.3	变量作用域	64
4.2	选择结构	65
4.2.1	if语句	65
4.2.2	switch语句	70

4.3	循环结构	72
4.3.1	for 循环	73
4.3.2	while 循环	77
4.3.3	程序重定向	80
4.4	综合应用	82
4.4.1	猜数游戏	82
4.4.2	排序	84
4.4.3	查找	88
第5章	类和对象	91
5.1	面向对象程序设计	91
5.1.1	面向对象的概念	91
5.1.2	面向对象的特性	92
5.2	创建自己的类	93
5.2.1	类的结构	93
5.2.2	成员变量	94
5.2.3	成员方法	95
5.2.4	构造方法	96
5.2.5	main()方法	97
5.3	对象的创建和使用	98
5.3.1	对象的创建	98
5.3.2	对象的使用	99
5.3.3	形参和实参	101
5.3.4	this 关键字	102
5.4	实例成员与类成员	103
5.4.1	static 关键字	104
5.4.2	实例变量和类变量	104
5.4.3	实例方法和类方法	105
5.4.4	静态代码块与构造代码块	105
5.5	内部类	107
5.5.1	内部类的概念	107
5.5.2	成员内部类	107
5.5.3	局部内部类	108
5.5.4	静态内部类	110
5.5.5	匿名内部类	111
第6章	继承和接口	112
6.1	继承	112
6.1.1	类的层次结构	112

6.1.2	类继承的实现	113
6.1.3	super 关键字	118
6.1.4	final 关键字	119
6.1.5	Object 类	120
6.2	抽象类	123
6.2.1	抽象类的定义	123
6.2.2	抽象类的实现	123
6.2.3	Number 抽象类	125
6.3	接口	126
6.3.1	接口的定义	126
6.3.2	接口的实现	127
6.3.3	CharSequence 接口	129
6.4	多态	130
6.4.1	方法的覆盖	130
6.4.2	变量的隐藏	132
6.4.3	instanceof 关键字	132
6.4.4	接口对象的实例化	133
第 7 章	异常处理	135
7.1	异常的基本概念	135
7.1.1	异常的产生	135
7.1.2	异常的处理	136
7.2	异常类	138
7.2.1	Throwable 类	138
7.2.2	Exception 类	139
7.3	处理异常	140
7.3.1	捕获并处理多种异常	140
7.3.2	异常与资源管理	141
7.4	抛出异常	142
7.4.1	方法体中抛出异常(throw)	142
7.4.2	方法声明中抛出异常(throws)	143
7.4.3	使用异常的原则	144
7.5	编写自己的异常类	145
第 8 章	数据容器	147
8.1	泛型	147
8.1.1	泛型的声明	147
8.1.2	泛型的使用	148
8.1.3	泛型环形队列的实现	149

8.2	枚举	151
8.2.1	声明枚举类型和枚举对象	151
8.2.2	枚举的使用	152
8.3	集合框架	152
8.3.1	迭代器接口 Iterable	153
8.3.2	集合接口 Collection	153
8.4	列表	154
8.4.1	List 继承结构	154
8.4.2	List 实现类	155
8.4.3	LinkedList 测试	157
8.4.4	Stack 测试	159
8.5	集	160
8.5.1	Set 继承结构	160
8.5.2	Set 的实现类	161
8.5.3	Set 测试	163
8.6	队列	165
8.6.1	Queue 继承结构	165
8.6.2	Queue 的实现类	166
8.7	映射	167
8.7.1	Map 继承结构	167
8.7.2	Map 的实现类	168
8.7.3	Map 测试	169
第 9 章	设计一个计算器	172
9.1	JFC 概述	172
9.1.1	AWT 和 Swing	172
9.1.2	容器与组件	174
9.2	容器类	176
9.2.1	窗口	176
9.2.2	JPanel 类	181
9.3	常用组件	183
9.3.1	JButton	183
9.3.2	Label	184
9.3.3	JTextField	185
9.4	事件处理	188
9.4.1	事件	188
9.4.2	事件处理	189
9.4.3	GUI 综合应用	194
9.5	计算器的设计与实现	196

9.5.1	计算器界面设计	197
9.5.2	框架初始化	198
9.5.3	事件处理程序	199
第 10 章	文本编辑器设计	206
10.1	文件	206
10.1.1	File 类	206
10.1.2	文件顺序读写	209
10.1.3	文件随机读写	215
10.2	界面布局设计	217
10.2.1	自由设计	218
10.2.2	布局设计	218
10.2.3	布局类型	219
10.2.4	综合布局	220
10.3	需要的组件	221
10.3.1	文件选择器	221
10.3.2	颜色选择器	222
10.3.3	字体	222
10.3.4	标准对话框	223
10.3.5	打印	224
10.4	编辑器实现	225
10.4.1	组件表	225
10.4.2	初始化	226
10.4.3	事件驱动方法	227
第 11 章	网络聊天室设计	234
11.1	网络通信	234
11.1.1	服务器(Server)和客户端(Client)	235
11.1.2	InetAddress 类	235
11.1.3	URL	238
11.1.4	Socket 类	240
11.1.5	ServerSocket 类	241
11.1.6	TCP 通信测试	242
11.2	多线程	245
11.2.1	线程	245
11.2.2	线程调度	247
11.2.3	线程同步	248
11.2.4	线程控制	249
11.3	界面设计	251

11.3.1	布局设计	251
11.3.2	界面外观	251
11.3.3	文本组件 JTextPane	253
11.3.4	列表组件 JList	254
11.3.5	组件列表	255
11.4	软件设计	257
11.4.1	软件线程	257
11.4.2	消息处理	262
第 12 章	Web 应用系统设计	271
12.1	数据库	271
12.1.1	JDBC	271
12.1.2	连接数据库	273
12.1.3	数据库分析	274
12.1.4	数据库操作	276
12.2	JavaFX	281
12.2.1	JavaFX 简述	282
12.2.2	JavaFX 入门	285
12.2.3	JavaFX 界面设计	286
12.2.4	JavaFX 事件处理	289
12.3	JavaFX 设计案例	291
12.3.1	数据库设计	292
12.3.2	界面设计	293
12.3.3	实现代码	295
12.3.4	JavaFX 部署	305

含有 Applet 的网页的 HTML 文件在代码中采用 <applet> 关键字标记出 Applet 部分, 当支持 Java 的网络浏览器遇到这样的标记时, 会将 Applet 部分下载到本地计算机上执行。

Java Applet 是采用 Java 语言来编写的小应用程序, 将其嵌入到网页中, 可以提高 Web 页面的交互能力和动态执行能力。当然, 本地计算机使用的浏览器必须是支持 Java 的网络浏览器。

在 Java Applet 中, 可以实现文字的字体和颜色的设计, 支持图形绘制, 支持动画和声音的插入, 可以实现丰富的人机交互功能。

2. Servlet

Servlet 的名词来自 Server Applet, 意思是“服务器小程序”。Servlet 是用 Java 编写的运行在服务器端的程序。其主要功能是响应浏览器客户端的交互, 并生成动态 Web 网页内容。

与 Applet 不同的是, Servlet 运行在服务器中, 而 Applet 与浏览器一起运行在客户端。同时 Applet 具有图形界面, 而 Servlet 没有图形界面。

在 Servlet 出现之前, 能为浏览器客户提供动态响应的技术是 CGI (Common Gateway Interface, 通用网关接口) 程序。通过 CGI, Web 服务器会根据浏览器客户端的请求启动不同的外部程序, 并将请求内容转发给该程序, 再通过外部程序生成对客户端的响应。

与 CGI 技术相比, Servlet 具有明显的优势, 主要表现在以下三个方面。

(1) 效率高。CGI 对于每一个用户请求都要启动一个新的进程, 而 Servlet 在服务器端仅需要运行一个 Java 虚拟机程序, 就可以通过不同的线程 (而不是进程) 对应多个不同的用户请求。

(2) 跨平台性。Servlet 通过使用 Java 语言来编写, 在运行 Java 环境的任何操作系统下都可以运行, 而 CGI 是需要针对不同的运行平台进行编译的。

(3) 功能强。相比于 CGI, Servlet 的功能更强。

Servlet 存在的问题是, 为了能够输出 HTML 格式内容, 需要编写大量重复代码, 为了弥补这一缺陷, Sun Microsystems 公司推出了 JSP (Java Server Pages) 技术。

3. JSP

JSP 是在 Sun Microsystems 公司的倡导下, 由多家公司参与制定的一种动态网页技术标准, 是集成在 Java EE 中的重要技术。JSP 技术的基础和核心是 Servlet 技术, 是为了提高 Servlet 的编程效率而制定的新技术标准。

JSP 的出现与微软推出的 ASP (Active Server Page) 形成了竞争关系, 两者都属于动态 Web 网页技术。这两种技术的差别主要体现在微软技术的封闭性和 Java 技术的开放性。ASP 技术主要应用于 Windows 平台, 不具备跨平台性, 而 JSP 却是具备跨平台性的, 在互联网时代, 跨平台性标志着具有更强大的生命力。

JAS 和 ASP 都是通过在网页文件中嵌入脚本代码来实现的, 不同之处在于 ASP 采用的是 JavaScript 或 VBScript, 而 JSP 嵌入的是 Java 代码, 由 Servlet 实现。JavaScript 是早期由 Netscape 公司设计的一种脚本语言, 因为 Netscape 与 Sun Microsystems 公司曾经有过的合作关系, Netscape 给它起了一个带 Java 字样的名字, 其实 JavaScript 并不是 Sun Microsystems 公司的技术, 但 JavaScript 在语法上的确与 Java 很相像。另一项相关的技术是 JScript, 是由微软制定的另一种脚本语言, 实际上可以说是微软版本的 JavaScript。

能够在互联网上传播并实现跨平台运行的技术,在网页文件中嵌入可执行的代码,在打开网页时将代码下载到客户端计算机执行。由于 Oak 的跨平台特征,Sun Microsystems 公司的开发团队决定将 Oak 设计为一种基于互联网环境的跨平台语言,在其中加入了可以嵌入网页并且可以随着网页文件在互联网上传输的 Applet 功能。

经过这样的改进,Sun Microsystems 公司于 1996 年将 Oak 的名称改为 Java,发布了第一个正式版本的 Java 开发工具包 Java Development Kit 1.0,即 JDK 1.0,并随后公布了 Java 的第一个即时编译器 JIT(Just-In-Time Compiler)。这样就构成了 Java 的第一个软件开发工具包 SDK(Software Development Kit)。

JDK 是一个用于 Java 开发的软件工具包,其中包含了 Java 开发和运行时所需要的支持工具软件。如果只是运行 Java 字节码文件,那么只需要有另一个软件包支持,即 JRE (Java Runtime Environment)支持即可,JRE 实际上是 JDK 的功能子集。

JDK 中主要包括:

(1) Java 类库,是 Sun Microsystems 公司为开发和运行 Java 程序而编写的,是针对不同运行平台的各种支持模块而设计的。Java 类库具有大量而丰富的功能,可以实现 Java 与运行平台的接口,所以又被称为 API(Application Programming Interface),与 JDK 一同发布的还包括 Java 类库的使用说明书,即 API 文档;

(2) Java 虚拟机(Java Virtual Machine,JVM),Java 若想在网络上实现跨平台运行及在网络上传输字节码文件,则在客户端计算机上需要一个能实现字节码文件解释和执行的软件,这个软件即 Java 虚拟机;

(3) Java 工具库,包括 Java 编译器(实现 Java 原程序代码到字节码文件的转换)、Java 打包软件(将开发的一个软件的相关类打包成一个文件)、Java 文档生成工具(将 Java 源程序中的注释文档提取并生成说明文档)等。

Java 的公布在当时的互联网领域获得了广泛的响应。在随后的几年里(1999 年之前),Java 对 JDK 做了多次升级。现在都习惯将这一段时间的 Java 版本称为 Java 第一版。在这段时间的 Java 技术逐步走向成熟,同时也被广泛地接受。

Java 第一版的代表性技术主要包括:

(1) 支持即时编译的 JVM,使得 Java 实现跨平台运行,完成了 Sun Microsystems 公司提出的“Write Once, Run Anywhere”的口号;

(2) 支持 Applet,在 HTML 网页文件中嵌入 Java 字节码指令,通过支持 Applet 的浏览器在客户计算机上执行,使原来只能显示文本信息的 Web 页面可以显示各种图片和动态图形,同时具备可交互性;

(3) AWT(Abstract Window ToolKit),一个能够兼容各种运行平台的图形用户界面接口。AWT 中的图形函数与操作系统所提供的图形函数之间存在着——对应的关系,这使得 Java 的跨平台特征包含应用程序前台界面。

1.1.2 Java 的成熟和发展

在 1998 年底,Sun Microsystems 公司发布了 JDK1.2,同时将其技术体系剪裁为面向不同应用的三个版本,并采用了新的命名格式。

- J2SE(Java 2 Platform, Standard Edition),是 Java 的标准版,用于面向桌面应用程序

含有 Applet 的网页的 HTML 文件在代码中采用 < applet > 关键字标记出 Applet 部分, 当支持 Java 的网络浏览器遇到这样的标记时, 会将 Applet 部分下载到本地计算机上执行。

Java Applet 是采用 Java 语言来编写的小应用程序, 将其嵌入到网页中, 可以提高 Web 页面的交互能力和动态执行能力。当然, 本地计算机使用的浏览器必须是支持 Java 的网络浏览器。

在 Java Applet 中, 可以实现文字的字体和颜色的设计, 支持图形绘制, 支持动画和声音的插入, 可以实现丰富的人机交互功能。

2. Servlet

Servlet 的名词来自 Server Applet, 意思是“服务器小程序”。Servlet 是用 Java 编写的运行在服务器端的程序。其主要功能是响应浏览器客户端的交互, 并生成动态 Web 网页内容。

与 Applet 不同的是, Servlet 运行在服务器中, 而 Applet 与浏览器一起运行在客户端。同时 Applet 具有图形界面, 而 Servlet 没有图形界面。

在 Servlet 出现之前, 能为浏览器客户提供动态响应的技术是 CGI (Common Gateway Interface, 通用网关接口) 程序。通过 CGI, Web 服务器会根据浏览器客户端的请求启动不同的外部程序, 并将请求内容转发给该程序, 再通过外部程序生成对客户端的响应。

与 CGI 技术相比, Servlet 具有明显的优势, 主要表现在以下三个方面。

(1) 效率高。CGI 对于每一个用户请求都要启动一个新的进程, 而 Servlet 在服务器端仅需要运行一个 Java 虚拟机程序, 就可以通过不同的线程 (而不是进程) 对应多个不同的用户请求。

(2) 跨平台性。Servlet 通过使用 Java 语言来编写, 在运行 Java 环境的任何操作系统下都可以运行, 而 CGI 是需要针对不同的运行平台进行编译的。

(3) 功能强。相比于 CGI, Servlet 的功能更强。

Servlet 存在的问题是, 为了能够输出 HTML 格式内容, 需要编写大量重复代码, 为了弥补这一缺陷, Sun Microsystems 公司推出了 JSP (Java Server Pages) 技术。

3. JSP

JSP 是在 Sun Microsystems 公司的倡导下, 由多家公司参与制定的一种动态网页技术标准, 是集成在 Java EE 中的重要技术。JSP 技术的基础和核心是 Servlet 技术, 是为了提高 Servlet 的编程效率而制定的新技术标准。

JSP 的出现与微软推出的 ASP (Active Server Page) 形成了竞争关系, 两者都属于动态 Web 网页技术。这两种技术的差别主要体现在微软技术的封闭性和 Java 技术的开放性。ASP 技术主要应用于 Windows 平台, 不具备跨平台性, 而 JSP 却是具备跨平台性的, 在互联网时代, 跨平台性标志着具有更强大的生命力。

JAS 和 ASP 都是通过在网页文件中嵌入脚本代码来实现的, 不同之处在于 ASP 采用的是 JavaScript 或 VBScript, 而 JSP 嵌入的是 Java 代码, 由 Servlet 实现。JavaScript 是早期由 Netscape 公司设计的一种脚本语言, 因为 Netscape 与 Sun Microsystems 公司曾经有过的合作关系, Netscape 给它起了一个带 Java 字样的名字, 其实 JavaScript 并不是 Sun Microsystems 公司的技术, 但 JavaScript 在语法上的确与 Java 很相像。另一项相关的技术是 JScript, 是由微软制定的另一种脚本语言, 实际上可以说是微软版本的 JavaScript。

所谓对象,就是指将数据和对该数据进行处理的程序代码封装起来,成为一个相互依存、不可分割的程序单位,这个程序单位被称为“类”。对于相同类型的对象进行分类、抽象后得的能对其共同特征的描述体就是类。在面向对象的程序设计中,类是对对象的抽象,而对象是类的一个实例。

Java 属于纯面向对象的程序设计语言,任何代码都要以类的形式出现。Java 程序从代码的角度看,一切皆为类;从运行的角度看,一切皆为对象。

面向对象技术使应用程序的开发变得简单易用,节省代码,同时在程序复杂度很高的情况下能保持代码的结构清晰和简单。

3. 简单

Java 程序借鉴了 C++ 程序的很多东西,包括运算符、流程控制语句等,这使得从语句层面来看,Java 的语句与 C++ 的语句几乎没有大的区别,这也使得 C++ 程序员可以较容易地掌握 Java。

Java 废弃了 C++ 中的指针等功能,这就避免了由于指针复杂性和多样化所带来的问题。可以这样理解,与 C++ 相比,Java 保持了更高层次的抽象。

Java 引入了“垃圾回收站”的功能。在程序中定义的各种对象都会占用内存空间,如果一个对象的成员在后面的代码中不再被引用,那么这个对象也就完成了它的使命,即成为了“垃圾”。在 C/C++ 中,需要编写代码释放垃圾所占用的空间,而在 Java 中,垃圾回收是自动完成的,不需要程序干预。

另外,Java 的运算器也相对简单,Java 中不存在 8 位和 16 位运算,仅有 32 位和 64 位运算,以及更高精度的运算。在基础数据类型方面,Java 不支持无符号整型数,所有整型数均为有符号数。在文字编码方面,Java 也更为简单,这是因为 Java 仅支持文字的 Unicode 编码。

4. 安全

在网络环境下,安全是一个重要的问题,Java 具有较好的安全结构和丰富的安全策略,代码在编译和实际运行过程中都会接受安全检查。这样做可以防止恶意程序入侵,同时取消指针机制后,使在对内存变量的访问时必须通过对象的实例变量来实现,可以有效地防止内存泄漏。

5. 多线程

Java 是通过多线程运行机制来支持多任务和并行处理。多线程机制使应用程序能够并行地执行,而且同步机制保证了对共享数据的正确操作。通过使用多线程机制,可以设计出不同的线程来完成应用程序的不同任务,这在完成具有用户交互功能的程序方面很有帮助。

1.2.2 Java 的相关技术

Java 的最重要技术特征是跨平台性,这使得 Java 在面向互联网的程序开发中得到了广泛的应用。

1. Applet

Applet 采用 Java 编程语言编写的小应用程序,不能独立地运行,而是需要嵌入到其他程序中。一般是将 Applet 包含在 Web 网页文件中,类似于在网页中嵌入图像的方式。在较早的 Java SE 版本中就开始支持 Applet。

Foundation 公司提供的通用 IDE 平台,它不是 Java 专用的,可以用于 Java 以外的其他编程语言,Eclipse 更适用于专业的开发人员使用,Android 应用软件开发时多用 Eclipse。

另一个 Java IDE 是 NetBeans,该平台是 Sun Microsystems 公司提供的。NetBeans 是一个开源的通用 IDE 平台,通过扩展不同的插件,可以实现 Java、PHP、C/C++ 等多种语言的开发。本教材的全部开发都是在 NetBeans 下完成的。

1.3.1 JDK 和 IDE 下载安装

通过浏览器访问 java.sun.com,可以打开 Oracle 的 Java 主页,单击 Java Downloads 按钮进入 Java 的下载页面。可以单独下载 JDK,也可以打包下载 Netbeans with JDK,后者将下载 NetBeans 和 Java SE 的 JDK 的安装软件包。

截止到 2017 年 5 月,NetBeans 的最高版本是 8.2,Java SE JDK 的最高版本是 8u131 (Java SE 8,第 131 次更新版)。阅读许可说明 (License) 后,勾选“Accept License Agreement”选项卡,选择安装平台 (Windows、Mac OS 或 Linux) 对应的安装包完成下载。打包下载的安装包中包括 JDK、NetBeans 和 Java SE 开发的插件。JDK/NetBeans 下载页面如图 1-1 所示。

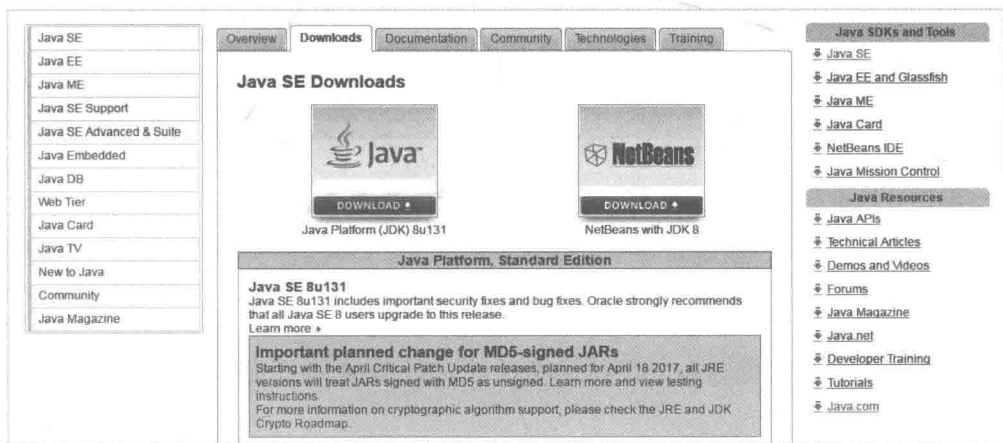


图 1-1 JDK/NetBeans 下载页面

如果下载的运行版本是 Windows 64 位版,下载的文件是: `jdk-8u131-nb-8_2-windows-x64.exe`,双击该文件,按照提示可以轻松地完成安装。

1.3.2 NetBeans 简介

安装完成后,打开 NetBeans 窗口,首先了解一下 NetBeans 的操作界面,如图 1-2 所示。

打开 NetBeans 窗口,呈现一个多窗口的集成开发界面。软件的界面风格与一般的应用程序一样。在分类菜单中可以找到所有的功能,菜单行下面是快捷键,使用最频繁的功能可以通过快捷键来实现。主要的窗口包括:

- 源程序: 编辑 Java 代码的窗口;
- 项目: 在 NetBeans 开发一个软件,需要创建一个项目,项目是开发特定应用软件所需要的源程序文件、库文件等多个文件的集合。项目窗口以树型结构显示一个项目