

第 1 章 Web 卓越工程师选择 Java 和 JSP 的理由

Java 语言是 Sun 公司发明的一种面向对象编程语言，最初是用来开发家用电器的分布式计算系统的，能实现电冰箱、电视机等电器之间的通信和控制。当时，出于复杂性和安全性的考虑，放弃了复杂的 C++ 语言，而设计了简单得多的 Oak 语言，该语言是 Java 语言的前身。Java 语言真正的兴起是从 Web 应用开始的，第 1 个 Web 应用产品就是 HotJava 浏览器，因而 Java 是 Web 卓越工程师历练的首选利器。而 JSP 作为 Java 在 Web 网站应用开发的重要衍生产品之一，也成为 Web 卓越工程师的重要工具。

1.1 Web 卓越工程师学习 Java 的理由

现在，Java 语言是十分红热的 Web 编程语言，Web 卓越工程师很容易找到学习 Java 的理由。从技术上，Java 语言是先进的、安全的、移植性好、易用的、强大的优秀 Web 编程语言；从经济上，Java 程序员可以找到薪水较高的工作，Java 程序员是热点职位之一。

1.1.1 Java 语言汲取了 C 语言和 C++ 语言的营养

Java 语言建立在 C++ 语言的基础上，基本形式采用 C++ 语言的编码风格，这样便于 C/C++ 程序员学习 Java。

在变量声明、参数传递、操作符、流控制、程序结构等方面，Java 语言继承了 C/C++ 语言的传统风格。

下面以最基本的例子为比较对象，比较 Java 语言和 C++ 语言实现“你好，世界！”问候的编程方法。用 Java 语言编写 HelloWorld 类，并通过主函数 main() 显示问候语，其源代码如下：

```
package project;
public class HelloWorld
{
    public HelloWorld()
    {
    }
    public static void main(String[] args)
    {
        HelloWorld helloworld = new HelloWorld();
        System.out.println("你好，世界！\n");
    }
}
```

用 VC++ 编写 HelloWorld 类，并通过主函数 main() 显示问候语，其源代码如下：

```
// HelloWorld.cpp : Defines the entry point for the console application.
class HelloWorld : public CWinApp
{
```

```

public:
    Helloworld();
};

Helloworld::Helloworld()
{
    // TODO: add construction code here,
    // Place all significant initialization in InitInstance
}

#include "stdafx.h"
int main(int argc, char* argv[])
{
    printf("你好，世界！\n");
    return 0;
}

```

这两个程序的运行结果分别如图 1.1 和图 1.2 所示。

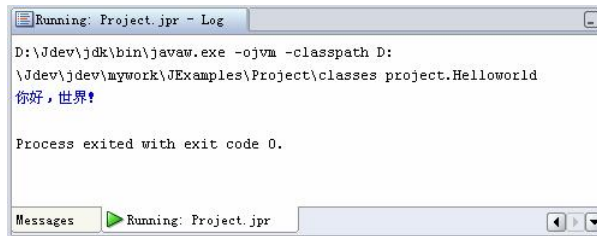


图 1.1 在 JDeveloper 开发环境中运行 Helloworld 类程序

在图 1.1 中，上述 Java 程序经过 javaw 集成开发工具编译、连接和运行，显示了“你好，世界！”问候语，并正常退出程序。

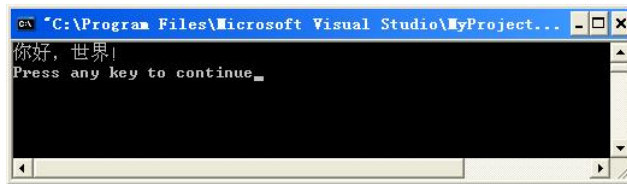


图 1.2 在 Visual C++ 6.0 开发环境中运行 Helloworld C++程序

在图 1.2 中，上述 C++程序在 DOS 环境中显示“你好，世界！”问候语，并提示按任意键就退出该程序。

1.1.2 Java 语言改进并简化了 C++语言

尽管 Java 语言使用了和 C/C++语言类似的编程风格，但是在易用性、安全性、多线程、可移植性等方面有了较大的改进和简化。

(1) Java 不再支持程序员使用指针。

指针是指含有另一个变量的存储地址的变量，指针可以指向一个包含数据的变量，也可

以指向另一个指针变量。虽然指针功能十分强大，但是指针给程序带来了安全隐患。由指针所进行的内存地址操作常会造成不可预知的错误，同时通过指针对某个内存地址进行显式类型转换后，可以访问 C++类的私有成员，从而破坏程序的安全性，造成系统的崩溃。

Java 语言不再向程序员开放指针的用法，指针只是作为 Java 语言的内部机制，不为程序员所控制，同时将数组、字符串作为类实现，从而解决了数组访问的越界错误问题。

(2) Java 支持更高的安全性。

Java 语言使用了字节编码技术，来校验从网上下载的 Java 程序是否完整和正确。Java 语言的内存自动管理增强了程序的安全性，减少了内存溢出的问题。在 C++程序中，不再使用的内存需要编程收回，程序员往往容易出现内存管理的错误；但在 Java 语言中，通过垃圾收集功能，内存一旦不再使用时就会自动释放。

Java 语言有一套比较成熟的异常检测与处理机制，能从系统层次上捕获各种异常，并通过编程处理和显示该异常。因此，Java 程序能检测编译异常和运行异常，比 C++程序具有更强的鲁棒性。

例如，下面分别用 Java 语言和 C++语言从并不存在的文件中读取数据，其 Java 程序的源代码如下：

```
package project;
import java.io.*; //导入 Java 的 IO 类包

public class JException
{
    public JException()
    {
        try
        {
            FileReader fr = new FileReader("d:\\Lab\\Book\\1.txt");
            int c = fr.read(); //从文件中读取一个字符
            while(c!=-1)
            {
                System.out.print((char)c);
                c = fr.read(); //从文件中继续读取数据
            }
            fr.close();
        }
        catch(IOException e) //捕获文件输入输出异常
        {
            System.out.println(e); //显示文件输入输出异常
        }
    }

    public static void main(String[] args)
    {
        JException je = new JException();
        System.out.println("程序正常结束！\n");
    }
}
```

```

}
}

```

上述 Java 程序的运行结果如图 1.3 所示，该程序会提示找不到该文件的异常。

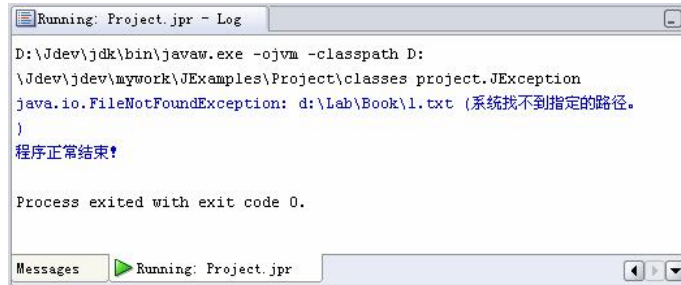


图 1.3 Java 程序提示找不到文件的输入输出异常

对于 C++程序，在 Visual C++ 6.0 开发环境中其源代码如下：

```

#include "stdafx.h"
int main(int argc, char* argv[])
{
    CFile file;
    char pbuf[100];
    file.Open("d:\\Lab\\Book\\1.txt", CFile::modeRead);
    file.Read(pbuf,100);
    printf(pbuf);
    file.Close();
    printf("程序正常结束! \n");
    return 0;
}

```

上述 VC 程序的运行结果如图 1.4 所示，该程序不会提示输入输出异常信息，而是出现错误，导致程序意外中止。



图 1.4 VC 程序不提示输入输出异常

(3) Java 采用更彻底的多线程编程技术。

有人说，C++语言的多线程编程是虚拟的，而 Java 语言的多线程编程是切实的。总之，Java

语言比 C++ 语言采用了更彻底的多线程编程技术，能在一个程序中同时执行多个任务。

(4) Java 在其他语法上进行了简化。

在 C/C++ 程序中，可以在所有类的外部定义全局变量，但在 Java 程序中只能在类中定义公用的、静态的变量来实现全局变量的作用。

在 C/C++ 程序中可以使用常导致程序复杂、出错的 goto 语句，但在 Java 程序中不能使用 goto 语句，并增加了异常处理语句 try、catch、finally、throw 等。

在 C/C++ 程序中，不同平台上的编译器对相同的基本数据类型 int、float 等支持不同字节长度，这就导致了代码的不可移植性；但在 Java 程序中，对这些数据类型进行了统一支持，保证了 Java 程序的平台无关性。

在 C/C++ 程序中使用头文件来声明类的原型、全局变量、库函数等，但在 Java 程序中不支持头文件，类的封装性更加完整，通过 import 语句实现与其他类的通信。

C/C++ 程序中使用结构和联合类型，但在 Java 程序中不支持结构和联合类型。

1.1.3 Java 的优势

Sun 公司的强大武器是 Java 技术本身，同时，Sun 和合作伙伴达成紧密合作，极大地支持软件开发团队。Sun 与联通宣布结成联盟，这是联通看到了 Java 在手机市场的专有性和 Java 具有的开放性和标准性。Java 欢迎竞争，因为只有这样才能最大限度地提高领域内的技术。

很多的编程语言在发展中并不是消失，而是转移到了其他领域中去，而 Java 的经久不衰，取决于 Java 的技术基础，如果问程序员为什么会选择 Java，他会说 Java 提供了多种功能，提供了方便的平台，是个足以吸引人的工具。推动 Java 最主要的因素是网络，Java 是以网络应用为基础的开发工具，这是它的长处。

在 PC 领域 Java 决不是弱者，在 PC 领域 Java 有很多应用，实际上 Java 应用很广泛，比如说人工智能游戏。在其他大的领域，Java 更是应用广泛，例如在汽车，铁路机车上的即时控制系统，Java 也广泛应用于军用方面。

J2ME 的规格在不断地发展，而不断增长的手机性能质量更是推动了 Java 语言在手机应用上的迅猛发展。在两年前，当时手机比的性能今天要差得远。现在用手机可以做的工作变得越来越有趣，这个领域有非常光明、非常令人振奋的前景。

Java 语言最大的优势在于，Java 是集成了许多人努力而创建的强有力的工具，可以提供很多功能。Java 满足网络服务标准，具有很强的安全性。Sun 花了很大的力气在解决和微软的互操作性上，遵循国际网络业的标准与微软工具进行交流，开发了大量相关工具。而 Sun 的 Web 服务未来将向扩大 Web 应用能力，扩大通用性和边缘应用上发展。

正如 Java 之父 James Gosling 所描述的：“当看到 Java 的客户通过 Java 完成了很多神奇的工作，像看到夏威夷火山上的观测台使用 Java 控制望远镜，看到荷兰健康医疗组织使用 Java 解决了保护隐私问题等等，那真是一种奇妙的感觉。”

学习 Java 编程的第 1 步是下载免费的 Java 开发工具 JDK，JDK 是 Java 编程世界的基础核心，该工具包括 Java 运行环境 JRE (Java Runtime Environment)、一套 Java 编译、测试工具和 Java 基础类库。JDeveloper 等 Java 开发工具都包括了某个版本的 JDK，目前 Sun 公司最新版本的 JDK 工具是 JDK 5.0，Java 运行环境 JRE 的最新版本是 1.5.0_07，下载地址为：http://www.java.com/zh_CN/download/windows_ie.jsp。

JDK 工具包是一种可执行的安装程序，默认安装路径为 C:\Program Files\Java，如图 1.5 所示。



图 1.5 安装 Java 2 运行环境

在图 1.5 中，双击 Sun 公司的 Java 运行环境 JRE 安装程序，启动了安装向导。遵循向导的操作步骤，就能顺利地安装该 Java 运行环境。

JDK 的编译、测试等工具主要包括 `java.exe`、`javac.exe`、`jar.exe` 和 `javadoc.exe` 可执行程序，其中 `java.exe` 工具用来启动 Java 虚拟机并执行 Java 类文件；`javac.exe` 工具用来编译 Java 源程序，生成 Java 类文件；`jar.exe` 工具用来对 Java 程序打包；`javadoc.exe` 工具用来生成 Java 文档。

学习 Java 语言的第 2 步就是熟悉 Java 开发工具的帮助文档，可以是在线的，即连接到 Sun 公司的文档中心，如图 1.6 所示；也可以是离线的，需要将帮助文档下载到本地。

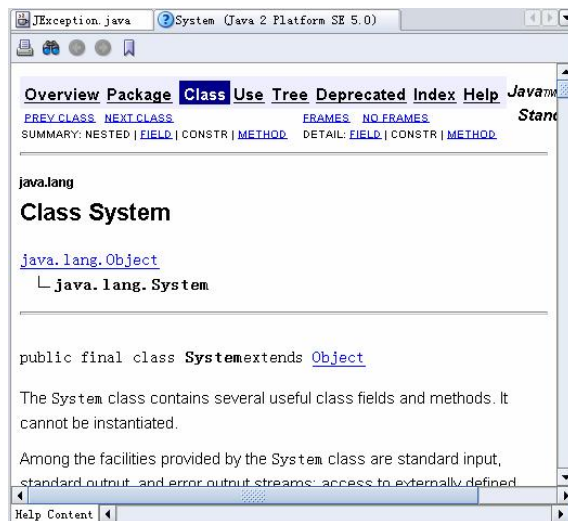


图 1.6 Java 帮助文档的在线版

在图 1.6 中，Java 2 平台的帮助系统包括概要（Overview）、包（Package）、类（Class）、用法（Use）、树（Tree）、Deprecated、索引（Index）和帮助（Help）几个栏目，类栏目显示

各个类的名称、层次关系。例如，System 类位于 java.lang 包中，是从 Object 类扩展而来的。System 类有一些有用的类域和方法，但它不能进行实例化。

Java 程序员进阶的第 3 步就是熟悉 Java 程序的应用需求，主要包括 Java 桌面应用、Web 环境的 Java 应用、Java 企业级应用、Java 嵌入式应用和 Java 并行应用，如图 1.7 所示。

对于 Java 桌面应用，仅仅需要 Java 运行环境的支持就足够了。

对于 Web 环境的 Java 应用，Java 支持的 Web 应用需要安装 JDK、应用服务器和数据库，因为 Web 应用分为 3 层。

在图 1.7 中，JDeveloper 开发工具以 Java 企业级应用为最基本的框架，Java 桌面应用和 Web 环境的 Java 应用只是该框架的一部分。

Web 应用的第 1 层是浏览器层，需要 Java Applet 支持，显示用户页面；第 2 层是 Web 服务器层，运行 Servlet 和 JSP 程序，有时需要访问数据库；第 3 层是数据库层，为 Java 程序提供后台数据库支持。

Java 企业级应用在 3 层 Web 应用结构的基础上又增加了 EJB 层等，EJB 层用来运行 EJB 程序，实现商务逻辑。

Java 嵌入式应用主要是指用于无线设备的嵌入式程序，需要使用嵌入式 Java 专用虚拟机。

Java 并行应用主要利用 Java 语言的多线程技术构建并行计算环境，需要在并行计算机或网格计算平台上运行。

Java 程序员升华的第 4 步就是多总结、多领悟，尽管编程不能全靠灵感，但是灵感确实对最优编程十分重要。灵感来源于平时的积累，这种积累不仅在于知识和经验的积累，还在于对经验和教训的总结和领悟，直觉真是很重要的。当 Java 程序员忘记了所有学到的套路，脑海中只有自己领悟的章法和秘诀时，该程序员的编程水平可以说是游刃有余了。

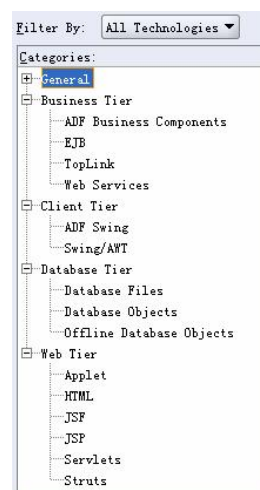


图 1.7 Java 应用开发的类型

1.2 Web 卓越工程师学习 JSP 的理由

JSP 是 Java Server Pages 的缩写，是 Sun 公司和其他公司一起建立和发布的动态网页技术标准，现已成为动态 Web 应用开发的主流开发技术之一。究其本质，JSP 是以 Java 平台为基础、以 Java 语言作为主要动态脚本语言的集成技术，其 Java 程序片段 (Scriptlet) 和 JSP 标记 (tag) 嵌入到传统的静态网页 HTML 代码中，形成动态的网页 (后缀名为 .jsp)。

1.2.1 JSP 与 HTML、ASP 和 PHP 比较的优势

如前所述，HTML 语言是基本的网页标记语言，单纯的 HTML 代码只能构成静态网页。静态网页缺乏交互性，只能显示某一时刻、某一对象的固定信息，其信息内容不会随着时间推动和对象变换而适应性变化，因而其应用十分局限，或者说只是网页基础。JSP 语言正是为了弥补动态显示内容的不足而设计的，其适应性、变换灵活性和安全性是 HTML 语言不能企及的。

ASP 是微软研发的动态网页设计技术，其脚本语言主要是 VBScript 和 JScript。VBScript 和 Jscript 程序简单、易学，因而 ASP 技术简便，有效，适用于快速开发动态网站应用。ASP 程序文件无需编译，直接在服务器上按照解释方式运行，用户的浏览器仅显示以 HTML 格式网页返回的运行结果，因此 ASP 程序运行与浏览器无关，只需浏览器支持 HTML 即可，也具有一定的安全性。如果 ASP 使用插件，就要求用户的浏览器支持这种插件。但是，由于微软产品的漏洞较多，黑客常常予以攻击，ASP 的安全性不及 JSP。另外，由于 ASP 每次都要按照解释方式运行代码，但 JSP 只在第一次运行程序执行编译，二次及以后的运行不需要再次编译，因而 ASP 二次的运行效率比 JSP 低得多。Java 的不限平台可移植性也让 JSP 在兼容性和可移植性方面胜于 ASP，特别是 ASP 仅限于微软操作系统的 IIS 支持下运转。

PHP 是市场份额仅次于 JSP 和 ASP 的动态网页开发技术，其兼容性和可移植性与 JSP 相当，不过需要安装和配置 PHP 服务器基础软件。PHP 语言便捷、简单易学，广泛支持多种数据库，支持面向对象的编程技术，其源代码完全公开免费。但是，JSP 与平台无关的特性也是 PHP 无法相比的，另外 JSP 的二次及以后运行无需编译，比 PHP 效率更高。

从标记形式来看，JSP 和 ASP 都是以<%开始，以%>结束，嵌入动态网页编程代码。PHP 以<?开始，以?>结束，嵌入 Perl、C 等代码。

由上可见，JSP 在与平台的无关性、代码执行方式和二次高效、支持的数据库连接、安全性、可重用性等方面具有较大优势，因此本书以 JSP 为主要工具介绍 Web 卓越工程师的编程案例。

1.2.2 JSP 的安装与配置

JSP 服务器安装之前，需要安装 Java 开发包 JDK，例如 1.1.3 小节介绍的 Java 2 JDK 和 JDeveloper 的安装。然后，可以安装 Tomcat 6.0，这是一种支持 JSP 编程的免费 Web 服务器软件，可从网站 <http://tomcat.apache.org/> 下载。JDK+Tomcat+JSP 是最流行的 JSP 开发模式之一，本书以 Tomcat 6.0 作为 Web 服务器。Tomcat 6.0 下载后得到安装文件，双击此安装文件，开始 Tomcat 6.0 的安装过程，如图 1.8 所示。



图 1.8 开始安装 Tomcat 6.0

然后，要签署 Tomcat 6.0 软件的使用协议，如图 1.9 所示。当然，只有单击【I Agree】按钮接受协议时，才会继续安装此软件。

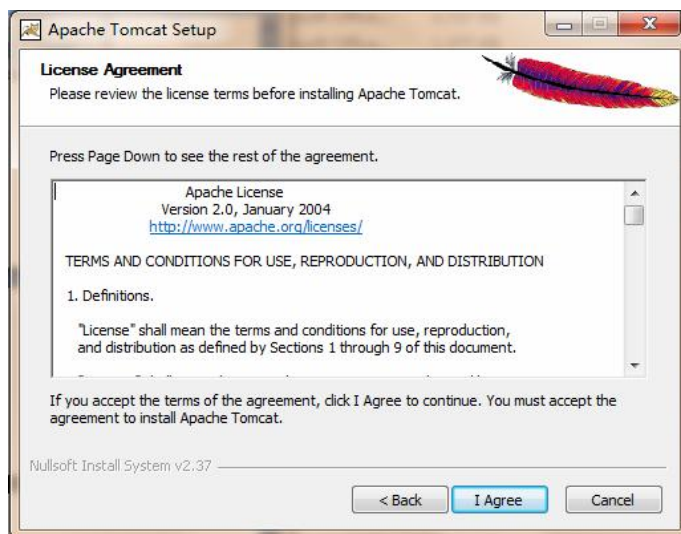


图 1.9 签署 Tomcat 6.0 软件的使用协议

接着，选择 Full（完全）安装模式，如图 1.10 所示。完全安装模式会安装以下主要组件：
①Tomcat 主软件；②开始菜单选项；③文档；④示例。

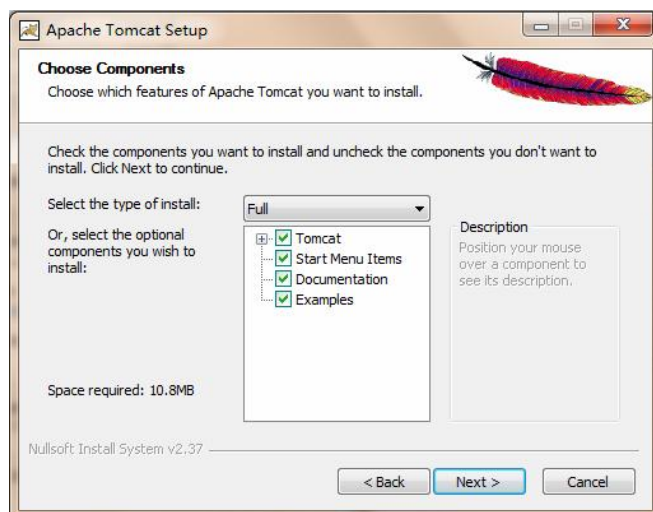


图 1.10 选择 Tomcat 6.0 软件的 Full（完全）安装模式

接下来，将其安装目录不妨设为 D:\tomcat6，即 Web 服务器的根目录，如图 1.11 所示。随后，就按照安装向导的提示完成安装过程即可。其配置过程是自动完成的，之后安装程序会自动启动 Tomcat 6.0。

为测试是否能够成功运行 JSP 程序，在 IE 浏览器的“地址”栏中输入下列地址：
http://localhost:8080/（或 http://127.0.0.1:8080/）

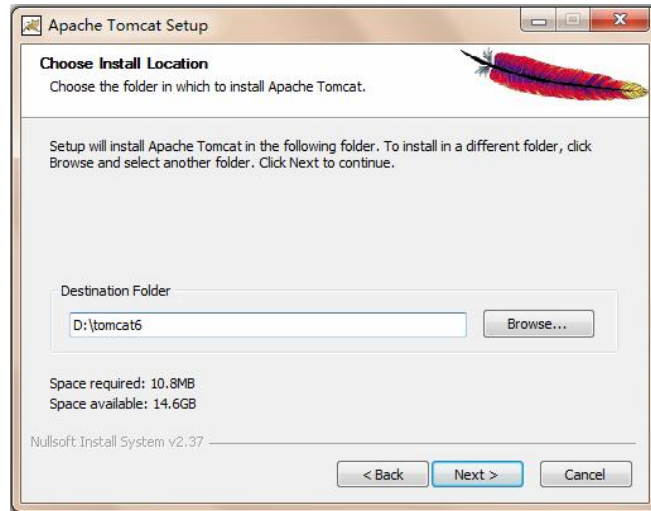


图 1.11 设定 Tomcat 6.0 软件安装的目标目录

当浏览器打开如图 1.12 所示的页面时，表示 Tomcat 6.0 成功启动，并且 JSP 程序也可成功运行。至此，JSP 开发环境已经成功被创建和配置。

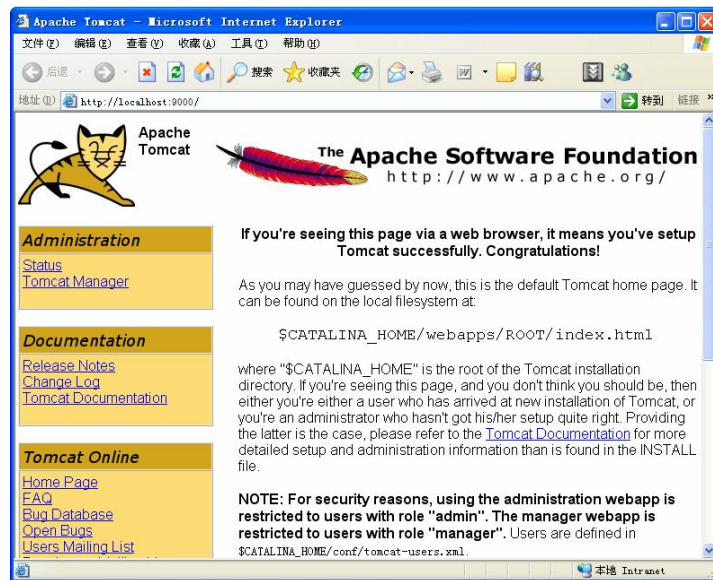


图 1.12 成功运行 Tomcat 6.0 的 JSP 程序

注意，127.0.0.1 为默认的、表示本机的特殊 IP 地址。8080 为 Tomcat 6.0 默认使用的端口号；如果需要的话，可以打开 D:\tomcat6\conf 目录下的文件 server.xml，并将 port 的值设置为需要的值（但不能与已有的端口号相冲突），如 9000。

```
<Connector port="9000" protocol="HTTP/1.1" maxThreads="150"
connectionTimeout="20000" redirectPort="8443" />
```

这样，保存 server.xml 并重新启动 Tomcat 6.0 后，输入下列地址，打开如图 1.12 的页面。
http://localhost:9000/（或 http://127.0.0.1:9000/）

1.3 Web 卓越工程师的成长之路

当认定 Web 卓越工程师作为主攻方向后，就要选择一条适合自己的 Web 卓越工程师成长之路，这样才能实现自己追求的 Web 卓越工程师价值。Web 卓越工程师之路是一条酸甜苦辣兼备的磨练之路，战胜了这条道路上的一个个难关就能充分享受 Web 卓越工程师的成就感和光明的“钱”途。

所谓万事开头难，Web 卓越工程师的成长之路就从第一个 Java 程序的学习开启，第 2 章将介绍 Java 编程开门的方法。之后，作为老练的程序员，建议在项目研发的参与中抓紧时间积极学习 Web 编程技术，利用一切时机积累项目开发经验，总结编程 BUG 调试的经验教训，改进编程习惯，优化 Web 网站结构，最后成为能在项目中独当一面或主持项目的 Web 卓越工程师。总之，项目是 Web 卓越工程师的孵化器，也是其茁壮成长的催化酶。

(1) 定义 Web 项目需求问题。

Web 卓越工程师是与政府、企事业单位的 Web 相关项目开发需求密切相关的，可以说是互联网时代应运而生的新职业。定义项目需求问题，是 Web 卓越工程师首先要学会做并擅长做的事。只有满足项目需求的 Web 编程，才是有效的、有价值的 Web 工程，否则可能会返工或以失败告终。

(2) 积极参与 Web 项目研发。

Web 卓越工程师的成长之路也是不断积累工程经验的过程，其工程经验的积累离不开 Web 研发项目的参与和主持。学习 Java 语言和 JSP 编程技术也是这样，看书一个月不如参与实际的 Web 研发项目一个月。因此，为了追求快而有效地成长为 Web 卓越工程师，必须与 Web 研发项目密切结合起来，全身心地投入项目研发的分工与协作中去，多总结多提升。

(3) Web 技术既要扎实学好基础，也要活学活用。

使用 Web 技术时，最好学扎实 Web 技术的基础知识。例如，一个刚入门的 Web 初学者正在编写第一个 JSP 实现的 Web 网页，可是浏览器打开该 JSP 网页时报错。当时，此 Web 初学者只知道这是一个 JSP 应用程序，因此认为一定是 JSP 程序代码的问题。实际上，这种判断是错误的。实际上是 Tomcat 服务器关闭了，没有启动。因此，重新启动 Tomcat 服务器后，该 JSP 网页程序就能在浏览器上运行了。

这个例子的启示是，如果 Web 卓越工程师在编写 Web 应用程序，那么他具备一些 Web 技术相关的基础知识才好。尽管 Web 卓越工程师不需要理解 Web 服务器的内部工作，但他的确需要主动了解一些 Web 服务器基础知识，以便能有效地在相关的技术领域内进行交流，并解决问题。程序调试是个技术活，有时也需要实践中积累灵感和经验，活学活用才是 Web 卓越工程师成长的王道。

(4) 多磨练 Web 项目开发技能。

无论 Web 程序员是生疏的新手，还是经验丰富的老手，都一定要多磨练 Web 项目开发技能，因为 Web 技术的知识和其他计算机知识一样容易折旧。Web 网络世界总是从一个版本到另一个版本不停地变化，Web 卓越工程师必须不断更新和精炼其技能。因为现在可能运行着无数种 Web 应用平台，所以不可能指望一个 Web 项目给所有的 Web 卓越工程师提供所有的 Web 技术知识和锻炼机遇。作为应用程序开发人员，通过自学、课堂也许还有认证，继续 Web

技术教育是 Web 卓越工程师的责任。在这个繁忙的世界中, Web 卓越工程师不能把所有的 Web 数据交互需要都依靠 DBA, Web 卓越工程师能做得越多, 越能自己应付 Web 技术应用需求, Web 卓越工程师就越能成为更好的应用程序支持专业人员。

总之, Web 卓越工程师应该抓住一切可以发展的机会和项目资源, 关键是 Web 卓越工程师需要确保自己尽一切努力成为更强大的 Web 技术专业人员。

1.4 小结

从 Java 和 JSP 的发展趋势和特点看, Web 卓越工程师可以依靠先进的、开放的和蒸蒸日上的 Java 技术和 JSP 技术打出一片编程天地, 未来是属于 Web 卓越工程师的。

从 Web 卓越工程师的成长之路看, Web 卓越工程师和其他程序员一样需要付出和不断努力, 循序渐进的学习是 Web 卓越工程师成长的必经之路。

以“华容道”手机游戏 Java 程序为实例, Java 编程是一门十分有趣而应用广泛的技术和艺术, 需要灵感, 需要实践, 更需要应用。第 3 章将会对此实例详细分析和设计, 验证 Java 游戏应用开发的规律和方法。