```
<Form action="" method=post name=form1>
  <Select name="fileName"  value="A.txt">
    <Option value="A.txt"> A.txt
    <Option value="B.txt"> B.txt
    <Option value="C.txt"> C.txt
  </Select>
  <Input type="submit" name="sub" value="确定">
</FORM>
  <jsp:setProperty  name= "test"  property="fileName" param="fileName" />
<BR>试题内容如下:
<BR> <jsp:getProperty  name= "test"  property="testContent"  /> <BR>
<FORM action="" method=post name=form2>
  在文本框输入全部题目的答案，答案之间不允许有空格:
  <BR><Input type=text name="selection" size=80>
  <BR><Input type=submit value="提交">
</FORM>
  <jsp:setProperty  name= "test"  property="selection"  />
<BR>试题的正确答案:  <jsp:getProperty  name= "test"  property="correctAnswer"  />
<BR>您提交的答案:  <jsp:getProperty  name= "test"  property="selection"  />
<BR>您的分数:  <jsp:getProperty  name= "test"  property="score"  />
</BODY></HTML>
```

6.5　文件上传

Web 应用经常提供文件上传功能，本节学习怎样使用 RandomAccessFile 类提供的功能来实现文件上传。

RandomAccessFile 类创建的流与前面的输入流、输出流不同，RandomAccessFile 类既不是输入流类 InputStream 的子类，也不是输出流类 OutputStram 的子类。习惯上，仍然称 RandomAccessFile 类创建的对象为一个流，RandomAccessFile 流的指向既可以作为源，也可以作为目的地。换句话说，当我们想对一个文件进行读写操作时，可以创建一个指向该文件的 RandomAccessFile 流，这样既可以从这个流中读取这个文件的数据，也通过这个流写入数据给这个文件。

RandomAccessFile 类的两个构造方法如下：

- RandomAccessFile(String name, String mode) ——参数 name 用来确定一个文件名，给出创建的流的源（也是流目的地），参数 mode 取“r”（只读）或“rw”（可读写），决定创建的流对文件的访问权。
- RandomAccessFile(File file, String mode) ——参数 file 是一个 File 对象，给出创建的流的源（也是流目的地），参数 mode 取“r”（只读）或“rw”（可读写），决定创建的流对文件的访问权。创建对象时应捕获 IOException 异常。

RandomAccessFile 类中有一个方法 seek(long a)，用来移动 RandomAccessFile 流指向的文件的指针，参数 a 确定文件指针距离文件开头的字节位置。流还可以调用 getFilePointer()方法获取当前文件的指针的位置（RandomAccessFile 类的一些方法见表 6.1），RandomAccessFile 流对文件的读写比顺序读写的文件输入/输出流更灵活。

表 6.1　RandomAccessFile 类的常用方法

方　法	描　述	方　法	描　述
close()	关闭文件	getFilePointer()	获取文件指针的位置
length()	获取文件的长度	read()	从文件中读取一个字节的数据
readBoolean()	从文件中读取一个布尔值，0 代表 false，其他值代表 true	readDouble()	从文件中读取一个双精度浮点值(8 字节)
readByte()	从文件中读取一个字节	readChar()	从文件中读取一个字符（2 字节）
readFloat()	从文件中读取一个单精度浮点值(4 字节)	readFully(byte b[])	读 b.length 字节放入数组 b，完全填满该数组
readInt()	从文件中读取一个 int 值（4 字节）	readLine()	从文件中读取一个文本行
readlong()	从文件中读取一个长型值（8 字节）	readShort()	从文件中读取一个短型值（2 字节）
readUTF()	从文件中读取一个 UTF 字符串	seek()	定位文件指针在文件中的位置
setLength(long newlength)	设置文件的长度	skipBytes(int n)	在文件中跳过给定数量的字节
write(byte b[])	写 b.length 个字节到文件	writeBoolean(boolean v)	把一个布尔值作为单字节值写入文件
writeByte(int v)	向文件写入一个字节	writeBytes(String s)	向文件写入一个字符串
writeChar(char c)	向文件写入一个字符	writeChars(String s)	向文件写入一个作为字符数据的字符串
writeDouble(double v)	向文件写入一个双精度浮点值	writeFloat(float v)	向文件写入一个单精度浮点值
writeInt(int v)	向文件写入一个 int 值	writeLong(long v)	向文件写入一个长型 int 值
writeShort(int v)	向文件写入一个短型 int 值	writeUTF(String s)	写入一个 UTF 字符串

JSP 页面提供 File 类型的表单，File 类型的表单可以让用户选择要上传的文件。File 类型表单的格式如下：

```
<FORM action= "接受上传文件的页面"  method= "post"  enctype=" multipart/form-data"
<Input type= "File"  name= "参数名字"  >
</FORM>
```

bean 负责将用户选择的文件上传到服务器。bean 可以让内置对象 request 调用方法 getInputStream()获得一个输入流，通过这个输入流读入客户上传的全部信息，包括文件的内容和表单域的信息。bean 可以从上传的全部信息中分离出文件的内容，并保存在服务器上。按照 HTTP，文件表单提交的信息中，前 4 行和后 5 行是表单本身的信息，中间部分才是客户提交的文件的内容。bean 通过使用 RandomAccessFile 流获取文件的内容，即去掉表单的信息。首先，bean 将客户提交的全部信息保存为一个临时文件，该文件的名字是客户的 session 对象的 id（不同客户的这个 id 是不同的），然后读取该临时文件的第 2 行，这一行中含有客户上传的文件的名字，获取这个名字，再获取第 4 行结束的位置，以及倒数第 6 行的结束位置，因为这两个位置之间的内容是上传文件的内容，然后将这部分内容存入文件，该文件的名字与客户上传的文件的名字保持一致。最后删除临时文件。

【例 6-7】

创建 bean 的源文件如下：

UpFile.java

```
package tom.jiafei;
import java.io.*;
import javax.servlet.http.*;
```

```
public class UpFile {
    HttpServletRequest request;
    HttpSession session;
    String upFileMessage="";
    public void setRequest(HttpServletRequest request) {
        this.request=request;
    }
    public void setSession(HttpSession session) {
        this.session=session;
    }
    public String getUpFileMessage() {
        String fileName=null;
        try{  String tempFileName=(String)session.getId();       //客户的 session 的 id
              File f1=new File("D:/apache-tomcat-5.5.20/webapps/chaper6",tempFileName);
              FileOutputStream o=new FileOutputStream(f1);
              InputStream in=request.getInputStream();
              byte b[]=new byte[10000];
              int n;
              while( (n=in.read(b))!=-1) {                       //将客户上传的全部信息存入
                  o.write(b,0,n);
              }
              o.close();
              in.close();
              RandomAccessFile random=new RandomAccessFile(f1,"r");
              int second=1;                                      //读出 f1 的第 2 行，析取出上传文件的名
              String secondLine=null;
              while(second<=2) {
                   secondLine=random.readLine();
                   second++;
              }
              //获取第 2 行中目录符号'\'最后出现的位置
              int position=secondLine.lastIndexOf('\\');
              //客户上传的文件的名字是:
              fileName=secondLine.substring(position+1,secondLine.length()-1);
              byte   cc[]=fileName.getBytes("ISO-8859-1");
              fileName=new String(cc);
              session.setAttribute("Name",fileName);             //供 show.jsp 页面使用
              random.seek(0);                                    //再定位到文件 f1 的开头
              //获取第 4 行回车符号的位置
              long   forthEndPosition=0;
              int forth=1;
              while((n=random.readByte())!=-1&&(forth<=4)) {
                   if(n=='\n') {
                        forthEndPosition=random.getFilePointer();
                        forth++;
                   }
              }
              //根据客户上传文件的名字，将该文件存入磁盘
              File f2= new File("D:/apache-tomcat-5.5.20/webapps/chaper6",fileName);
              RandomAccessFile random2=new RandomAccessFile(f2,"rw");
              //确定出文件 f1 中包含客户上传的文件的内容的最后位置，即倒数第 6 行
```

```
                    random.seek(random.length());
                    long endPosition=random.getFilePointer();
                    long mark=endPosition;
                    int j=1;
                    while((mark>=0)&&(j<=6)) {
                        mark--;
                        random.seek(mark);
                        n=random.readByte();
                        if(n=='\n') {
                            endPosition=random.getFilePointer();
                            j++;
                        }
                    }
                    //将 random 流指向文件 f1 的第 4 行结束的位置
                    random.seek(forthEndPosition);
                    long startPoint=random.getFilePointer();
                    //从 f1 读出客户上传的文件存入 f2(读取从第 4 行结束位置和倒数第 6 行之间的内容)
                    while(startPoint<endPosition-1) {
                        n=random.readByte();
                        random2.write(n);
                        startPoint=random.getFilePointer();
                    }
                    random2.close();
                    random.close();
                    f1.delete();                                    //删除临时文件
                    upFileMessage=fileName+" Successfully UpLoad";
                    return upFileMessage;
                }
                catch(Exception exp) {
                    if(fileName!=null) {
                        upFileMessage=fileName+" Fail to UpLoad";
                        return upFileMessage;
                    }
                    else {
                        upFileMessage="";
                        return upFileMessage;
                    }
                }
            }
        }
```

JSP 页面文件如下：

upfile.jsp

```
<%@ page contentType="text/html;charset=GB2312" %>
<%@ page import="tom.jiafei.UpFile" %>
<jsp:useBean id="upFile" class="tom.jiafei.UpFile" scope="session" />
<HTML><BODY> <P>选择要上传的文件：<BR>
  <FORM action="" method="post" enctype="multipart/form-data">
     <Input type=FILE name="boy" size="45">
     <BR> <Input type="submit" name ="g" value="提交">
  </FORM>
```

```
    <%   upFile.setRequest(request);
         upFile.setSession(session);
    %>
    <jsp:getProperty   name="upFile" property="upFileMessage"/>
    <P>如果上传的是图像文件，可单击超链接查看图像:
    <BR><A href="show.jsp"> 查看图像</A>
  </BODY></HTML>
```

show.jsp

```
  <%@ page contentType="text/html;Charset=GB2312" %>
  <jsp:useBean id="upFile" class="tom.jiafei.UpFile" scope="session" />
  <HTML><BODY>
    <%   String pic=(String)session.getAttribute("Name");
         out.print(pic);
         out.print("<image src="+pic+">");
    %>
  </BODY></HTML>
```

6.6　文件下载

Tomcat 5.5 服务器提供了方便的下载功能。只需让内置对象 response 调用方法 setHeader，添加下载的头给客户的浏览器即可，浏览器收到该头后就会打开相应的下载对话框。response 调用 setHeade()方法添加下载头的格式如下：

```
response.setHeader("Content-disposition","attachment;filename="下载的文件名字");
```

【例 6-8】　客户在 JSP 页面选择一个文件，JSP 页面调用 bean 下载所选择的文件。

创建 bean 的源文件如下：

DownLoadFile.java

```
package tom.jiafei;
import java.io.*;
import javax.servlet.http.*;
public class DownLoadFile {
    HttpServletResponse response;
    String fileName;
    public void setResponse(HttpServletResponse response) {
        this.response=response;
    }
    public String getFileName() {
        return fileName;
    }
    public void setFileName(String s) {
        fileName=s;
        File fileLoad=new File("F:/2000",fileName);
        //客户使用下载文件的对话框:
        response.setHeader("Content-disposition","attachment;filename="+fileName);
        //下载的文件:
        try{
            //读取文件,并发送给用户下载:
            FileInputStream in=new FileInputStream(fileLoad);
            OutputStream o=response.getOutputStream();
```

```
                int n=0;
                byte b[]=new byte[500];
                while((n=in.read(b))!=-1)
                    o.write(b,0,n);
                o.close();
                in.close();
            }
            catch(Exception exp){ }
        }
    }
```

JSP 页面文件如下：

downfile.jsp

```
<%@ page contentType="text/html;charset=GB2312" %>
<%@ page import="tom.jiafei.DownLoadFile" %>
<%@ page import="java.io.*" %>
<jsp:useBean id="downFile" class="tom.jiafei.DownLoadFile" scope="page" />
<HTML><BODY> <P>选择要下载的文件:
  <FORM action="">
    <Select name="fileName" >
       <Option value="book.zip">book.zip
       <Option value="A.java">A.java
       <Option value="B.jsp">B.jsp
    </Select>
    <Input type="submit" value="提交你的选择" name="submit">
  </FORM>
  <%   downFile.setResponse(response);
  %>
<jsp:setProperty   name="downFile" property="fileName" param="fileName"/>
</BODY></HTML>
```

习 题 6

1. File 类对象的主要作用是什么？
2. File 类对象怎样获取文件的长度？
3. RandomAccessFile 类创建的流在读写文件时有什么特点？
4. 按行读取文件的关键技术是什么？
5. 编写 2 个 JSP 页面 a.jsp 和 b.jsp。a.jsp 通过表单提交一个 Web 服务目录的名字给 b.jsp，b.jsp 根据 a.jsp 提交的名字调用一个 bean，在 Tomcat 服务器创建一个 Web 服务目录。
6. 编写 2 个 JSP 页面 input.jsp 和 read.jsp。input.jsp 通过表单提交一个目录和该目录下的一个文件名给 read.jsp，read.jsp 根据 input.jsp 提交的目录和文件名调用一个 bean 读取文件的内容。