

快速开发时可以使用的Java文件工具方法 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/289/2021_2022__E5_BF_AB_E9_80_9F_E5_BC_80_E5_c104_289956.htm

Java I/O系统被设计为功能强大并便于扩展，但是当将一个简单的文件加载到一个字符串时它有时候显得太强大了。如果你已经厌倦了文件处理代码并想提高开发速度，使用简单的文件加载和文件保存程序。首先创建一个静态(static)的方法，给出一个字符串的文件名，将把文件的内容加载到一个字符串或者返回一个null表明什么也得不到。可以读入的文件的大小最多为MAXINT，因为它被加载到一个数组中。大文件的内容不能被加载到内容中。代码如下：/** * 加载给定文件名的内容，如果文件不可加载返回null。 */ static public

```
String  
loadFile(String filename) { File file = new File( filename ). try { Reader rdr = new  
FileReader( file ). long sz = file.length(). // can only read in things of  
MAXINT length char[] ch = new char[(int)sz]. rdr.read(ch).  
rdr.close(). return new  
String(ch). } catch(IOException ioe) { return null.  
}}
```

因为这段代码是设计作为类库代码的，因此当有IOException抛出时静态方法没有打印堆栈跟踪（stack trace）或者输出到日志，它也没有抛出一个异常，目的也是更快的开发速度。这是一种折衷。第二个程序是用于保存的上面的那个方法的变种，获取一个字符串的文件名和要写入的字符串并将那个字符串作为文件保存。如果文件已经存在则覆

盖原来的内容。如果文件路径上的目录不存在则创建那些目录，最后返回true和false来表明操作是否成功。同样，字符串中的数据的内容是有限制的，并且缺少好的错误报告。抛出一个运行时异常或者使用内部的日志工具是好的解决方法。代码如下：/** * 将字符串的内容作为文件存入给定的文件中 * * @return完成操作时返回true */ static public boolean saveFile(java/lang/String.java.html" target="_blank">String filename, java/lang/String.java.html" target="_blank">String contents) { try { File file = new File(filename). if(file.getParent() != null) { new File(file.getParent()).mkdirs(). } Writer wtr = new FileWriter(file). char[] ch = contents.toCharArray(). wtr.write(ch). wtr.close(). return true. } catch(IOException ioe) { return false. } } 提高创建这些方法，代码的质量下降了，因为它不能再处理底层的東西。然而开发的分离和速度都提高了。增加一个FileException是提高质量的一个方法但是牺牲了开发的速度。当代码只需要良好而不是完美时拥有这样的类库类是很不错的。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com