

第 1 章课后练习

☒ 选择题

1. 以下何者适合使用 GET 请求来发送？

- (A) 使用者名称、密码
- (B) 检视论譚页面
- (C) 信用卡资料
- (D) 查询数据的分页

答案：B、D

提示：敏感性数据不应使用 GET 来发送。论譚页面、数据分页可以使用 GET，便于使用者设定为书签。

2. 以下何者应该使用 POST 请求来发送？

- (A) 使用者名称、密码
- (B) 档案上传
- (C) 搜寻引擎的结果画面
- (D) BLOG 文件

答案：A、B、D

提示：敏感性数据、大量数据不应使用 GET 来发送。搜寻引擎结果画面可以使用 GET，便于使用者设定为书签。

3. HTTP 哪个请求方式，请求参数会出现在网址列上？

- (A) GET
- (B) POST

答：A

4. 以下何者适合使用 GET 请求来发送？

- (A) 检视静态页面
- (B) 查询商品数据
- (C) 新增商品资料
- (D) 删除商品数据

答案：A、B

提示：这题在考等幂操作，不改变服务器状态的查询应使用 GET。

5. 以下何者应该使用 POST 请求来发送?

- (A) 查询商品数据
- (B) 新增商品资料
- (C) 更新商品数据
- (D) 删除商品数据

答案: B、C、D

提示: 这题在考非等幂操作, 会改变服务器状态的查询应使用 POST。

6. 以下何者是属于客户端执行的程序?

- (A) JSP
- (B) JavaScript
- (C) Servlet
- (D) Applet

答案: B、D

7. Servlet/JSP 主要是属于哪个 Java 平台的规范之中?

- (A) Java SE
- (B) Java ME
- (C) Java EE

答: C

8. Servlet/JSP 必须基于哪一类型的容器才能提供服务?

- (A) Applet 容器
- (B) 应用程序客户端容器
- (C) Web 容器
- (D) EJB 容器

答: C

9. Web 容器在收到浏览器请求时, 会如何处理请求?

- (A) 使用单一执行绪处理所有请求
- (B) 一个请求就建立一个执行绪来处理请求
- (C) 一个请求就建立一个行程来处理请求
- (D) 一个请求就执行一个容器来处理请求

答: B

10. Java EE 中各技术标准最后将由什么文件明订规范？

(A) JCP

(B) JSR

(C) JDK

答：B

提示：JCP 是组织名称。JDK 是 Java 开发工具箱名称。

第 2 章课后练习

☒ 选择题

1. 若要针对 HTTP 请求撰写 Servlet 类别，以下何者是正确的作法？
 - (A) 实作 Servlet 界面
 - (B) 继承 GenericServlet
 - (C) 继承 HttpServlet
 - (D) 直接定义一个结尾名称为 Servlet 的类别

答案：C

2. 续上题，如何针对 HTTP 的 GET 请求进行处理与响应？
 - (A) 重新定义 service() 方法
 - (B) 重新定义 doGet() 方法
 - (C) 定义一个方法名称为 doService()
 - (D) 定义一个方法名称为 get()

答案：B

提示：不建议也不应重新定义 HttpServlet 所继承下来的 service() 方法。

3. HttpServlet 是定义在哪个套件之中？
 - (A) javax.servlet
 - (B) javax.servlet.http
 - (C) java.http
 - (D) javax.http

答案：C

提示：与 Servlet 定义相关的类别或接口都位于 javax.servlet 套件之中，像是 Servlet、GenericServlet、ServletRequest、ServletResponse 等。与 HTTP 定义相关的类别或接口都位于 javax.servlet.http 套件之中，像是 HttpServlet、HttpServletRequest、HttpServletResponse 等。

4. 你在 web.xml 中定义了以下的内容：

```
<servlet>
  <servlet-name>Goodbye</servlet-name>
  <servlet-class>cc.openhome.LogutServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>GoodBye</servlet-name>
```

```
<url-pattern>/goodbye</url-pattern>
</servlet-mapping>
```

哪个 URL 可以正确的要求 Servlet 进行请求处理？

- (A) /GoodBye
- (B) /goodbye.do
- (C) /LoguotServlet
- (D) /goodbye

答案：D

5. 在 Web 容器中，以下哪两个类别的实例分别代表 HTTP 请求与响应对象？

- (A) HttpRequest
- (B) HttpServletRequest
- (C) HttpServletResponse
- (D) HttpPrintWriter

答案：B、C

6. 在 Web 应用程序中，何者负责将 HTTP 请求转换为 HttpServletRequest 物件？

- (A) Servlet 物件
- (B) HTTP 服务器
- (C) Web 容器
- (D) JSP 网页

答案：C

7. 在 Web 应用程序的档案与目录结构中，web.xml 是直接放置在哪个目录之中？

- (A) WEB-INF 目录
- (B) conf 目录
- (C) lib 目录
- (D) classes 目录

答案：A

8. 你在 web.xml 中定义了以下的内容：

```
<servlet>
    <servlet-name>HelloServlet</servlet-name>
    <java-class>cc.openhome.HelloServlet</java-class>
</servlet>
<servlet-mapping>
```

```
<mapping-name>HelloServlet</mapping-name>
<url-pattern>/hello</url-pattern>
</servlet-mapping>
```

这个 web.xml 中的定义有哪些错误？

- (A) <url-pattern>标签中的设定一定要用.do 作结尾
- (B) <mapping-name>标签应改为<servlet-name>，结尾标签名称也要修改
- (C) <java-name>标签应改为<servlet-class>，结尾标签名称也要修改
- (D) <servlet>标签应改为<servlet-definition>，结尾标签名称也要修改

答案：B、C

提示：<url-pattern>可以无需.do 结尾，也可以用万用字符（wildcard），例如*.do 将会让所有以.do 结尾的请求由该 Servlet 进行处理。

9. 在 MVC 架构中，谁负责通知应用程序客户端，应用程序本身有状态改变？

- (A) 模型（Model）
- (B) 视图（View）
- (C) 控制器（Controller）

答案：A

10. MVC 与 Model 2 架构最大的差别在于？

- (A) Model 2 架构的视图是由 HTML 组成
- (B) Model 2 架构中的模型无法通知视图状态已更新
- (C) MVC 架构是基于请求／响应模型
- (D) MVC 架构只能用于单机应用程序

答案：B

提示：Model 2 架构中呈现视图的技术并没有限制。MVC 架构也可以应用于 3-tier 的网络架构中，其中模型通知视图时，可以透过 Socket 联机来进行。

☒ 实作题

1. 讲师提示：这个练习在让学生动手实作 HttpServlet 的继承、web.xml 的定义，并能够自行查询在线 API 文件，了解 HttpServletRequest 有哪些方法可以利用。

2. 提示：这个练习在让学生动作撰写窗体（学生必须自行学习基本的 HTML），了解 GET 与 POST 如何在 Servlet 中进行处理，学生必须重新定义 doPost () 方法，并了解如何在 Servlet 中撰写判断分支来呈现不同条件下的结果画面。

第 3 章课后练习

课后练习

☒ 选择题

1. 以下的空格应该填入哪个方法？

```
response.setContentType("text/html;charset=UTF-8");  
PrintWriter out = response._____  
out.println("<html>");  
...
```

- (A) `getPrintWriter()`
- (B) `getWriter()`
- (C) `getBufferedWriter()`
- (D) `getOutputWriter()`

答案：B

2. 请问以下的 Servlet 程序片段输出结果为何？

```
out.println("第一个 Servlet 程序");  
out.flush();  
request.getRequestDispatcher("message.jsp")  
    .forward(request, response);  
out.println("Hello!World!");
```

- (A) 显示"第一个 Servlet 程序"后转发 message.jsp
- (B) 显示"第一个 Servlet 程序"与"Hello!World!"
- (C) 直接转发给 message.jsp 进行响应
- (D) 丢出 `IllegalStateException`

答案：A

提示：在 `flush()` 之后响应已确认，此时再 `forward()` 就会发生 `IllegalStateException`。

3. 你将 secret.jsp 档案放在 WEB-INF 目录中，哪个方式或程序片段可以正确让 secret.jsp 进行响应？

- (A) 使用浏览器请求 `/WEB-INF/secret.jsp`
- (B) `request.getRequestDispatcher("/WEB-INF/secret.jsp")` 并进行 `forward()`

- (C) 使用 `response.sendRedirect("/WEB-INF/secret.jsp")`
- (D) 使用 `response.sendError("/WEB-INF/secret.jsp")`

答案: B

4. 如果想知道使用者所使用的浏览器版本等相关讯息, 可以执行哪一段程序代码:

- (A) `request.getHeaderParameter("User-Agent")`
- (B) `request.getParameter("User-Agent")`
- (C) `request.getHeader("User-Agent")`
- (D) `request.getRequestHeader("User-Agent")`

答案: C

5. 如果想取得输出串流对象对浏览器输出位数据, 应该撰写哪一段程序代码?

- (A) `ResponseStream out = response.getResponseStream();`
- (B) `ResponseStream out = response.getStream();`
- (C) `ResponseStream out = response.getOutputStream();`
- (D) `ServletOutputStream out = response.getOutputStream();`

答案: D

6. 哪几个程序代码可以取得 password 请求参数的值?

- (A) `request.getParameter("password");`
- (B) `request.getParameters("password")[0];`
- (C) `request.getParameterValues("password")[0];`
- (D) `request.getRequestParameter("password");`

答案: A、C

提示: `getParameterValues()` 会用字符串数组传回指定的请求参数值, 如果请求参数只有一个值, 则透过指定索引值 0 也可取得该值。

7. 下面这个程序代码片段会输出什么结果?

```
PrintWriter writer = response.getWriter();  
writer.println("第一个 Servlet 程序");  
OutputStream stream = response.getOutputStream();  
stream.println("第一个 Servlet 程序".getBytes());
```

- (A) 浏览器会看到两段"第一个 Servlet 程序"的文字
- (B) 浏览器会看到一段"第一个 Servlet 程序"的文字
- (C) 丢出 `IllegalStateException`
- (D) 由于没有正确地设定内容类型 (`content-type`), 浏览器会提示另存新档

答案: C

提示: 同一个请求周期中, HttpServletResponse 的 `getWriter()` 与 `getOutputStream()` 只能择一使用, 否则会丢出 `IllegalStateException`。

8. 如果要设置响应的内容类型标头, 哪几个选项是正确的作法?

- (A) `response.setHeader("Content-Type", "text/html");`
- (B) `response.setContentType("text/html");`
- (C) `response.addHeader("Content-Type", "text/html");`
- (D) `response.setContentHeader("text/html");`

答案: A、B、C

9. 下面哪个选项, 可以正确地追加自订标头 "MyHead" 的值?

- (A) `response.setHeader("MyHeader", "Value2");`
- (B) `response.appendHeader("MyHeader", "Value2");`
- (C) `response.addHeader("MyHeader", "Value2");`
- (D) `response.insertHeader("MyHeader", "Value2");`

答案: C

10. 哪几个方法是定义在 `HttpServletRequest` 中, 而非由 `ServletRequest` 中继承而来?

- (A) `getMethod()`
- (B) `getHeader()`
- (C) `getParameter()`
- (D) `getCookies()`

答案: A、B、C

提示: 与 HTTP 请求有关的方法, 是定义在 `HttpServletRequest` 中, 与通用请求有关的, 是定义在 `ServletRequest` 中, 例如 `getParameter()`。

☒ 实作题:

1. 请实作一个 Web 应用程序, 可以将使用者所发送的 name 请求参数值画在一张图片上 (参考下图, 底图可任选)。

提示: 可在搜寻引擎下关键词「动态产生图片」寻找解答。参考 <http://caterpillar.onlyfun.net/Gossip/JSPServlet/DynamicImage.htm>, 画图的部分不难, 主要让学生练习如何取得请求参数、如何使用 `ServletOutputStream` 以及利用搜寻引擎寻找解答的能力。



图 3.20 根据使用者输入动态产生图片内容

2. 请实作一个 Web 应用程序，可动态产生使用者登入密码（参考下图，仅需先确作出动态产生密码图片的功能即可，送出窗体后的密码验证功能还不用实现）。

提示：主要是上一题练习的延伸，随机产生四个数字，再画到图片上。这个练习可结合 Session 属性的运用，就可实作出登入的部份，在第 5 章讲到 Session 对象时会再要求实作登入的机制。

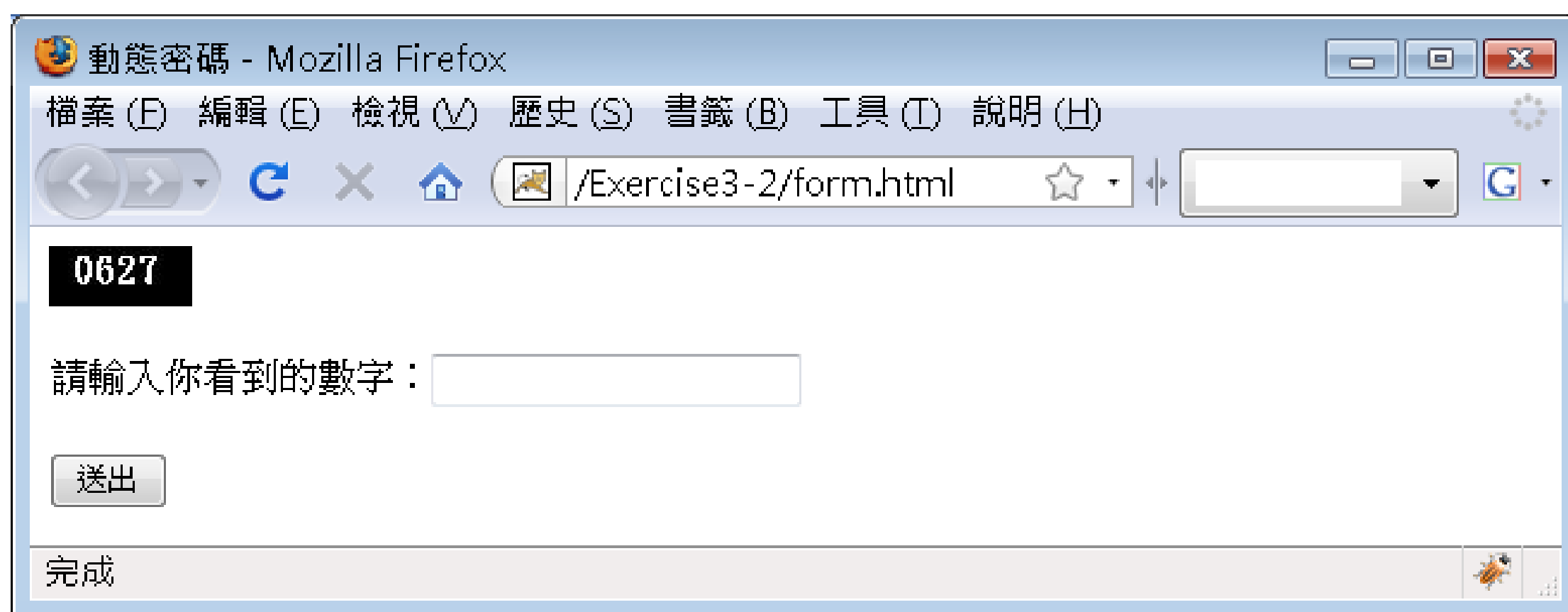


图 3.21 动态产生登入密码

第 4 章课后练习

课后练习

☒ 选择题

1. 如果是整个应用程序会共享的数据，则适合存放在哪个对象之中成为属性？
(A) ServletConfig
(B) ServletContext
(C) ServletRequest
(D) Session

答案：B

2. 如果要取得 ServletContext 初始参数，则可以执行哪个方法？
(A) getContextParameter()
(B) getParameter()
(C) getInitParameter()
(D) getAttribute()

答案：C

3. 假设一段程序代码如下，其中 PARAM 为设定于 web.xml 中的初始参数：

```
public class SomeServlet extends HttpServlet {  
    private String param;  
    public SomeServlet() {  
        param = getInitParameter("PARAM");  
    }  
    ...  
}
```

以下何者正确？

- (A) param 被设定为 web.xml 中的初始参数值
- (B) 无法通过编译
- (C) 应该改用 getServletParameter() 方法
- (D) 发生 NullPointerException

答案：D

提示：getInitParameter()也是透过 ServletConfig 来取得初始参数，在建构式执行时，由于尚未进行 Servlet 初始化，ServletConfig 尚未经由 init

()方法传入 Servlet 之中，所以建构式中尝试取得初始参数时就会发生 NullPointerException。

4. 继承 HttpServlet 之后，若要进行 Servlet 初始化，重新定义哪个方法才是正确的作法？

- (A) `public void init(ServletConfig config) throws ServletException;`
- (B) `public void init() throws ServletException`
- (C) `public String getInitParameter(String name)`
- (D) `public Enumeration getInitParameterNames()`

答案：B

5. 关于 ServletContext 初始参数，以下哪两项是 web.xml 中正确的设定？

- (A) `<init-param>`
 `<param-name>PARAM</param-name>`
 `<param-value>VALUE</param-value>`
 `</init-param>`
- (B) `<context-param>`
 `<param-name>PARAM</param-name>`
 `<param-value>VALUE</param-value>`
 `</context-param>`
- (C) `<init-param>`
 `<param-name>PARAM1</param-name>`
 `<param-value>VALUE1</param-value>`
 `<param-name>PARAM1</param-name>`
 `<param-value>VALUE1</param-value>`
 `</init-param>`
- (D) `<context-param>`
 `<param-name>PARAM1</param-name>`
 `<param-value>VALUE1</param-value>`
 `</context-param>`
 `<context-param>`
 `<param-name>PARAM2</param-name>`
 `<param-value>VALUE2</param-value>`
 `</context-param>`

答案：B、D

6. 若要读取 WEB-INF/data.txt 档案的数据，以下哪个是正确的作法？

- (A) `FileInputStream input =
 new FileInputStream("/WEB-INF/data.txt");`
- (B) `FileInputStream input = new FileInputStream(
 new File("/WEB-INF/data.txt"));`
- (C) `InputStream input =
 getResourceAsStream("/WEB-INF/data.txt");`
- (D) `FileInputStream input =
 getResourceAsStream("/WEB-INF/data.txt");`

答案: C

7. `ServletContext` 定义与属性相关的方法是哪三个?

- (A) `getAttribute();`
- (B) `setAttribute();`
- (C) `deleteAttribute()`
- (D) `removeAttribute();`

答案: A、B、D

8. 若 `web.xml` 中有以下的设定:

```
<servlet>  
  <servlet-name>Some</servlet-name>  
  <servlet-class>cc.openhome.SomeServlet</servlet-class>  
  <init-param>  
    <param-name>param1</param-name>  
    <param-value>1</param-value>  
  </init-param>  
  <init-param>  
    <param-name>param2</param-name>  
    <param-value>2</param-value>  
  </init-param>  
</servlet>
```

而某个 `Servlet` 撰写了以下的程序代码:

```
public void init() throws ServletException {  
  super.init();  
  int param1 = getInitParameter("param1");  
  int param2 = getInitParameter("param2");  
}
```

以下何者描述正确?

- (A) 第二对`<param-name>`、`<param-value>`可以直接撰写在第一个

<init-param>之中

- (B) param1 的值为 1, param2 的值为 1
- (C) 应该在建构式中进行初始参数的取得
- (D) Servlet 类别原始档无法通过编译

答案: D

提示: getInitParameter()取得的是字符串。

9. 哪几个对象提供有 getAttribute()方法?

- (A) ServletRequest
- (B) HttpServlet
- (C) ServletConfig
- (D) ServletContext

答案: A、D

提示: 事实上还有一个, HttpSession 对象。只不过到本章还没介绍到 HttpSession 的使用。

10. 在 web.xml 中有以下的设定:

```
<web-app ...>
    ...
    <listener>
        <listener-class>cc.openhome.SomeListener</listener-class>
    </listener>
    ...
</web-app>
```

以下何者正确?

- (A) SomeListener 类别要实作 ServletContextListener 接口
- (B) 每生成一个 Servlet 实例就会呼叫 ServletContextListener 的 init()方法
- (C) Web 应用程序初始化之后, 会呼叫 ServletContextListener 的 contextInitialized()方法
- (D) 不需要<listener-class>这个标签

答案: A、C

☒ 实作题

1. 请使用目前所学得的 Servlet 相关技巧, 实作一个在线留言版程序, 其中必须包括以下的功能:

- (1). 有个档案会储存留言, 应用程序初始时, 必须从该档案中加载留言记录。
- (2). 「观看留言」功能, 每笔留言中包括了留言者的头像、名称与留言讯息。

下图是个简单的示范。

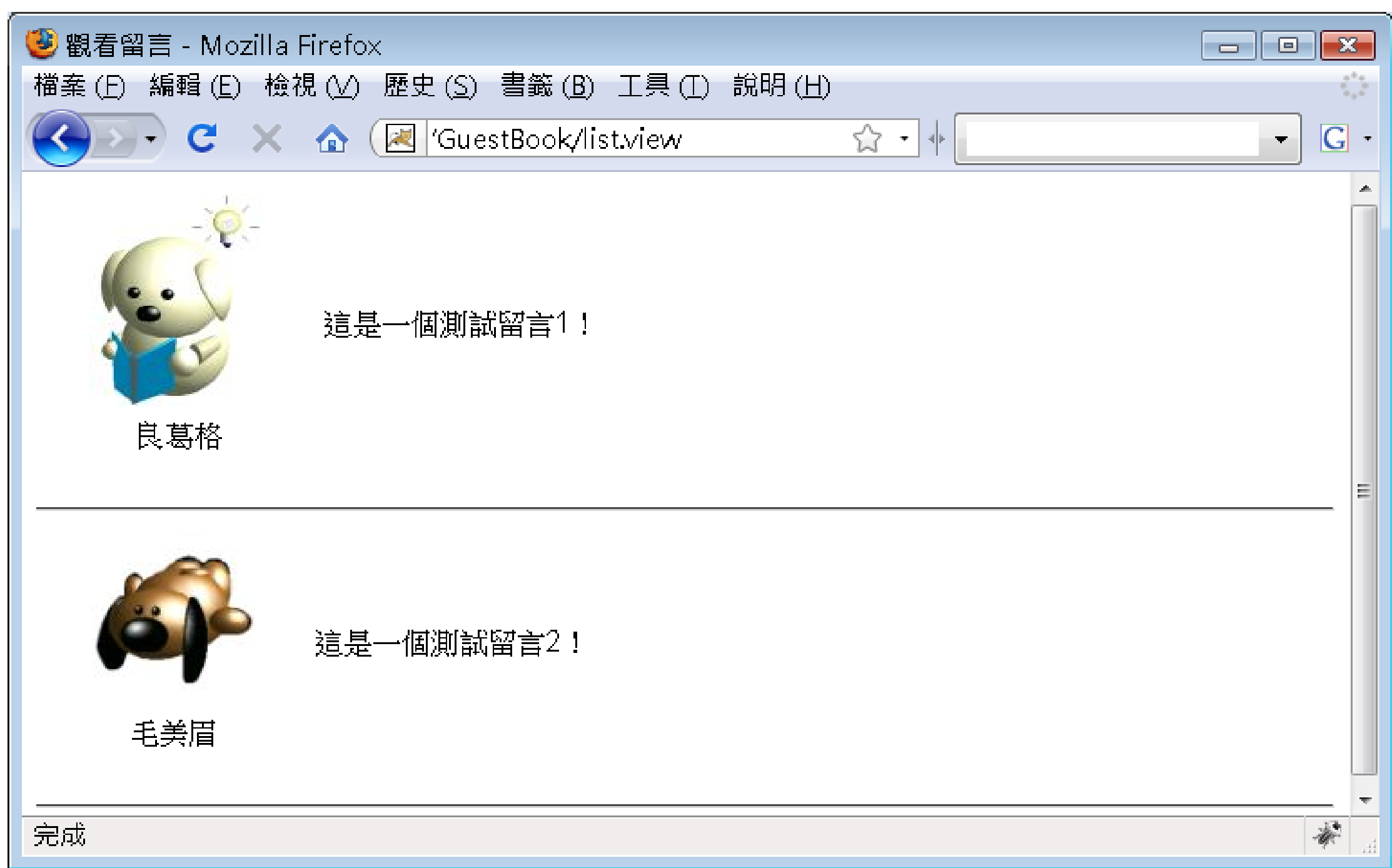


图 4.8 观看留言

(3). 动态「留言窗体」功能，新增留言时使用的窗体。其中包括了输入留言者名称、留言的字段，并可以让使用者选取头像。头像存放的目录可以由 Servlet 初始参数设定。窗体必须可以自动显示头像存放目录中的图片，有多少图片就显示多少个头像。若新增留言失败也会将请求转发回窗体，此时要显示错误讯息以及使用者先前填写的名称与留言。下图是个简单的示范。



图 4.9 新增留言窗体

(4). 「新增留言」功能，必须作基本的请求参数检查。留言失败及成功的 URL，必须可以由 Servlet 初始参数来设定。留言成功时必须显示留言

成功讯息、使用者名称、留言与头像。下图是个简单的示范。



图 4.10 新增留言成功

提示：这个练习可以让学生模仿书中在线书签的制作流程与方式。可建议学生先如第 3 章实作简单的「观看留言」及「新增留言」功能。再进一步根据第 4 章改写，利用 `ServletConfig` 读取 `Servlet` 初始参数，利用 `ServletContext` 读取应用程序初始参数等。

第 5 章课后练习

☒ 选择题

1. 在浏览器禁用 Cookie 的情况下，哪些机制仍可以用于会话管理？
(A) HttpSession
(B) URL 重写
(C) 隐藏字段
(D) Cookie API

答案：B、C

2. 如何设定 Cookie 的有效期限？
(A) 使用 `setMaxAge()`
(B) 使用 `setMaxInactive()`
(C) 使用 `setMaxInactiveInterval()`
(D) 在 `web.xml` 中设定 `<cookie-timeout>`

答案：A

3. `web.xml` 中 `<session-timeout>` 的设定单位是？
(A) 时
(B) 分
(C) 秒
(D) 毫秒

答案：B

4. 何者可以实作使用者的自动登入机制？
(A) HttpSession
(B) Cookie
(C) HttpServletRequest
(D) URL 重写

答案：B

5. 关于 HttpSession 的 `setMaxInactiveInterval()` 方法，以下描述何者错误？
(A) 设定的单位是「分」
(B) 用来设定 HttpSession 在浏览器多久没活动后失效
(C) 用来设定 Cookie 的失效时间

(D) 会覆盖 web.xml 中<session-timeout>的设定

答案：A、C

6. 关于 HttpSession 的描述，以下何者错误？

- (A) 关闭浏览器后 HttpSession 就会被回收
- (B) 可以执行 invalidate() 使之失效
- (C) 不同的浏览器使用相同的 Session ID，也会取得不同的 HttpSession 对象
- (D) HttpSession 的 setMaxInactiveInterval() 方法设定为-1 将永不失效

答案：A、C

提示：关闭浏览器后立即失效的是储存 Session ID 的 Cookie，不是 HttpSession 对象。

7. 哪几个对象提供有 setAttribute() 方法？

- (A) ServletRequest
- (B) HttpSession
- (C) ServletConfig
- (D) ServletContext

答案：A、B、D

8. 关于 Servlet/JSP 的 Session ID，何者正确？

- (A) 预设使用 Cookie 来储存 Session ID
- (B) Cookie 的名称是 JSESSIONID
- (C) 在禁用 Cookie 时，可以使用 URL 重写来发送 Session ID
- (D) 必须自行呼叫 HttpSession 的 getId() 方可产生

答案：A、C

提示：Session ID 的 Cookie 名称是厂商各自实作的，JSESSIONID 在 Tomcat 上的实作名称。容器会自动帮你产生 Session ID。

9. 哪几个对象的属性设定可以跨越请求范围？

- (A) ServletRequest
- (B) HttpSession
- (C) ServletConfig
- (D) ServletContext

答案：B、D

10. HttpSession 失效的时机为何？

- (A) 执行 invalidate()方法
- (B) 下次请求超过 web.xml 中<session-timeout>的设定时间
- (C) 存活时间超过 setMaxInactiveInterval()方法设定的时间
- (D) 关闭浏览器

答案：A、B

提示：无论是<session-timeout>或 setMaxInactiveInterval()所设定的时间，都是指浏览器没有再次请求活动的时间，而不是指 HttpSession 本身的存活时间。

☒ 实作题

1. 请实作一个 Web 应用程序，可动态产生使用者登入密码，送出窗体后必须通过密码验证才可观看到使用者页面。

提示：此题仍第 3 章课后练习第 2 个实作题之延伸。

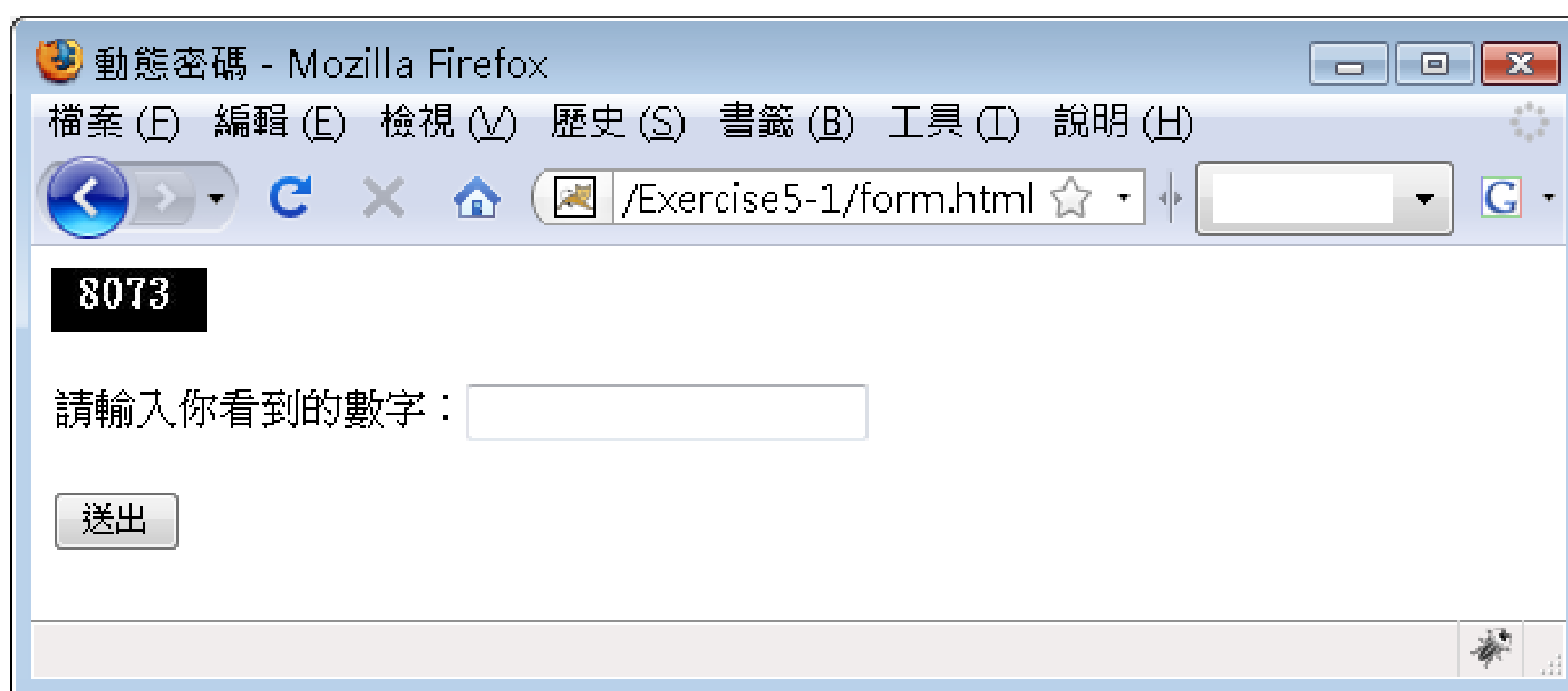


图 5.10 图片验证

2. 实作一个登入窗体，如果使用者核取「记住名称、密码」，则下次造访窗体时，将会自动在名称、密码字段填入上次登入时所使用的值。

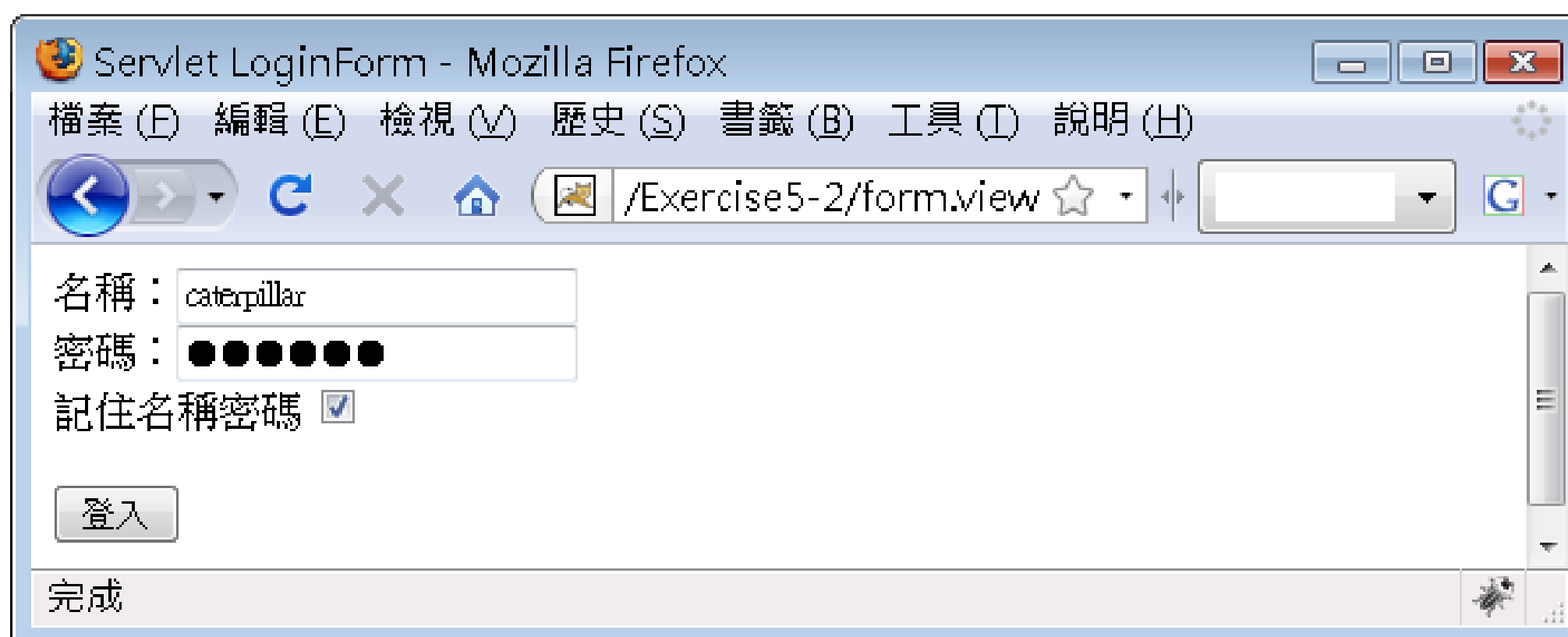


图 5.11 记住名称、密码

第 6 章课后练习

☒ 选择题

1. 关于过滤器的描述，以下何者正确？

- (A) Filter 接口定义了 init()、service()与 destroy()方法
- (B) 会传入 ServletRequest 与 ServletResponse 至 Filter
- (C) 要执行下一个过滤器，必须执行 FilterChain 的 next()方法
- (D) 如果要取得初始参数，要使用 FilterConfig 物件

答案：B、D

2. 关于以下 web.xml 的设定：

```
<filter>
  <filter-name>SecurityFilter</filter-name>
  <filter-class>cc.openhome.SecurityFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>SecurityFilter</filter-name>
  <url-pattern>*.do</url-pattern>
  <init-param>
    <param-name>USER</param-name>
    <param-value>caterpillar</param-value>
  </init-param>
  <init-param>
    <param-name>PASSWORD</param-name>
    <param-value>123456</param-value>
  </init-param>
</filter-mapping>
```

以下描述何者正确？

- (A) 根据<filter-mapping>，过滤器会套用在所有以.do 结尾的 URL 请求
- (B) 可以透过 FilterConfig 来读取 USER 与 PASSWORD 初始参数
- (C) 初始参数设定位置错误，<init-param>等标签应放在<filter>标签之中
- (D) <url-pattern>应改为<servlet-name>才是正确设定

答案：A、C

提示：由于初始参数的设定错误，无法透过 FilterConfig 取得初始参数。

3. 关于以下的 web.xml 设定：

```
<filter-mapping>
    <filter-name>SecurityFilter</filter-name>
    <url-pattern>/add.do</url-pattern>
</filter-mapping>
<filter-mapping>
    <filter-name>SecurityFilter</filter-name>
    <url-pattern>*.view</url-pattern>
</filter-mapping>
<filter-mapping>
    <filter-name>CharacterFilter</filter-name>
    <url-pattern>/add.do</url-pattern>
</filter-mapping>
```

以下描述何者正确？

- (A) 当请求 add.do 时，只会套用 SecurityFilter
- (B) 当请求 add.do 时，过滤器的执行顺序为 SecurityFilter、CharacterFilter
- (C) 当请求 list.view 时，过滤器的执行顺序为 SecurityFilter、CharacterFilter
- (D) 当请求 list.view 时，只会套用 SecurityFilter

答案：B、D

4. 关于 FilterChain 的描述，何者正确？

- (A) 如果不呼叫 FilterChain 的 doFilter() 方法，则请求略过接下来的过滤器而直接交给 Servlet
- (B) 如果有下一个过滤器，呼叫 FilterChain 的 doFilter() 方法，会将请求交给下一个过滤器
- (C) 如果没有下一个过滤器，呼叫 FilterChain 的 doFilter() 方法，会将请求交给 Servlet
- (D) 如果没有下一个过滤器，呼叫 FilterChain 的 doFilter() 方法没有作用

答案：B、C

5. 关于 FilterConfig 的描述，何者错误？

- (A) 会在 Filter 界面的 init() 方法呼叫时传入
- (B) 为 web.xml 中 <filter> 设定的代表对象
- (C) 可读取 <servlet> 标签中 <init-param> 所设定的初始参数
- (D) 可使用 getInitParameter() 方法读取初始参数

答案：C

6. 以下的程序代码将实作请求包裹器：

```
public class MyRequestWrapper _____ {  
    public MyRequestWrapper(HttpServletRequest request) {  
        super(requset);  
    }  
    ...  
}
```

请问空白处应该填上哪个程序代码片段？

- (A) implements ServletRequest
- (B) extends ServletRequestWrapper
- (C) implements HttpServletRequest
- (D) extends HttpServletRequestWrapper

答案：D

提示：因为建构式中直接使用传入的 HttpServletRequest 对象及 super() 来呼叫父类别建构式，所以一定是继承 HttpServletRequestWrapper。

7. 关于请求包裹器，以下描述何者正确？

- (A) 可以实作 ServletRequest 界面
- (B) 可以继承 ServletRequestWrapper 类别
- (C) 一定要继承 ServletRequestWrapper 类别
- (D) HttpServletRequestWrapper 是 ServletRequestWrapper 的子类别

答案：A、B、D

提示：要自己实作 ServletRequest 接口来实作请求包裹器也是可以的，只是所有的方法都要自行实作。

8. 关于 HttpServletRequestWrapper 与 HttpServletResponseWrapper 的描述，何者有误？

- (A) 分别实作了 HttpServletRequest 界面与 HttpServletResponse 界面
- (B) 分别继承了 ServletRequestWrapper 与 ServletResponseWrapper 类别
- (C) 实作时，至少要重新定义一个父类别中的方法
- (D) 实作时必须在建构式中呼叫父类别建构式

答案：C

9. 在开发过滤器时，以下何者是正确的观念？
- (A) 必须考虑前后过滤器之间的关系
 - (B) 挂上过滤器后不改变应用程序原有的功能
 - (C) 设计 Servlet 时必须考虑到未来加装过滤器的需求
 - (D) 每个过滤器要设计为独立互不影响的组件

答案：B、D

10. 关于 Filter 界面上的 doFilter()方法之说明，何者有误？
- (A) 会传入两个参数 HttpServletRequest、HttpServletResponse
 - (B) 会传入三个参数 HttpServletRequest、HttpServletResponse、FilterChain
 - (C) 前一个过滤器呼叫 FilterChain 的 doFilter()后，会执行目前过滤器的 doFilter()方法
 - (D) 前一个过滤器的 doFilter()执行过后，会执行目前过滤器的 doFilter()方法

答案：A、B、D

提示：Filter 的 doFilter()方法上三个参数是 ServletRequest、ServletResponse、FilterChain。前一个过滤器的 doFilter()执行时，当中必须呼叫 FilterChain 的 doFilter()方法，才会执行目前过滤器的 doFilter()方法。

☒ 实作题

1. 模仿本单元综合练习，重构第 4 章元课后练习中的留言版程序，将从档案读写留言版讯息的职责封装至一个 MessageService 类别中。。
2. 你的留言版程序不允许使用者输入 HTML 标签，但可以允许使用者输入一些代码作些简单的样式。例如：
 - ☒ [b]粗体[/b]
 - ☒ [i]斜体[/i]
 - ☒ [big]放大字体[/big]
 - ☒ [small]缩小字体[/small]

HTML 的过滤功能，你可以直接使用本章所开发的字符过滤器，并且请另行开发一个过滤器来完成代码替换的功能。

第 7 章课后练习

☒ 选择题

1. 以下哪几个监听器，必须在 web.xml 中设定？

- (A) HttpSessionListener
- (B) RequestListener
- (C) ServletContextListener
- (D) ServletAttributeListener

答案：A、C

提示：没有 B、D 这两种监听器。

2. 以下何者为监听器的事件类型之一？

- (A) HttpSessionEvent
- (B) RequestAttributeEvent
- (C) ServletContextAttributeEvent
- (D) ServletContextEvent

答案：A、C、D

3. 以下监听器中，不需要在 web.xml 中设定的是？

- (A) HttpSessionListener
- (B) HttpSessionBindingListener
- (C) ServletContextListener
- (D) ServletAttributeListener

答案：B

4. 以下哪几个事件类型，拥有 getName() 与 getValue() 方法？

- (A) HttpSessionEvent
- (B) ServletRequestAttributeEvent
- (C) ServletContextAttributeEvent
- (D) HttpSessionBindingListener

答案：B、C、D。

5. 你有一段程序代码片段：

```
HttpSession session = request.getSession();  
User user = new User();
```

```
session.setAttribute("user", user);
```

以下何种作法，可以让你在不修改程序代码的情况下，实作统计在线人数？

- (A) 实作 HttpSessionBindingListener
- (B) 实作 HttpSessionListener
- (C) 实作 HttpSessionActivationListener
- (D) 以上皆非

答案：B

提示：题目要求不修改原有的程序代码，所以 B 是唯一的可能性。

6. 在 web.xml 中有以下的设定：

```
<servlet>
    <servlet-name>Some</servlet-name>
    <servlet-class>cc.openhome.SomeServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet>
    <servlet-name>Other</servlet-name>
    <servlet-class>cc.openhome.OtherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet>
    <servlet-name>AnOther</servlet-name>
    <servlet-class>cc.openhome.AnOtherServlet</servlet-class>
    <load-on-startup>2</load-on-startup>
</servlet>
```

请问以下描述何者正确？

- (A) 容器会产生两个 AnotherServlet 实例
- (B) 容器会先初始化 SomeServlet，再初始化 OtherServlet
- (C) 容器会先初始化 AnOtherServlet，然后才是其它 Servlet
- (D) 容器在请求来到时，才会初始化对应的 Servlet

答案：B

7. 哪个 URL 模式设定方式，可以符合/guest/list.view 的请求？

- (A) <url-pattern>*.view</view>
- (B) <url-pattern>/guest/*.view</view>
- (C) <url-pattern>/guest/*</view>

(D) <url-pattern>/guest/list.view</view>

答案: A、C、D

8. 如果想要请求 guest 目录时, 自动显示欢迎档案, 则以下空格可以填上什么?

```
<web-app ...>
  <welcome-file-list>
    <welcome-file>_____</welcome-file>
  </welcome-file-list>
</web-app>
```

- (A) /guest/index.html
- (B) /index.html
- (C) default.jsp
- (D) default.*

答案: C

9. 关于以下 web.xml 中的设定, 何者描述正确?

```
<web-app ...>
  <error-page>
    <exception-type>NullPointerException</exception-type>
    <location>report.view</location>
  </error-page>
</web-app>
```

- (A) 发生 NullPointerException 时会转发 report.view 处理
- (B) 例外型态必须指定完全符合名称 (Full Qualified Name)
- (C) report.view 必须改为 /report.view
- (D) 例外类型不可以指定特定的例外类别名称

答案: B、C

10. 在过滤器设定中, 哪个 URL 模式设定方式, 在 /guest/list.view 的请求时会触发过滤器?

- (A) <url-pattern>*.view</view>
- (B) <url-pattern>/guest/*.view</view>
- (C) <url-pattern>/guest/*</view>
- (D) <url-pattern>/guest/list.view</view>

答案: A、C、D

提示: 过滤器的 <url-pattern> 设定原则与 Servlet 相同。

☒ 实作题

1. 请扩充 7.1.1 节中的范例，不仅统计在线人数，还可以在页面上显示目前登入使用者的名称、浏览器信息、最后活动时间。

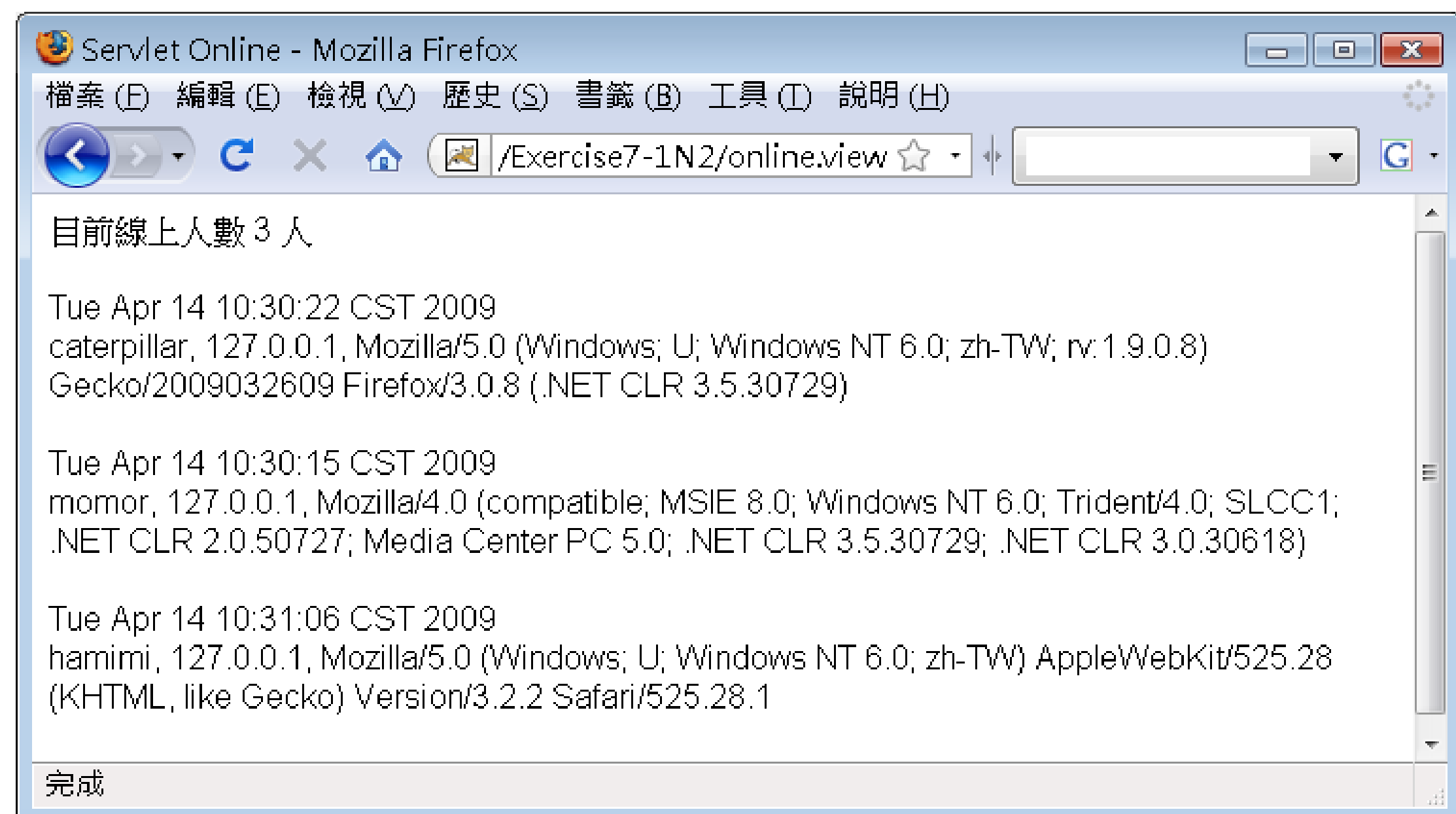


图 7.14 在线使用者信息

2. 在 7.1.3 节中,使用 HttpSessionBindingListener 来实作使用者登入、注销的日志记录功能,请改用 HttpSessionAttributeListener 来实作这个功能。

第 8 章课后练习

☒ 选择题

1. 关于 JSP 的描述，哪个正确？

- (A) JSP 是直译式的网页，与 Servlet 无关
- (B) JSP 会先转译为.java，然后编译为.class 载入容器
- (C) JSP 会直接由容器动态生成 Servlet 实例，无需转译
- (D) JSP 是丢到浏览器端，由浏览器进行直译

答案：B

2. 关于 JSP 的描述，哪个正确？

- (A) 要在 JSP 中撰写 Java 程序代码，必须重新定义 `_jspService()`
- (B) 重新定义 `jspInit()` 来作 JSP 初次载入容器的初始化动作
- (C) 重新定义 `jspDestroy()` 来作 JSP 从容器销毁时的结尾动作
- (D) 要在 JSP 中撰写 Java 程序代码，必须重新定义 `service()`

答案：B、C

3. 如果想要在 JSP 中定义方法，应该使用哪个 JSP 元素？

- (A) `<% %>`
- (B) `<%= %>`
- (C) `<%! %>`
- (D) `<%- - - %>`

答案：C

4. 当 JSP 中有撰写中文时，而执行结果出现乱码，必须检查 page 指示元素的哪些属性设定是否正确？

- (A) `contentType`
- (B) `language`
- (C) `extends`
- (D) `pageEncoding`

答案：A

提示：基本上还得检查一下编辑器的预设编码与 `pageEncoding` 是否一致。

5. 哪个 JSP 隐含对象，转译后对应 ServletContext 物件？

- (A) pageContext
- (B) config
- (C) page
- (D) application

答案：D

6. 在 Session 范围中以名称"bean"放置了一个 JavaBean 属性，JavaBean 上有个 getMessage() 方法，请问以下哪个方式呼叫 getMessage() 以取得讯息并显示出来？

- (A) `<jsp:getProperty name="bean" property="message">`
- (B) `${requestScope.bean.message}`
- (C) `<%= request.getBean().getMessage() %>`
- (D) `${bean.message}`

答案：B、D

提示：<jsp:getProperty>前必须先使用<jsp:useBean>并指定 scope 为 request，否则<jsp:getProperty>只会在 page 范围中寻找属性。

7. 在 Web 应用程序中有以下的程序代码，执行后转发至某个 JSP 网页：

```
Map map = new HashMap();  
map.put("user", "caterpillar");  
map.put("role", "admin");  
request.setAttribute("login", map);
```

以下哪个选项可以正确地使用 EL 取得 map 中的值？

- (A) `${map.user}`
- (B) `${map["role"]}`
- (C) `${login.user}`
- (D) `${login[role]}`

答案：C

提示：选项 D 必须改为`${login["role"]}`才是正确。

8. 在 Web 应用程序中有以下的程序代码，执行后转发至某个 JSP 网页：

```
Map map = new HashMap();  
map.put("local.role", "admin");  
request.setAttribute("login", map);
```

以下哪个选项可以正确地使用 EL 取得 map 中的值？

- (A) `${map.local.role}`
- (B) `${login.local.role}`
- (C) `${map["local.role"]}`
- (D) `${login["local.role"]}`

答案：D

9. 在 Web 应用程序中有以下的程序代码，执行后转发至某个 JSP 网页：

```
List names = new ArrayList();  
names.add("caterpillar");  
request.setAttribute("names", names);
```

以下哪个选项可以正确地使用 EL 取得 List 中的值？

- (A) `${names.0}`
- (B) `${names[0]}`
- (C) `${names.[0]}`
- (D) `${names["0"]}`

答案：B、D

提示：在[]指定索引值时可以用双引号，所以 D 也是正确的。

10. 以下何者不是 EL 隐含物件？

- (A) `param`
- (B) `request`
- (C) `pageContext`
- (D) `cookie`

答案：B

☒ 实作题

1. 先前章节实作在线留言版时，使用 Servlet 来实现画面的输出，请将之改为 JSP，并尽量使用本章所学得的 JSP 元素，减少 Scriptlet 的使用。

第 9 章课后练习

☒ 选择题

1. 哪些 JSTL 标签可用来进行 Java 程序中 if、if...else 的功能?

- (A) <c:if>
- (B) <c:else>
- (C) <c:when>
- (D) <c:otherwise>

答案: A、C、D

2. 如果打算使用 request 对象的 setCharacterEncoding() 方法设定字符编码处理方式, 则以下哪个 JSTL 标签可以让你不使用 Scriptlet?

- (A) <c:if>
- (B) <c:set>
- (C) <c:out>
- (D) <c:url>

答案: B

3. 如果 taglib 设定如下:

```
<%@taglib prefix="x" uri="http://openhome.cc/magic/x"%>
```

则以下何者会是使用自订标签的正确方式?

- (A) <x:if>
- (B) <magic:forEach>
- (C) <if/>
- (D) <x:if/>

答案: D

4. 在使用 Tag File 自订标签时, 若 JSP 中有以下的内容:

```
<%@taglib prefix="html" tagdir="/WEB-INF/tagFile" %>
```

以下何者描述正确?

- (A) 可以<html:Errors/>的方式使用自订标签
- (B) 可以<htmlErrors/>的方式使用自订标签
- (C) 可以<Errors/>的方式使用自订标签
- (D) taglib 的定义有误, 无法使用自订标签

答案: D

提示: 如果是 Tag File, 则 tagdir 只有可能是 /WEB-INF/tags 或其子资料夹。

5. 关于 tag 指示元素的 body-content 属性, 以下说明何者正确?
- (A) 可设定的值有 JSP、scriptless、empty 与 tagdependent
 - (B) 默认值是 scriptless
 - (C) 设定为 tagdependent 时, 本体内容将不作任何处理直接传入 Tag File 中
 - (D) 如果本体中要执行 Scriptlet, 则要设定为 JSP

答案: B、C

提示: 可设定的值只有 scriptless、empty 与 tagdependent, 预设是 scriptless。无论如何, Tag File 的本体是绝不能有 Scriptlet。

6. 在继承 SimpleTagSupport 后, doTag() 的实作如下:

```
public void doTag() throws JspException {
    try {
        if(test) {
            getJspBody()._____ ;
        }
    } catch (java.io.IOException ex) {
        throw new JspException("执行错误", ex);
    }
}
```

如果 test 为 true 时, 将执行本体内容并输出结果至浏览器, 则空白部份应填入?

- (A) invoke()
- (B) invoke(new JspWriter())
- (C) invoke(null)
- (D) invoke(new PrintWriter())

答案: C

7. 在继承 SimpleTagSupport 后, doTag() 的实作如下:

```
public void doTag() throws JspException {
    try {
        if(test) {
            // ...
        }
        else {
            throw new _____ ;
        }
    }
}
```

```
    } catch (java.io.IOException ex) {  
        throw new JspException("执行错误", ex);  
    }  
}
```

如果 test 为 false 时，希望能中断 JSP 后续页面的处理？则空白部份应填入？

- (A) SkipPageException()
- (B) IOException()
- (C) ServletException()
- (D) Exception()

答案：A

8. 在继承 TagSupport 后，有个程序片段实作如下：

```
public int _____ throws JspException {  
    if(test) {  
        // ...  
        return EVAL_PAGE;  
    }  
    return SKIP_PAGE;  
}
```

如果 test 为 false 时，希望能中断 JSP 后续页面的处理？则空白部份应是实作哪个方法？

- (E) doTag()
- (F) doStartTag()
- (G) doEndTag()
- (H) doAfterBody()

答案：G

9. 在继承 BodyTagSupport 类别实作自订标签时，基本上 doStartTag() 方法可以传回的有效值有哪几个？

- (A) SKIP_BODY
- (B) EVAL_BODY_BUFFERED
- (C) EVAL_BODY_INCLUDE
- (D) EVAL_PAGE

答案：A、B、C

10. 在 TagSupport 类别的实作中, doStartTag()、doEndTag() 的预设传回值是?

- (A) SKIP_BODY、EVAL_PAGE
- (B) EVAL_BODY_INCLUDE、EVAL_PAGE
- (C) EVAL_BODY_INCLUDE、SKIP_PAGE
- (D) SKIP_BODY、SKIP_PAGE

答案: A

☒ 实作题

1. 请开发一个自订标签, 可以如下使用:

```
<g:eachImage var="image" dir="/avatars">
    <br>
</g:eachImage>
```

你可以指定某个目录, 这个自订卷标将取得该目录下所有图片的路径, 并设定给 var 所指定的变量名称, 之后在标签本体中可以使用该名称 (像上例使用 \${image} 搭配 卷标将图片显示在浏览器上)。

2. 请将第 8 章课后练习的留言版程序, JSP 页面中的 Scriptlet 使用 EL、JSTL 或自订标签取代, 让 JSP 页面中不出现任何的 Scriptlet。

第 10 章课后练习

☒ 选择题

1. 哪几个 Web 容器提供的验证方式，会出现对话框让使用者输入名称、密码？
(A) BASIC
(B) FORM
(C) DIGEST
(D) CLIENT_CERT

答案：A、C、D

2. 以下何者验证方式，有机会自行设计登入画面与注销机制？
(A) BASIC
(B) FORM
(C) DIGEST
(D) CLIENT_CERT

答案：B

3. 在使用容器窗体验证时，<form>的 action 属性应设置为？
(A) j_security
(B) j_check
(C) j_security_check
(D) 没有规范

答案：C

4. 假设有以下的 web.xml 设定档：

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Login Required</web-resource-name>
    <url-pattern>/delete.jsp</url-pattern>
    <url-pattern>/delete.do</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>admin</role-name>
```

```
</auth-constraint>
</security-constraint>
```

以下描述何者正确？

- (A) 只有 admin 角色才可以存取/delete.jsp 与/delete.do
- (B) admin 角色只允许使用 GET 与 POST 方法
- (C) GET 与 POST 方法只允许 admin 使用
- (D) 其它角色可使用 GET、POST 以外的 HTTP 方法存取

答案：A、C、D

提示：admin 可以使用所有的 HTTP 方法来存取/delete.jsp 与/delete.do。

5. 如果要保证数据传输时的完整性与机密性，必须在 <security_constraint>中使用哪两个标签？

- (A) <user-data-constraint>
- (B) <transport-guarantee>
- (C) <auth-constraint>
- (D) <form-login-config>

答案：A、B

6. 假设有以下的 web.xml 设定档：

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Login Required</web-resource-name>
    <url-pattern>/delete.jsp</url-pattern>
    <url-pattern>/delete.do</url-pattern>
  </web-resource-collection>
  <auth-constraint/>
</security-constraint>
```

以下描述何者正确？

- (A) 所有角色都可以存取/delete.jsp 与/delete.do
- (B) 存取/delete.jsp 与/delete.do 任何 HTTP 方法都是允许的
- (C) 没有任何角色可以存取/delete.jsp 与/delete.do
- (D) 这个设定片段是错误的，无法部署

答案：C

7. 验证成功但授权失败，则会出现哪种状态码及讯息？

- (A) 404 Not Found
- (B) 401 Unauthorized

- (C) 403 Forbidden
- (D) 500 Error

答案：C

8. 哪几个 Web 容器提供的验证方式，会需要使用到凭证？

- (A) BASIC
- (B) FORM
- (C) DIGEST
- (D) CLIENT_CERT

答案：D

9. HttpServletRequest 中的 isUserInRole() 可以协助达到安全的哪项要求？

- (A) 验证
- (B) 授权
- (C) 机密性
- (D) 完整性

答案：B

提示：会使用到 isUserInRole() 方法的 URL，基本上会先透过宣告式安全的验证及授权，再透过 isUserInRole() 作更细部的授权控制。

10. 以下何者是 <transport-guarantee> 的合法设定值？

- (A) NONE
- (B) CONFIDENTIAL
- (C) INTEGRAL
- (D) SSL

答案：A、B、C

☒ 实作题

1. 以第 9 章的课后练习为基础，假设你将开发一个删除留言的功能，这个功能只能登入且具备 admin 角色的使用者才可以使用，若删除留言需要用到 /delete.jsp 与 /delete.do 两个 URL，请使用容器宣告式窗体验证对这两个 URL 加以保护。

第 11 章课后练习

☒ 选择题

1. 哪些 JDBC 驱动程序可以有跨平台的特性？

- (A) TYPE 1
- (B) TYPE 2
- (C) TYPE 3
- (D) TYPE 4

答案：C、D

2. 哪种 JDBC 驱动程序是基于数据库所提供的 API 来进行实作？

- (A) TYPE 1
- (B) TYPE 2
- (C) TYPE 3
- (D) TYPE 4

答案：B

3. JDBC 相关接口或类别，是放在哪个套件之下加以管理？

- (A) `java.lang`
- (B) `javax.sql`
- (C) `java.sql`
- (D) `java.util`

答案：C

4. 使用 JDBC 时，通常会需要处理哪个受检例外（Checked Exception）？

- (A) `RuntimeException`
- (B) `SQLException`
- (C) `DBException`
- (D) `DataException`

答案：B

5. 关于 `Connection` 的描述，何者正确？

- (A) 可以从 `DriverManager` 上取得 `Connection`
- (B) 可以从 `DataSource` 上取得 `Connection`
- (C) 在方法结束之后 `Connection` 会自动关闭

(D) Connection 是执行绪安全 (Thread-safe)

答案: A、B

提示: Connection 不是执行绪安全的, 不可以在多执行绪环境下共享 Connection, 以免发生数据错乱的问题。

6. 使用 Statement 来执行 SELECT 等查询用的 SQL 指令时, 应使用下列哪个方法?

- (A) executeSQL()
- (B) executeQuery()
- (C) executeUpdate()
- (D) executeFind()

答案: B

7. 哪个对象正确使用下, 可以适当地避免 SQL Injection 的问题?

- (A) Statement
- (B) ResultSet
- (C) PreparedStatement
- (D) Command

答案: B

8. 取得 Connection 之后, 如何取得 Statement 对象?

- (A) conn.createStatement()
- (B) conn.buildStatement()
- (C) conn.getStatement()
- (D) conn.createSQLStatement()

答案: A

9. 以下描述何者有误?

- (A) 使用 Statement 一定会发生 SQL Injection
- (B) 使用 PreparedStatement 就不会发生 SQL Injection
- (C) 不使用 Connection 时必须加以关闭
- (D) ResultSet 代表查询的结果集合

答案: A、B

提示: 只要有正确的使用观念, 即使用 Statement 也不一定会发生 SQL Injection。如果使用不正确, 使用 PreparedStatement 也有可能发生 SQL Injection。

10. 使用 Statement 的 executeQuery()方法，会传回哪个型态？

- (A) int
- (B) boolean
- (C) ResultSet
- (D) Table

答案：C

☒ 实作题

1. 以第 9 章的课后练习为基础，将先前的留言版程序，改用数据库储存留言版讯息，并新增「删除留言」功能。
2. 请继续扩充留言版程序，使之具备响应留言、删除留言、分页显示等功能。

第 12 章课后练习

☒ 选择题

1. 当继承 `HttpServlet` 后，重新定义 `doGet()` 方法，以在 GET 请求来到时执行，这实际上是哪个模式的实现结果？
 - (A) Factory Method 模式
 - (B) Template Method 模式
 - (C) Command 模式
 - (D) Proxy 模式

答案：B

2. 在 `web.xml` 中定义使用者登入验证过滤器、效能过滤器，就可以有拦截请求，这是哪个模式的实现？
 - (A) Business Delegate 模式
 - (B) Interceptor Filter 模式
 - (C) Service Locator 模式
 - (D) Front Controller 模式

答案：B

3. 以下哪几个属于 Java EE 模式？
 - (A) Business Delegate 模式
 - (B) Interceptor Filter 模式
 - (C) Service Locator 模式
 - (D) Front Controller 模式

答案：A、B、C、D

4. 想要降低前端所采用技术及后端服务对象的耦合度，例如为免日后前端从 `Servlet`/`JSP` 改用 `JSF` 时必须对后端对象作出修改，则适合采用哪个模式？
 - (A) Business Delegate 模式
 - (B) Interceptor Filter 模式
 - (C) Service Locator 模式
 - (D) Front Controller 模式

答案：A

5. 在分布式的应用当中，你希望对某个对象隐藏取得远程服务对象的细节（也许是透过 `RMI`、`JNDI` 等），则应采用哪个模式？

- (A) Business Delegate 模式
- (B) Interceptor Filter 模式
- (C) Service Locator 模式
- (D) Front Controller 模式

答案：C