软件复用课程设计

Client-Server 应用程序

复用文档

File构件

小组： Team10

成员： 谢志杰 1352975

计鹏玥 1352914

王思尧 1352896

谢明玥 1352937

徐锦程 1353012

[1． 构件简介 1](#_Toc13922)

[2． 接口说明 1](#_Toc27135)

[2.1 SaveToFile.java 1](#_Toc26796)

[(1) SaveToFile() 1](#_Toc3597)

[(2) SaveToFile(String s) 1](#_Toc12257)

[(3) getBufferedWriter() 2](#_Toc29163)

[(4) write(String str) 2](#_Toc23786)

[(5) getDirectoryPath() 2](#_Toc20870)

[(6) finalized() 3](#_Toc14363)

[3．Demo 3](#_Toc18319)

# 构件简介

|  |  |
| --- | --- |
|  | **获得配置信息 － File** |
| 构件介绍 | 这一构件的主要功能是保存客户端与服务端收到的所有消息到文件，且对保存文件的格式不限，存储路径也可以自己配置。 |
| 功能说明 | 用户可以使用该构件来保存消息到文件。 |

# 接口说明

## 2.1 SaveToFile.java

注意：应先导入File包。

### (1) SaveToFile()

用于获取配置信息所指定的文件路径，无参数与返回值。

**作用描述：**

实例化存储消息到文件的对象，存储路径从配置文件中来获取。

**使用方法：**

对于一个实例化的对象，可以直接调用该方法，该方法没有输入参数和返回值，形如：

SaveToFile saveToFile= new SaveToFile();

**注意事项：**

此方法是一个构造函数，在调用该方法前必须先有一个非空的配置文件。

### (2) SaveToFile(String s)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| s | String | 文件的路径 |

**作用描述：**

实例化存储消息到文件的对象，存储路径从传入的字符串来获取。

**使用方法：**

传入一个字符串，表示存储文件的路径，将文件保存到该路径下，形如：

SaveToFile saveToFile= new SaveToFile(“D:\\”);

**注意事项：**

传入的字符串表示的存储路径必须存在，否则会抛出异常。

### (3) getBufferedWriter()

获取输出流，并创建文件。

**作用描述：**

该方法获取当前时间和文路径作为保存文件的目录路径名，并判断该文件是否已经存在，如果不存在，则可以创建文件，否则显示创建文件失败。

**使用方法：**

创建指定的文件以便写入输出流中的内容，形如：

saveToFile.getBufferedWriter();

**注意事项：**

文件路径需要存在且有效。

### (4) write(String str)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| str | String | 准备写入文件的内容 |

**作用描述：**

将获得的字符串流写入文件。

**使用方法：**

对于字符串流进行文件写入，且每写入一条消息就会进行换行，形如：

saveToFile.write(“hello world”);

### (5) getDirectoryPath()

|  |  |  |
| --- | --- | --- |
| 返回值 | 类型 | 含义 |
| directoryPath | String | 文件夹路径 |

**作用描述：**

获取文件夹目录，并返回该字符串值。

**使用方法：**

对于实例化的保存文件对象，可以直接调用该方法，获取到文件目录，该值为字符串类型，形如：

String dirPath = saveToFile.getdirectoryPath();

### (6) finalized()

关闭文件输入流。

**作用描述：**

在文件写入完毕后关闭文件输入流。

**使用方法：**

当文件写入完成后，执行该函数以关闭文件输入流，若为完成就关闭，则会报错，形如：

saveToFile.finalized();

**注意事项：**

文件写入完成后才可以关闭文件输入流。

## 2.2 FileUtils.java

注意：应先导入File包。

### (1) filenameWithoutExt(String fileNameWithExt)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| fileNameWithExt | String | 包含扩展文件名的文件名称 |

|  |  |  |
| --- | --- | --- |
| 返回值 | 类型 | 含义 |
| fileNameWithoutExt | String | 不包含扩展文件名的文件名称 |

**作用描述：**

去除文件的扩展名。

**使用方法：**

对于一个实例化的对象，可以直接调用该方法，参数为包含扩展文件名的文件名称，返回值为不包含扩展文件名的文件名称，形如：

String filenameWithoutExt = fileUtils.filenameWithoutExt(“fileSrc.txt”);

**注意事项：**

参数代表的文件名称必须包含扩展名。

### (2) renameFileExtension(String source, String newExtension)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| source | String | 文件源 |
| newExtension | String | 新的扩展文件名 |

|  |  |  |
| --- | --- | --- |
| 返回值 | 类型 | 含义 |
| File(source).renameTo(new File(target)) | Boolean | 判断是否对文件扩展名重命名成功 |

**作用描述：**

更改文件的扩展名。

**使用方法：**

对于一个实例化的对象，可以直接调用该方法，参数为文件源和新的文件扩展名，返回值判断了判断是否对文件扩展名重命名成功，形如：

Boolean renameFileExt = fileUtils.renameFileExtension(“D://fileSrc”, “pdf”);

**注意事项：**

文件源合法有效，且文件扩展名已经事先被移除。

### (3) getFileExtension(String f)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| f | String | 最近的文件 |

|  |  |  |
| --- | --- | --- |
| 返回值 | 类型 | 含义 |
| ext | String | 新的扩展文件名 |

**作用描述：**

获得最近的文件扩展名。

**使用方法：**

对于一个实例化的对象，可以直接调用该方法，参数为最近使用文件名，返回值为新的扩展文件名，形如：

String currentExtension = getFileExtension(source);

**注意事项：**

参数f代表的文件必须具有后缀名。

# 3．Demo