**import** java.io.ByteArrayOutputStream;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.io.OutputStream;

**import** java.util.Iterator;

**import** java.util.Map;

**import** javax.servlet.http.HttpServletResponse;

**import** org.apache.poi.hwpf.HWPFDocument;

**import** org.apache.poi.hwpf.model.FieldsDocumentPart;

**import** org.apache.poi.hwpf.usermodel.Field;

**import** org.apache.poi.hwpf.usermodel.Fields;

**import** org.apache.poi.hwpf.usermodel.Range;

**import** org.apache.poi.hwpf.usermodel.Table;

**import** org.apache.poi.hwpf.usermodel.TableIterator;

**import** org.apache.poi.hwpf.usermodel.TableRow;

**public** **class** WordUtil {

**public** **static** **void** readwriteWord(String filePath, String downPath,

Map<String, String> map, **int**[] num, String downFileName) {

//读取word模板

FileInputStream in = **null**;

**try** {

in = **new** FileInputStream(**new** File(filePath));

} **catch** (FileNotFoundException e1) {

e1.printStackTrace();

}

HWPFDocument hdt = **null**;

**try** {

hdt = **new** HWPFDocument(in);

} **catch** (IOException e1) {

e1.printStackTrace();

}

Fields fields = hdt.getFields();

Iterator<Field> it = fields.getFields(FieldsDocumentPart.*MAIN*)

.iterator();

**while** (it.hasNext()) {

System.*out*.println(it.next().getType());

}

//读取word表格内容

**try** {

Range range = hdt.getRange();//得到文档的读取范围

TableIterator it2 = **new** TableIterator(range);

//迭代文档中的表格

**int** tabCount = 0;

**while** (it2.hasNext()) {

//System.out.println(" 第几个表格 "+tabCount);

//System.out.println(num[tabCount] +" 行");

Table tb2 = (Table) it2.next();

//迭代行，默认从0开始

**for** (**int** i = 0; i < tb2.numRows(); i++) {

TableRow tr = tb2.getRow(i);

// System.out.println(" fu "+num[tabCount] +" 行");

**if** (num[tabCount] < i && i < 7) {

tr.delete();

}

} //end for

tabCount++;

} //end while

//替换word表格内容

**for** (Map.Entry<String, String> entry : map.entrySet()) {

range.replaceText("$" + entry.getKey().trim() + "$", entry

.getValue());

}

// System.out.println("替换后------------："+range.text().trim());

} **catch** (Exception e) {

e.printStackTrace();

}

// System.out.println("--------------------------------------------------------------------------------------");

ByteArrayOutputStream ostream = **new** ByteArrayOutputStream();

String fileName = downFileName;

fileName += ".doc";

String pathAndName = downPath + fileName;

File file = **new** File(pathAndName);

**if** (file.canRead()) {

file.delete();

}

FileOutputStream out = **null**;

**try** {

out = **new** FileOutputStream(pathAndName, **true**);

} **catch** (FileNotFoundException e) {

e.printStackTrace();

}

**try** {

hdt.write(ostream);

} **catch** (IOException e) {

e.printStackTrace();

}

//输出字节流

**try** {

out.write(ostream.toByteArray());

} **catch** (IOException e) {

e.printStackTrace();

}

**try** {

out.close();

} **catch** (IOException e) {

e.printStackTrace();

}

**try** {

ostream.close();

} **catch** (IOException e) {

e.printStackTrace();

}

}

/\*\*

\* 实现对word读取和修改操作(输出文件流下载方式)

\* **@param** response 响应,设置生成的文件类型,文件头编码方式和文件名,以及输出

\* **@param** filePath word模板路径和名称

\* **@param** map 待填充的数据，从数据库读取

\*/

**public** **static** **void** readwriteWord(HttpServletResponse response,

String filePath, Map<String, String> map) {

//读取word模板文件

//String fileDir = new File(base.getFile(),"http://www.cnblogs.com/http://www.cnblogs.com/../doc/").getCanonicalPath();

//FileInputStream in = new FileInputStream(new File(fileDir+"/laokboke.doc"));

FileInputStream in;

HWPFDocument hdt = **null**;

**try** {

in = **new** FileInputStream(**new** File(filePath));

hdt = **new** HWPFDocument(in);

} **catch** (Exception e1) {

e1.printStackTrace();

}

Fields fields = hdt.getFields();

Iterator<Field> it = fields.getFields(FieldsDocumentPart.*MAIN*)

.iterator();

**while** (it.hasNext()) {

System.*out*.println(it.next().getType());

}

//替换读取到的word模板内容的指定字段

Range range = hdt.getRange();

**for** (Map.Entry<String, String> entry : map.entrySet()) {

range.replaceText("$" + entry.getKey() + "$", entry.getValue());

}

//输出word内容文件流，提供下载

response.reset();

response.setContentType("application/x-msdownload");

String fileName = "" + System.*currentTimeMillis*() + ".doc";

response.addHeader("Content-Disposition", "attachment; filename="

+ fileName);

ByteArrayOutputStream ostream = **new** ByteArrayOutputStream();

OutputStream servletOS = **null**;

**try** {

servletOS = response.getOutputStream();

hdt.write(ostream);

servletOS.write(ostream.toByteArray());

servletOS.flush();

servletOS.close();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

注：以上代码需要poi包，www.findjar.com 可以下载。