

## 第一章

- 1、 JSP 引擎是支持 JSP 程序的 Web 容器，负责运行 JSP，并将有关结果发送到客户端。
- 2、 目前流行的 JSP 引擎之一是 Tomcat 安装 Tomcat 服务器，首先要安装 JDK。
- 3、 JSP 页面必须保存在 Web 服务目录中，Tomcat 服务器的 webapps 下的目录都可以作为 Web 服务目录。如果想让 webapps 以外的其他的目录作为 Web 服务目录，必须要修改 Tomcat 服务器下 conf 文件夹中的 **server.xml** 文件，**并重新启动 Tomcat 服务器。**
- 4、 当服务器上的一个 JSP 页面被第一次请求执行时，服务器上的 JSP 引擎首先将 JSP 页面文件转译成一个 Java 文件，再将这个 Java 文件编译生成字节码文件，然后通过执行字节码文件响应客户的请求。
- 5、 当多个客户请求一个 JSP 页面时，Tomcat 服务器为每个客户启动一个线程，该线程负责执行常驻内存的字节码文件来响应相应客户的请求。这些线程由 Tomcat 服务器来管理。
- 6、 JSP 与 Servlet 的关系，JSP 是转换成 Servlet 来运行的。

## 第二章

- 1、 JSP 页面的构成：普通的 HTML 标记（客户端浏览器执行）、JSP 标记、成员变量和方法声明、Java 程序片、Java 表达式（JSP 引擎处理并将结果发送给用户浏览器）
- 2、 成员变量为所有用户共享，任何用户对成员变量的操作都会影响其他用户，synchronized 关键字保证一次只有一个线程执行
- 3、 多用户访问 JSP 页面，其程序片会被执行多次，分别在不同线程中，其局部变量互不干扰。
- 4、 page 指令标记用来定义整个 JSP 页面的一些属性，常用的有 contentType 和 import。
- 5、 include 指令标记在编译阶段就处理所需要的文件，被处理的文件在逻辑与语法上依赖于当前 JSP 页面，优点是速度快；include 动作标记是在 JSP 页面运行时才处理文件，在逻辑与语法上独立于当前 JSP 页面，更加灵活。
- 6、 注意成员变量和方法的声明和普通 Java 程序片内变量声明的区别：`<%! %>`、`<% %>`
- 7、 在多次执行的情况下，成员变量和局部变量的值会如何变化，这个很重要，也很容易错。
- 8、 JSP 页面中 Java 程序片是顺序执行的，某个 Java 程序片中的变量在后续的程序片中依然有效。
- 9、 JSP 的注释在客户端不可见，HTML 的可见
- 10、 Page 指令标记的 contentType 只能指定一个值
- 11、 Forward 动作标记是用于转向新的页面，转后之后地址栏不变，param 动作标记是传递参数的

## 第三章

- 1、 HTTP 通信协议是用户与服务器之间一种提交请求信息与响应信息的无状态通信协议；在 JSP 中，内置对象 request 封装了请求信息，使用 `getParamter(String s)` 获取提交的信息；response 对象请求作出响应，向用户发送数据。

- 2、 HTTP 是一种**无状态协议**，无法记忆连接的有关信息。通过 session 对象记录连接的信息，同一用户在同一 Web 服务目录中的各个页面的 session 是相同的，在不同 Web 服务目录中 Session 互不相同。
- 3、 Session 生存周期依赖于关闭浏览器、调用 invalidate() 方法、最长的“发呆”时间。
- 4、 内置 application 对象由服务负责创建，每个 Web 服务目录下的 application 对象被所有访问该服务目录的用户共享，不同 Web 服务目录下的 application 对象互不相同。
- 5、 HTTP 是一种**无状态协议**，无法记忆连接的有关信息。通过 session 对象记录连接的信息，同一用户在同一 Web 服务目录中的各个页面的 session 是相同的，在不同 Web 服务目录中 Session 互不相同。
- 6、 Session 生存周期依赖于关闭浏览器、关闭服务器、调用 invalidate() 方法、最长的“发呆”时间，最长发呆时间通过修改 Tomcat 的 Conf 目录的 Web.xml
- 7、 内置 application 对象由服务负责创建，每个 Web 服务目录下的 application 对象被所有访问该服务目录的用户共享，不同 Web 服务目录下的 application 对象互不相同。
- 8、 ISO-8859-1 编码不支持中文，需要 gb2312、utf-8、utf-16
- 9、 FORM 表单可以 POST 和 GET 两种方式提交
- 10、 sendRedirect 重定向之后，地址栏会变。
- 11、 URL 重写实际在 URL 的后面加上了 JSESSIONID，把 Session 的 ID 传递过去。
- 12、 Out.print 和 Out.println 的区别

## 第四章

- 1、 Javabeen 是一个可重复使用的软件组件，是遵循一定标准、用 Java 语言编写的一个类，该类的一个实例称作一个 Javabeen。
- 2、 一个 JSP 页面可以将数据的处理过程指派给一个或几个 bean 来完成，我们只需在 JSP 页面中调用这个 bean 即可。在 JSP 页面中调用 bean 可以将数据的处理代码从页面中分离出来，实现代码服用，更有效维护一个 Web 应用
- 3、 bean 的生命周期分为 page、request、session 和 application，一定要注意他们之间的区别
- 4、 Javabeen 中方法都是 public
- 5、 构造方法是无参
- 6、 setProperty 可以通过表达式或者字符串给一个属性赋值
- 7、 setProperty 也可以通过表单的方式给一个或多个属性赋值
- 8、 如何写一个简单的 Javabeen 的类，给你几个属性，至少要知道怎么写一个 JavaBean

## 第五章

- 1、 Servlet 对象第一次被请求加载时，服务器创建一个 servlet 对象，这个对象调用 init 方法完成必要的初始化工作。init 方法只被调用一次，当后续的客户请求该 servlet 对象服务时，服务器将启动一个新的线程，在该线程中 servlet 对象调用 service 方法响应客户的请求，调用过程运行在不同的线程中，互不干扰。
- 2、 Init、service、destroy 之间的区别

- 3、Servlet 类继承的 service 方法检查 HTTP 请求类型 (Get 、Post 等)，并在 service 方法中根据用户的请求方式，对应地再调用 doGet 或 doPost 方法。因此，Servlet 类不必重写 service 方法，直接继承该方法即可，可以在 Servlet 类中重写 doPost 或 doGet 方法来响应用户的请求。
- 4、RequestDispatcher 对象可以把用户对当前 JSP 页面或 servlet 的请求转发给另一个 JSP 页面或 servlet，而且将用户对当前 JSP 页面或 servlet 的请求和响应传递给所转发的 JSP 页面或 servlet。也就是说，当前页面所要转发的目标页面或 servlet 对象可以使用 request 获取用户提交的数据
- 5、注意请求转发和重定向的区别
- 6、重写 init 方法一定要加 super.init(cofnig);
- 7、Javax.servlet.http 包不在 JDK 核心类库中，因此需要将 Tomcat 安装目录 lib 子目录中的 servlet-api.jar 文件复制到 Tomcat 服务器所使用的 JDK 的扩展目录中，比如，复制到 D:\jdk1.7\jre\lib\ext 中
- 8、Servlet 的工作原理

## 第六章

- 1、MVC 模式的核心思想是有效的组合“视图”、“模型”和“控制器”。在 JSP 技术中，视图是一个或多个 JSP 页面，其作用主要是向控制器提交必要的数据和为模型提供数据显示；模型是一个或多个 JavaBean 对象，用于存储数据；控制器是一个或多个 servlet 对象，根据视图提交的要求进行数据处理操作，并将有关的结果存储到 Javabeen 中，然后 servlet 使用重定向或转发方式请求视图中的某个 JSP 页面更新显示。
- 2、在 MVC 模式中，模型也可以由控制器负责创建和初始化
- 3、一定要注意模型的生命周期：Request、Session、Apppication
- 4、当用 servlet 创建 JavaBean 时，就可以使用 JavaBean 类的带参数构造方法，类的方法命名继续保留“get”规则，但可以不遵守“set”规则
- 5、Request 周期的 JavaBean, Servlet 只能使用请求转发显示 JavaBean 中的数据
- 6、MVC 模式下 JSP 页面中使用 JavaBean 的时候 Type 和 Class 的区别

## 第七章

- 1、JSP 使用 JDBC 提供的 API 和数据库进行交互信息。JDBC 技术在数据库开发中占有很重要的地位，JDBC 操作不同的数据库仅仅是连接方式上的差异而已，使用 JDBC 的应用程序一旦和数据库建立连接，就可以使用 JDBC 提供的 API 操作数据库。
- 2、当查询 ResultSet 对象中的数据时，不可以关闭和数据库的连接。
- 3、使用 PreparedStatement 对象可以提高操作数据库的效率
- 4、MySQL 的命令行语句，如何启动、如何创建数据、如何创建表、删除数据库和表、更新数据
- 5、JDBC 和数据库连接的两个步骤：加载驱动、建立链接
- 6、加载 MySQL 驱动 Class.forName("com.mysql.jdbc.Driver");
- 7、DriverManager 类、Connection 接口、Statement 接口、PreparedStatement 接口、ResultSet 接口的作用
- 8、通过 JDC 操作数据库的步骤
- 9、ResultSet 结果集可以使用 getXxx 方法获得字段值 (列值)，将位置索引 (第一列使用 1，第二列使用 2 等等) 或列名传递给 getXxx 方法的参数即可

- 10、总是可以使用 `getString()` 返回字段值的串表示
- 11、元数据的作用
- 12、`type`、`concurrency` 的取值情况，以及分别是什么含义
- 13、如何使用结果集更新、插入记录
- 14、预处理语句中通配符的使用
- 15、事务操作的几个步骤

## 第八章

- 1、输入流的指向称为源，程序从指向源的输入流中读取源中的数据。而输出流的指向是数据要去的目的地，程序通过向输出流中写入数据把信息送往目的地。
- 2、`FileInputStream` 和 `FileReader` 流都顺序地读取文件，只要不关闭流，每次调用 `read` 方法就顺序地读取源中其余的内容，直到源的末尾或流被关闭。二者的区别是，`FileInputStream` 流以字节 (byte) 为单位读取文件；`FileReader` 流以字符 (char) 为单位读取文件。
- 3、`FileOutputStream` 流和 `FileWriter` 顺序地写文件，只要不关闭流，每次调用 `writer` 方法就顺序地向输出流写入内容，直到流被关闭。二者的区别是，`FileOutputStream` 流以字节 (byte) 为单位写文件；`FileWriter` 流以字符 (char) 为单位写文件。
- 4、`RandomAccessFile` 流的指向既可以作为源也可以作为目的地，在读写文件时可以调用 `seek` 方法改变读写位置。
- 5、`File` 类的对象主要用来获取文件本身的一些信息，例如文件所在的目录、文件的长度、文件读写权限等，不涉及对文件的读写操作
- 6、使用输入流 4 个基本步骤
- 7、使用输出流 4 个基本步骤
- 8、缓冲流是如何使用

```
FileReader inOne=new FileReader("studen.txt");    //创建字符输入流
BufferedReader inTwo=new BufferedReader(inOne);    //创建缓冲输入流
FileWriter tofile=new FileWriter("hello.txt");    //创建字符输出流
BufferWriter out=new BufferWriter(tofile);    //创建缓冲输入流
```
- 9、字节输出流如何选择具有刷新功能的构造方法
- 10、文件上传下载的时候是如何获得输入、输出流的

## 第九章

- 1、XML 文件是由标记构成的文本文件。XML 文件有且仅有一个根标记，其他标记都必须封装在根标记中。文件的标记必须是树型结构，非空标记必须由“开始标记”与“结束标记”组成，空标记没有“开始标记”和“结束标记”。
- 2、DOM 解析器在内存中按树型结构组织数据，DOM 解析器通过读入 XML 文件在内存中建立一棵“树”，XML 文件的标记、标记的文本内容都会和内存中“树”的某个节点相对应。
- 3、SAX 解析器根据从文件中解析出的数据产生相应的事件，并报告这个事件给事件处理器，事件处理器就会处理所发现的数据。
- 4、通过将 XML 文件和一个 CSS 样式表文件相关联，可以方便地显示 XML 文件中的标记所含有文本，是如何关联的？`xml-stylesheet`
- 5、XML 标记名称区分大小写

- 6、 XML 文件是由标记构成的文本文件。标记的名称可以由字母、数字、下划线“\_”、点号“.”或连字符“-”组成，但必须以字母或下划线开头
- 7、 CSS 样式表的使用方式