第1单元　项目导引

一、单元概述

本单元引入贯穿教材的 Web 应用系统——网络点餐系统，从项目的构思、设计、实施和运行四个方面对项目进行全面的剖析和介绍，以此引出基于 Java 的 Web 开发技术。

二、所支撑的职业技能

通过本单元的学习，全面了解网络点餐系统，理解基于 Java 的 Web 开发技术，重点掌握 JSP 的基本概念和运行原理。

三、单元重点与难点

重点：

(1)基于 Java 的 Web 开发技术。

(2)JSP 的运行原理。

难点：

JSP 的运行原理。

重点及难点学习指导建议：

先了解基于 Java 的 Web 开发技术包含哪些，然后对 JSP 的基本概念和运行原理进行识记。

四、知识单元正文

随着互联网的飞速发展，人们足不出户就可以完成很多事情：购物、订票、求医、交友、缴费等等，极大地方便了人们的生活，甚至人们坐在家中通过网络就可以实现点餐、付款、等待送餐等一系列服务。

教材以 Web 应用系统——网络点餐系统——为基石，全面讲解基于 Java 的 Web 开发技术，并运用所介绍的技术完成系统的开发。

1.1　项目构思

1.1.1　系统特性概述

本系统的开发就是让更多的人通过互联网知道商户的存在，为商户带来更多的客户和

利润，争取所有通过搜索引擎进行有效搜索的用户都能快速便捷地找到商户，同时所有互联网用户都是系统信息的受体，均可视为潜在客户。

系统为普通用户提供了菜品预览功能，注册和登录功能；为登录用户提供了菜品搜索功能，点餐功能，点餐信息维护功能及个人信息维护功能；为管理员提供了菜品信息，菜品分类信息，用户信息和点餐信息的管理功能。

1.1.2 用户特点

对于销售类型的系统，用户面非常广，学历高低也不尽相同，为此系统开发的时候应充分考虑因受体用户不同而造成使用困难的这一情况，在开发初期就针对不同用户的特点进行优化分级。系统的最终用户分为管理员用户和普通用户两类：其中管理员用户应具有一定的计算机专业知识；普通用户不需要专业的计算机知识，只需要对互联网有简单的了解就能运用此系统进行购物。同时系统是对外服务类型的，访问量根据市场的具体情况不同而变化，预计使用频度大约 100 人次/天。

1.1.3 系统运行环境

(1)主机类型：虚拟主机或独享主机；

(2)网络类型：城域网或广域网两种；

(3)存储器容量：20GB；

(4)操作系统：支持 Windows 7 及以上操作系统。

1.1.4 系统功能描述

(1)用户在未登录的状态下可以浏览点餐系统中的热点菜品(按点餐率升序排列)，今日特价菜品和厨师推荐菜品。

(2)用户可以登录点餐系统，如果是新用户则需要先注册再登录。

(3)登录后的用户可以按照菜品分类浏览系统中的所有菜品，也可以按照菜品名称搜索菜品，并可以批量挑选喜爱的菜品加入点餐车。

(4)登录后的用户可以查看自己的点餐车，浏览所点的菜品，并可以批量删除。

(5)登录后的用户可以修改自己的资料，包括密码、电话和地址信息。

(6)登录后的用户可以退出点餐系统。

(7)系统提供一个管理员用户，管理员登录后可以行使管理功能。

(8)登录后的管理员可以对用户进行管理，可以浏览所有用户的信息，也可以按照用户名查找用户信息，并对用户进行删除。

(9)登录后的管理员可以对菜品分类进行管理，可以浏览所有菜品分类，并对菜品分类进行增加、删除和修改。

(10)登录后的管理员可以对菜品信息进行管理，可以浏览所有菜品的信息，并对菜品进行增加、删除和修改。

(11)登录后的管理员可以查看所有用户的点餐情况。

(12)未登录的用户禁止访问登录用户的功能，非法访问时，系统会给出登录提示，登录

后方可继续操作。

1.1.5　非功能需求

除了最基本的增加、删除、修改和查询四类操作外，针对不同范围的使用，系统应当能够承受不同数据量的访问连接，所以对计算机性能的要求，最低也应该是在 Windows 7 以上。系统的响应时间应该在用户可以接受的范围之内，一般在操作之后的四五秒内，按照要求输出用户所需的结果。

系统能够区分不同的用户，保护用户的隐私和区分不同的使用权限，对数据进行安全保护。同时也会对数据库进行备份，限制操作人员的权限，不同权限人员，数据对其开放级别不同，保证数据的安全性。

1.2　项目设计

1.2.1　体系结构设计

基于 Web 的应用系统开发可以采用两种体系结构，一种是 C/S 架构，另外一种是 B/S 架构。

C/S(Client/Server)，即客户机和服务器结构。通过它可以充分利用两端硬件环境的优势，将任务合理分配到 Client 端和 Server 端来实现，降低了系统的通信开销。应用此种结构，需要为客户机和服务器分别编写不同的软件。

C/S 架构的优势在于应用服务器运行数据负荷较轻，数据的存储管理功能较为透明。它的劣势在于高昂的维护成本且投资大，传统的 C/S 结构的软件需要针对不同的操作系统开发不同版本的软件，由于产品的更新换代十分快，代价高和低效率已经不适应工作需要。在 Java 这样的跨平台语言出现之后，B/S 架构更是猛烈冲击 C/S，并对其形成威胁和挑战。

B/S(Browser/Server)，即浏览器和服务器结构。它是随着 Internet 技术的兴起，对 C/S 结构的一种变化或者改进的结构。在这种结构下，用户工作界面是通过 WWW 浏览器来实现，极少部分业务逻辑在前端(Browser)实现，主要的业务逻辑在服务器端(Server)实现，这样就大大简化了客户端电脑载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本。

总的说来，B/S 结构同传统的 C/S 结构相比，其优点在于：

(1)B/S 是一种瘦客户机模式，客户端软件仅须安装浏览器，且对客户端硬件配置要求较低。

(2)标准统一，维护相对简单。HTML 是 Web 信息的组织方式，所有 Web 服务器和浏览器都遵循这个国际标准，使用 B/S 方式，可以将开发人员集中在服务器端，只须开发和维护服务器端应用程序，而服务器上的应用程序可通过网络浏览器在客户端上执行，从而充分发挥开发人员的群体优势，应用软件的维护也相对简单。

(3)无须开发客户端软件。浏览器软件可以从 Internet 上免费得到，对于安装了

Windows 操作系统的客户机来说，只要使用内置的网络协议和浏览器即可。

(4)跨平台支持。由于采用统一的通信协议，并且浏览器及服务器软件可以支持多平台，所以可以方便地在企业异构平台运行。

(5)浏览器界面易学易用，使用者无须太多技术知识。

综上因素考虑，网络点餐系统的开发选择 B/S 架构。

1.2.2 系统模块设计

依据项目构思的描述，将网络点餐系统划分为 3 个模块，分别是：公共模块，用户模块和管理员模块，如图 1-1 所示。

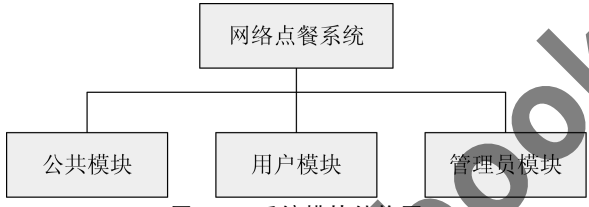

图 1-1 系统模块结构图

公共模块分为首页、登录、注册、结果页和退出 5 个部分，如图 1-2 所示。

图 1-2 公共模块细化图

用户模块分为用户首页(即点餐页面)、点餐车页面和修改用户资料 3 个部分，如图 1-3 所示。

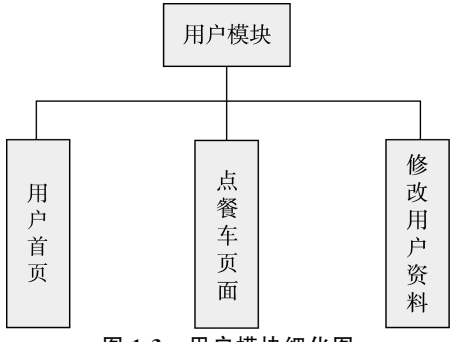

图 1-3 用户模块细化图

管理员模块分为管理员首页、用户管理、菜品分类管理、菜品管理、查看用户点餐情况 5 个部分，其中用户管理包括用户浏览和用户删除；菜品分类管理包括菜品分类浏览、菜品分类添加、菜品分类修改和菜品分类删除；菜品管理包括菜品浏览、菜品添加、菜品修改和菜品删除，如图 1-4 所示。

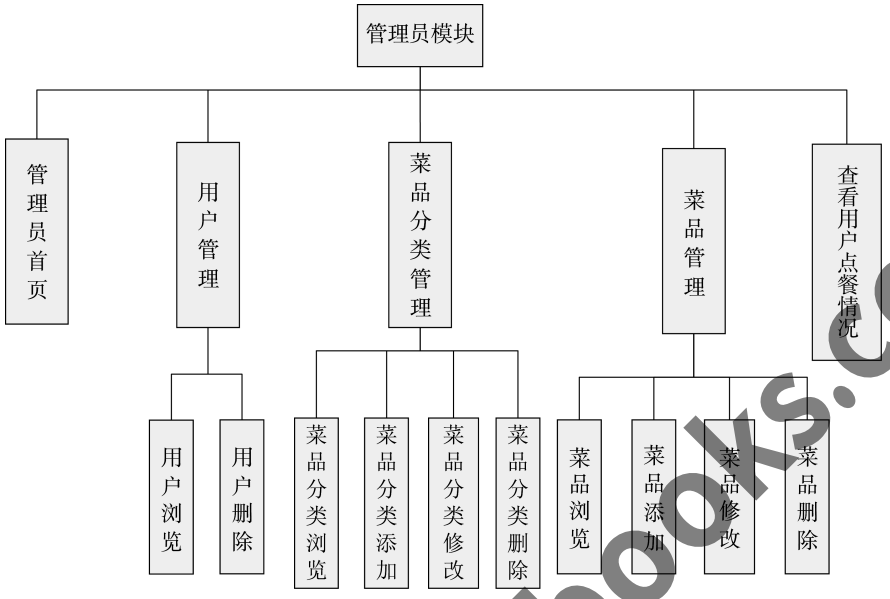

图 1-4　管理员模块细化图

1.2.3　界面原型

网络点餐系统作为大众使用的应用系统，其界面设计应遵循简洁美观、方便易用的基本原则。系统从首页开始，通过点击不同的超级链接或按钮转入其他页面，下面介绍一些主要的功能页面图及相关说明。

网络点餐系统的首页显示热点菜品、今日特价和厨师推荐菜品，每种菜品均提供超级链接以访问菜品详细信息，首页中还提供了登录和注册点餐系统的超级链接，如图 1-5 所示。

图 1-5　网络点餐系统首页

在图 1-5 中点击菜品名称的超级链接将进入菜品详细信息显示页面，如图 1-6 所示。

图 1-6　菜品详细信息页面

在图 1-5 中点击“登录”超级链接将弹出登录窗口，如图 1-7 所示。

图 1-7　登录窗口

在图 1-5 中点击“注册”超级链接将弹出注册窗口，如图 1-8 所示。

图 1-8　注册窗口

在图 1-7 中填写登录信息，点击“确定”按钮后，如果登录失败，则弹出如图 1-9 所示的提

示窗口；登录成功后，如果是普通用户身份，将进入用户首页面，如图 1-10 所示；如果是管理员身份，将进入管理员首页面，如图 1-11 所示。登录用户点击图 1-10 中的“网络点餐系统”超级链接将跳转到如图 1-12 所示的网络点餐系统首页，但此时导航栏中含有操作链接，与图 1-5 略有不同。

图 1-9　登录失败页面

图 1-10　用户首页面

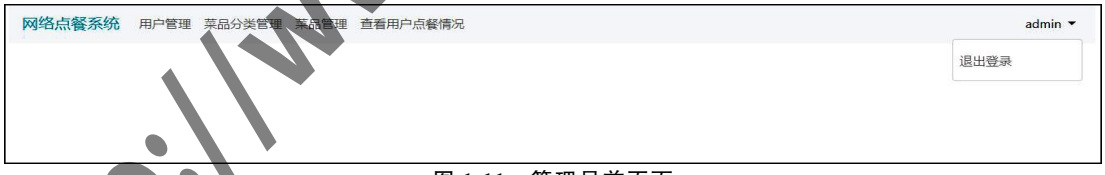

图 1-11　管理员首页面

图 1-12　登录用户返回点餐系统首页

在图 1-10 中选择菜品后，点击“将菜品添加到点餐车”按钮将进入如图 1-13 所示的结果页面，在结果页面中点击“确定”按钮将进入我的点餐页面，如图 1-14 所示。

图 1-13　加入点餐车结果页面

图 1-14　用户查看点餐车页面

在图 1-14 我的点餐页面，可以选择菜品从点餐车删除，这里不再截图赘述。

在图 1-11 的管理员首页面中点击导航条中的“菜品管理”超级链接将进入菜品管理页面，如图 1-15 所示。

图 1-15　菜品管理页面

在图 1-15 中，可以按菜名和分类查询菜品信息，还可以对菜品信息进行添加、修改和删除操作。在图 1-15 中，点击“添加菜品”按钮，页面如图 1-16 所示；点击“修改”按钮，页面如图 1-17 所示；点击“删除”按钮，弹出删除确认对话框，如图 1-18 所示。

图 1-16　添加菜品页面

图 1-17　修改菜品页面

图 1-18　删除确认对话框

管理员的其他管理功能界面和菜品管理界面相似，这里不再截图赘述。

在图 1-11 的管理员首页面中点击导航条中的“查看用户点餐情况”超级链接将进入用户点餐列表页面，如图 1-19 所示。

图 1-19　用户点餐列表页面

1.2.4　数据库设计

从项目构思出发，结合界面原型，从中抽取关键要素，分析出本项目主要具备 3 个实体，分别是系统用户、菜品分类和菜品。分析这 3 个实体之间的关系，设计出如图 1-20 所示的 E-R 模型。

对 E-R 图分析转换后，提取出 4 张数据库表，分别是：用户表、菜品分类表、菜品表和点餐表，表的具体设计见表 1-1～表 1-4。

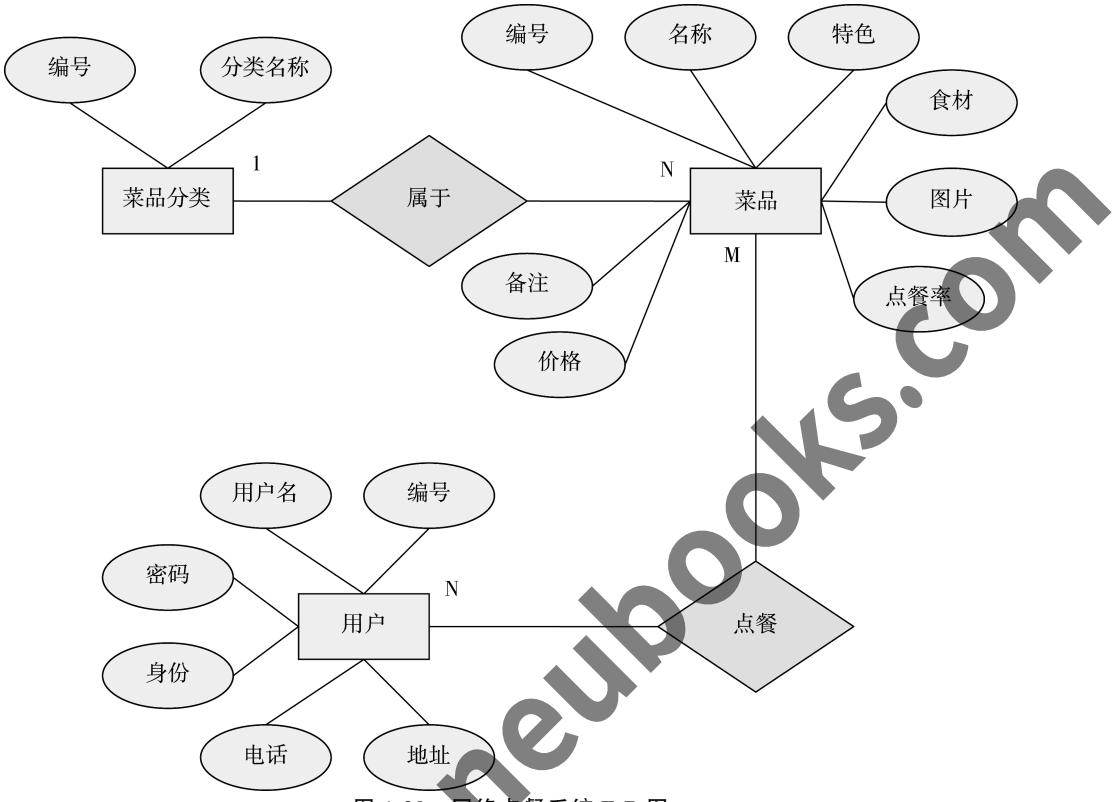

图 1-20　网络点餐系统 E-R 图

表 1-1　　user(用户表)

字段名	数据类型	主键	可为空	是否外键	其他约束	说明
id	int	是	否	否	无	用户编号
username	varchar(20)	否	否	否	唯一键	用户名
password	varchar(20)	否	否	否	无	密码
ident	char(1)	否	否	否	无	用户身份 (0:普通用户,1:管理员)
telephone	varchar(20)	否	否	否	无	电话
address	varchar(50)	否	否	否	无	地址

表 1-2　　foodtype(菜品分类表)

字段名	数据类型	主键	可为空	是否外键	其他约束	说明
id	int	是	否	否	无	分类编号
typename	varchar(20)	否	否	否	唯一键	分类名称

表 1-3 food(菜品表)

字段名	数据类型	主键	可为空	是否外键	其他约束	含义
id	int	是	否	否	无	菜品编号
foodname	varchar(20)	否	否	否	唯一键	菜品名称
feature	varchar(100)	否	是	否	无	特色
material	varchar(100)	否	是	否	无	食材
price	int	否	否	否	无	价格
type	int	否	否	是	无	分类
picture	varchar(40)	否	是	否	无	图片
hits	int	否	否	否	无	点餐率
comment	int	否	否	否	无	备注(-1:正常菜,0:厨师推荐菜品,正整数:特价菜品价格)

表 1-4 diningcar(点餐表)

字段名	数据类型	主键	可为空	是否外键	其他约束	含义
id	int	是	否	否	无	点餐编号
userid	int	否	否	是	无	用户编号
foodid	int	否	否	是	无	菜品编号

1.3 项目实施

1.3.1 开发技术的选择

开发 Web 应用系统的技术有很多,目前流行的 Web 开发技术包括 PHP、ASP. NET 和 JSP 等等。

PHP(Hypertext Preprocessor)是一种嵌入 HTML 页面中的脚本语言。它大量地借用 C 语言和 Perl 语言的语法,并结合 PHP 自己的特性,使 Web 开发者能够快速地编写动态页面。PHP 是完全免费的开源产品,PHP 经常和 MySQL 搭配使用。PHP 易学易用,但在复杂的大型项目上的开发和维护都比较困难,因此更适合于编写比较小型的业余网站。

ASP. NET 是 Microsoft. NET 的一部分,它不仅仅是 Active Server Page(ASP)的下一个版本,它还提供了一个统一的 Web 开发模型,其中包括开发人员生成企业级 Web 应用程序所需的各种服务。ASP. NET 的语法在很大程度上与 ASP 兼容,同时它还提供一种新的编程模型和结构,可生成伸缩性和稳定性更好的应用程序,并提供更好的安全保护。ASP. NET 可以无缝地与 WYSIWYG HTML 编辑器和其他编程工具(包括 Microsoft Visual Studio. NET)一起工作,这使得 Web 开发变得更加方便。ASP. NET 运行于 IIS 之上,这是

个曾无数次遭受攻击的服务器，很多 IT 专业人士已经拒绝将他们的网络暴露于 IIS Web 服务器之下。

JSP(Java Server Pages)是由前 Sun 公司(已被 Oracle 收购)倡导、许多公司参与一起建立的一种动态网页技术标准。JSP 技术有点类似 ASP 技术，它是在传统的网页 HTML 文件中插入 Java 程序段和 JSP 标签，从而形成 JSP 文件。由于 JSP 是基于 Java 的，所以它也有 Java 语言的最大优点——平台无关性，也就是所谓的“一次编写，随处运行”。除了这个优点，JSP 的效率以及安全性也是相当惊人的。

在网络点餐系统的开发过程中，本教材选择了 JSP 技术，它是基于 Java 的 Web 开发技术的一部分，下面将对基于 Java 的 Web 开发技术做详细介绍。

1.3.2　基于 Java 的 Web 开发技术

基于 Java 的 Web 开发技术包含很多内容，这里重点概述 Servlet、JSP 和 JDBC 技术。

1. Servlet

Servlet 是使用 Java Servlet 应用程序设计接口及相关类和方法的 Java 程序，Servlet 扩展了服务器的功能以处理请求并生成响应。Servlet 是由服务器端调用和执行的 Java 类，是小型的、与平台无关的 Java 类，它被编译成结构中立的字节码，由基于 Java 的 Web 服务器动态加载和执行。Servlet 通过容器实现的 request 和 response 实例与网页客户交互。Servlet 的主要功能在于交互式地浏览和修改数据，生成动态 Web 内容。

在网络点餐系统的实现中，Servlet 主要用来接收用户的请求，根据用户的请求地址来决定调用哪个方法处理请求，并决定最终将哪个响应页面发送给用户。

Servlet 技术将在第 3 单元详细介绍。

2. JSP

JSP 是一种建立在 Servlet 规范提供的功能之上的动态网页技术，它是通过在 HTML 文件中嵌入 Java 代码和 JSP 标签来产生动态内容。

执行 JSP 程序首先需要一个 JSP 的运行环境，也就是 JSP 容器(也是 Servlet 容器)，比较常用的 JSP 容器有 Tomcat、Resin 和 Websphere 等。当用户第一次请求某个 JSP 文件时，容器首先检查 JSP 文件的语法是否正确，然后将 JSP 文件转换成 Servlet 源文件，并将 Servlet 源文件编译成字节码文件。接下来，容器加载转换后的 Servlet 类，实例化一个该类的对象处理客户端的请求，请求处理完成后，容器将 HTML 格式的响应信息发送给客户端，整个执行过程如图 1-21 所示。

JSP 具有以下特点：

(1)将内容的生成和显示分离

使用 JSP 技术，Web 页面开发人员可以使用 HTML 或者 XML 标签来设计和格式化最终页面，使用 JSP 标签或者 Java 代码来生成页面上的动态内容。在服务器端，JSP 容器解释 JSP 标签和 Java 代码，生成所请求的内容，并且将结果以 HTML(或者 XML)页面的形式发送回浏览器。这样既有助于保护代码，又保证了任何基于 HTML 的 Web 浏览器的完全可用性。

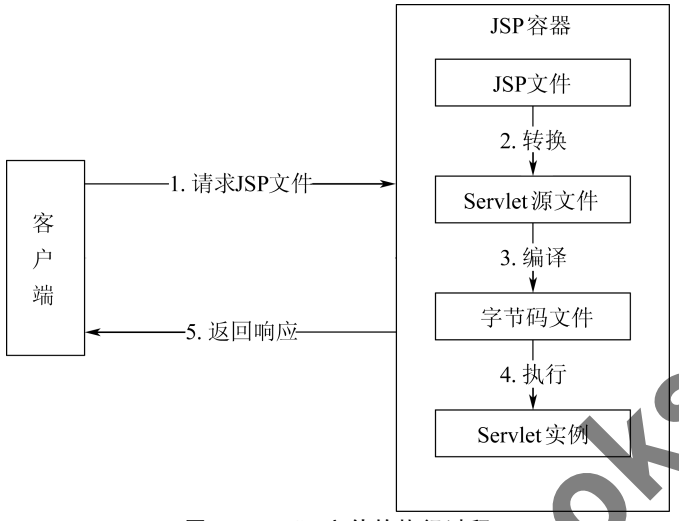

图 1-21　JSP 文件的执行过程

(2)使用可重用的组件

绝大多数 JSP 页面依赖于可重用的、跨平台的组件来执行应用程序所要求的更为复杂的功能。开发人员能够共享这些执行通用操作的组件,或者使得这些组件为更多的客户所使用。基于组件的方法加速了总体开发过程,并且使得各种组织可以在现有的技能和优化结果的基础上继续开发。

(3)采用标签简化页面开发

Web 网页开发人员不一定都是熟悉 Java 语言的程序员。JSP 技术能够将许多功能封装起来,成为一个自定义的标签,这些功能是完全根据 XML 的标准来制订的。标准的 JSP 标签能够访问和实例化 JavaBeans 组件,设置或者检索组件属性,下载 Applet,以及执行用其他方法更难于编码和耗时的功能。第三方开发人员和其他人员可以为常用功能创建自己的标签库。这使得 Web 页面开发人员能够使用标签来执行特定的功能而不需要掌握全部 Java 语言。

(4)完善的存储管理和安全性

由于 JSP 页面的内置脚本语言是 Java 语言,而且所有的 JSP 页面都要被转换成 Servlet,所以 JSP 页面就具有 Java 技术的所有优势,包括健壮的存储管理、安全性以及跨平台性。

(5)一次编写,各处运行

作为 Java 平台的一部分,JSP 技术拥有 Java 语言“一次编写,随处运行”的特点。这一点对企业用户尤其重要,当企业更换服务器平台时,并不影响之前所投入的成本、人力所开发的 JSP 应用程序。

在网络点餐系统的实现中,JSP 页面用来接收 Servlet 返回的数据信息,并生成给用户的响应页面。

JSP 技术将在第 4 单元详细介绍。

3. JDBC

JDBC(Java DataBase Connectivity)是 Sun 公司提供的一组用来按照统一方式访问数据

库的 API。它向程序员提供了独立于数据库的统一接口，可以使开发人员不必考虑所用的特定数据库便可编写应用程序，实现了 Java 与数据库的互联。与其他的数据库编程环境相比，JDBC 能够提供对数据库的跨平台存取，可以将数据从一个数据库移植到另一个数据库。

在网络点餐系统的实现中，利用 JDBC API 实现与数据库的连接并进行相应的数据库操作。

JDBC 技术将在第 6 单元详细介绍。

1.3.3　开发和运行环境的选择

基于 Java 的 Web 开发和运行环境有很多，本教材中选择了 JDK8、Tomcat 9、Eclipse4.12、MySQL8.0 和 Navicat12 作为开发、调试和运行程序的工具和平台。

1. JDK

Servlet、JSP、Tomcat 和 Eclipse 的运行离不开 JDK，所以基于 Java 的 Web 开发必须首先安装 JDK。

第 2 单元将详述 JDK8 的安装过程。

2. Tomcat

Tomcat 是一个免费的、开放源代码的 Web 应用服务器软件，同时 Tomcat 也是目前世界上用量最大的 Java 服务器软件，主要用来支持运行 Servlet 和 JSP。

Tomcat 是 Apache 软件基金会(Apache Software Foundation)的 Jakarta 项目中的一个核心项目，由 Apache、Sun(已被 Oracle 收购)和其他一些公司及个人共同开发而成。由于有了 Sun 的参与和支持，最新的 Servlet 和 JSP 规范总是能在 Tomcat 中得到体现，Tomcat9 支持最新的 Servlet3.1 和 JSP2.3 规范。因为 Tomcat 技术先进、性能稳定，而且免费，因而深受 Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的 Web 应用服务器。

Tomcat 很受广大程序员的喜欢，因为它运行时占用的系统资源小，扩展性好，支持负载平衡与邮件服务等开发应用系统常用的功能；而且它还在不断的改进和完善中，任何一个感兴趣的程序员都可以更改它或在其中加入新的功能。

第 2 单元将详述 Tomcat9 的安装和使用过程。

3. Eclipse

Eclipse 最初是由 IBM 公司开发的替代商业软件 Visual Age for Java 的下一代 IDE 开发环境，2001 年 11 月贡献给开源社区，现在它由非盈利软件供应商联盟 Eclipse 基金会(Eclipse Foundation)管理。

Eclipse 是著名的、跨平台的集成开发环境，它是一个开放源代码的、基于 Java 的可扩展开发平台。Eclipse 灵活、易扩展，可以免费下载使用，是目前最为常用的 Java Web 开发环境。

第 2 单元将详述 Eclipse4.12 的安装和使用过程。

4. MySQL

MySQL 是一个跨平台的开源关系型数据库管理系统，目前 MySQL 被广泛地应用在 Internet 上的中小型网站中。由于其体积小、速度快，尤其是开放源码这一特点，许多中小型网站都选择了 MySQL 作为网站数据库。

第 2 单元将详述 MySQL8.0 的安装和配置过程。

5. Navicat

由于 MySQL 数据库只提供了命令行管理工具，为了方便管理和使用，需要为其安装一个图形用户界面的管理工具。Navicat 是一个强大的 MySQL 数据库管理和开发工具。Navicat 为专业开发者提供了一套强大的工具，同时它对于新用户仍然是易学易用。使用 Navicat，用户可完全控制 MySQL 数据库和显示不同的管理资料，包括一个多功能的图形化管理用户和访问权限的管理工具，方便将数据从一个数据库移转到另一个数据库中，进行档案备份。Navicat 支持中文，有免费版本提供。

第 2 单元将详述 Navicat12 的安装和使用过程。

1.3.4 编码实现

网络点餐系统的编码实现将贯穿在全教材的各个单元中，这里不再统一概述。

1.4 项目运行

网络点餐系统的预期运行效果将与界面设计中的页面效果图相同，为了更适合实际应用的需要，读者可以在此系统的基础上添加订单生成模块、订单管理模块、信息统计模块、分页处理模块等作为扩展。

五、习题

(1)什么是 JSP？它的运行原理是什么？

(2)基于 Java 的 Web 开发技术主要有哪些？

(3)自选购物网站进行调研，比如：淘宝网、京东、唯品会、网易严选等，分析其核心用户功能，并推测管理端功能，撰写如 1.1 节项目构思中系统功能描述的文档。

第 2 单元　开发和运行环境的安装和使用

一、单元概述

本单元详细介绍 Java Web 开发环境及运行环境的安装、配置和使用。

二、所支撑的职业技能

通过本单元的学习，掌握 JDK、Tomcat、Eclipse、MySQL 和 Navicat 的安装、配置及使用，并能熟练运用开发和运行环境进行 Java Web 项目的开发、部署及访问。

三、单元重点与难点

重点：

运用开发和运行环境进行 Java Web 项目的开发、部署及访问。

难点：

Java Web 开发环境的配置和使用。

重点及难点学习指导建议：

先了解开发环境的安装顺序，并依次完成安装，然后将 JDK 和 Tomcat 正确地关联到 Eclipse 中，最后通过 Java Web 项目的开发、部署和访问掌握开发和运行环境的运用。

四、知识单元正文

2.1　JDK 的安装

JDK 的安装文件 jdk-8u271-windows-x64.exe 可以从 Oracle 公司的网站免费下载。JDK8 的安装步骤如下：

(1)执行安装文件 jdk-8u271-windows-x64.exe，打开如下窗口，如图 2-1 所示。

(2)在图 2-1 中点击【下一步】按钮后，选择安装路径及要安装的程序功能。默认安装路径为“C:\Program Files\Java\jdk1.8.0_271”，如需改变安装路径，则点击【更改】按钮，如图 2-2 所示。

(3)在图 2-2 中点击【下一步】按钮后，正式开始执行 JDK 的安装，如图 2-3 所示。

图 2-1 JDK 安装图示 1

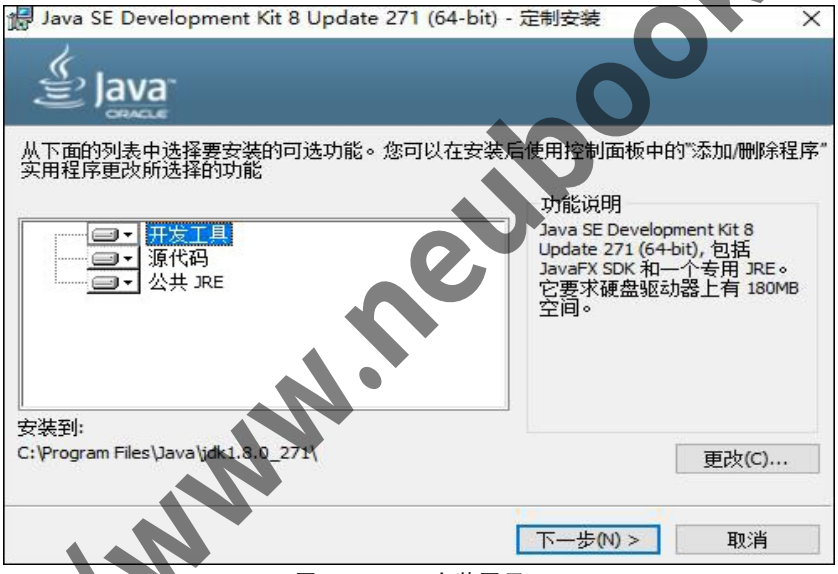

图 2-2 JDK 安装图示 2

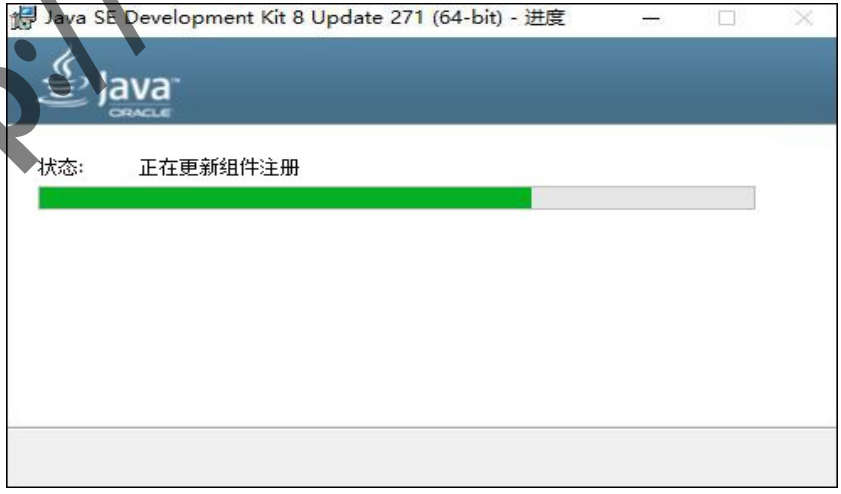

图 2-3 JDK 安装图示 3

(4)选择 JRE 的安装,默认安装路径为“C:\Program Files\Java\jre1.8.0_271”,如需改变安装路径,则点击【更改】按钮,如图 2-4 所示。

图 2-4　JDK 安装图示 4

(5)在图 2-4 中点击【下一步】按钮后,正式开始执行 JRE 的安装,如图 2-5 所示。

图 2-5　JDK 安装图示 5

(6)安装成功后,出现“安装完成”的提示,点击【关闭】按钮结束安装,如图 2-6 所示。

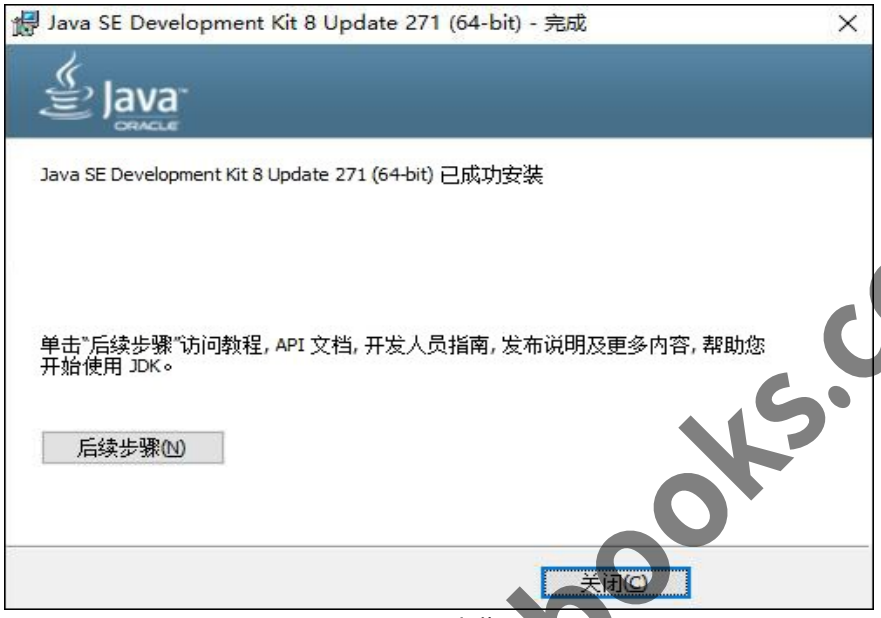

图 2-6 JDK 安装图示 6

2.2 Tomcat 的安装和使用

2.2.1 Tomcat 的安装

Tomcat 的安装文件 apache-tomcat-9.0.40-windows-x64.zip 可以从 APACHE 公司的官方网站上免费下载。

apache-tomcat-9.0.40 为绿色版本，只需要将 apache-tomcat-9.0.40.zip 解压到任意指定的目录下即可。解压完毕后，为使得 Tomcat 启动时可以找到已经安装的 JDK，需要在【我的电脑】→【属性】→【高级系统设置】→【环境变量】→【系统变量】→【新建】，设置变量名为 JAVA_HOME，其值设定为已经安装的 JDK 的主目录（如果 JDK 采用的是默认安装目录，则其值是 2.1 节中安装的 JDK 路径 C:\Program Files\Java\jdk1.8.0_271），如图 2-7 所示。

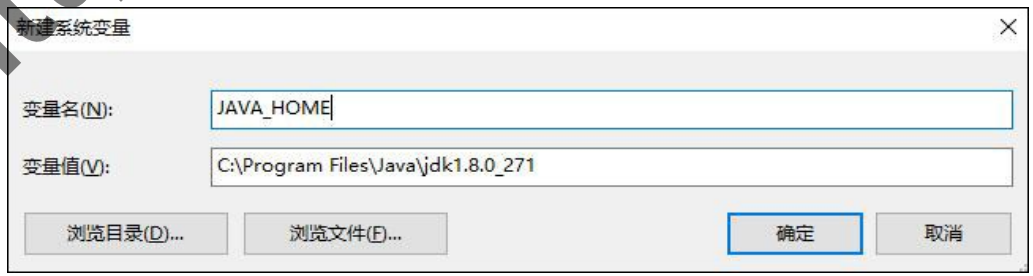

图 2-7 设置系统变量 JAVA_HOME

启动和关闭 Tomcat 服务器的文件分别为 Tomcat 主目录下的 bin 目录下的 startup.

bat 和 shutdown. bat。

执行 startup. bat，启动 Tomcat，正常的启动界面如图 2-8 所示。Tomcat 正常启动后，打开浏览器，在地址栏里输入 URL：http://localhost:8080，将看到如图 2-9 所示的页面。

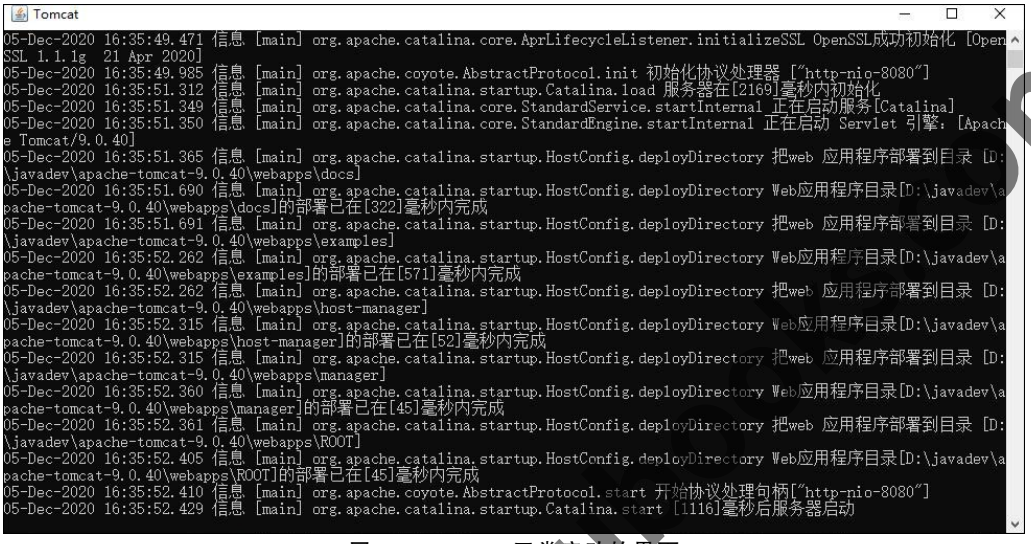

图 2-8　Tomcat 正常启动的界面

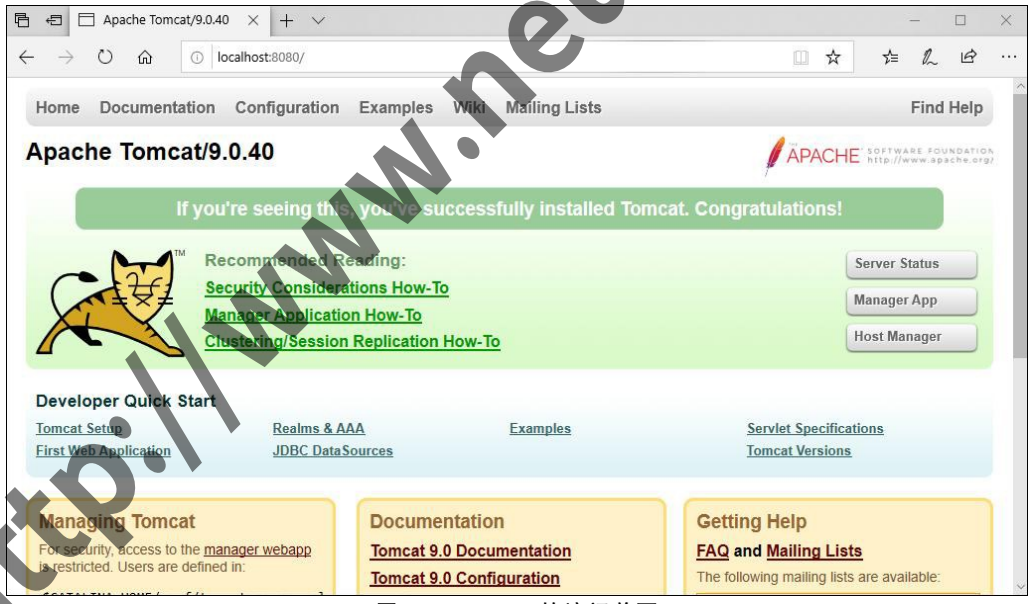

图 2-9　Tomcat 的访问首页

Tomcat 提供服务的默认端口号是 8080，有时它会与其他应用程序的端口号发生冲突，这种情况下 Tomcat 将无法正常启动，启动界面如图 2-10 所示。

这时，为保证 Tomcat 的正常启动，需要修改 Tomcat 的端口号为任意未被占用的端口号。修改 Tomcat 端口号的位置在 Tomcat 主目录下的 conf 目录下的 server. xml 文件中，如图 2-11 所示。

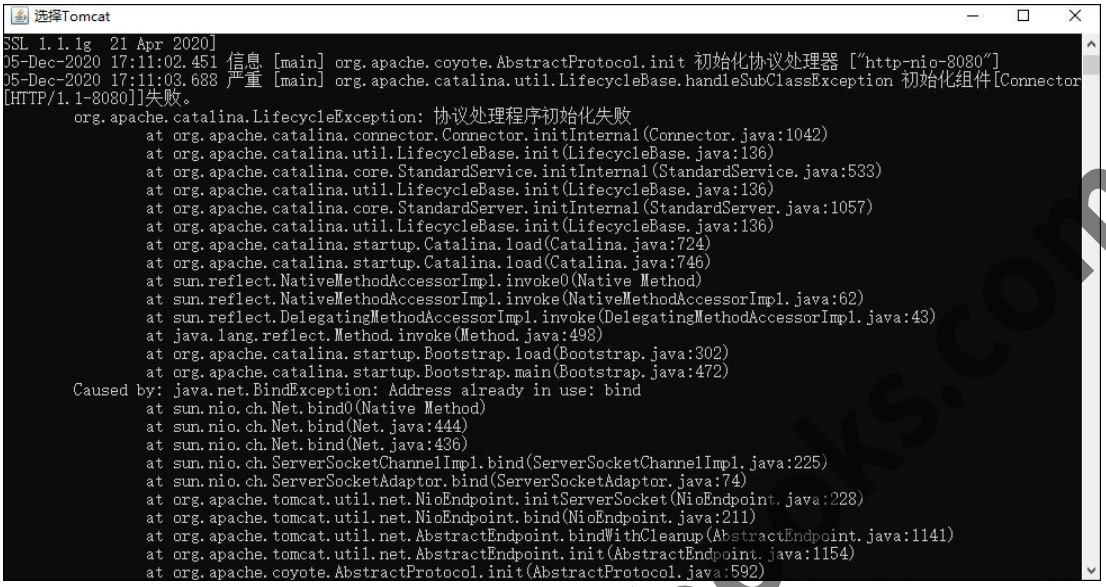

图 2-10　Tomcat 启动异常界面

```
62  ····<!-- A "Connector" represents an endpoint by which requests are received
63  ·····and responses are returned. Documentation at :
64  ·····Java HTTP Connector: /docs/config/http.html
65  ·····Java AJP  Connector: /docs/config/ajp.html
66  ·····APR (HTTP/AJP) Connector: docs/apr.html
67  ·····Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
68  ····-->
69  ····<Connector port="8080" protocol="HTTP/1.1"
70  ·····connectionTimeout="20000"
71  ·····redirectPort="8443" />
72  ····<!-- A "Connector" using the shared thread pool-->
73  ····<!--
74  ····<Connector executor="tomcatThreadPool"
75  ·····port="8080" protocol="HTTP/1.1"
76  ·····connectionTimeout="20000"
77  ·····redirectPort="8443" />
78  ····-->
```

图 2-11　在 server.xml 中修改端口号的位置

2.2.2　Tomcat 的目录结构

Tomcat 的主目录下有若干目录，表 2-1 描述了这些目录的用途。

表 2-1　Tomcat 9 的目录结构

目录	描述
\bin	放置启动和关闭 Tomcat 的可执行文件和批处理文件
\lib	放置 Tomcat 运行需要加载的 jar 包
\conf	放置 Tomcat 主要的配置文件
\logs	放置 Tomcat 的日志文件
\temp	放置 Tomcat 运行时产生的临时文件
\webapps	放置 Web 应用的目录
\work	放置 JSP 页面转换成对应的 Servlet 的目录

2.2.3　在 Tomcat 下开发 Web 应用

在 Tomcat 下开发 Web 应用，首先需要在 webapps 目录下建立一个新目录。webapps 目录是 Tomcat 存放 Web 应用的地方，它下面有若干个子目录。一般来说，每个目录对应一个 Web 应用，这个目录存放关于这个 Web 应用的所有 JSP 文件、Servlet 及 JavaBeans 等。为了能有效地访问 Web 应用，需要在 Web 应用的目录下建立 WEB-INF 目录，并在其下建立 web.xml 文件。

在 webapps 目录下建立 firstWebApp 目录，在此目录下用记事本新建一个 index.html 文件，用记事本打开该文件后键入“<h1>我的第一个 Web 应用</h1>”，保存后关闭。在 firstWebApp 目录下建立 WEB-INF 目录，并建立如下内容的 web.xml 文件：

```
<? xml version="1.0" encoding="UTF-8"? >
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                      http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
  version="4.0"
  metadata-complete="true">
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>
</web-app>
```

如上的操作创建了一个 Web 应用 firstWebApp，启动 Tomcat 后，打开浏览器，在地址栏里输入 URL：http://localhost:8080/firstWebApp，访问这个 Web 应用，将看到如图 2-12 所示的页面。

图 2-12　firstWebApp 的访问首页

如果将 webapps 目录下的 ROOT 目录中的 index.jsp、所有图片、样式表文件 tomcat.css 等全部拷贝到 firstWebApp 文件夹中，此时在浏览器地址栏中键入 URL：http://localhost:8080/firstWebApp/index.jsp，将会打开如图 2-13 所示页面。

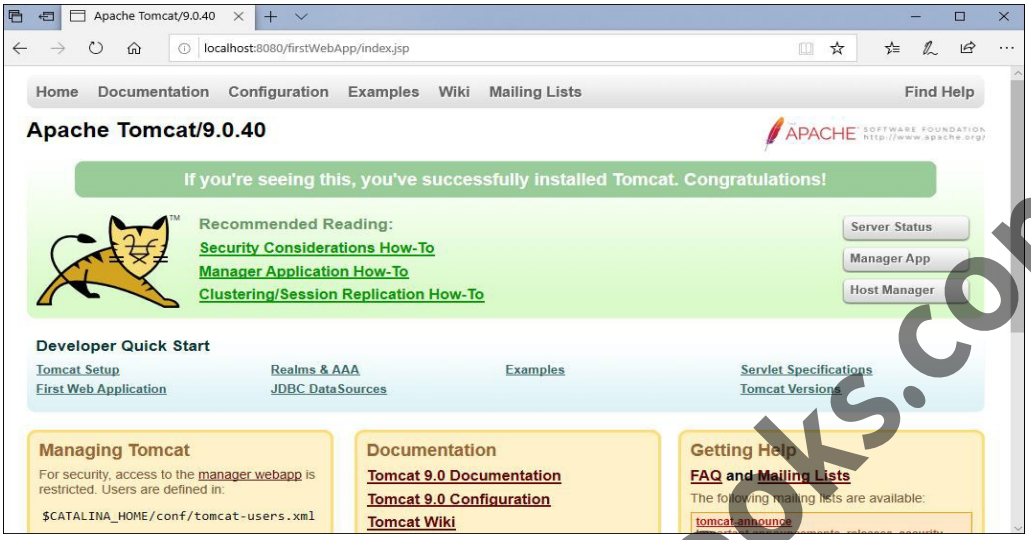

图 2-13　firstWebApp 的访问首页

2.3　Eclipse 的安装和使用

2.3.1　Eclipse 的安装与启动

Eclipse 的安装文件 eclipse-inst-win64.exe 可以从 Eclipse 官网上免费下载。安装与启动步骤如下：

(1)执行安装文件 eclipse-inst-win64.exe，打开如下窗口，如图 2-14 所示。

图 2-14　Eclipse 安装界面 1

(2)点击“Eclipse IDE for Java EE Developers”后,打开如下窗口,选择安装路径及开始菜单桌面快捷方式,如图 2-15 所示。

图 2-15　Eclipse 安装界面 2

(3)在图 2-15 中点击【INSTALL】按钮后,打开如下窗口,确认用户许可,如图 2-16 所示。

图 2-16　Eclipse 安装界面 3

(4)在图 2-16 中点击【Accept Now】按钮后,进入安装页面,如图 2-17 所示。

图 2-17 Eclipse 安装界面 4

(5)安装完毕后出现如图 2-18 图所示界面,点击【LAUNCH】按钮后即可打开 Eclipse。

图 2-18 Eclipse 安装成功界面

(6)点击图 2-18 中的【LAUNCH】按钮后,打开如图 2-19 所示界面选择工作区,即所开发的应用存放的位置,点击【Browse…】按钮可更改默认工作区。

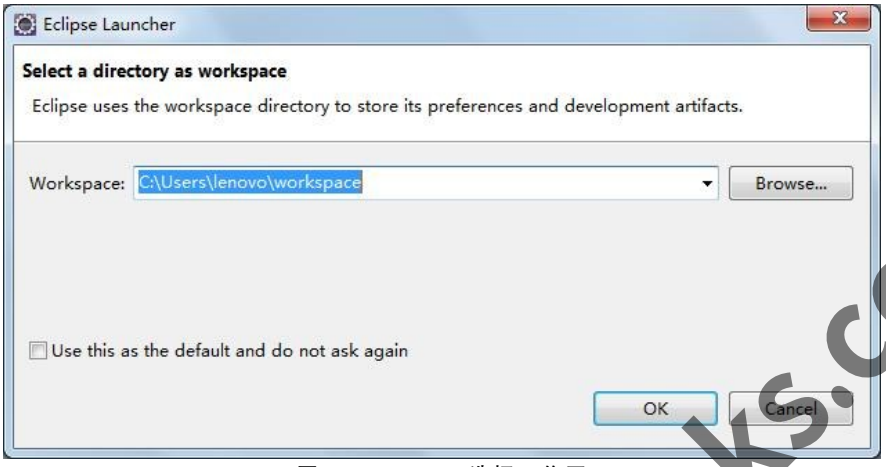

图 2-19　Eclipse 选择工作区

(7)在图 2-19 中点击【OK】按钮后,打开如图 2-20 所示的启动界面,成功启动后出现欢迎页面,如图 2-21 所示。

图 2-20　Eclipse 启动界面

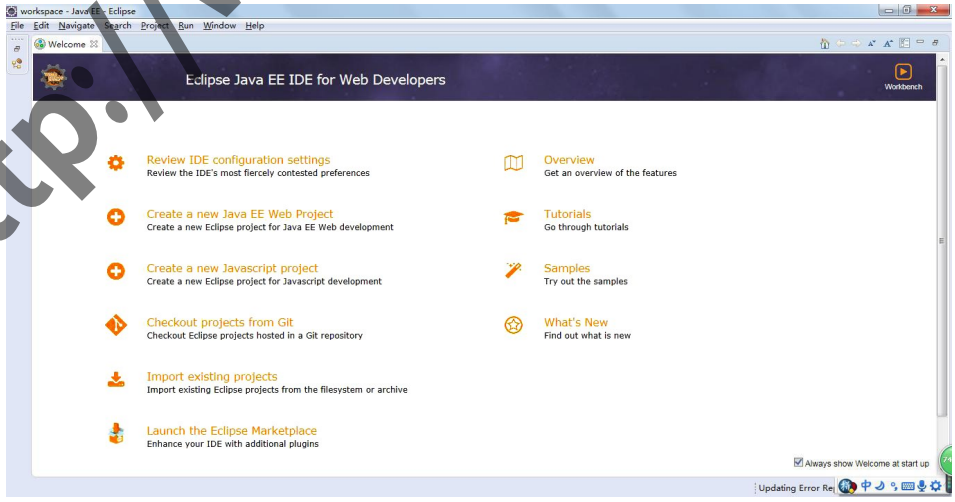

图 2-21　Eclipse 欢迎页面

2.3.2 在 Eclipse 中配置 JRE

(1)点击【Window】菜单中的【Preferences】菜单项,打开如图 2-22 所示界面。

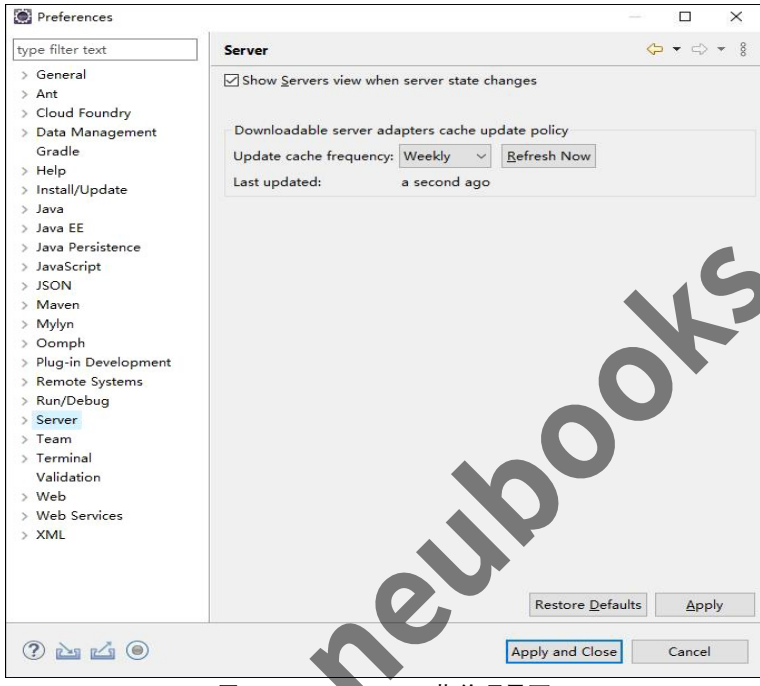

图 2-22 Preferences 菜单项界面

(2)在图 2-22 中点击左侧【Java】列表项,点开【Installed JREs】,打开如图 2-23 所示界面。如果右侧窗口中已出现电脑中安装的 JRE,则不需另行配置,点击【OK】按钮即可;如列表中未出现,则需要点击【Add…】按钮,手动添加。

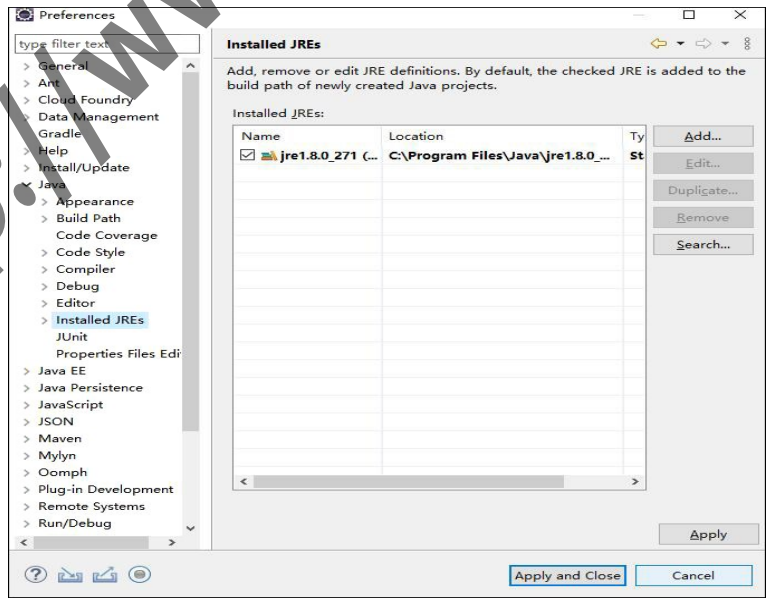

图 2-23 配置 JRE 界面

2.3.3　在 Eclipse 中配置、启动 Tomcat

(1)如图 2-24 所示，点开左侧列表项中的【Server】，选择【Runtime Environments】，打开如图 2-25 所示界面。

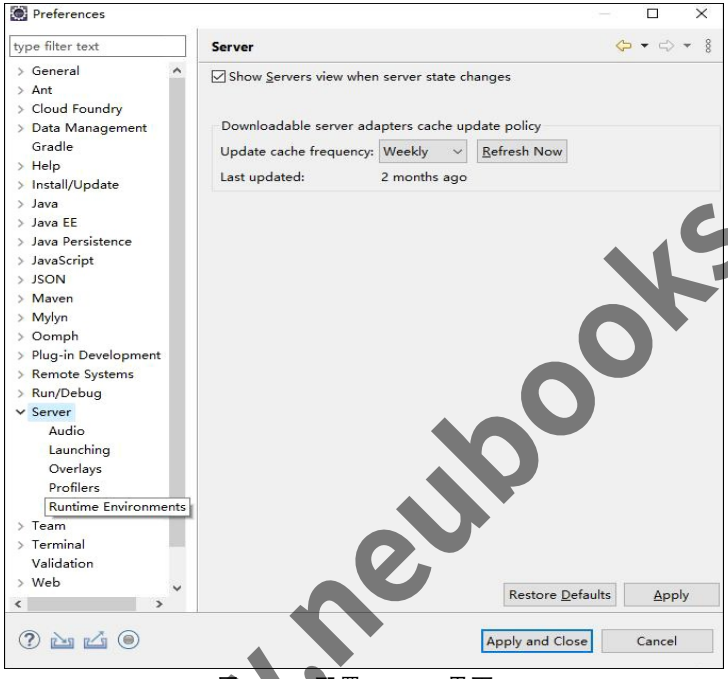

图 2-24　配置 Tomcat 界面 1

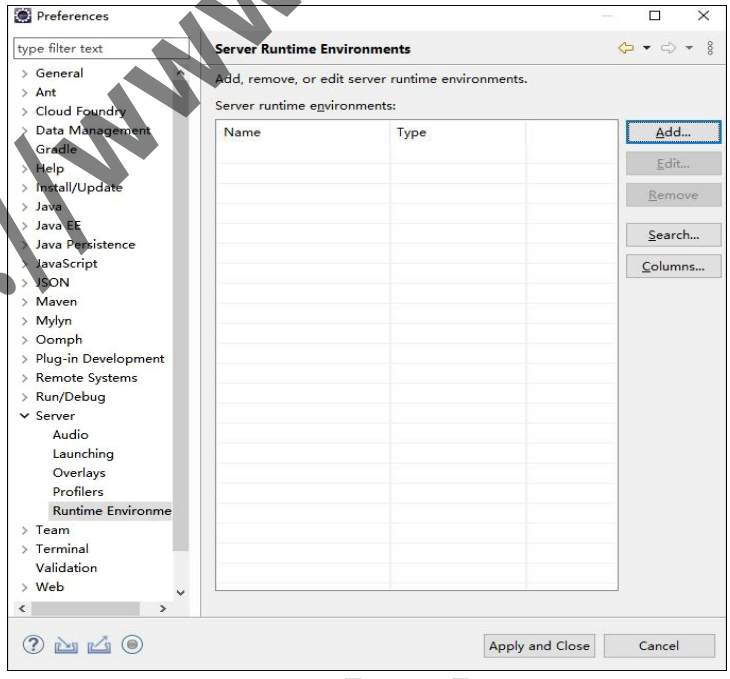

图 2-25　配置 Tomcat 界面 2

(2)在图 2-25 中点击【Add…】按钮,打开如图 2-26 所示界面。

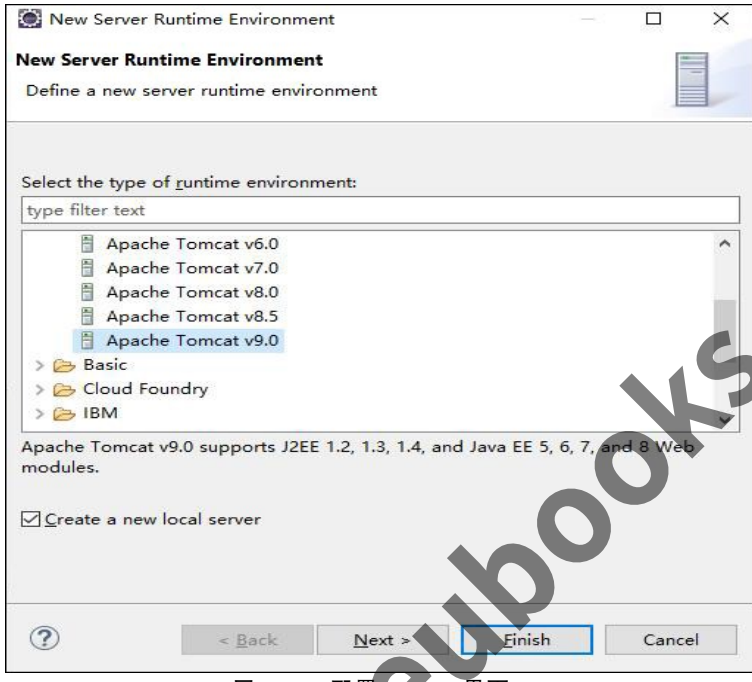

图 2-26　配置 Tomcat 界面 3

(3)在图 2-26 所示图中选择“Apache Tomcat V9.0”,并勾选“Create a new local server”,点击【Next】按钮,打开如图 2-27 界面,选择电脑中 Tomcat 安装目录,如图 2-28 所示。

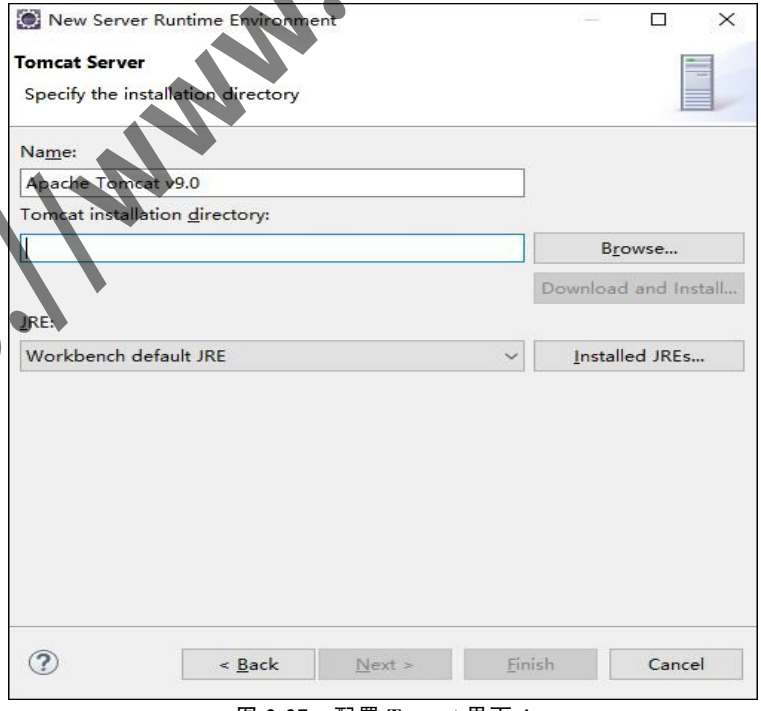

图 2-27　配置 Tomcat 界面 4

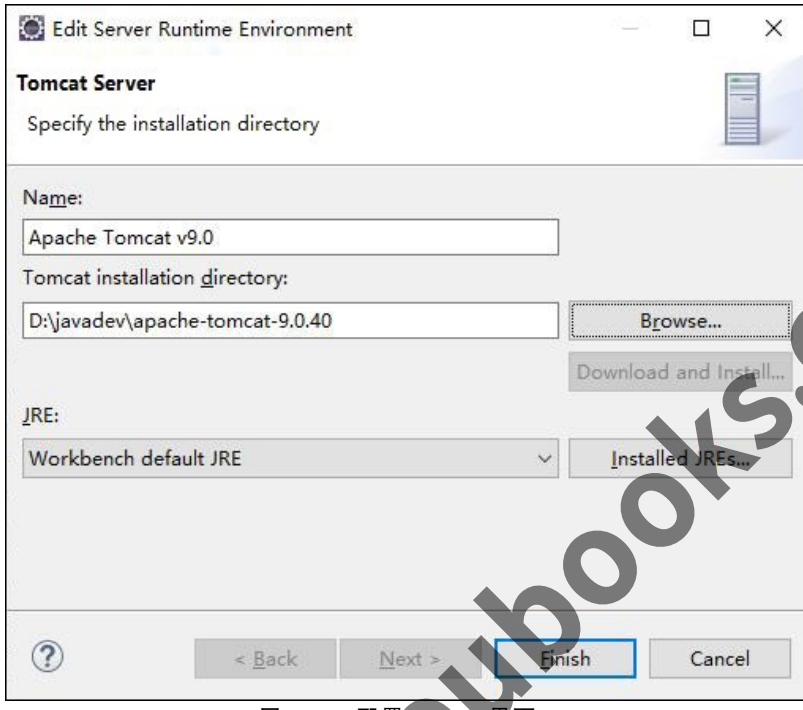

图 2-28　配置 Tomcat 界面 5

(4)在图 2-28 中点击【Finish】后完成配置，在 Eclipse 的 Server 视图中即可看到刚刚创建的服务器，如图 2-29 所示。

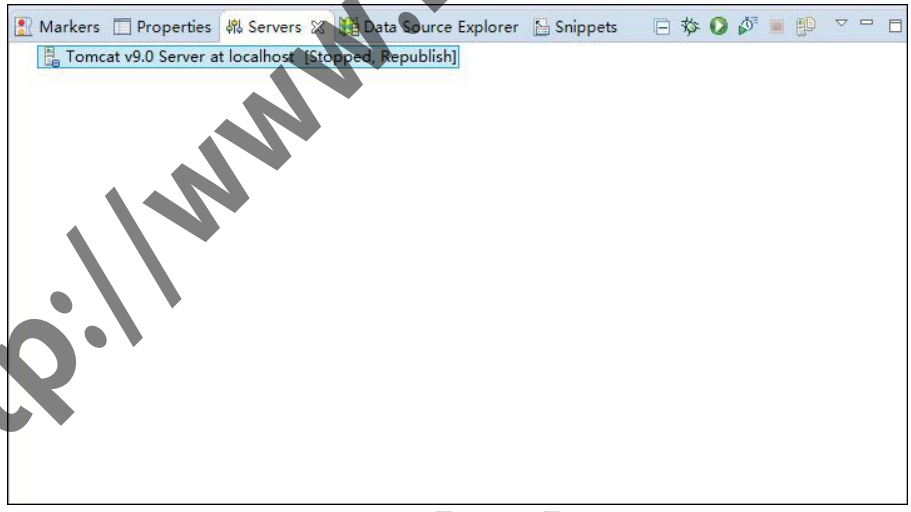

图 2-29　配置 Tomcat 界面 6

(5)双击图 2-29 中的“Tomcat v9.0 Server at localhost [Stopped,Republish]”打开服务器设置窗口，这里主要设置“Server Locations”中的 Server path 和 Deploy path，其中 Server path 选择第 2 项即 Tomcat 安装目录，并且将 Deploy path 由“wtpwebapps”改为“webapps”，如图 2-30 所示，设置完成保存(Ctrl+S)关闭。

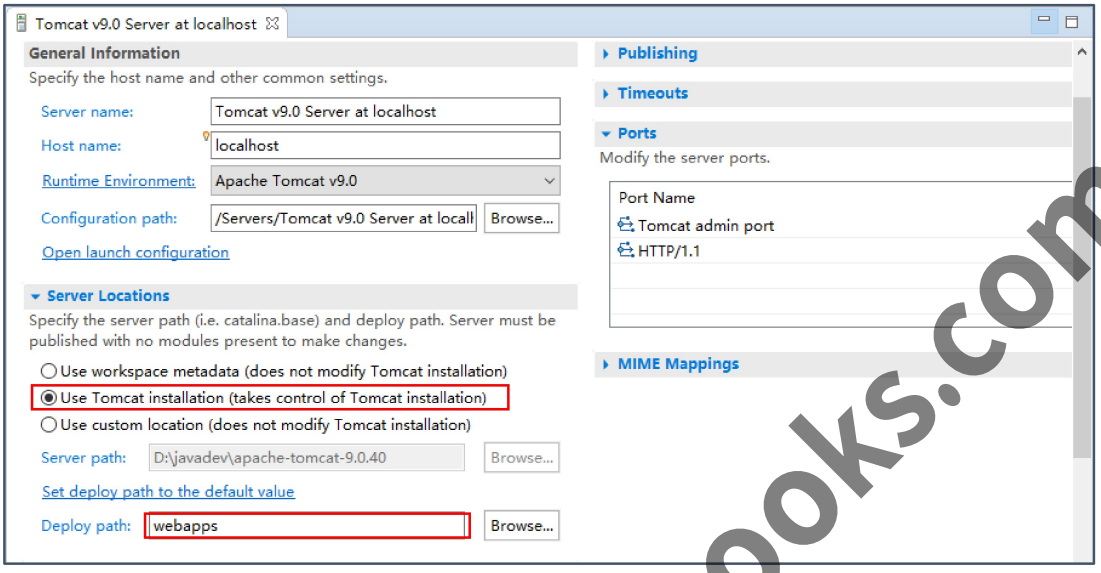

图 2-30　配置 Tomcat 界面 7

(6)点击顶端工具栏或 Server 视图中的  按钮,即可启动 Tomcat。启动后控制台窗口如图 2-31 所示。

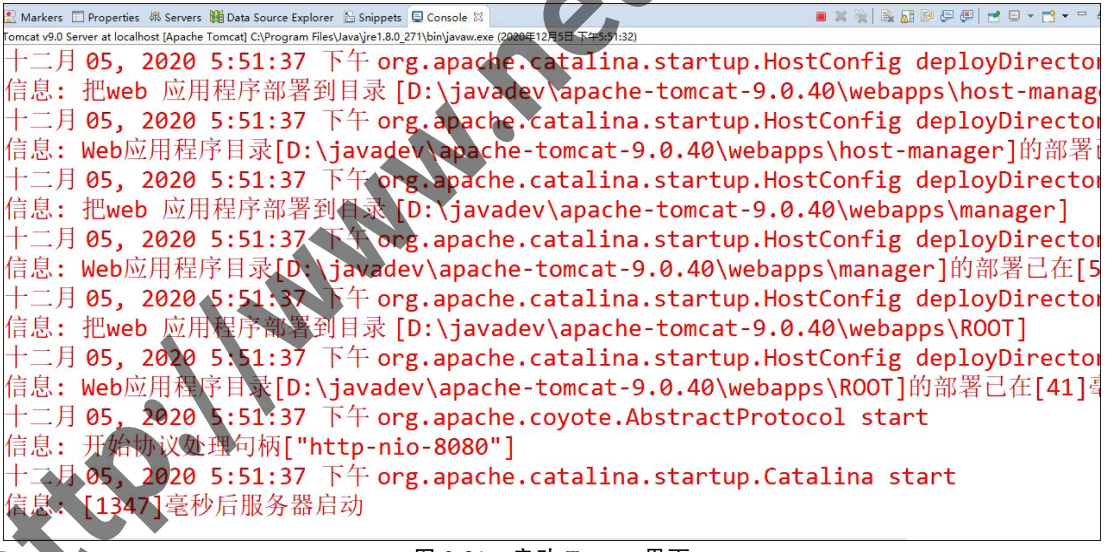

图 2-31　启动 Tomcat 界面

Tomcat 启动后,可以直接打开浏览器访问,也可以点击工具栏中的  按钮打开 Eclipse 的浏览器窗口,在地址栏中输入 URL:http://localhost:8080 进行访问测试。

2.3.4　在 Eclipse 中开发 Web 应用

(1)在菜单栏中选择【File】→【New】→【Dynamic Web Project】,新建 Dynamic Web Project,在图 2-32 所示的界面中,输入 Project name“Hello”。

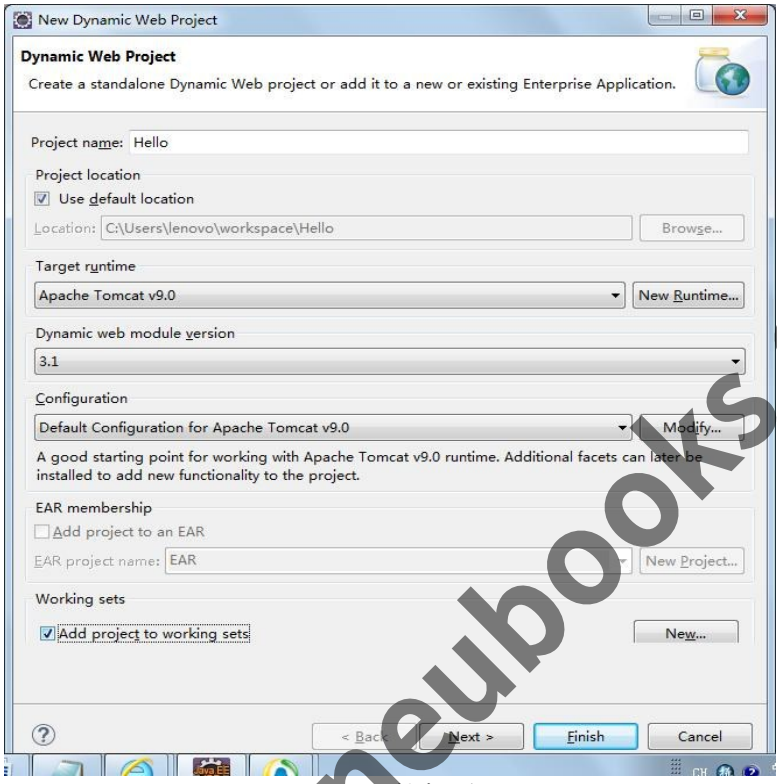

图 2-32　创建工程

(2)点击图 2-32 中的【Next】按钮后，设置 src 输出目录为 WebContent\WEB-INF\classes，如图 2-33 所示。

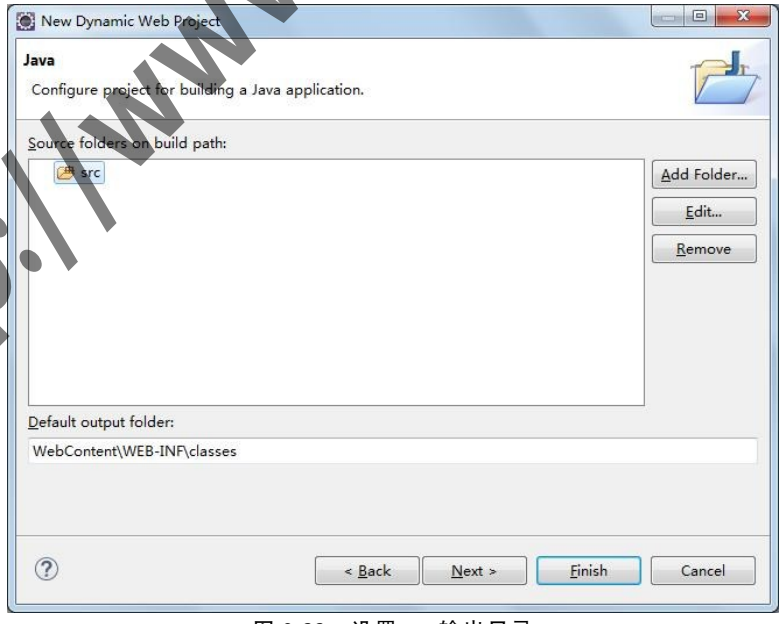

图 2-33　设置 src 输出目录

(3)点击图 2-33 中的【Next】按钮后，打开设置项目发布的根路径，如图 2-34 所示。

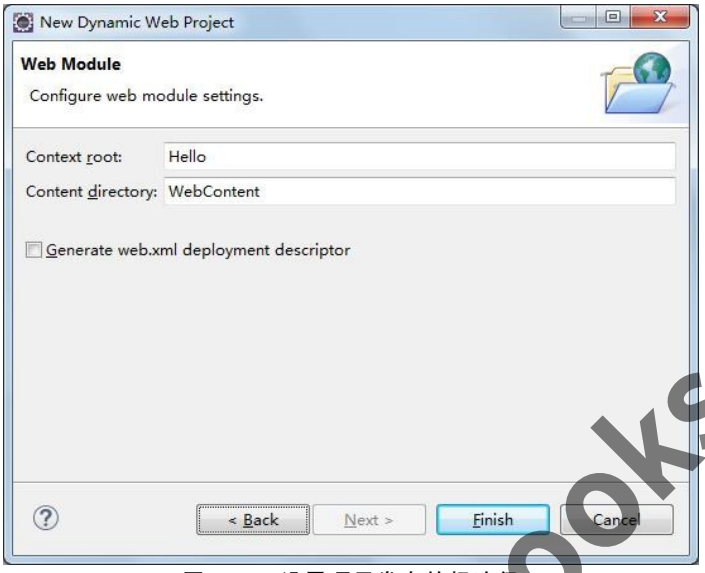

图 2-34　设置项目发布的根路径

(4)点击图 2-34 中的【Finish】按钮后，在 Eclipse 的导航中能看到刚刚创建的工程，如图 2-35 所示。

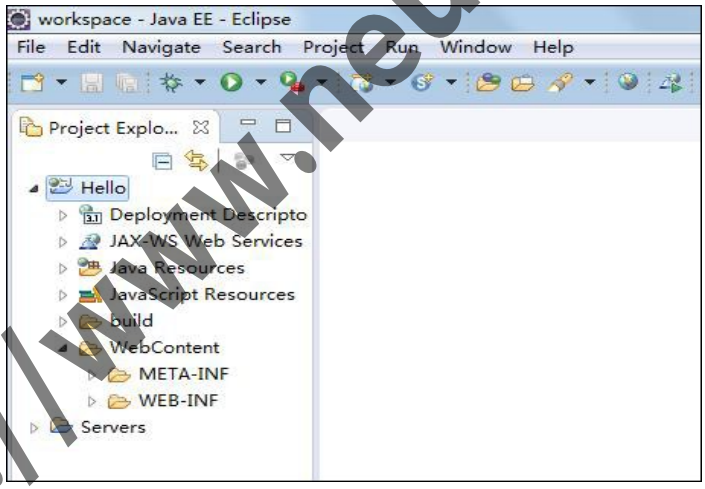

图 2-35　工程目录

(5)在图 2-35 中的 Hello 工程的 WebContent 目录上右键新建 JSP 文件，如图 2-36 所示，输入文件名字“hello.jsp”。

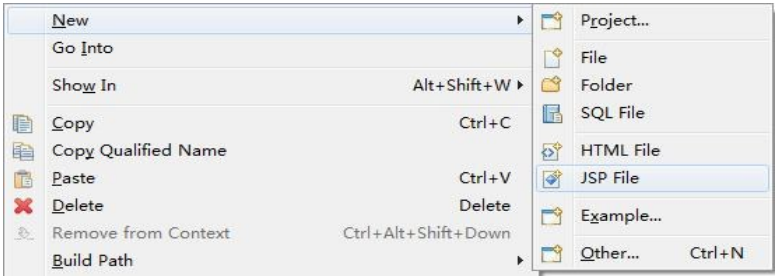

图 2-36　在工程下建立文件

(6)成功得到 hello.jsp 编辑界面,在<body>标记下方输入"Eclipse 下的第一个页面",将编码方式改为"GBK",如图 2-37 所示。

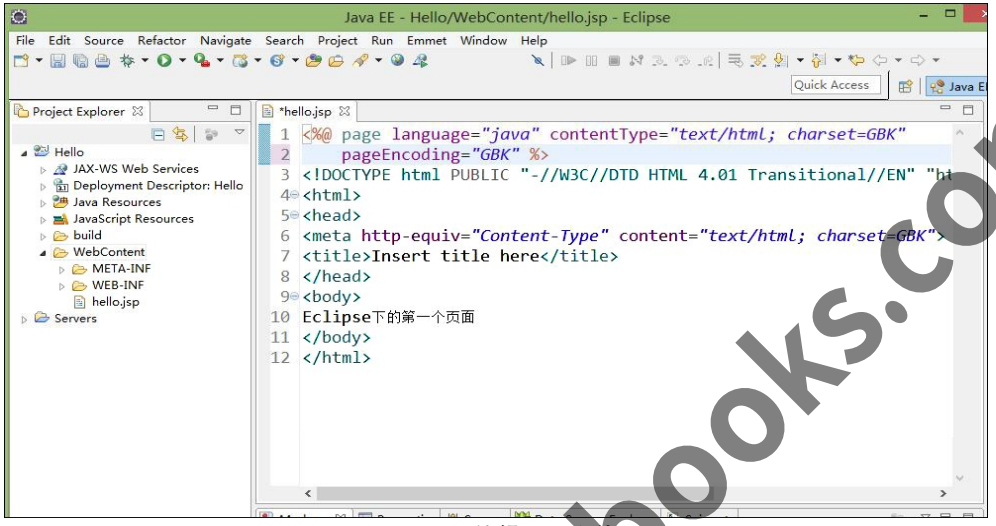

图 2-37　编辑 JSP 文件

(7)保存 JSP 文件后,在文件编辑区右键【Run As】→【Run on Server】,出现图 2-38 所示服务器选择界面。

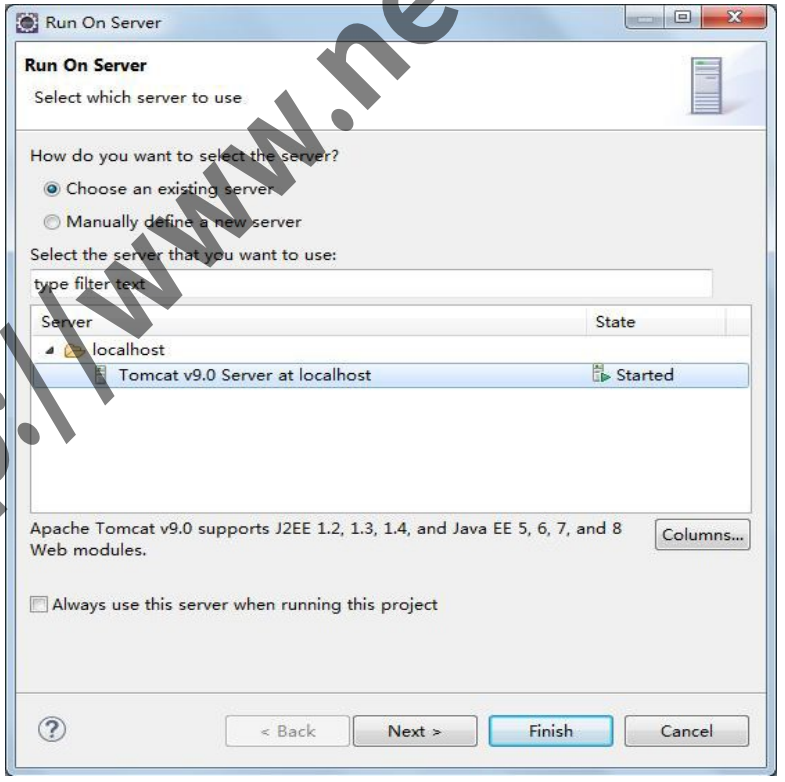

图 2-38　运行 JSP 程序

(8)在图 2-38 所示界面中点击【Finish】按钮后,自动打开内置浏览器,显示页面运行结果,