**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:<%! %> 、

<% %>

1. 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，也和容易错。
2. 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。
3. JSP的注释在客户端不可见，HTML的可见
4. Page指令标记的contentType只能指定一个值
5. 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. H**TTP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. Javabean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Javabean。
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. Javabean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Javabean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Javabean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. FileOutStream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，FileOutStream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**

**BufferReader inTwo=new BufferReader(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**

1. 字节输出流如何选择具有刷新功能的构造方法
2. 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的文本，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