

使用Java操作文本文件的方法详解 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E4_BD_BF_E7_94_A8Java_c104_144628.htm

最初Java是不支持对文本文件的处理的，为了弥补这个缺憾而引入了Reader和Writer两个类，这两个类都是抽象类，Writer中write(char[] ch,int off,int

length)，flush()和close()方法为抽象方法，Reader中read(char[] ch,int off,int length)和close()方法是抽象方法。子类应该分别实现他们。当我们读写文本文件的时候，采用Reader是非常方便的，比如FileReader，InputStreamReader和BufferedReader。

其中最重要的类是InputStreamReader，它是字节转换为字符的桥梁。你可以在构造器重指定编码的方式，如果不指定的话将采用底层操作系统的默认编码方式，例如GBK等。当使用FileReader读取文件的时候。

```
FileReader fr = new
```

```
FileReader("ming.txt"); int ch = 0; while((ch = fr.read()) != -1) {
```

```
System.out.print((char)ch); }
```

其中read()方法返回的是读取得下一个字符。当然你也可以使用read(char[] ch,int off,int length)这和处理二进制文件的时候类似，不多说了。如果使用

InputStreamReader来读取文件的时候 while((ch =

```
isr.read()) != -1) { System.out.print((char)ch); }
```

这和FileReader并没有什么区别，事实上在FileReader中的方法都是从InputStreamReader中继承过来的。read()方法是比较好费时间的，如果为了提高效率我们可以使用BufferedReader

对Reader进行包装，这样可以提高读取得速度，我们可以一行一行的读取文本，使用readLine()方法。

```
BufferedReader br =
```

```
new BufferedReader(new InputStreamReader(new
```

FileInputStream("ming.txt"))). String data = null. while((data = br.readLine())!=null) { System.out.println(data). } 当你明白了如何用Reader来读取文本文件的时候那么用Writer写文件同样非常简单。有一点需要注意，当你写文件的时候，为了提高效率，写入的数据会先放入缓冲区，然后写入文件。因此有时候你需要主动调用flush()方法。与上面对应的写文件的方法为：

```
FileWriter fw = new FileWriter("hello.txt"). String s = "hello world". fw.write(s,0,s.length()). fw.flush(). OutputStreamWriter osw = new OutputStreamWriter(new FileOutputStream("hello2.txt")). osw.write(s,0,s.length()). osw.flush(). PrintWriter pw = new PrintWriter(new OutputStreamWriter(new FileOutputStream("hello3.txt")),true). pw.println(s).
```

不要忘记用完流！下面是个小例子，帮助新手理解。其实有的时候java的IO系统是需要我们多记记的，不然哪天就生疏了。

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