

第二章 第一个例子：猜数字

学习一项新技术，最好的入门方法是参照一个完整的例子，从头到尾做一遍。本章为读者设计了一个非常简单的入门例子，使用 JBuilder2005 做为开发工具，将开发 Struts 程序的整个过程展现给读者。为了提高读者的兴趣，笔者选择了猜数字这个游戏做为本章的主线。

猜数字就是系统随机生成一个整数，用户通过不断和程序沟通最终猜出准确的数字。用户每一次向程序提交自己输入的数字后，程序会反馈信息给用户，信息包括“猜中”，“太大了”，“太小了”三种。

在开始这个例程前，先向读者交待一下 Struts 的工作流程，算是给读者一个大体上的印象，具体的细节在后面的章节中会说明。

2.1 Struts的业务流程

Struts 的业务流程是基于 Model2 的流程基础上的。下面来粗略的了解一下它，为能正确理解后面的例子做一个铺垫。假设现在有以下情景：用户正在一个用 Struts 技术构建的网站的主页浏览，主页上有一个登录表单，用户填好登录名和密码，单击登录按钮。该点击激活了以下一系列过程：

(1) 用户的请求以 HTTP 方式传输到服务器上，接收请求的是 ActionServlet。

(2) ActionServlet 接收到请求后，会查找 struts-config.xml 文件来确定服务器上是否有用户请求的操作，此处用户请求的操作应为登录操作。如果没有，则返回一个用户请求无效的出错信息。

(3) 当 ActionServlet 找到用户请求的 Action 后，首先将用户输入的表单参数打包成一个 ActionForm 对象，这个 ActionForm 对象其实也就是一个 JavaBean，里面包含两个字段分别是用户名和密码。接着 ActionServlet 再根据 struts-config.xml 中的配置信息决定是否要执行 ActionForm 对象中的 Validate 方法。若 Validate 方法执行有错，则返回，否则，继续下一步。

(4) 系统生成一个用户所请求的 Action 的实例对象，将前面的 ActionForm 对象传递给它，运行它的 execute()方法。这一步其实就是用户登录的控制器，在执行 execute()方法时可以调用后台模型验证登录名和密码是否正确等等信息。

(5) execute()执行结束前会生成一个 ActionForward 类型的对象并将之返回给 ActionServlet，该对象的作用是告诉 ActionServlet 下一步应该跳转到哪里，假如后台模型检验用户名和密码正确，则 ActionForward 就代表跳转到一个登录成功的界面。ActionServlet 将对之进行分析，其实就相当于接收到一个新的请求，重复 2 至 5 的过程，直到将某个界面返回用户为止。

以上就是 Struts 的基本工作流程，可以看出 struts-config.xml 在整个流程中起到了一个类似站点地图的作用，它记录了所有可能的请求跳转。其实，在 Web 容器加载 Struts 应用程序后，struts-config.xml 文件就被首先读入内存成为了一个 ActionMapping 对象，前面所

说的查找 struts-config.xml 文件，实际上是查找 ActionMapping 对象。对于初学者来说可以不必深究其中的细节，后面的章节将会详细说明该文件。

2.2 快速开始

本节包括“猜数字”这个程序的具体开发过程，开发工具选用的是 JBuilder2005。指导读者一步步尝试 Struts 应用开发的基本操作，理解基本的 Struts 应用结构。

2.2.1 JBuilder2005的配置

JBuilder2005 是 Borland 公司开发的一套 Java IDE 工具，它的强大功能在这里就不多说了，相信所有使用 IDE 进行 Java 程序开发的程序员们应该都听说过它。它的安装很简单，只需按照默认的步骤就可以完成。以下主要说明如何正确配置它来完成“猜数字”这个例子。

1. 配置JDK

启动 JBuilder，在 JBuilder 主界面单击【Project】|【Default Project Properties】命令菜单就会出现配置默认工程属性的对话框，如图 2-1 所示。



图2-1 工程属性对话框

在左边的滚动窗口中选中“Path”项，右边出现“Path”的属性页，点击“JDK”一项右边的按钮，出现如图 2-2 所示界面。



图2-2 选择JDK

此处选择 JBuilder 自带的 JDK 1.4.2_04-b05 版本，单击【OK】按钮选中。

2. 配置Web服务器

在 JBuilder 主界面单击【Project】|【Default Project Properties】命令菜单，在左边的滚动窗口中选择“Server”项，出现如图 2-3 所示的画面。



图2-3 配置server

在右边的页面中选择“Tomcat 5.0”作为“猜数字”的 Web 服务器。该 Tomcat 是 JBuilder2005 自带的，版本为 5.0.27。

2.2.2 创建工程和Web模块

当 JBuilder 配置完毕后，就可以开始创建猜数字 Web 应用程序了。

(1) 第一步的工作是创建一个工程，点击 JBuilder 主窗口的【File】|【New Project】，将会出现如图 2-4 所示界面。该界面用于给工程命名，并选择工程的存放目录。给工程命名为“ch2”，路径可以按照自己的情况设置。单击【Next】按钮。



图2-4 创建工程第一步

(2) 这一步主要是设置工程的输出路径和一些附属信息，均按默认值设置。以后如果还有需要，可以在菜单的【Project】|【Project Properties】中再设置这些信息。此时单击【Finish】按钮完成创建。

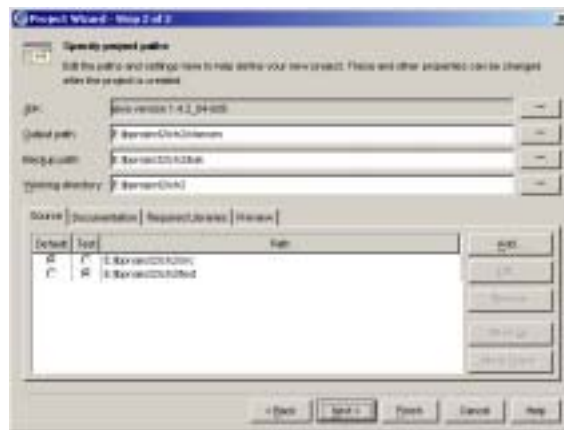


图2-5 创建工程第二步

(3) 在创建完工程后，可以看到 JBuilder 主界面左边的工程视图中出现了工程的图标，此时里面什么都没有。由于我们的猜数字应该是一个基于 Java Web 的应用程序，所以下面我们要创建一个 Web 模块。

单击【File】|【New】命令菜单，出现如图 2-6 所示的对话框，在左边的子窗口中选中“Web”子页，再在右边的窗口中选中“WebModule”，单击【OK】按钮。



图2-6 新建一个WebModule

(4) 系统会提示是否从已有的目录中导入一个 Web Module 还是创建一个新的 Web Module，此处选择“Create Empty Web Module”，并单击【Next】按钮。



图2-7 新建WebModule第一步

(5) 在“Name”文本框中输入“GuessNumber”为该 Web Module 命名，其他的按照默认设置，如图 2-8 所示。单击【Next】按钮。

(6) 在这一步中，勾选“Struts 1.1”复选框，表示要建一个 Struts 的 Web 应用程序。单击【Finish】按钮完成创建。此时，一个 Web 模块就建立好了，可以在主界面左边的工程窗口中看到它。



图2-8 新建WebModule第二步

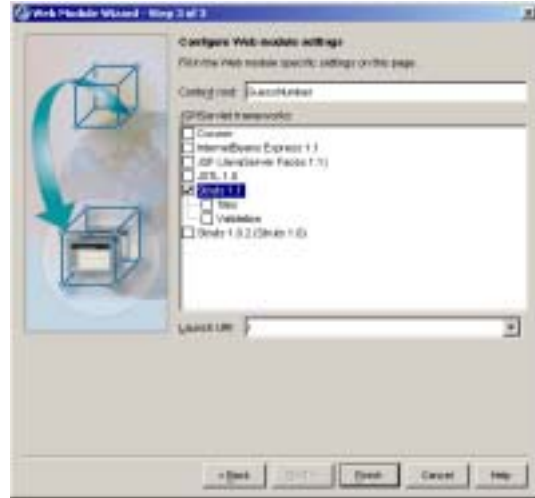


图2-9 新建WebModule第三步

2.2.3 CreateNumberActionForm和createNumberAction

下面就要开始创建应用程序的各种对象了。选择的顺序是先创建 ActionForm，然后是 Action，最后创建与之相关的 JSP 页面。这里一共需要创建两个 ActionForm 和两个 Action，第一个 ActionForm 的作用是用于向后台的创建随机数的 Action 提交你所猜的数字在什么范围内，第二个 ActionForm 的作用就是向后台判断是否猜对的 Action 提交所猜的数字。首先来创建 CreateNumberActionForm.java。

1. 创建CreateNumberActionForm

(1) 单击【File】|【New】命令菜单，出现如图 2-10 所示的对话框，在左边的子窗口中选中“Web”子页，再在右边的窗口中选中“ActionForm”项，单击【OK】按钮。



图2-10 新建一个ActionForm

(2) 在其中的“ActionForm”项的文本框中输入“GuessNumberActionForm”，以此作为 ActionForm 的类名。在“Package”项的文本框中输入“ch2”。单击【Next】按钮。

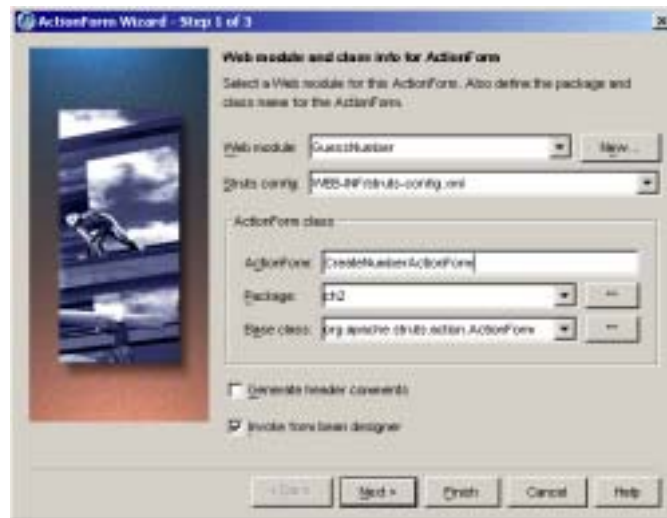


图2-11 新建ActionForm第一步

(3) 这一步是向 ActionForm 中填加域，可以通过单击【Add】按钮进行添加。本例中添加一个叫做 digits 的域，类型为 java.lang.String，它的作用是传递用户所选择的随机数生成范围。当添加好域后，点击【Next】按钮。

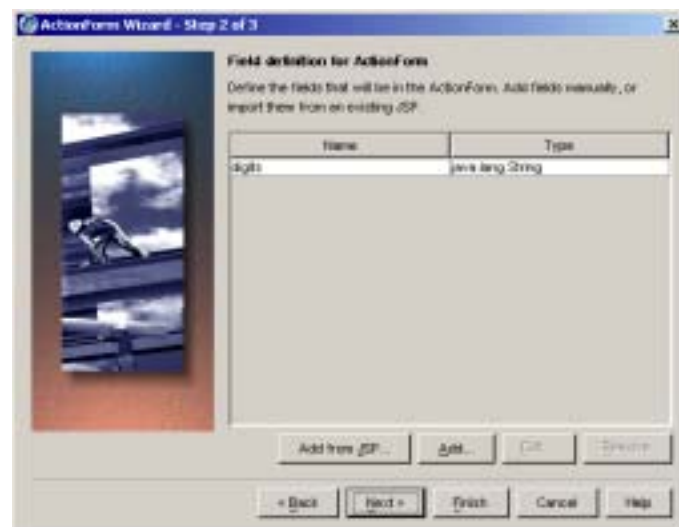


图2-12 新建ActionForm第二步

(4) 这一步是确定 ActionForm 的名称，并将之添加到 struts-config.xml 配置文件中去。如图 2-13 所示，在“FormBean Name”项中填入“createNumberActionForm”，然后点击【Finish】按钮，就创建了 ActionForm。

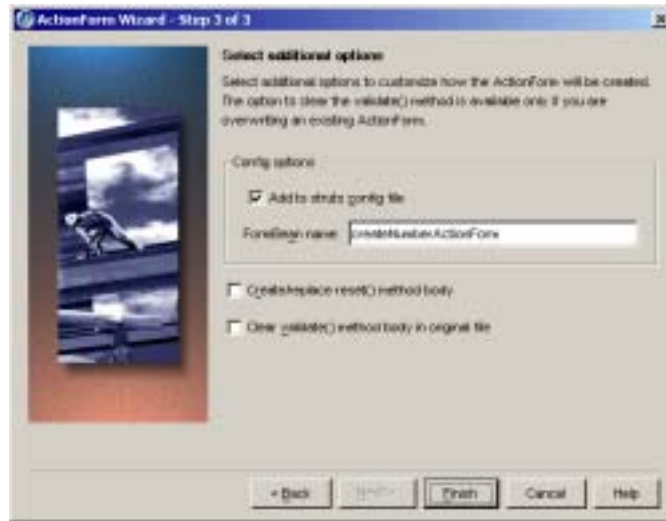


图2-13 新建ActionForm第三步

现在，打开这个 ActionForm 文件，代码如下所示：

代码 2.1

```
package ch2;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionMapping;
import javax.servlet.http.HttpServletRequest;

public class CreateNumberActionForm extends ActionForm {
    //此处就是刚才添加的digits域
    private String digits;
    //get方法
    public String getDigits() {
        return digits;
    }
    //set方法
    public void setDigits(String digits) {
        this.digits = digits;
    }
    //验证方法，未使用
    public ActionErrors validate(ActionMapping actionMapping,
        HttpServletRequest httpRequest) {
        return null;
    }
    //重设方法，未使用
    public void reset(ActionMapping actionMapping,
        HttpServletRequest servletRequest) {
    }
}
```

可以看到，在 ActionForm 类的内部还有两个 Validate 方法和 reset 方法，有关这两个方

法的详解放在后面说明。

2. 创建CreateNumberAction

创建好 CreateNumberActionForm 后，就可以创建 Action 了。

(1) 单击【File】|【New】命令菜单，选中“Web”子页，再选择“Action”。单击【Next】按钮。



图2-14 新建一个Action

(2) 这一步主要是提示输入所要建立的 action 的类名称和包名。此处为 Action 命名为“createNumberAction”，包名“ch2”。单击【Next】按钮。



图2-15 新建Action第一步

(3) 这一步是定义 Action 的一些属性，其中“Action path”项使用默认的“/createNumberAction”，“FormBean Name”项就是刚才创建的 ActionForm 的名称“createNumberActionForm”（注意，这不是类名。此处的名称第一个字母是小写，与前面

的 ActionForm 的名称相对应。), scope 选择 “ session ”, Validate FormBean 选择 “ false ”, 然后点击【Finish】按钮, 这个 Action 就创建完成。

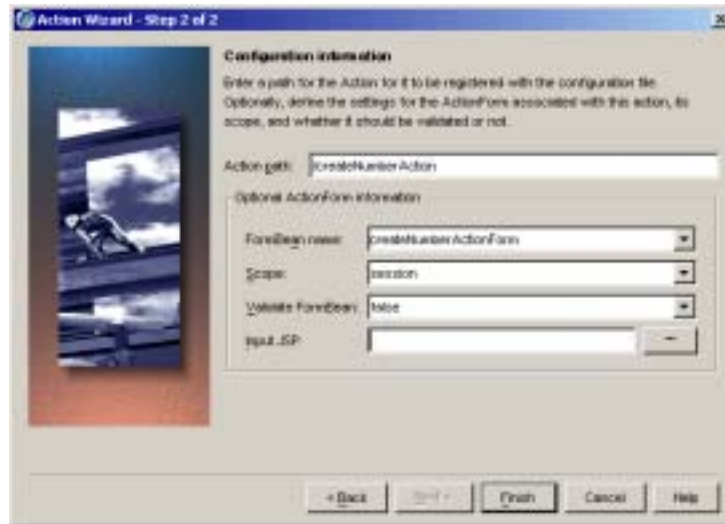


图2-16 新建Action第二步

现在可以在左边的工程视图中看到它。此时, 这个类还什么功能也没有, 无法随机生成一个整数。为其添加代码, 使其具有随机生成整数的功能, 并能完成相关的操作, 代码如 2.2 所示:

代码 2.2

```
package ch2;
import org.apache.struts.action.*;
import javax.servlet.http.*;
public class createNumberAction extends Action {
    public ActionForward execute(ActionMapping mapping,
        ActionForm actionForm,
        HttpServletRequest servletRequest,
        HttpServletResponse servletResponse) {
        /**
         * 此处的CreateNumberActionForm就是前面创建的ActionForm
         * 在Action中首先将它获取, 后面就可以获取它所传的值
         */
        CreateNumberActionForm cform = (CreateNumberActionForm)actionForm;

        //获取用户session
        HttpSession session = servletRequest.getSession();
        //去除会话中的message属性
        session.removeAttribute("message");
        //从ActionForm中获取表单传来的值, 并将之转为int型
        int digits = Integer.parseInt(cform.getDigits());
        //随机生成一个范围在0到digits之间的整数
        int secret_number = (int)(Math.random()*digits);
```

```

        //把随机生成的数放进用户session中
        session.setAttribute("secret_number",String.valueOf(secret_number
    ));
        //返回猜数字的页面
        return mapping.findForward("guess");
    }
}

```

2.2.4 JudgeActionForm和judgeAction

接下来要创建处理用户所猜数字是否正确的的 ActionForm 和 Action，分别为 JudgeActionForm 和 judgeAction。其中 JudgeActionForm 中传递的值为用户所猜的数字，judgeAction 中根据用户所猜的数字与存在 session 中的数字的比较，返回一个“message”标记存于 session 中。

具体的创建步骤不再赘述，与前面基本相同。值得注意的是在创建 JudgeActionForm 时，需通过“add”按钮添加一个域叫做“guess_number”，该域类型为 java.lang.String。然后，为该 ActionForm 取名为“judgeActionForm”。在创建 judgeAction 时将“judgeActionForm”这个名称当作对应的 FormBean 的名称，scope 选择“session”，Validate FormBean 选择“false”。

下面给出 judgeAction 的代码如 2.3 所示：

代码 2.3

```

package ch2;
import org.apache.struts.action.*;
import javax.servlet.http.*;

public class judgeAction extends Action {
    public ActionForward execute(ActionMapping mapping,
                                ActionForm actionForm,
                                HttpServletRequest servletRequest,
                                HttpServletResponse servletResponse) {
        /**
         * 获取传入的ActionForm
         */
        JudgeActionForm jForm = (JudgeActionForm) actionForm;
        //获取用户session
        HttpSession session = servletRequest.getSession();
        //从传入的ActionForm中得到用户所猜的数字，并转为int型
        int guess_number = Integer.parseInt(jForm.getGuess_number());
        //从session中获取随机生成的数字
        int secret_number =
Integer.parseInt((String)session.getAttribute("secret_number"));
        //比较用户输入的数字和随机生成的数字
        if (guess_number == secret_number){
            //若猜中，则在session中的“message”属性放入“equal”标记
            session.setAttribute("message","equal");
        }else if (guess_number > secret_number){
            //若太大，则在session中的“message”属性放入“bigger”标记

```

```

        session.setAttribute("message","bigger");
    }else{
        //若太小,则在session中的“message”属性放入“small”标记
        session.setAttribute("message","smaller");
    }

    //返回猜数字的页面
    return mapping.findForward("guess");
}
}

```

2.2.5 Struts-config.xml 文件

在 createNumberAction 和 judgeAction 的最后一步都有一句

```
return mapping.findForward("guess");
```

这句代码的作用是转发到猜数字的页面。那么,系统是如何从“guess”这样一个参数获取要转发的页面的信息呢?

这就是 struts-config.xml 要做的事。在前面已经说过, struts-config.xml 就像一个站点地图,主控 Servlet 通过查找它来获取程序转发规则。

打开位于 WEB-INF 目录下的 struts-config.xml 文件,可以看到前面建立好的 Action 和 ActionForm 都已经被写入了该文件。此处只需添加一个 Global-forward 标记就可以实现转发, struts-config.xml 的完整代码如 2.4 所示。

代码 2.4

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD Struts
Configuration 1.1//EN"
"http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">
<struts-config>
<!-- 此处定义了ActionForm的信息 -->
<form-beans>
<form-bean name="judgeActionForm" type="ch2.JudgeActionForm" />
<form-bean name="createNumberActionForm"
type="ch2.CreateNumberActionForm" />
</form-beans>
<!-- 此处定义了转发信息 -->
<global-forwards>
<forward name="guess" path="/guess.jsp" />
</global-forwards>
<!-- 此处定义了Action的信息 -->
<action-mappings>
<action name="judgeActionForm" path="/judgeAction" scope="session"
type="ch2.judgeAction" />
<action name="createNumberActionForm" path="/createNumberAction"
scope="session" type="ch2.createNumberAction"/>
</action-mappings>
</struts-config>

```

2.2.6 创建guess.jsp

下面创建猜数字的页面 guess.jsp。单击【File】|【New】命令菜单，选择 Web 子页，再选择“JSP”，出现如图 2-17 所示画面。



图2-17 创建JSP页面

在其中 Name 项中输入 JSP 页面的名字 guess，然后直接点完成就可以了。下面给出 JSP 页面的代码。

代码 2.5

```
<!-- 此处包含两个struts标记库，分别是html和bean标记库 -->
<% taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<% taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<% page contentType="text/html; charset=GB2312" %>
<html:html>
<head>
  <title>
    guess.jsp
  </title>
</head>
<body>
<h1><bean:message key="page" /></h1>
<!-- 第一个表单，用于向createNumberAction提交随机数的生成范围 -->
<html:form action="/createNumberAction.do" method="POST">
  <h3><bean:message key="page.choose" /></h3>
<!-- 此处的属性和CreateNumberActionForm中的属性一样都是digits -->
<html:select property="digits">
  <html:option value="10">10</html:option>
  <html:option value="100">100</html:option>
  <html:option value="1000">1000</html:option>
</html:select>
<html:submit><bean:message key="page.new" /></html:submit><br>
</html:form>
```

```

<hr>
<!-- 第二个表单，用于向judgeAction提交用户猜的数字 -->
<html:form action="/judgeAction.do" method="POST">
<!-- 此处的属性名和JudgeActionForm中的属性一样都是guess_number -->
<html:text property="guess_number" value=""/>
<html:submit><bean:message key="page.guess"/></html:submit><br>
</html:form>
<hr>
<!-- 页面显示判断结果的部分 -->
<%
    String message = null;

    //从session中获取judgeAction存入的标记
    message = (String)session.getAttribute("message");
    if (message != null) {
        //若用户猜中了
        if (message.equals("equal")){
%>
            <h1><bean:message key="result.equal"/></h1>
<%
        }

        //若猜的数比实际的数大
        else if (message.equals("bigger")){
%>
            <h1><bean:message key="result.big"/></h1>
<%
        }
        //若猜的数比实际的数小
        else {
%>
            <h1><bean:message key="result.small"/></h1>
<%
        }
    }
%>
</body>
</html:html>

```

这个页面分为三个部分。第一部分是一个表单，该表单对应的 Action 是 createNumberAction，它的作用是传递给后台产生随机数的范围，并在后台生成一个新的随机数。第二部分还是一个表单，它接收用户的输入，并将用户所猜数字传到 judgeAction 进行判断。第三部分是一段 Java 程序片断，它从 session 属性中读出 judgeAction 存放的“message”标记，并根据标记的不同显示不同的提示。

在上述程序中，有许多地方用到了<bean:message key="">这样的标签。这个标签是属性 struts 的 bean 标签库中的标签，它的作用是显示一段文本，一般是和资源文件联合在一起使用的，其中的“key”属性指定了资源文件中的内容的标识。下面就看看为该页面创建的资源文件。

2.2.7 创建资源文件

JBuilder2005 在创建 Web Module 时自动创建了名为 ApplicationResources.properties 的资源文件，并在 web.xml 中配置了它。在源程序的根路径上可以看到这个文件，打开它，此时它是空的。

以下给出它的代码：

代码 2.6

```
result.equal = \u606d\u559c\u4f60\u5927\u7b54\u5b99\u4e86\u5927
result.small = \u592a\u5c0f\u4e86\u5927\u7b54\u5b99\u4e86\u5927
result.big = \u592a\u5927\u4e86\u5927\u7b54\u5b99\u4e86\u5927
page = \u731c\u6570\u5b57\u7684\u9875
page.choose = \u9009\u62e9\u6570\u5b57\u7684\u8303\u56f4
page.new = \u5f00\u59cb\u4e00\u6b21\u65b0\u7684\u731c\u6570
page.guess = \u731c\u731c\u770b
```

读者可能感到奇怪，这里的字符编码为什么不是直接用中文，而采用了上面看到的 Unicode 形式的编码。原因是 Java 考虑到国际化的需要，将其所有显示的代码都采用了 Unicode 形式编码。其原始文件如下所示。

代码 2.7

```
result.equal = 恭喜你，答对了！
result.small = 太小了，重新猜吧！
result.big = 太大了，重新猜吧！

page = 猜数字的页面
page.choose = 选择数字的范围
page.new = 开始一次新的猜数
page.guess = 猜猜看
```

前面的 guess.jsp 中就用到了这个资源文件内的编码。例如<bean:message key="page"/> 标签，它实际显示的就是“猜数字的页面”这样几个字，其他的依此类推。

2.3 运行结果

下面就可以运行该 Web 应用了。点击 JBuilder 主面板上的 Run 按钮，JBuilder 会在其内部打开 Web 服务器（此处打开的是 Tomcat），并加载该 Web 模块。

在浏览器中输入 <http://localhost:8080/GuessNumber/guess.jsp> 会看到如图 2-18 所示画面：选择随机生成数的范围为 10，点击“开始一次新的猜数”。此时系统就会随机生成一个数，用户就可以开始猜数字了。运行结果如下所示。

当所猜的数字比生成的数字小时，如图 2-19 所示。



图2-18 主页



图2-19 太小

当所猜的数字比生成的数字大时,如图 2-20 所示。当猜中生成的数字时,如图 2-21 所示。



图2-20 太大



图2-21 猜中了

2.4 本章小结

本章使用 JBuilder2005 开发了一个简单的小游戏。从这个程序中读者可以了解 Struts 应用程序设计中的几个关键组件 ActionForm, Action, JSP 以及资源文件的开发过程。当然,这里的开发过程并没有涉及到 Struts 全部的组件和功能,同时也没有讲太多的原理。我们的希望是读者能对使用一个开发工具构建 Struts 应用程序有一个最初的认识。相信读者按照我们提供的步骤和代码一步步的做就可以完成这个例子。后续各章将详细讲述 Struts 的各个组件以及它们的原理。