

```

package com.jiang.UpAndDown;

import org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.FileItemFactory;
import org.apache.commons.fileupload.FileUploadBase;
import org.apache.commons.fileupload.FileUploadException;
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.File;
import java.io.IOException;
import java.util.Iterator;
import java.util.List;

public class UploadServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        request.setCharacterEncoding("GBK");
        response.setCharacterEncoding("GBK");
        response.setContentType("text/html;charset=GBK");
        //上传
        boolean isMutipart= ServletFileUpload.isMultipartContent(request);
        if(isMutipart){//判断前台form是否有mutipart属性
            DiskFileItemFactory factory=new DiskFileItemFactory();
            ServletFileUpload upload=new ServletFileUpload(factory);

            //限制条件放在parse之前
            //控制上传文件时 用到的临时文件的大小(用DiskFileItemFactory来控制)
            factory.setSizeThreshold(10240); //设置临时的缓冲文件大小为10240B
            factory.setRepository(new File("D:\\docs")); //设置临时目录
            //控制上传单个文件的大小 20kb
            upload.setSizeMax(20480); //字节B
            try {
                List<FileItem> items=upload.parseRequest(request); //通过解析request获
得全部字段，并保留到items集合中
                Iterator<FileItem> iter= items.iterator();
                while(iter.hasNext()) {
                    FileItem item = iter.next();
                    String itemName = item.getFieldName();
                    int sno=0;

```

```

String sname=null;
//判断前台是否是普通form表达属性 (sno sname) ,还是文件字段
if (item.isFormField()) {
    if (itemName.equals("sno")) {
        sno=Integer.parseInt(item.getString("GBK"));
    } else if (itemName.equals("sname")) {
        sname=item.getString("GBK");
    } else {
        System.out.println("普通字段");
    }
} else { //picture
    //文件上传
    //文件名 getFieldName是获取普通字段的Name值
    //getName是获取文件名
    String name=item.getName();
    //获取文件内容(并上传)
    //定义文件上传的内容(一般为服务器)
    String
path=request.getSession().getServletContext().getRealPath("upload");
    String ext=name.substring(name.indexOf(".")+1);
    if(!(ext.equals("png")||ext.equals("gif")||ext.equals("jpg"))){
        System.out.println("图片类型有误, 请上传图片");
    }
    File file=new File(path,name);
    item.write(file);

    System.out.println("上传成功");
    return ;
}
}
} catch (FileUploadBase.SizeLimitExceededException e){
    e.printStackTrace();
} catch (FileUploadException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
}
}

protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    doPost(request, response);
}
}

```

