第 1 章　JSP 和 Web 应用程序

本章导读

JSP 是一种开发 Web 应用程序的新技术。自发布以来，它一直受到密切的关注。为什么 JSP 发展如此迅猛，原因之一是它基于 Java 技术，而 Java 极适合企业级计算。另一个原因在于 JSP 支持强大的 Web 应用程序开发模式，它可以把外观呈现与处理过程分隔开来，使得擅长图形制作、布局的网页设计师与精通服务器端技术（例如多线程资源池、数据库和高速缓存）的程序员能够协调地工作。尽管其他一些技术也支持类似的开发模式，例如 ASP、PHP 和 ColdFusion，但它们当中没有一种能提供 JSP 的所有优点。

本章首先简要介绍 JSP 及其发展状况，接着介绍 JSP 开发环境的配置以及常用开发工具，最后讲解 Web 应用程序和软件体系结构。

1.1　JSP 概述

JSP（Java Server Pages）是由 Sun 公司于 1999 年 6 月在 Java 语言基础上开发出来的一种动态网页制作技术，在 Sun 正式发布 JSP 之后，这种新的 Web 应用开发技术很快引起了人们的关注。

1.1.1　什么是 JSP

网络技术日新月异，细心的网友会发现许多网页文件扩展名不再只是“.htm”，还有“.jsp”、“.asp”等，这些都是采用动态网页技术制作出来的。

早期的动态网页主要采用 CGI 技术，但由于编程困难、效率低下、修改复杂，所以有逐渐被新技术取代的趋势。目前颇受关注的几种新技术有 PHP、ASP 和 JSP。

JSP 是一种动态网页技术标准，在传统的网页 HTML 文件（*.htm，*.html）中加入 Java 程序片段和 JSP 标记（tag），就构成了 JSP 网页（*.jsp）。Web 服务器在遇到访问 JSP 网页的请求时，首先执行其中的程序片段，然后将执行结果以 HTML 的形式返回给客户。程序片段可以操作数据库、重新定向网页等，这就是建立动态网站所需要的功能。所有程序操作都在服务器端执行，网络上传送给客户端的仅是得到的结果，对客户浏览器的要求很低，可以实现无 Plugin，无 ActiveX，无 Java Applet，甚至无 Frame。

ASP 和 JSP 的区别主要有以下两点：一是 ASP 的编程语言是 VBScript 之类的脚本语言，JSP 使用的是 Java、JavaScript 等；二是 ASP 与 JSP 这两种技术的语言引擎用完全不同的方式处理页面中嵌入的程序代码。在 ASP 下，VBScript 代码被 ASP 引擎解释执行；在 JSP 下，代码被编译成 Servlet 并由 Java 虚拟机执行，这种编译操作仅在对 JSP 页面的第一次请求时发生。

1.1.2 JSP 的优点

JSP 技术在多个方面加速了动态 Web 页面的开发，它具有很多优点。首先，它可以将内容的生成和显示进行分离。使用 JSP 技术，Web 页面开发人员可以使用 HTML 或者 XML 标识来设计和格式化最终页面；使用 JSP 标识或者小脚本来生成页面上的动态内容。生成内容的逻辑被封装在标识和 JavaBean 组件中，并且捆绑在小脚本中，所有的脚本在服务器端运行。如果核心逻辑被封装在标识和 Bean 中，那么其他人，如 Web 管理人员和页面设计者，能够编辑和使用 JSP 页面，而不影响内容的生成。在服务器端，JSP 引擎解释 JSP 标识和小脚本，生成所请求的内容（例如，通过访问 JavaBean 组件，使用 JDBC TM 技术访问数据库，或者包含文件），并且将结果以 HTML（或者 XML）页面的形式发送回浏览器。这有助于作者保护自己的代码，而又保证任何基于 HTML 的 Web 浏览器的完全可用性。

其次，强调可重用的组件。绝大多数 JSP 页面依赖于可重用的、跨平台的组件（JavaBean 或者 Enterprise JavaBean TM 组件）来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件，或者使得这些组件为更多的使用者或者客户团体所使用。基于组件的方法加速了总体开发过程，并且使得各种组织在他们现有的技能和优化结果的开发努力中得到平衡。

第三，采用标识简化页面开发。Web 页面开发人员不会都是熟悉脚本语言的编程人员。JSP 技术封装了许多功能，这些功能可以在 XML 标识中进行动态内容的生成。标准的 JSP 标识能够访问和实例化 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。

1.2 JSP 开发环境的配置

要学习 JSP 开发，必须先搭建一个符合 JSP 规范的开发环境。Sun 推出的 JSP 是一种执行于服务器端的动态网页开发技术，它基于 Java 技术。执行 JSP 时需要在 Web 服务器上架设一个编译 JSP 网页的引擎。配置 JSP 环境可以有多种途径，但主要工作就是安装和配置 Web 服务器和 JSP 引擎。本书以实用为原则，介绍了以 JDK+Tomcat 配置 JSP 环境的方法。

（1）JDK。Java 的软件开发工具，是 Java 应用程序的基础。JSP 是基于 Java 技术的，所以配置 JSP 环境之前必须要安装 JDK。本书使用的版本是 j2sdk1.4.1，可以到 Sun 公司的网站免费下载，网址是：http://java.sun.com/j2se/1.4.1/download.html。

（2）Tomcat 服务器。Tomcat 服务器是 Apache 组织开发的一种 JSP 引擎，本身具有 Web 服务器的功能，可以作为独立的 Web 服务器来使用。同时该软件也是免费的，对于初学者来说，Tomcat 是一个很不错的选择。本书使用的版本是 Tomcat 5.0.28，下载网址是：http://apache.linuxforum.net/dist/jakarta/Tomcat-5/v5.0.28/bin/jakarta-Tomcat-5.0.28.exe。

1.2.1 JDK 的安装和配置

下载好 JDK，单击安装，选择好安装路径，正确安装在计算机上。安装完成后右击桌面上的“我的电脑”，选择“属性”，然后选择“高级”里面的“环境变量”，在打开的界面中需要设置三个变量 JAVA_HOME、Path、CLASSPATH，在没安装过 JDK 的环境下，Path 变量是

本来存在的。而 JAVA_HOME 和 CLASSPATH 并不存在，需要新建。

（1）新建一个系统变量，变量名为 JAVA_HOME，顾名思义该变量是作用就是声明 Java 的安装路径，笔者的安装路径为“D:\jdk1.5.0_03”。新建 JAVA_HOME 变量，如图 1.1 所示。

图 1.1　新建 JAVA_HOME 变量截图

（2）在系统变量里面找到 Path，然后单击“编辑”，Path 变量的含义就是系统在任何路径下都可以识别 Java 命令，在其变量值处填入“D:\jdk1.5.0_03\bin;”，修改 Path 变量，如图 1.2 所示。

图 1.2　修改 Path 变量截图

（3）再新建一个用户变量，在变量名上填写 CLASS_PATH，该变量的含义是为 Java 加载类（class or lib）路径，只有类在 CLASSPATH 中，Java 命令才能识别。其值为“.; D:\jdk1.5.0_03\lib\tools.jar”，新建 CLASSPATH 用户变量，如图 1.3 所示。

图 1.3　新建用户变量截图

以上三个变量设置完毕，则单击“确定”按钮直至属性窗口消失，接下来是验证安装是否成功。先打开“开始”→“运行”，输入 cmd，进入 dos 系统界面。然后输入 java-version 命令，如果安装成功，系统会显示 java version jdk"1.4.08"（不同版本号则显示不同）。显示界面如图 1.4 所示。

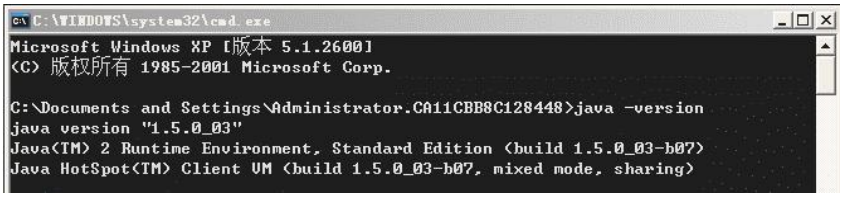

图 1.4　显示版本信息截图

1.2.2 Tomcat 的安装与配置

下载好 jakarta-Tomcat-5.0.28.exe 后运行，如果要改变安装路径，可以在这个步骤操作，笔者的安装路径为 D:\Tomcat 5.0。安装程序会自动搜索 JDK 的安装路径，如果没有正确显示，则可以手工修改，同时访问端口号也可以更改，默认是 8080，笔者为 9090。接下来就开始拷贝文件了，成功安装后，程序会提示启动 Tomcat 并查看 readme 文档。

如果下载的是 zip 压缩包格式的，直接解压缩到 D:\Tomcat 5.0（当然也可以是其他目录），并且按照前面设置环境变量的方法设置新的系统环境 TOMCAT_HOME，值为安装路径 D:\Tomcat 5.0。默认访问端口是 8080，如果想改动，可以在文件夹 conf 下找到 server.xml 文件，用记事本打开，查找到 8080，然后改成你想设置的端口号即可。

至此安装与配置都已完成，重启计算机，在 Tomcat 的安装文件夹 bin 里找到 startup.bat 文件，双击即可启动 Tomcat，打开浏览器输入 http://localhost:8080（如端口更改，则将 8080 改成你所更改的数字，下同）即可看到 Tomcat 的相关信息，如果不能打开如图 1.5 所示的页面，则表示没有正确配置好。

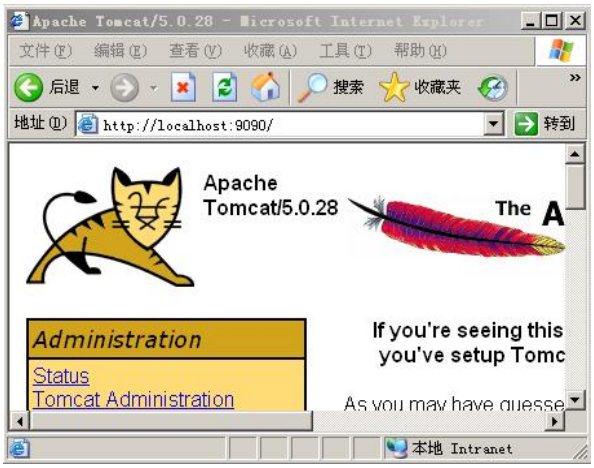

图 1.5 Tomcat 欢迎页面

打开文本编辑器，如记事本，输入下列代码，并保存为 test.jsp（注意扩展名为.jsp）。

```
<%@ page contentType="text/html; charset=gb2312" language="java"  errorPage="" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>JSP 测试页面</title>
</head>
<body>
<**输出字符串"Hello World!"*>
<%out.println("<h1> Hello World! </h1>");%>
</body>
</html>
```

把 test.jsp 放在 D:\Tomcat 5.0\webapps\目录下，在地址栏中输入 http://localhost:8080/

test.jsp，如果浏览器中显示“Hello World!”，则说明已经成功运行了 JSP 页面运行结果如图 1.6 所示。

图 1.6　JSP 测试页面

1.2.3　Tomcat 的目录结构

/bin 关闭 Tomcat 的脚本文件。
/conf 存放 Tomcat 的配置文件。
/server 三个子目录 classes，lib，webapps。
/server/lib 存放 Tomcat 所需要的各种 jar 文件（不能被其他 Web 服务器访问）。
/server/webapps 存放 Tomcat 两个自带 Web 应用 admin 应用和 manager 应用。
/common/lib 存放 Tomcat 服务器以及所有 Web 应用都可以访问的 jar 文件。
/shared/lib 存放所有 Web 应用都可以访问的 jar 文件（不能被 Tomcat 访问）。
/logs 存放 Tomcat 日志文件。
/webapps 当发布 Web 应用时，默认情况下把 Web 文件夹放于此目录下。
/work jsp 生成的 servlet 置在此目录下。
注意：*在每个 Web 应用下/wen-inf/lib 下也可以放 jar 文件，但只对当前应用有效。*

由于 Tomcat 本身具有 Web 服务器的功能，因此不必安装 Apache。但其处理静态 HTML 页面的速度比不上 Apache，且其作为 Web 服务器的功能远不如 Apache，因此把 Apache 和 Tomcat 集成起来，用 Apache 充当 Web 服务器，而 Tomcat 作为专用的 JSP 引擎。这种方案的配置比较复杂，但是能让 Apache 和 Tomcat 完美整合，实现强大的功能。有兴趣的读者可以查看相关资料进行设置。

1.3　JSP 常用开发工具

JSP 引擎搭建起来后就可以着手使用开发工具进行 JSP 的编程了，现下流行的 JSP 开发工具主要有 Eclipse、JBuilder、NetBeans、EditPlus、Ultraedit、Dreamweaver 等。最简单的方法是用记事本创建 JSP 文件，然后将文件拷贝到 Webapps 目录下运行。这里主要介绍 EditPlus、Eclipse 两种工具的一些基本情况，在第 2 章中再详细介绍 Dreamweaver，可以参照其各自的特点，结合自身开发环境选择合适的开发工具。

1.3.1 EditPlus

EditPlus 是一款功能非常强大的文本编辑工具，它支持自定义工具组、自定义文件类型等功能，对于从事程序设计和网页制作的工作者实在是不可或缺！

EditPlus、Ultraedit、记事本是很多高手坚持使用的开发工具，其中 EditPlus 最为方便，可以只把它当作高彩显示代码的工具。EditPlus 支持 HTML、CSS、PHP、ASP、JSP、Perl、C/C++、Java、JavaScript 和 VBScript 的语法加亮，还可以自己扩展定制。不仅如此，EditPlus 经过设置后还能直接编译和运行 Java 等程序，读者可以在网上查阅相关的资料。

EditPlus 的运行界面如图 1.7 所示。

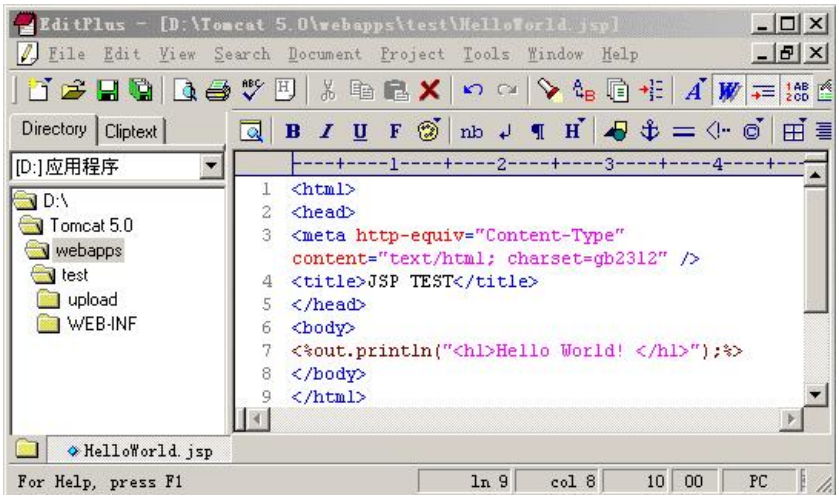

图 1.7 EditPlus 的运行界面

1.3.2 Eclipse

目前 Java 开发领域的各种集成开发环境（IDE）呈现出百花齐放的局面，在所有的 IDE 中，Eclipse 可以说是最有发展前途的产品之一。Eclipse 最初由 OTI 和 IBM 两家公司的 IDE 产品开发组创建，起始于 1999 年 4 月。IBM 提供了最初的 Eclipse 代码基础，包括 Platform、JDT 和 PDE。目前由 IBM 牵头，围绕着 Eclipse 项目已经形成一个庞大的 Eclipse 联盟，有 150 多家软件公司参与到 Eclipse 项目中，其中包括 Borland、Rational Software、Red Hat 及 Sybase，最近 Oracle 也计划加入到 Eclipse 联盟中。

Eclipse 是一个开放的开发平台，通过插件系统，可以拥有几乎无限的扩展能力，因此越来越多的程序员使用它来开发程序，它也是笔者喜欢的开发工具。鉴于本书讲述的是 JSP 的开发，将重点讲述怎样使用 Eclipse 开发 JSP。

首先，下载 Eclipse 的 Win32 安装文件，官方网站 http://www.eclipse.org/提供了较新版本 Eclipse 3.0.1 的下载，直接运行 eclipse.exe，程序会自动找到 JDK 并完成相应的配置。下载 Sysdeo Eclipse Tomcat 3.1.0，它是 Tomcat 在 Eclipse 上的一个插件，一直以来用它做 Eclpise 下 Tomcat 的启动开发平台，解压 TomcatPluginV31beta.zip 到 Eclipse 安装目录下的 plugins 目录中。

下面开始配置 Eclipse。

（1）启动 Eclipse。

（2）打开菜单 Windows→Preferences。

（3）在左侧选择 Tomcat，可以看到右侧出现一些表单。

（4）Tomcat Version 选择 Version 5.0.x，Tomcat Home 选择 Tomcat 的安装路径，Configuration File 中会自动填入 Tomcat 的配置文件 server.xml。

（5）展开左侧的 Tomcat 菜单，选择 JVM Settings，JRE 选择 Detected VM，单击 Apply 后单击 OK。

（6）工具栏中应该多了一个小猫的图标，如果没有的话，选择菜单 Windows→Custmize Perspective，展开 other 选项，在 Tomcat 上打勾即可。

（7）单击 Start Tomcat 按钮，Tomcat 便在 console 中启动了。

运行界面如图 1.8 所示。

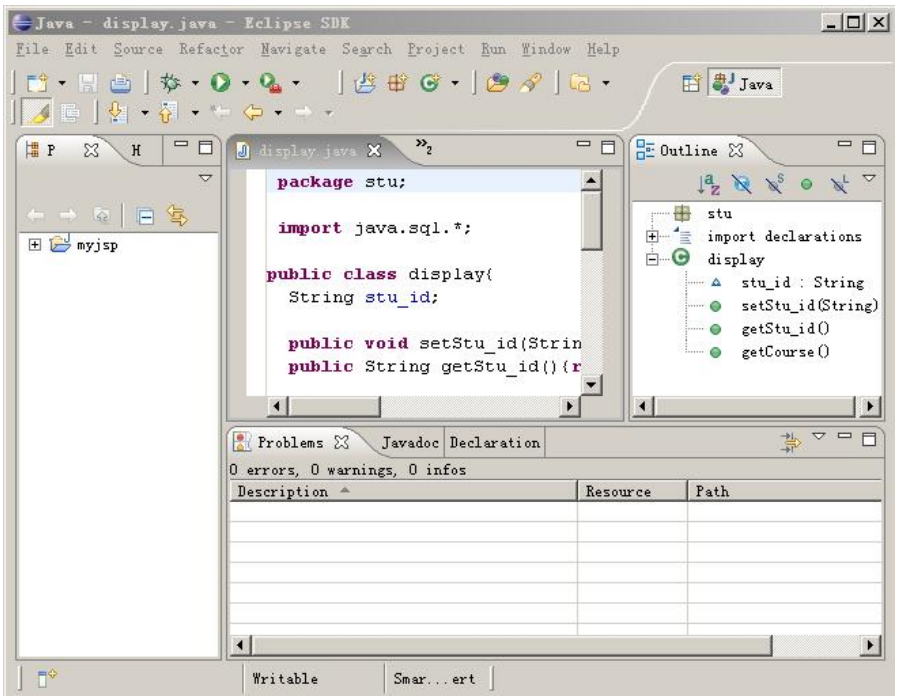

图 1.8　Eclipse 的运行界面

1.4　Web 应用程序

前面的几节中已经使用过 Web 应用程序术语，其所指的既不是一个真正意义上的 Web 网站，又不是一个传统的应用程序。换句话说，它是一些 Web 网页和用来完成某些任务的其他资源的一个集合。这些网页存储在 Web 服务器上，其部分内容或全部内容是未确定的。只有当用户请求 Web 服务器中的某个页面时，才确定该页的最终内容。因为页面内容基于用户的操作，随请求的不同而变化，所以这种页称为动态页面，反之则称为静态页面。因而 Web 应用程序是一组静态和动态 Web 页的集合。

静态页面是当接到用户请求时不会发生更改的页，Web 服务器将该页发送到浏览器，不对其进行修改。相反，将动态页面发送到浏览器之前，服务器将对该页进行修改。页面发生更改是称其为动态页面的原因。例如，读者可以设计一个页来显示学生名单，而这些信息（例如学生姓名和结果）在接到请求时根据查询条件再确定。

建立 Web 应用程序是为了解决多种问题，Web 应用程序的一般用途如下：

（1）用户可以快速方便地在一个内容丰富的 Web 站点上查找信息。

（2）收集、保存和分析用户提供的数据。

（3）对内容不断变化的 Web 站点进行更新。

1.4.1 Web 应用程序的工作原理

1.4.1.1 处理静态页面的工作原理

一般的 Web 站点由一组相关的 HTML 页面和文件组成，这些页面和文件驻留在运行 Web 服务器的计算机上。当用户单击 Web 页上的某个链接或在浏览器中选择一个书签、或在浏览器的“地址”文本框中输入一个 URL 并单击“转到”时，便生成一个页面请求。

静态页面的内容由网页设计人员确定，当接到请求时，内容不发生更改。页面的每一行代码都是在将页面放置到服务器之前由设计人员编写好的。严格来说，“静态”页可能不是完全静态的。例如，一个鼠标经过图像或一个 Flash 影片可以使静态页活动起来。但是，本系统所说的静态页面是指发送到浏览器时不进行修改的页面。

当 Web 服务器接收到对静态页的请求时，服务器将读取该请求，查找该页，然后将其发送到请求浏览器，处理流程如图 1.9 所示。

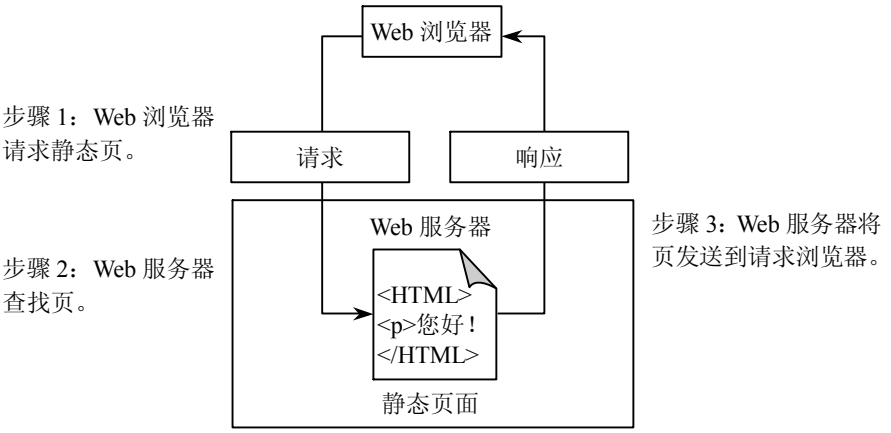

图 1.9 静态页面处理流程

1.4.1.2 处理动态页面的工作原理

当 Web 服务器接收到对常规 Web 页的请求时，服务器将该页发送到请求浏览器，而不进行进一步的处理。当 Web 服务器接收到对动态页的请求时，它将作出不同的反应，将该页传递给一个负责完成页面的特殊软件扩展，这个特殊软件叫做应用程序服务器。

应用程序服务器读取页上的代码，根据代码中的指令完成页面，所得的结果将是一个静态页，应用程序服务器将该页传递回 Web 服务器，然后 Web 服务器将该页发送到浏览器。当

该页到达时，浏览器得到的全部内容都是纯 HTML，处理流程如图 1.10 所示。读者可以通过浏览器查看源文件得到该 HTML 文件。

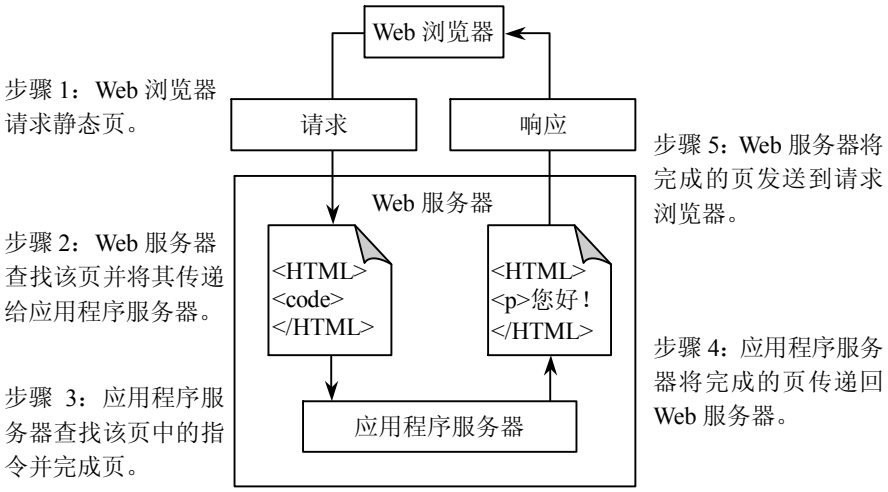

图 1.10　动态页面处理流程

JSP 页面的执行过程是通过 JSP 引擎把 JSP 标记符、JSP 页中的 Java 代码甚至连同静态 HTML 内容都转换为大块的 Java 代码，JSP 引擎和 JDK 在这里充当应用程序服务器的角色。这些代码块被 JSP 引擎组织到用户看不到的 Java Servlet 中去，然后 Servlet 自动把它们编译成 Java 字节码。这样，当网站的访问者请求一个 JSP 页时，在他不知道的情况下，一个已经生成的、预编译过的 Servlet 实际上将完成所有的工作。JSP 引擎将该页传回 Web 服务器，然后 Web 服务器将该页发送到浏览器。

1.4.2　Web 服务器和应用程序服务器

通俗地说，Web 服务器传送页面使浏览器可以浏览，而应用程序服务器提供的是客户端应用程序可以调用的方法。确切一点，Web 服务器专门处理 HTTP 请求，而应用程序服务器是通过很多协议来为应用程序提供事务逻辑处理。

Web 服务器可以解析 HTTP 协议。当 Web 服务器接收到一个 HTTP 请求时，会返回一个 HTTP 响应，例如送回一个 HTML 页面。为了处理一个请求，Web 服务器可以响应一个静态页面或图片，进行页面跳转。把动态响应的产生委托给一些其他的程序例如 CGI 脚本、JSP 脚本、ASP 脚本、服务器端 JavaScript，或者一些其他的服务器端技术。无论它们的目的如何，这些服务器端的程序通常产生一个 HTML 的响应来让浏览器浏览。

Web 服务器的代理模型非常简单。当一个请求被送到 Web 服务器时，它只单纯地把请求传递给可以很好地处理请求的程序（服务器端脚本）。Web 服务器仅仅提供一个可以执行服务器端程序和返回响应的环境，而不会超出职能范围。而服务器端程序通常具有事务处理、数据库连接和消息发送等功能。

应用程序服务器通过各种协议（包括 HTTP 协议）把商业逻辑暴露给客户端应用程序。应用程序服务器提供访问商业逻辑的途径以供客户端应用程序使用。应用程序使用此商业逻辑就像调用对象的一个方法（或过程语言中的一个函数）一样。

应用程序服务器的客户端可能会运行在一台 PC、一个 Web 服务器或者甚至是其他的应用程序服务器上。在应用程序服务器与其客户端之间来回穿梭的信息不仅仅局限于简单的显示标记。相反，这种信息就是程序逻辑。正是由于这种逻辑取得了数据和方法调用的形式而不是静态 HTML，所以客户端才可以随心所欲地使用这种被暴露的商业逻辑。

另外，现在大多数应用程序服务器也包含了 Web 服务器，这就意味着可以把 Web 服务器当作是应用程序服务器的一个子集。虽然应用程序服务器包含了 Web 服务器的功能，但是开发者很少把应用程序服务器部署成这种功能。相反，如果需要，他们通常会把 Web 服务器和应用程序服务器独立配置。这种功能的分离有助于提高性能，而且给最佳产品的选取留有余地。

1.5 软件编程体系

当今世界科学技术飞速发展，尤其以通信、计算机、网络为代表的互联网技术更是日新月异。由于计算机互联网在政治、经济、生活等各个领域的发展、运用以及网络的迅速普及和全社会对网络的依赖程度，计算机网络已经成为国家的经济基础和命脉，成为社会和经济发展的强大动力，其地位越来越重要。但是，由于主流技术研发企业和用户对“B/S”和“C/S”技术谁优谁劣、谁代表技术潮流发展等问题争论不休。因此本节就此两项技术发展变化和应用前景做些探讨，供读者参考。

1.5.1 什么是 C/S 和 B/S

1.5.1.1 什么是 C/S 结构

C/S （Client/Server）结构，简单地说就是传统意义上拥有客户端和服务器端的网络软件或系统，可以用 VB 或 VC 等语言开发，比如最常用的 QQ 就是 C/S 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。目前大多数应用软件系统都是 Client/Server 形式的两层结构，由于现在的软件应用系统正在向分布式的 Web 应用发展，Web 和 Client/Server 应用都可以进行同样的业务处理，应用不同的模块共享逻辑组件；因此，内部的和外部的用户都可以访问新的和现有的应用系统，通过现有应用系统中的逻辑可以扩展出新的应用系统。这也就是目前应用系统的发展方向。

1.5.1.2 什么是 B/S 结构

B/S（Browser/Server）结构即浏览器和服务器结构。它是随着 Internet 技术的兴起，由 C/S 结构进行改进而形成的结构。在这种结构下，用户工作界面通过 WWW 浏览器来实现，极少部分事务逻辑在前端实现，主要事务逻辑在服务器端实现，形成所谓三层结构。这样就大大简化了客户端的负荷，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。

以目前的技术看，局域网建立 B/S 结构的网络应用，并通过 Internet/Intranet 模式下数据库的应用，相对易于把握、成本也是较低的。它是一次到位的开发，能实现不同的人员，从不同的地点，以不同的接入方式（比如 LAN，WAN，Internet/Intranet 等）访问和操作共同的数据库；它能有效地保护数据平台和管理访问权限，服务器数据库也很安全。特别是在 Java 这样的跨平台语言出现之后，B/S 架构管理软件更是方便、快捷、高效。

1.5.2　C/S 和 B/S 之比较

C/S 和 B/S 是当今世界开发模式的两大主流技术。C/S 由美国 Borland 公司最早研发，B/S 由美国微软公司研发。目前，这两项技术已被世界各国所掌握，国内公司以 C/S 和 B/S 技术开发出的产品也很多。这两种技术都有自己一定的市场份额和客户群，各家企业都说自己的管理软件架构技术功能强大、先进、方便，都能举出各自的客户群体，可谓仁者见仁，智者见智。

1.5.2.1　C/S 架构软件的优势与劣势

1．应用服务器运行数据负荷较轻

最简单的 C/S 体系结构的数据库应用由两部分组成，即客户应用程序和数据库服务器程序。二者可分别称为前台程序与后台程序。运行数据库服务器程序的机器，也称为应用服务器。一旦服务器程序被启动，就随时等待响应客户程序发来的请求；客户应用程序运行在用户自己的计算机上，对应于数据库服务器，可称为客户端，当需要对数据库中的数据进行任何操作时，客户程序就自动寻找服务器程序，并向其发出请求，服务器程序根据预定的规则作出应答，送回结果。应用服务器运行数据负荷较轻。

2．数据的存储管理功能较为透明

在数据库应用中，数据的存储管理功能，是由服务器程序和客户应用程序分别独立进行的。前台主要应用可以违反的规则，并且常把那些不同的（不管是已知还是未知的）运行数据分散到服务器程序中实现。所有这些，对于工作在前台程序上的最终用户，都是“透明”的，他们无须过问（通常也无法干涉）背后的过程，就可以完成自己的一切工作。在客户服务器架构的应用中，前台程序并不“瘦小”，很多事情可以自行完成，而并不都交给了服务器和网络。

3．C/S 架构的劣势是高昂的维护成本且投资大

首先，采用 C/S 架构，要选择适当的数据库平台来实现数据库数据的真正“统一”，使分布于两地的数据同步，完全交由数据库系统去管理，但逻辑上两地的操作者要直接访问同一个数据库才能有效实现。有这样一些问题，如果需要建立“实时”的数据同步，就必须在两地间建立实时的通信连接，保持两地的数据库服务器在线运行，网络管理工作人员既要对服务器进行维护和管理，又要对客户端进行维护和管理，这需要高昂的投资和复杂的技术支持，维护成本很高，维护任务量大。

其次，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，因此传统的 C/S 结构的软件代价高，效率低，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 架构，并对其形成威胁和挑战。

1.5.2.2　B/S 架构软件的优势与劣势

1．维护和升级方式简单

目前，软件系统的改进和升级越来越频繁，B/S 架构的产品明显体现着更为方便的特性。对一个稍微大一些的单位来说，如果需要系统管理人员在几百甚至上千部电脑之间来回奔跑，效率和工作量是可想而知的，但 B/S 架构的软件只需要管理服务器就行了，所有的客户端只是浏览器，根本不需要做任何的维护。无论用户的规模有多大，有多少分支机构都不会增加任

何维护和升级的工作量，所有的操作只需要针对服务器进行；如果是异地，只需要把服务器连接专网即可，实现远程维护、升级和共享。因此客户机越来越“瘦”，而服务器越来越“胖”是将来信息化发展的主流方向。今后，软件升级和维护会越来越容易，而使用起来会越来越简单，这对用户人力、物力、时间、费用的节省是显而易见的。因此，维护和升级革命的方式是“瘦”客户机，“胖”服务器。

2．成本降低，选择更多

大家都知道 Windows 占领了桌面操作系统的大部分市场，浏览器成为了标准配置，但在服务器操作系统上 Windows 并不是处于绝对的统治地位。现在的趋势是凡使用 B/S 架构的应用管理软件，只需安装在 Linux 服务器上即可，而且安全性高。所以服务器操作系统的选择是很多的，不管它选用哪种操作系统都可以让大部分人使用 Windows 作为桌面操作系统，这就使得免费的 Linux 操作系统快速发展起来。除了 Linux 操作系统是免费的以外，数据库也可以选用免费产品，这种几乎全免费的平台非常盛行。

比如说很多人每天上“网易”网，只要安装了浏览器即可，并不需要了解“网易”的服务器用的是什么操作系统，而事实上很多网站确实没有使用 Windows 操作系统，但用户的计算机本身安装的大部分是 Windows 操作系统。

3．应用服务器运行数据负荷较重

由于 B/S 架构管理软件只安装在服务器（Server）端上，网络管理人员只需要管理服务器就行了，用户界面主要事务逻辑在服务器（Server）端完全通过 WWW 浏览器实现，极少部分事务逻辑在前端（Browser）实现，所有的客户端只有浏览器，网络管理人员只需要对客户端做硬件维护。但是，应用服务器运行数据负荷较重，一旦发生服务器“崩溃”等问题，后果不堪设想。因此，许多单位都备有数据库存储服务器，以防万一。

本章小结

JSP（Java Server Pages）是由 Sun 公司基于 Java 语言开发出的一种动态网页制作技术，JSP 技术在多个方面加速了动态 Web 页面的开发。JSP 技术实际上是通过 JSP 引擎把 JSP 标记符、JSP 页中的 Java 代码甚至连同静态 HTML 内容都转换为大块的 Java 代码。这些代码块被 JSP 引擎组织到用户看不到的 Java Servlet 中去，然后 Servlet 自动把它们编译成 Java 字节码。由于是 JSP 引擎自动生成并编译 Servlet，不用程序员动手编译代码，JSP 就能提供高效的性能和快速开发所需的灵活性。

Web 应用程序所指的既不是一个真正意义上的 Web 网站，又不是一个传统的应用程序。它是一些 Web 网页和用来完成某些任务的其他资源的集合。Web 服务器传送页面使浏览器可以浏览，而应用程序服务器提供的是客户端应用程序可以调用的方法。确切一点，Web 服务器专门处理 HTTP 请求，而应用程序服务器通过很多协议来为应用程序提供事务逻辑处理。

C/S（Client/Server）结构是传统意义上拥有客户端和服务器端的网络软件或系统，B/S（Browser/Server）结构即浏览器和服务器结构，它是随着 Internet 技术的兴起，对 C/S 结构的进行改进而形成的结构。

课后习题

一、填空题

1．JSP 的全称是_____，它是由_____公司于 1999 年 6 月基于_____语言开发出来的一种动态网页制作技术。

2．JSP 网页文件的后缀名为_____。

3．JSP 技术实际上是通过_____把 JSP 标记符、JSP 页中的 Java 代码甚至连同静态 HTML 内容都转换为大块的_____代码。

4．配置 JDK 时，需要设置的三个变量分别有_____、_____、_____。

5．页面最终内容基于用户的操作随请求的不同而变化，这种页面称为_____。

6．C/S 结构即_____结构，B/S 结构即_____结构，它们是当今开发模式的两大主流技术。

二、选择题

1．JSP 是由（　　）公司开发的。

A．Microsoft　　B．Sun　　C．IBM　　D．Apache

2．JSP 文件应放在 Tomcat 的文件夹（　　）下。

A．/conf　　B．/bin　　C．/server　　D．/webapps

3．Tomcat 的默认访问端口是（　　）。

A．8080　　B．8088　　C．9090　　D．9099

4．Eclipse 是 Java 开发的（　　）。

A．开发工具包　　B．IDE（集成开发环境）

C．应用程序服务器　　D．Web 服务器

5．以下不属于 B/S 结构特点的是（　　）。

A．维护和升级方式简单　　B．成本降低，选择更多

C．应用服务器运行数据负荷较重　　D．维护成本高且投资大

三、问答题

1．JSP 与 ASP 相比有哪些优势，自身有哪些优点？

2．如何修改 Tomcat 的访问端口号？

3．动态页面与静态页面有何区别，它们最大的不同是什么？

4．Web 应用程序的一般用途有哪些？

5．B/S 结构与 C/S 结构最关键的区别是什么？