



高等院校计算机应用系列教材

Java Web 程序设计教程 (第二版)(微课版)

和孟佯 主 编

赵国桦 副主编

清华大学出版社
北 京



内 容 简 介

Java Web开发是当今互联网领域最为流行的开发方式之一，Java Web应用程序的规模和复杂度不断提高，作为开发人员需要掌握更多的技术和工具来满足不断变化的需求。全书共分为13章，包括Web应用开发概述、Java EE开发及运行环境、JSP与Servlet、使用JSP标签库、Struts 2框架基础、SQL与JDBC、Hibernate框架基础、Hibernate性能优化、Spring框架基础、Spring Web MVC、Spring MVC整合Hibernate及RESTful Web服务等内容。

本书力求通俗易懂，采用了大量的实例演示和案例分析，主要面向Java Web应用开发的初学者，也适合作为各种Java Web开发培训班的教材、高等院校Java Web程序设计课程的教材，还可作为Java Web应用开发人员的参考资料。

本书配套的电子课件、实例源文件和习题答案可以到<http://www.tupwk.com.cn/downpage>网站下载，也可以扫描前言中的二维码获取。扫描前言中的视频二维码可以直接观看教学视频。

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。举报：010-62782989，beiqinquan@tup.tsinghua.edu.cn。

图书在版编目(CIP)数据

Java Web程序设计教程：微课版 / 和孟伴主编. —2版. —北京：清华大学出版社，2023.8
高等院校计算机应用系列教材
ISBN 978-7-302-64248-0

I. ①J… II. ①和… III. ①JAVA语言—程序设计—高等学校—教材 IV. ①TP312.8

中国国家版本馆 CIP 数据核字 (2023) 第 136020 号

责任编辑：胡辰浩

封面设计：高娟妮

版式设计：孔祥峰

责任校对：成凤进

责任印制：丛怀宇

出版发行：清华大学出版社

网 址：<https://www.tup.com.cn>，<https://www.wqxuetang.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-83470000 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

印 装 者：三河市龙大印装有限公司

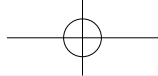
经 销：全国新华书店

开 本：185mm×260mm 印 张：24 字 数：569千字

版 次：2017年6月第1版 2023年11月第2版 印 次：2023年11月第1次印刷

定 价：86.00元

产品编号：093352-01



前

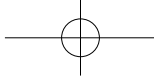
言

Java自诞生以来，便成为全球最流行、使用最广泛的Web开发语言之一，至今仍保持着同样的地位。随着Java语言的推广和应用，基于Java的各种针对Web开发的框架技术应运而生。本书以Java Web开发环境为背景，从开发环境的搭建讲起，遵循“从简单到复杂”“从抽象到具体”的原则，介绍Java Web开发的核心技术，以及Web开发的基本步骤和基本方法。

本书共13章，第1章内容是Web应用开发概述，主要介绍Web开发的基本概念和使用Java开发Web应用的相关技术与常用开发环境。第2章介绍Java EE开发环境的搭建，包括JDK、Tomcat及Eclipse的下载、安装与配置。第3章介绍JSP和Servlet，包括JSP的3个编译指令、7个动作指令和9个内置对象，以及Servlet的工作原理、过滤器与监听器等。JSP和Servlet是开发Java Web应用程序的两种基本技术，一些主流框架都是以这些基本技术为基础演变而来的。第4章介绍JSTL标签库的使用。第5章介绍Struts 2框架的基本知识，包括MVC框架的相关内容、Struts 2的工作流程、Action的工作原理以及拦截器工作机制和用法等。第6章介绍SQL与JDBC，这是应用程序与数据库交互的基础。第7章和第8章介绍Hibernate框架，包括Hibernate环境的搭建、Hibernate的工作流程、事务控制及缓存机制等。第9章开始讲解Spring框架，介绍Spring的IoC与AOP。第10章介绍Spring Web MVC，它是Spring框架中用于Web应用开发的一个模块，是当今流行的Web开发框架之一，也是本书的重点。第11章介绍Spring MVC与Hibernate的整合，这也是实际项目中应用最广泛的一种框架组件。第12章介绍RESTful Web服务，包括如何创建和测试RESTful Web服务。第13章是一个综合实例，使用Spring MVC + Hibernate框架开发了一个简易的图书馆管理系统。

本书内容丰富、结构合理、思路清晰、语言简练流畅、示例翔实并附有教学视频。每章的开始部分都概述了该章的作用和内容，指出该章的学习目标。正文部分结合每章的知识点和关键技术，穿插了大量极富实用价值的示例，所有示例都在Eclipse + Tomcat 8.5 + JDK 1.8环境下调试运行通过。每章的末尾都有本章小结，总结本章的内容、重点与难点；安排了有针对性的思考和练习，帮助读者巩固本章所学内容，提高读者的实际动手能力；同时录制了教学视频，对每章的知识点进行深入讲解和总结。

本书主要面向Java Web应用开发的初学者，也适合作为各种Web开发培训班的教材、高等院校Java Web开发相关课程的教材，还可作为Web应用程序开发人员的参考资料。



本书由和孟佺任主编，赵国桦任副主编，此外参与本书编写的人员还有李金阳、张世豪等。

在本书的编写过程中参考了相关文献，在此向这些文献的作者深表感谢。由于作者水平有限，书中难免有不足之处，恳请专家和广大读者批评指正。我们的电话是010-62796045，信箱是992116@qq.com。

本书配套的电子课件、实例源文件和习题答案可以到<http://www.tupwk.com.cn/downpage>网站下载，也可以扫描下方二维码获取。扫描下方二维码“看视频”二维码可以直接观看教学视频。

扫描下载



配套资源

扫一扫

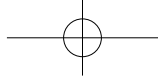


看视频

作者
2023年5月

目 录

| | |
|-------------------------------------|---|
| 第1章 Web应用开发概述 1 | 第3章 JSP与Servlet 38 |
| 1.1 Web应用概述 1 | 3.1 一个简单的JSP+Servlet应用 38 |
| 1.1.1 Web与Web应用 2 | 3.1.1 创建Servlet类 39 |
| 1.1.2 Web应用是如何运行的 4 | 3.1.2 部署Servlet 41 |
| 1.1.3 服务器端开发技术对比 5 | 3.1.3 创建JSP文件 43 |
| 1.2 使用Java开发Web应用 7 | 3.2 JSP技术初步 44 |
| 1.2.1 Java语言简介 7 | 3.2.1 JSP的工作原理 44 |
| 1.2.2 丰富的框架技术 9 | 3.2.2 JSP文件中的内容 45 |
| 1.2.3 Java Web应用的核心技术 10 | 3.2.3 JSP的内置对象 52 |
| 1.2.4 使用Ajax和jQuery提升用户体验 11 | 3.2.4 JSP中的中文显示问题 58 |
| 1.3 集成开发环境介绍 12 | 3.3 Servlet的开发与应用 63 |
| 1.3.1 常用的Java IDE 12 | 3.3.1 Servlet的生命周期 63 |
| 1.3.2 Web服务器汇总 14 | 3.3.2 使用HttpServletRequest 67 |
| 1.4 本章小结 15 | 3.3.3 使用HttpServletResponse 69 |
| 1.5 思考和练习 15 | 3.3.4 使用过滤器 71 |
| | 3.3.5 使用监听器 76 |
| 第2章 Java EE开发及运行环境 16 | 3.4 使用jQuery异步请求数据 79 |
| 2.1 下载并安装JDK 16 | 3.4.1 下载jQuery库 80 |
| 2.1.1 安装JDK 17 | 3.4.2 jQuery与Ajax 80 |
| 2.1.2 配置环境变量 18 | 3.5 本章小结 86 |
| 2.1.3 JDK的简单使用 19 | 3.6 思考和练习 86 |
| 2.2 Tomcat的安装与配置 20 | |
| 2.2.1 下载并安装Tomcat 20 | 第4章 使用JSP标签库 87 |
| 2.2.2 在Tomcat中部署和卸载应用程序 23 | 4.1 JSP标准标签库(JSTL) 87 |
| 2.3 Eclipse开发环境介绍 26 | 4.1.1 下载JSTL安装包 88 |
| 2.3.1 安装Eclipse 26 | 4.1.2 表达式语言(EL) 88 |
| 2.3.2 使用Eclipse新建Java EE应用 26 | 4.1.3 使用核心标签库(C名称空间) 92 |
| 2.3.3 在Eclipse中使用Tomcat 30 | 4.1.4 使用国际化和格式化标签库 (FMT命名空间) 101 |
| 2.3.4 Eclipse的常用快捷键 35 | 4.1.5 使用SQL标签库(SQL名称空间) 102 |
| 2.4 本章小结 37 | 4.1.6 使用XML标签库(X名称空间) 104 |
| 2.5 思考和练习 37 | |



| | |
|---------------------------|------------|
| 4.1.7 使用JSTL函数 | 104 |
| 4.2 使用JSTL开发用户管理系统 | 105 |
| 4.2.1 数据库设计 | 105 |
| 4.2.2 开发与实现 | 105 |
| 4.2.3 部署并测试应用 | 109 |
| 4.3 自定义标签库 | 110 |
| 4.3.1 一个最简单的自定义标签 | 111 |
| 4.3.2 访问标签体 | 114 |
| 4.3.3 自定义标签属性 | 115 |
| 4.4 本章小结 | 116 |
| 4.5 思考和练习 | 116 |

第5章 Struts 2框架基础 117

| | |
|-----------------------|------------|
| 5.1 MVC框架 | 117 |
| 5.1.1 框架内容 | 118 |
| 5.1.2 框架和设计模式的区别 | 119 |
| 5.1.3 MVC的优点 | 120 |
| 5.2 Struts 2基础 | 120 |
| 5.2.1 Struts 2中的MVC | 121 |
| 5.2.2 Struts 2的工作流程 | 122 |
| 5.2.3 一个简单的Struts 2应用 | 123 |
| 5.2.4 Action详解 | 126 |
| 5.2.5 struts.xml配置详解 | 127 |
| 5.2.6 Struts 2标签库 | 131 |
| 5.3 拦截器 | 134 |
| 5.3.1 拦截器的工作机制 | 134 |
| 5.3.2 Struts 2内置拦截器 | 135 |
| 5.3.3 自定义拦截器 | 138 |
| 5.4 本章小结 | 140 |
| 5.5 思考和练习 | 140 |

第6章 SQL与JDBC 141

| | |
|----------------------|------------|
| 6.1 准备关系数据库 | 141 |
| 6.1.1 下载并安装MySQL | 141 |
| 6.1.2 安装MySQL JDBC驱动 | 143 |
| 6.2 SQL语言简介 | 144 |
| 6.2.1 SQL概述 | 144 |
| 6.2.2 SQL数据类型 | 144 |
| 6.2.3 常用SQL语句 | 145 |
| 6.3 JDBC简介 | 150 |

| | |
|-------------------|------------|
| 6.3.1 JDBC概述 | 150 |
| 6.3.2 JDBC驱动程序 | 151 |
| 6.3.3 使用JDBC操作数据库 | 152 |
| 6.3.4 JDBC示例 | 155 |
| 6.4 本章小结 | 163 |
| 6.5 思考和练习 | 163 |

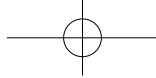
第7章 Hibernate框架基础 164

| | |
|---------------------------------|------------|
| 7.1 ORM简介 | 164 |
| 7.1.1 应用ORM的意义 | 165 |
| 7.1.2 流行的ORM框架 | 165 |
| 7.1.3 使用Hibernate ORM的原因 | 167 |
| 7.2 一个简单的Hibernate应用 | 167 |
| 7.2.1 下载Hibernate资源包 | 167 |
| 7.2.2 在Eclipse中引入Hibernate的JAR包 | 168 |
| 7.2.3 使用Hibernate代替JDBC | 171 |
| 7.3 认识Hibernate ORM | 176 |
| 7.3.1 Hibernate的框架结构 | 176 |
| 7.3.2 Hibernate配置文件详解 | 179 |
| 7.3.3 使用Hibernate映射文件 | 181 |
| 7.3.4 Hibernate的工作流程 | 183 |
| 7.4 Hibernate查询 | 184 |
| 7.4.1 Hibernate的检索策略 | 185 |
| 7.4.2 Hibernate的关联查询 | 191 |
| 7.4.3 Hibernate的查询方式 | 199 |
| 7.5 本章小结 | 202 |
| 7.6 思考和练习 | 202 |

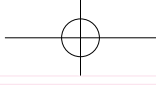
第8章 Hibernate性能优化 203

| | |
|------------------------------|------------|
| 8.1 Hibernate事务与并发 | 203 |
| 8.1.1 什么是事务 | 203 |
| 8.1.2 Hibernate Session和事务范围 | 204 |
| 8.1.3 并发控制 | 205 |
| 8.2 Hibernate缓存 | 212 |
| 8.2.1 Hibernate缓存概述 | 212 |
| 8.2.2 应用一级缓存 | 213 |
| 8.2.3 应用二级缓存 | 214 |
| 8.2.4 查询缓存 | 218 |
| 8.3 本章小结 | 220 |

| | | |
|--|--------------------------|-----|
| 8.4 | 思考和练习 | 220 |
| 第9章 Spring框架基础 221 | | |
| 9.1 | Spring框架简介 | 221 |
| 9.1.1 | Spring概述 | 222 |
| 9.1.2 | Spring框架的模块结构 | 223 |
| 9.1.3 | Spring 5.x新特性 | 225 |
| 9.2 | 从Hello World开始 | 227 |
| 9.2.1 | 下载Spring资源包 | 227 |
| 9.2.2 | 基于Spring的Hello World | 228 |
| 9.3 | Spring IoC | 230 |
| 9.3.1 | IoC基础 | 230 |
| 9.3.2 | IoC容器 | 232 |
| 9.3.3 | bean的装配 | 233 |
| 9.4 | Spring AOP | 241 |
| 9.4.1 | 什么是AOP | 241 |
| 9.4.2 | AOP相关概念 | 241 |
| 9.4.3 | 使用Spring的通知 | 242 |
| 9.5 | 本章小结 | 245 |
| 9.6 | 思考和练习 | 245 |
| 第10章 Spring Web MVC 246 | | |
| 10.1 | Spring Web MVC入门 | 246 |
| 10.1.1 | Spring Web MVC是什么 | 247 |
| 10.1.2 | 为什么使用Spring Web MVC | 247 |
| 10.1.3 | Spring Web MVC的工作流程 | 248 |
| 10.1.4 | Spring MVC的Hello World程序 | 249 |
| 10.2 | 深入学习Spring Web MVC | 252 |
| 10.2.1 | 启动Spring MVC | 252 |
| 10.2.2 | DispatcherServlet组件类 | 255 |
| 10.2.3 | 使用@RequestMapping | 257 |
| 10.2.4 | 控制器方法的参数 | 261 |
| 10.2.5 | 控制器方法的返回类型 | 269 |
| 10.2.6 | 模型与视图 | 271 |
| 10.2.7 | Spring MVC的表单标签库 | 282 |
| 10.3 | 本章小结 | 286 |
| 10.4 | 思考和练习 | 287 |
| 第11章 Spring MVC整合 Hibernate 288 | | |
| 11.1 | Spring提供的DAO支持 | 288 |
| 11.1.1 | J2EE应用的3层架构 | 288 |
| 11.1.2 | Spring的DAO理念 | 289 |
| 11.1.3 | 使用@Repository注解 | 290 |
| 11.2 | Spring MVC整合Hibernate 5 | 291 |
| 11.2.1 | 新建工程 | 291 |
| 11.2.2 | 创建实体类 | 292 |
| 11.2.3 | 创建Dao层 | 296 |
| 11.2.4 | 创建Service层 | 301 |
| 11.2.5 | 创建Controller控制器 | 303 |
| 11.2.6 | 创建JSP页面 | 305 |
| 11.2.7 | 配置Spring和Hibernate | 307 |
| 11.2.8 | 引入aspectjweaver.JAR包 | 314 |
| 11.2.9 | 项目运行结果 | 315 |
| 11.3 | 本章小结 | 315 |
| 11.4 | 思考和练习 | 315 |
| 第12章 RESTful Web服务 316 | | |
| 12.1 | Web服务概述 | 316 |
| 12.1.1 | 基于SOAP的Web服务 | 317 |
| 12.1.2 | RESTful Web服务概述 | 318 |
| 12.2 | 创建RESTful Web服务 | 324 |
| 12.2.1 | 新建工程 | 324 |
| 12.2.2 | 创建实体类、DAO层和Service层 | 325 |
| 12.2.3 | 基于REST的控制器 | 327 |
| 12.2.4 | 添加配置信息 | 329 |
| 12.3 | 测试RESTful Web服务 | 330 |
| 12.3.1 | 使用RESTClient Firefox插件 | 330 |
| 12.3.2 | 使用REST模板编写REST客户端 | 333 |
| 12.4 | 本章小结 | 335 |
| 12.5 | 思考和练习 | 335 |
| 第13章 图书馆管理系统 336 | | |
| 13.1 | 系统概述 | 336 |
| 13.1.1 | 项目背景 | 336 |
| 13.1.2 | 需求分析 | 337 |



| | | | |
|---------------------------------|-----|------------------------|------------|
| 13.2 数据库设计 | 337 | 13.3.7 图书管理功能 | 359 |
| 13.2.1 系统E-R图 | 337 | 13.3.8 读者登录及操作功能 | 365 |
| 13.2.2 数据表设计 | 338 | 13.4 系统运行结果 | 368 |
| 13.3 系统设计与实现 | 340 | 13.4.1 管理员操作页面 | 368 |
| 13.3.1 搭建系统框架 | 341 | 13.4.2 读者操作页面 | 371 |
| 13.3.2 配置Spring与Hibernate | 342 | 13.5 本章小结 | 372 |
| 13.3.3 创建实体类 | 343 | 13.6 思考和练习 | 372 |
| 13.3.4 管理员登录功能 | 343 | 参考文献 | 373 |
| 13.3.5 管理员管理功能 | 348 | | |
| 13.3.6 读者管理功能 | 356 | | |



第 1 章

Web 应用开发概述

Web使用超文本技术将Internet上的资源以页面的形式展示出来，Web应用是一种使用HTTP作为核心通信协议，通过Internet让Web浏览器和服务器通信的计算机程序，为浏览者在Internet上查找和浏览信息提供了图形化的、易于访问的直观界面。Java是由Sun公司推出的能够跨越多平台的、可移植性较高的一种面向对象的编程语言，特征丰富，功能强大，是一种应用范围较广泛的开发语言，特别是在Web程序开发方面。本章将从Web应用的基本概念开始，讲述与Java Web应用开发相关的技术和常用框架，最后介绍常用的集成开发环境与Web服务器。

本章学习目标

- 理解Web和Web应用的基本概念
- 掌握Web应用的工作原理
- 了解服务器端开发技术
- 了解Java语言的特点与发展前景
- 熟悉常用的Java Web框架技术
- 了解常用的集成开发环境

1.1 Web应用概述

Internet采用超文本和超媒体的信息组织方式，将信息的链接扩展到整个Internet上。而Web就是一种超文本信息系统，它使得文本不再像一本书一样是固定的、线性的，而是可以从一个位置跳到另一个位置并从中获取更多的信息，或是转到别的主题上。想要了解某一个主题的内容只要在这个主题上点一下，就可以跳转到包含这一主题的文档上。正是由于这种多连接性，它被称为Web。



1.1.1 Web与Web应用

Web应用是一种可以通过Web访问的应用程序。那么, Web是如何发展而来的呢? 什么样的应用才是Web应用呢?

1. Web的发展

Web即全球广域网(World Wide Web), 也称为万维网, 它是一种基于超文本和HTTP的、全球性的、动态交互的、跨平台的分布式图形信息系统。Web是建立在Internet上的一种网络服务, 其使用超文本技术将Internet上的资源以页面的形式展示出来, 为浏览者在Internet上查找和浏览信息提供了图形化的、易于访问的直观界面。Web上的资源十分丰富, 包括图片、文本、声音、视频等多媒体元素。通常所说的网页是一个包含HTML标签的纯文本文件(文件扩展名为.html、.htm、.asp、.aspx、.php或.jsp等), 它可以存放在世界某个角落的某一台计算机中, 是万维网中的一“页”。

Web在组成上包括以下两部分。

- 服务器: 通常包括物理设备和软件应用程序, 物理设备是指存放供用户访问的信息资源的远程计算机; 软件应用程序是指能够根据用户的请求将信息资源传递给用户的应用程序, 如Apache服务器。
- 客户端: 通常是指客户使用的本地计算机, 通过客户端浏览器向服务器发送请求, 然后接收并显示服务器传递过来的信息资源。

发展到今天, Web共经历了三个阶段: Web 1.0、Web 2.0和Web 3.0。其中Web 1.0被称为Internet第一代, 指的是2003年以前的Internet模式。在Web 1.0时代, Internet采用的是技术创新主导模式, 其主要特征是大量使用静态的HTML网页来发布信息, 并开始使用浏览器来获取信息, 这个时候主要是单向的信息传递。通过万维网, 互联网上的资源可以在一个网页里比较直观地表示出来, 而且在网页上各种资源之间可以任意链接。而Web 2.0则是以Internet为平台, 以用户为灵魂, 允许多人参与, 以可读、可写的模式成为Internet新的发展趋势。在Web 2.0中, 软件被当成一种服务, Internet从一系列网站演化成成熟的为最终用户提供网络应用的服务平台, 强调用户的参与、在线的网络协作、数据存储的网络化、社会关系网络、RSS应用以及文件的共享等成为Web 2.0发展的主要支撑和表现。如果说Web 1.0的本质是联合, 那么Web 2.0的本质就是互动, 它让网民更多地参与信息产品的创造、传播和分享, 而这个过程是有价值的。

在Web 2.0的基础上, 又有人提出了Web 3.0, 不过对Web 3.0的争议比较大, 常见的对Web 3.0的解读是: 网站内的信息可以直接和其他网站的相关信息进行交互, 能通过第三方信息平台同时对多家网站的信息进行整合使用; 用户在互联网上拥有自己的数据, 并能不同网站上使用; 完全基于Web, 用浏览器即可实现复杂系统程序才能实现的系统功能; 用户数据审计后, 同步于网络数据。总体而言, Web 3.0更多的不仅仅是一种技术上的革新, 而是以统一的通信协议, 通过更加简洁的方式, 为用户提供更为个性化的互联网信息资讯定制的一种技术整合。Web 3.0将会是互联网发展中由技术创新走向用户理念创新的关键一步。不管Web 3.0最终将向何处发展, 但不可否认的是, 从Web诞生至今, 它不仅改变

着人们联系、交流、获取知识的方式，而且也在改变着商业运行模式，已经成为人们日常生活和工作不可缺少的一部分。

2. 什么是Web应用

Web应用，顾名思义，是运行在Web上的应用程序。但是反过来，运行在Web上的应用程序都是Web应用吗？答案是否定的。这里所说的Web应用是指运行在网络上，以浏览器作为通用客户端的应用程序，在许多地方又被称为B/S(Browser/Server，浏览器/服务器)模式的应用。我们通过浏览器可以访问百度、淘宝、网易等网站，这些就是Web应用程序(简称Web应用)。最初，这些网站上的内容都是由静态页面组成的，页面上包含一些文本、图片等信息资源，用户可以通过链接来浏览信息。采用静态页面的缺陷非常多，如不能与用户进行交互，不能实时更新Web上的内容，因此像搜索引擎、股票行情等许多功能无法实现。于是出现了动态页面，即根据不同的用户或在不同的时间，呈现给用户的信息资源也不相同。这里的动态内容就是由Web应用程序来实现的。

Web应用程序是一种使用HTTP(Hypertext Transfer Protocol，超文本传输协议)作为核心通信协议，通过Internet让Web浏览器和服务器通信的计算机程序。

一个Web应用程序是由完成特定任务的各种Web组件构成的并通过Web将服务展示给外界。在实际应用中，Web应用程序由多个Servlet、JSP页面、HTML文件及图像文件等组成。所有这些组件相互协调，从而为用户提供一组完整的服务。

Web应用中的每一次数据交换都要涉及客户端和服务端两个层面。因此，Web应用程序的开发技术分为客户端开发技术和服务器端开发技术两种。

3. 客户端开发技术

常用的客户端开发技术包括如下内容。

- **HTML**：超文本标记语言，是Web的描述语言。无论哪种动态页面开发技术，都无法摆脱HTML的影子。HTML是所有动态页面开发技术的基础，这些动态页面开发技术无非是在静态HTML页面的基础上添加了动态的可以交互的内容。
- **CSS**：层叠样式表(Cascading Style Sheets)，也就是通常所说的样式表，是用于增强控制网页样式并允许将样式信息与网页内容分离的一种标记性语言。通过使用CSS，开发者可以方便、灵活地设置网页中不同元素的外观属性，通过这些设置可以使网页在外观上达到一个更高的级别。
- **ActiveX**：一个集成平台，使用ActiveX可以方便地在Web页中插入多媒体效果、交互式对象、复杂程序等。
- **JavaScript**：一种简单的脚本语言，可以在浏览器中直接运行，JavaScript的出现给静态的HTML网页带来很大的变化。JavaScript增加了HTML网页的互动性，使以前单调的静态页面变得富有交互性。它可以在浏览器端实现一系列动态的功能，仅仅依靠浏览器就可以完成一些与用户的互动。
- **jQuery**：一个快速、简洁的JavaScript框架，是继Prototype之后又一个优秀的JavaScript代码库。jQuery具有独特的链式语法和短小清晰的多功能接口；具有高



效灵活的CSS选择器，并且可对CSS选择器进行扩展；拥有便捷的插件扩展机制和丰富的插件。

- 其他：VBScript、Applet等。

4. 服务器端开发技术

常用的服务器端开发技术包括如下内容。

- **Servlet/JSP**：一种独立于平台和协议的Java应用程序，可以生成动态Web页面。它担当Web浏览器或其他发出请求的HTTP客户端程序，是与HTTP服务器上的数据库或应用程序之间的中间层。JSP是一种建立在Servlet规范提供的功能之上的动态网页技术。JSP文件在用户第一次请求时，会被编译成Servlet，然后由这个Servlet处理用户的请求，所以JSP可以被看成运行时的Servlet。
- **PHP**：“超文本预处理器”，是在服务器端执行的脚本语言，尤其适用于Web开发并可嵌入HTML中。该语言当初创建的主要目标是让开发人员快速编写出优质的Web网站。
- **ASP/ASP.NET**：微软公司推出的用于构建Windows服务器平台上的Web应用程序。
- 其他：CGI、Perl、ISAPI等。

5. 应用程序的模式

应用程序有C/S、B/S两种模式。C/S(Client/Server)是客户端/服务器端程序。也就是说，这类程序一般独立运行。而B/S就是浏览器端/服务器端应用程序，这类应用程序一般借助IE等浏览器来运行。Web应用程序一般采用B/S模式。Web应用程序首先是“应用程序”，和用标准的程序语言(如C、C++等)编写出来的程序没有本质上的不同。然而Web应用程序又有自己独特的地方，就是它是基于Web的，而不是采用传统方法运行的。换句话说，它是典型的浏览器/服务器架构的产物。

采用B/S模式的Web应用程序分为3层结构。

- **界面层**：主要用于接收用户的请求，以及数据的返回，为用户提供管理系统的访问。
- **业务逻辑层**：主要负责对业务逻辑和功能的操作，也就是把一些数据层的操作进行组合。该层结构将浏览器和数据层屏蔽，安全性更高。
- **数据访问层**：主要看数据层里面是否包含逻辑处理，实际上它的各个函数主要完成对数据文件的各种操作，而不必管其他操作对这三层所进行的明确分割，并在逻辑上使其独立。

随着Internet和手机上网的普及，Web应用程序已经成为目前最流行的应用程序类型。

1.1.2 Web应用是如何运行的

随着网络的普及，大家对上网的过程越来越熟悉，上网的一般过程如下：① 打开浏览器；② 输入某个网址；③ 经过一段时间的等待，浏览器显示要访问的信息。

然后可以在网页上继续进行其他操作。例如：在网页上单击超链接，访问其他内容；

或者在网页中输入一些信息，然后单击搜索按钮，等待浏览器中内容的再次更新。

不管是在地址栏中输入地址，还是单击超链接或者单击搜索按钮，都需要等待浏览器中内容的更新。等待浏览器内容更新的过程，实际上就是浏览器访问 Web 应用的过程。这个过程如下：

(1) 浏览器根据我们输入的地址找到相应的服务器，不同的网站对应不同的服务器。这个服务器通常称为 Web 服务器，可以接收浏览器发送的请求。

(2) Web 服务器根据请求的内容调用不同的服务器端程序，服务器端程序通常也是一个服务器，被称为应用服务器。

(3) 应用服务器接收到请求之后，查找相应的文件，加载并执行相应的任务。如果涉及数据处理，则需要与数据库服务器交互。

(4) 处理完数据，将处理结果返回给应用服务器，服务器端程序的执行结果通常是 HTML 文档。

(5) 应用服务器把执行的结果返回给 Web 服务器，Web 服务器再把这个结果返回给客户端浏览器。

(6) 浏览器解析 HTML 文档，然后把解析后的网页显示给最终用户。

Web 应用的工作原理如图 1-1 所示。

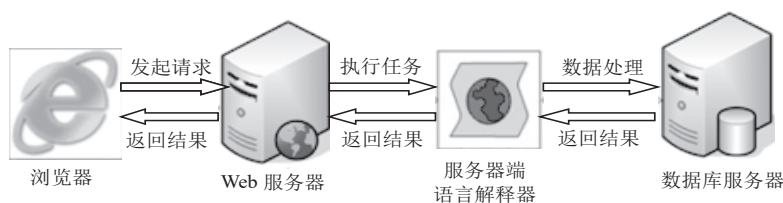


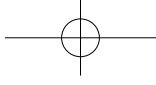
图1-1 Web应用的工作原理

1.1.3 服务器端开发技术对比

静态网页是网站建设的基础，早期的网站一般都是由静态网页制作的，网站只能实现静态的信息展示。静态网页是没有后台数据库、不含程序和不可交互的网页。静态网页相对更新起来比较麻烦，适用于一般更新较少的展示型网站。实际上静态也不是完全静态，它也可以出现各种动态的效果，如 GIF 格式的动画、Flash、滚动字幕等。但是静态网页不能与客户进行互动，不能满足用户不同的需要。为了满足不同用户各种各样的需求，需要网站或 Web 应用程序具有收集并处理用户需求的功能，这就有了后来的一系列动态页面语言。

动态页面，是指可以和用户进行交互，能根据用户的输入信息产生相应响应的页面，能满足这种需求的语言可以称之为动态语言。

较早的动态网页技术主要使用 CGI，现在常用的动态网页技术有 ASP/ASP.NET、JSP、PHP 等。下面分别介绍这几种动态语言。



1. CGI

在互联网发展的早期,动态网页技术主要使用CGI(Common Gate Interface,通用网关接口)。CGI程序被用来解释处理表单中的输入信息,并在服务器中产生对应的操作处理,或是把处理结果返回给客户端的浏览器,从而可以给静态的HTML网页添加动态的功能。CGI的优点是可以用很多语言编写,如C、C++、VB和Perl语言,在语言的选择上有很大的灵活性。最常用的CGI开发语言为Perl。CGI的主要缺点是维护复杂,运行效率也比较低。因此在一段时间以后,CGI逐渐被其他新的动态网页技术所替代。

2. ASP/ASP.NET

ASP是由微软公司推出的一种动态网页语言,它可以将用户的HTTP请求传入ASP的解释器中。这个解释器对这些ASP脚本进行分析和执行,然后从服务器返回处理的结果,从而实现了与用户交互的功能。ASP的语法比较简单,对编程基础没有很高的要求,所以很容易上手,而且微软公司提供的开发环境的功能十分强大,这更是降低了ASP程序开发的难度。但是ASP也有自身的缺点。ASP在本质上还是一种脚本语言,除了使用大量的组件,没有其他办法提高效率,而且ASP只能运行在Windows环境中。这样,Windows自身的一些限制就制约了ASP的发挥,这些都是使用ASP时无法回避的弊端。

ASP.NET又称为ASP+,它不仅仅是ASP的简单升级,更是由微软公司推出的新一代脚本语言。ASP.NET是基于.NET Framework的Web开发平台,不但吸收了ASP以前版本的最大优点,并参照Java、VB语言的开发优势加入了许多新的特色,同时也修正了以前的ASP版本的运行错误。ASP.NET具备开发网站应用程序的一切解决方案,包括验证、缓存、状态管理、调试和部署等全部功能。在代码撰写方面,它的特色是将页面逻辑和业务逻辑分开,分离程序代码与显示的内容,让丰富多彩的网页更容易撰写,同时使程序代码看起来更简洁。

3. JSP

JSP(Java Server Pages)是由Sun Microsystems公司主导创建的一种动态网页技术标准。JSP部署于网络服务器上,可以响应客户端发送的请求,并根据请求内容动态地生成HTML、XML或其他格式文档的Web网页,然后返回给请求者。JSP技术以Java语言作为脚本语言,为用户的HTTP请求提供服务,并能与服务器上的其他Java程序共同处理复杂的业务需求。

JSP将Java代码和特定变动内容嵌入静态的页面中,实现以静态页面为模板,动态生成其中的部分内容。JSP引入了被称为“JSP动作”的XML标签,用来调用内建功能。另外,可以创建JSP标签库,然后像使用标准HTML或XML标签一样使用它们。标签库能增强功能和服务器性能,而且不受跨平台问题的限制。

由于JSP中使用的是Java语法,因此Java语言的所有优势都可以在JSP中体现出来。尤其是J2EE中的强大功能,更是成为JSP语言发展的强大后盾。

4. PHP

PHP和JSP类似,都是可以嵌套到HTML中的语言,不同之处在于:PHP的语法比较

独特，它混合了C、Java等多种语法中的优秀部分，而且PHP网页的执行速度要比CGI和ASP页面等快很多。在PHP中，提供了对常见数据库的支持，例如SQL Server、MySQL、Oracle、Sybase等，这种内置的方法使PHP中的数据库操作变得异常简单。而且PHP程序可以在IIS和Apache中运行，提供对多种操作系统平台的支持。

但是PHP也存在一些劣势，PHP开发运行环境的配置比较复杂，而且PHP是开源的产品，缺乏正规的商业支持。这些因素在一定程度上限制了PHP的进一步发展。

总之，各种动态语言都有着自身的优势和劣势，开发人员可以根据客户的需求来选择具体的语言。

1.2 使用Java开发Web应用

Java提供的JSP和Servlet是开发Web应用的两项引人注目的技术，同时它的开源项目也是层出不穷，如Web框架Struts、Struts 2等，持久层框架Hibernate、iBATIS等，J2EE框架Spring，模板引擎Velocity、FreeMarker等。

1.2.1 Java语言简介

Java是一种跨平台的面向对象的编程语言，由Sun公司于1995年推出。Java语言自从问世以来，受到越来越多开发者的喜爱。在Java语言出现以前，很难想象在Windows环境下编写的程序可以不加修改就能在Linux系统中运行，因为计算机硬件只识别机器指令，而不同操作系统中的机器指令是有所不同的。所以，要把一种平台下的程序迁移到另一种平台，必须针对目标平台进行修改。如果想要程序运行在不同的操作系统上，就要求程序设计语言能够跨平台，可以跨越不同的硬件、软件环境，而Java语言就能够满足这种要求。

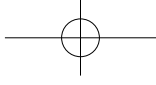
1. Java语言的特点

Java是一种优秀的面向对象编程语言。在Java语言中，有着健壮的安全设计，它的结构是中立的，可以移植到不同的系统平台。优秀的多线程设计也是Java语言的一大特色。

目前，Java语言最大的用途就是Web应用的开发。使用Java语言不用考虑系统平台的差异，在一种系统下开发的应用系统，可以不加任何修改就能运行在另一种不同的系统中。例如，开发人员在Windows平台下开发的Web应用程序，可以直接部署在Linux或UNIX服务器系统中。

Java语言之所以如此受欢迎，是由其自身的优点决定的。以下简单介绍Java语言的特性。

- 平台无关性：在Java中，并不是直接把源文件编译成硬件可以识别的机器指令。Java的编译器把Java源代码编译为字节码文件，这种字节码文件就是编译Java源程序时得到的class类文件，执行这种类文件的是Java虚拟机。Java虚拟机是软件模拟出的计算机，可以执行编译Java源文件得到的中间码文件，而各种平台的差异就是由Java虚拟机来处理的，从而实现了可以在各种平台上运行Java程序的目的。



- **安全性：**Java语言放弃了C/C++中的指针操作。在Java中，没有显式提供指针操作，不提供对存储器空间直接访问的方法，这样就可以保证系统的地址空间不会被有意或无意破坏。而且经过这样的处理，也可以避免系统资源的泄漏。例如在C/C++中，如果指针不及时释放，就会占用系统内存空间，而Java提供了一套有效的资源回收策略，会自动回收不再使用的系统资源，从而保证了系统的安全性和稳定性。
- **面向对象：**面向对象是现代软件开发中的主流技术，Java语言继承了C++面向对象的理论，并简化了这种面向对象的技术，去掉了一些复杂的技术，例如多继承、运算符的重载等功能。在Java程序中，所有的操作都是在对象的基础上实现的，为了实现模块化和信息的隐藏，Java语言采用了将功能代码封装的处理方法，Java语言对继承性的实现使功能代码可以重复利用。用户可以把具体的功能代码封装成自定义的类，从而实现了对代码的重用。
- **异常处理：**Java中的异常处理可以帮助用户定位处理各种错误，从而大大缩短了Java应用程序的开发周期。而且这种异常策略可以捕捉到程序中的所有异常，针对不同的异常，用户可以采取具体的处理方法，从而保证了应用程序在用户的控制下运行，保证了程序的稳定和健壮。
- **稳健性：**Java是一种强类型语言，它允许扩展编译时检查潜在类型不匹配问题的功能。Java要求显式的方法声明，它不支持C风格的隐式声明。这些严格的要求保证编译程序能捕捉调用错误，确保程序可靠。

2. Java语言的发展

Java语言和Java平台的发展是一段漫长而富于传奇的历史。从20世纪90年代中期发明开始，Java已经经历了许多变化，也遇到过许多争论。在早期，Java被称为Java开发工具包或JDK，是一门与平台(由一组必需的应用程序编程接口API组成)紧密耦合的语言。

针对不同的开发市场，Sun公司将Java划分为3个技术平台，它们分别是Java SE、Java EE和Java ME。

- **Java SE(Java Platform Standard Edition, Java标准版)**，是为开发普通桌面和商务应用程序提供的解决方案。Java SE是3个平台中最核心的部分，Java EE和Java ME都是从Java SE的基础上发展而来的，Java SE平台中包括了Java最核心的类库，如集合、IO、数据库连接以及网络编程等。
- **Java EE(Java Platform Enterprise Edition, Java企业版)**，是为开发企业级应用程序提供的解决方案。Java EE可以被看作一个技术平台，该平台用于开发、装配以及部署企业级应用程序，其中主要包括Servlet、JSP、JavaBean、JDBC、EJB、Web Service等技术。
- **Java ME(Java Platform Micro Edition, Java小型版)**，是为开发电子消费产品和嵌入式设备提供的解决方案。Java ME主要用于小型数字电子设备上软件程序的开发，例如，为手机增加新的游戏和通讯录管理功能。此外，Java ME提供了HTTP等高级Internet协议，使移动电话能以C/S模式直接访问Internet的全部信息，提供最高效率的无线交流。

Sun公司提供了一套Java开发环境,简称JDK(Java Development Kit),它是整个Java的核心,其中包括Java编译器、Java运行工具、Java文档生成工具、Java打包工具等。为了满足用户日新月异的需求,JDK的版本也在不断地升级。

Sun公司除了提供JDK,还提供了一种JRE(Java Runtime Environment)工具,它是提供给普通用户使用的Java运行环境。由于用户只需要运行事先编写好的程序,不需要自己动手编写程序,因此JRE工具中只包含Java运行工具。不包含Java编译工具。值得一提的是,为了方便使用,Sun公司在其JDK工具中自带了一个JRE工具,也就是说,开发环境中包含运行环境。这样一来,开发人员只需要在计算机上安装JDK即可。

3. 企业级Java的诞生

随着Internet的发展和Web应用程序的流行,Sun公司已经意识到应用程序开发对高级开发工具的需求。1998年,就在J2SE 1.2发布之前,Sun宣布正在开发一个称为Java专业版本或JPE的产品。同时Sun还研发了一种称为Servlet的技术,这是一个能够处理HTTP请求的小型应用程序。

Servlet和JPE在经历过几次内部迭代过程之后,Sun于1999年12月12日发布了Java 2平台的企业版(J2EE, Java 2 Enterprise Edition),版本为1.2。J2EE包含J2SE中的类,并且还包含用于开发企业级应用的类,例如EJB、Servlet、JSP、XML、事务控制等。在随后发布的版本中,J2EE迅速成为对J2SE的补充,并且随着多年的发展,一些组件已经被认为必须从J2EE迁移到J2SE中。随着版本的不断升级,从JDK 5.0开始,不再叫J2SE和J2EE了,而改名为Java SE和Java EE了,因为那个“2”已经失去了其本应该有的意义。

4. Java语言的发展前景

虽然说Java语言并不是为网络环境设计的,但是Java语言目前还是主要被用于网络环境中,尤其是在服务器端的程序设计中,Java语言的地位是其他动态语言所无法替代的。尤其是在B/S开发模式盛行的今天,Java语言的地位更是举足轻重。在Java EE中,提供了优秀的B/S应用程序的解决方案。再加上Java语言跨平台、简单易用等特性,用户自然会选择Java语言进行开发。随着网络技术的急速发展,Java语言必然会取得更大的发展,在这个复杂的网络环境中,Java语言有着广阔的前景。

1.2.2 丰富的框架技术

框架其实就是可重用的设计架构,应用框架强调的是软件的设计重用性和系统的可扩充性,以缩短大型应用软件系统的开发周期,提高开发效率和质量。

使用Java开发Web应用的常用框架有很多,下面简要介绍一些比较常用的框架。

1.Struts 2

Struts 2以WebWork优秀的设计思想为核心,吸收了Struts框架的部分优点,提供了一个更加整洁的MVC设计模式实现的Web应用程序框架。



同时Struts 2引入了几个新的框架特性,例如从逻辑中分离出横切关注点的拦截器、减少或者消除配置文件、贯穿整个框架的强大表达式语言、支持可变更和可重用的基于MVC模式的标签API,Struts 2充分利用了从其他MVC框架学到的经验和教训,使得 Struts 2框架更加清晰、灵活。

2. Spring

Spring是一个以IoC(Inversion of Control,控制反转)和AOP(Aspect Oriented Programming,面向切面编程)为核心的轻量级容器框架。它提供了一系列的Java EE开发解决方案,包括表示层的Spring MVC、持久层的Spring JDBC、业务层事务管理等众多的企业级应用技术。目前,Java Web应用开发的主流框架就是Spring。

3. Hibernate

Hibernate是一个ORM(Object Relational Mapping,对象-关系映射)框架,它对JDBC进行了轻量级封装。通过使用Hibernate框架,开发人员能够以面向对象的思维方式来操作数据库。Hibernate框架需要开发人员创建一系列的持久化层,每个类的属性都可被简单看作和一张数据库表的属性一一对应,也可以实现关系数据库的各种表件关联的对应。当我们需要相关操作时,可以不用关注数据库表,也不用再一行行查询数据库,只需要持久化类就可以完成增删改查功能。

4. Apache Shiro

Apache Shiro是功能强大并且容易集成的开源权限框架。Apache Shiro提供了认证、授权、加密和会话管理功能,将复杂的问题隐藏起来,提供清晰直观的API使开发者可以很轻松地开发自己的程序安全代码,并且在实现此目标时无须依赖第三方的框架、容器或服务。此外,它也能做到与这些环境的整合,使其在任何环境下都可拿来使用。Apache Shiro自身提供了对Spring的良好支持。

5. SiteMesh

SiteMesh是一个用来在JSP中实现页面布局和装饰的框架组件,利用它可以将网页的内容和页面结构分离,以达到共享页面结构的目的。

SiteMesh基于Servlet的过滤流,通过截取response,并进行装饰后再交付给客户。

除了上面介绍的这些框架,Java中还有很多框架,在应用开发中可根据实际的需求来选择使用。

1.2.3 Java Web应用的核心技术

Java Web应用的核心技术包括以下几个方面。

- JSP: 进行输入和输出的基本手段。JSP以脚本文件的形式存在,主要由HTML代码、客户端脚本(JavaScript等)、JSP的标签和指令、自定义标签库构成。下面是一个典型的JSP示例。

```

<%@ page contentType="text/html;charset=gb2312"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
学生信息如下:
<table>
<tr>
<th>学号</th>
<th>姓名</th>
<th>电话</th>
</tr>
<c:forEach items="{stu}" var="stu">
<tr>
<td>${stu.sid}</td>
<td>${stu.sname}</td>
<td>${stu.phone}</td>
</tr>
</c:forEach>
</table>

```

指令

HTML代码

标准标签库标签

HTML代码

在HTML中嵌套表达式语句

HTML代码

标准标签库标签

HTML代码

- **JavaBean**: 完成功能的处理。JavaBean就是Java中普通的Java类, 所以没有特殊的地方。Java Web技术中提供了多个与JavaBean操作相关的标签。
- **Servlet**: 对应用的流程进行控制。Servlet以Java文件的形式存在, 所以Servlet也是Java类, 是特殊的Java类, 在Java Web技术中主要完成控制功能, 负责协调JSP页面和完成功能的JavaBean之间的关系。
- **JDBC**: 与数据库进行交互不可缺少的技术。严格来讲, JDBC不属于Java Web技术, 但是在Java Web中不可避免地要使用JDBC, 所以JDBC也算是Java Web开发中比较重要的技术之一。
- **JSTL和表达式语言(Expression Language, EL)**: 完成对JSP页面中各种信息的控制和输出。JSTL和表达式语言是在JSP 2.0之后引入的, 主要是为了方便用户在JSP页面中使用常用功能。其典型应用是信息的输出, 因为JSP界面的主要功能就是展示信息, 使用表达式语言使得信息的显示非常简单。例如, 上面的 JSP 代码中的 `${stu.sid}`, 完成的功能是从请求中获取stu对象的sid属性。如果使用Java代码, 就没有这么简单了。另外, JSTL中提供了大量常用的功能, 例如选择结构和循环结构, 在上面的 JSP 例子中就使用了 `<c:forEach>` 标签来完成循环控制。

1.2.4 使用Ajax和jQuery提升用户体验

JavaScript的出现使网页和用户之间出现了一种实时性、动态的、交互性的关系, 这在一定程度上减轻了服务器的负载量, 为客户提供更流畅的浏览效果。

Ajax(Asynchronous JavaScript and XML)被称为异步的JavaScript与XML, 它是一种支持异步请求的技术, 可以使用JavaScript向服务器提出请求并处理响应, 而不阻塞用户。它最大的优点是在不重新加载整个页面的情况下, 可以与服务器交换数据并更新部分网页内容, 让用户感觉不到与服务器的交互过程, 从而获得更好的用户体验。

Ajax的核心是XMLHttpRequest对象, XMLHttpRequest是由微软公司开发的可以在不刷



新页面的情况下直接进行脚本与服务器间通信的技术。

XMLHttpRequest在发送请求的时候,有同步与异步两种方式。同步方式是请求发出后,一直到收到服务器返回的数据为止,浏览器进程被阻塞,在页面上什么事也做不了。而异步方式则不会阻塞浏览器进程,在服务器端返回数据并触发回调函数之前,用户依然可以在该页面上进行其他操作。Ajax的核心是异步方式,而同步方式只有在极其特殊的情况下才会被用到。

Ajax的基本流程可以概括为:在页面上由JavaScript脚本设置好服务器端的URL、必要的查询参数和回调函数之后,向服务器发出请求,服务器在处理请求之后将处理结果返回给页面,触发事先绑定的回调函数。这样,页面脚本如果想要改变一个区域的内容,只需要通过Ajax向服务器请求与该区域有关的少量数据,在回调函数中将该区域的内容替换即可,不需要刷新整个页面。

jQuery是一个快速、简洁的JavaScript框架,它简化了JavaScript开发。jQuery中的j代表JavaScript,Query是“查询”的意思。也就是说,这个库的意图是基于JavaScript的查询,查询的目标是DOM(文档对象模型)结构中的Node(节点)。网页上的所有内容都是节点,包括文档节点、元素节点、文本节点、注释节点、属性节点等。而jQuery查询主要针对的是元素节点,如段落(<p>)、表格(<table>)等,同时jQuery还可以用attr方法方便地对元素节点的属性进行读取/设置。

此外, jQuery还提供了浏览器兼容、样式读写、事件绑定与执行、动画等特性,后来又加入了Ajax、Promise等,再加上方便的插件编写机制,对整个JavaScript生态圈产生了重大影响,可以说是JavaScript历史上影响力最大的一个库。

1.3 集成开发环境介绍

通常情况下,开发应用程序都要使用IDE(Integrated Development Environment, 集成开发环境),IDE能提高应用程序的开发效率。本节将介绍Java Web应用开发常用的IDE和Web服务器。

1.3.1 常用的Java IDE

IDE是一种用于辅助开发人员开发应用程序的应用软件,一般包括代码编辑器、编译器、调试器和图形用户界面。有的还包括版本控制系统、性能分析器等更多辅助工具,因此IDE具有编写、编译、调试等多种功能。正是基于这些功能,使用IDE开发应用程序才能大大减轻程序员的工作,缩短项目的开发周期,从而提高应用程序的开发效率。

IDE的种类非常多,有的IDE能同时支持多种应用程序的开发,例如, Eclipse能用于Java、PHP、C++等多种语言开发;有的IDE只针对特定语言的开发,如JBuilder只能用于Java开发, Zend Studio只能用于PHP开发。本节将介绍Java应用开发常用的IDE。

1. IntelliJ IDEA

IntelliJ IDEA是用于Java语言的集成开发环境，旨在最大限度地提高开发人员的生产力。通过提供巧妙的代码完成静态代码分析和重构，它可以执行例行和重复的任务。IntelliJ IDEA是跨平台的IDE，可在Windows、macOS和Linux上提供一致的体验。IntelliJ IDEA在业界被公认为最好的Java开发工具之一，尤其在智能代码助手、代码自动提示、重构、J2EE支持、各类版本工具(git、svn、GitHub等)、JUnit、CVS整合、代码分析、创新的GUI设计等方面，具有超强的功能。IntelliJ IDEA的开发界面如图1-2所示。

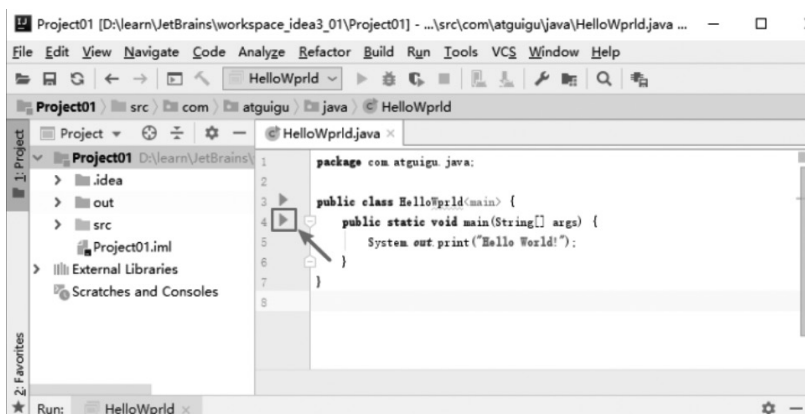


图1-2 IntelliJ IDEA的开发界面

该软件的官方下载地址为<http://www.jetbrains.com/idea/download/index.html>。它的旗舰版本还支持HTML、CSS、PHP、MySQL、Python等。免费版只支持Java等少数语言。

2. Eclipse

Eclipse是著名的跨平台集成开发环境(IDE)。最初主要用于Java语言开发，通过安装不同的插件，Eclipse可以支持不同的计算机语言，比如C++和Python等开发工具。Eclipse的本身只是一个框架平台，但是众多插件的支持使得Eclipse拥有其他功能相对固定的IDE软件很难具有的灵活性。许多软件开发商以Eclipse为框架开发自己的IDE。Eclipse最早是由IBM公司开发的，后来IBM公司将Eclipse作为一个开发源代码的项目，献给了开源组织Eclipse.org，但仍由IBM公司的子公司OTI(主要从事Eclipse开发)继续Eclipse的开发。从2018年9月开始，Eclipse每3个月发布一个版本，并且版本代号不再延续天文星体名称，直接使用年份跟月份。其官方下载地址为<http://www.eclipse.org/downloads/>，开发界面如图1-3所示。

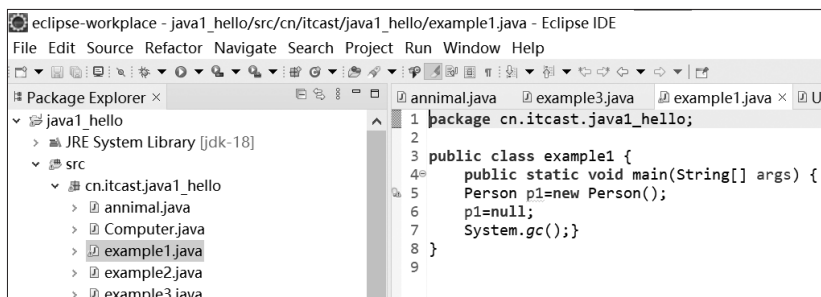
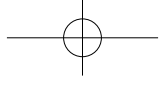


图1-3 Eclipse的开发界面



MyEclipse是一个十分优秀的用于开发Java EE的Eclipse插件集合，MyEclipse的功能非常强大，支持也十分广泛，尤其是对各种开源产品的支持十分不错。MyEclipse可以支持Java Servlet、Ajax、JSP、JSF、Struts、Spring、Hibernate、EJB 3、JDBC数据库链接工具等多项功能。可以说，MyEclipse是几乎囊括了目前所有主流开源产品的专属Eclipse开发工具。

3. Lightly

Lightly作为功能强大的集成开发工具，同时兼备文本编辑器的轻量以及集成开发工具的各项功能。用户只需在浏览器中打开 Lightly，即可在线体验完整的集成开发工具。Lightly集成开发工具目前已上线的功能包括：语法高亮、智能提示、自动补全，提高编程效率，免去死记硬背、重复打字等麻烦；多语言适配，支持Python、Java、C、C++、HTML/CSS、JavaScript、PHP或Go等编程语言；第三方库支持，一键安装线上包及本地包。

1.3.2 Web服务器汇总

Web服务器一般是指网站服务器，可以向浏览器等Web客户端提供文档。Web服务器不仅能够存储信息，还能在用户通过Web浏览器提供的信息的基础上运行脚本和程序。Web服务器不仅可以放置网站文件，让全世界网友浏览，还可以放置数据文件，让全世界网友下载。开发Java Web应用所采用的服务器主要是与JSP/Servlet兼容的Web服务器，常用的服务器有如下几种。

- **Apache服务器：**世界使用排名第一的Web服务器软件。它可以运行在几乎所有广泛使用的计算机平台上，由于其跨平台和安全性被广泛使用，因此是最流行的Web服务器端软件之一。Apache的特点是简单、速度快、性能稳定，对静态页面的处理非常高效，并可作为代理服务器来使用。
- **Tomcat服务器：**由Apache基金组织提供的一种Web服务器，提供对JSP和Servlet的支持，通过安装的插件，同样可以提供对PHP语言的支持。但是Tomcat只是一个轻量级的Java Web容器，像EJB这样的服务在Tomcat中是不能运行的。它是初学者学习开发JSP应用的首选。
- **Nginx服务器：**一款高性能的HTTP和反向代理服务器，也是电子邮件(IMAP/POP3)代理服务器，并在一个BSD-like协议下发行。Nginx服务器的特点是占用内存少，并发能力强。事实上，Nginx的并发能力确实在同类型的网页服务器中表现较好，使用Nginx的网站有百度、京东、新浪、网易、腾讯、淘宝等。在连接高并发的情况下，Nginx是Apache服务器不错的替代品，能够支持高达5万个并发连接数的响应。
- **Jetty服务器：**目前比较被看好的一款Servlet服务器。该服务器的架构比较简单，但在可扩展性方面表现得非常灵活。它有一个基本数据模型，这个数据模型就是Handler，所有可以被扩展的组件都可以作为Handler添加到Server中，Jetty就是帮助用户管理这些Handler数据模型，以便于更迅捷的开发。
- **Lighttpd服务器：**一个德国人领导开发的开源Web服务器软件，其根本的目的是提供一个专门针对高性能网站，安全、快速、兼容性好并且灵活的Web Server环

境。它具有非常低的内存开销、CPU占用率低、效能好等特点。Lighttpd是众多OpenSource轻量级的Web Server中较为优秀的一个。它支持FastCGI、CGI、Auth、输出压缩、URL重写、Alias等重要功能。Apache之所以流行，很大程度上是因为功能丰富，在Lighttpd上很多功能都有相应的实现，这点对于Apache的用户是非常重要的。

- Resin服务器：Caucho公司的产品，是一个非常流行的支持Servlet和JSP的服务器，速度非常快。Resin本身包含了一个支持HTML的Web服务器，这使它不仅可以显示动态内容，而且显示静态内容的能力也毫不逊色，因此许多网站都使用Resin服务器来构建。
- JBoss服务器：一个开源的重量级Java Web服务器，是一个遵从Java EE规范、开放源代码的纯Java EJB服务器，对J2EE有很好的支持。JBoss采用JML API实现软件模块的集成与管理，其核心服务又是提供EJB服务器，不包含Servlet和JSP的Web容器，不过它可以和Tomcat完美结合。
- WebLogic服务器：用于开发、集成、部署和管理大型分布式Web应用、网络应用和数据库应用的Java应用服务器。它将Java的动态功能和Java Enterprise标准的安全性引入大型网络应用的开发、集成、部署和管理之中。WebLogic支持企业级的、多层次的和完全分布式的Web应用，并且服务器的配置简单、界面友好。对于那些正在寻求能够提供Java平台所拥有的一切应用服务器的用户来说，WebLogic是十分理想的选择。

1.4 本章小结

本章对Java Web应用开发的一些基本知识做了简要介绍。首先介绍了Web和Web应用的概念，以及Web应用的工作原理；接下来讲解了使用Java开发Web应用的优势和核心技术；最后对使用Java开发Web应用的常用集成开发环境进行了对比和简介。通过本章的学习，读者应对Web应用开发有个基本的理解，掌握Web应用的工作原理，熟悉常用的Java Web应用的核心技术，了解常用的Java IDE。

1.5 思考和练习

1. 什么是Web应用？
2. 简述Web应用的工作原理。
3. 常用的客户端开发技术有哪些？
4. 简述Java语言的特点。
5. Java Web应用的核心技术有哪些？
6. Web服务器的用途是什么？说出几个常见的Java Web服务器。