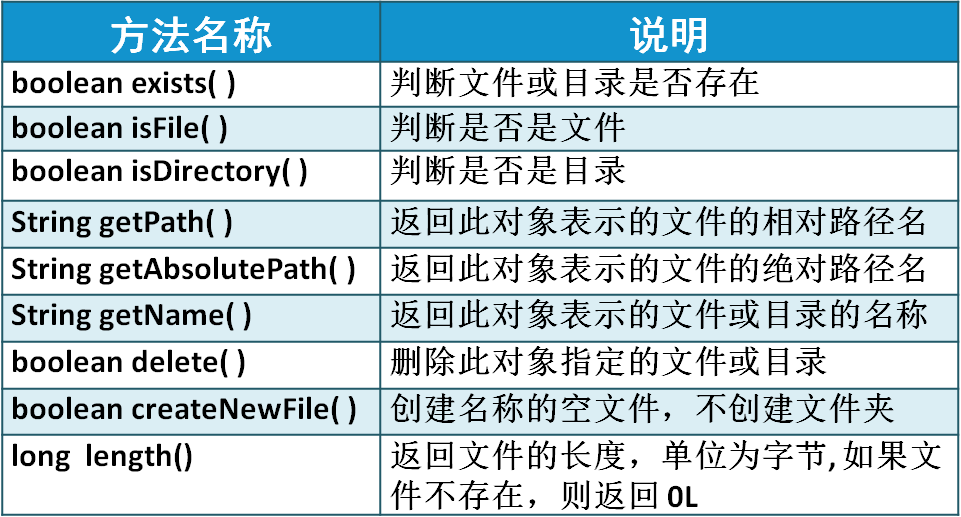
1. **使用File类操作文件或目录属性**

自行编写程序分别验证查看文件属性、创建和删除文件的实际效果。



**2、使用FileInputStream类读取文本文件**

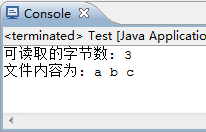
需求说明：按照如下的四个步骤，使用FileInputStream类读取文本文件,效果如下图所示。

（1）引入相关的类，java.io.IOException和java.io.FileInputStream；

（2）构造文件输入流FileInputStream 对象，比如为：FileInputStream fis= new FileInputStream("c:\\test.txt");；

（3）读取文本文件的数据，fis.available();fis.read();；

（4）关闭文件流对象：fis.close();；



**3、使用FileOutputStream类向文本文件写数据**

需求说明：按照如下的四个步骤，使用FileOutputStream类写文本文件,写入口程序执行效果如下图所示。

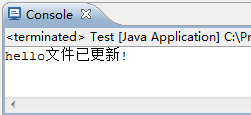
（1）引入相关的类，java.io.IOException和java.io.FileOutputStream；

（2）构造文件输出流FileOutputStream 对象，比如为：

FileOutputStream fos = new FileOutputStream("text.txt");；

（3）把数据写入文本文件：String str ="好好学习Java";byte[] words = str.getBytes();fos.write(words, 0, words.length);；

（4）关闭文件流对象：fos.close();；



**4、复制文本文件**

需求说明：文件“我的青春谁做主.txt”位于D盘根目录下，要求将此文件的内容，按照如下步骤复制到C:\myFile\my Prime.txt中，复制完成后输出如下图所示信息，检查文件是否已经写入目标位置。

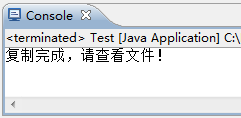
（1）创建FileInputStream输入流对,负责读取D:/ 我的青春谁做主.txt文件；

（2）创建FileOutputStream输出流对象；

（3）创建中转站数组byte[] words,存放每次读取的内容；

（4）通过while循环实现文件读取；

（5）操作完成后，记得关闭流；



**5、使用FileReader读取文本文件**

需求说明：按照如下的四个步骤，使用FileReader读取文件，文件内容为“这是作业5”,程序执行效果如下图所示。

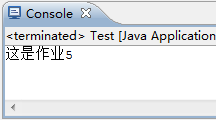
（1）引入相关的类，java.io.IOException和java.io.FileOutputStream和import java.io.FileReader;；

（2）创建FileReader对象，比如为：

Reader fr= new FileReader("D:\\myDoc\\简介.txt");；

（3）读取文本文件的数据，fr.read();；

（4）关闭相关的流对象：fr.close();；



**6、使用BufferedReader和FileReader读取文本文件**

需求说明：按照如下的四个步骤，使用BufferedReader和FileReader读取文本文件，文件内容为“这是hello”,程序执行效果如下图所示。

（1）引入相关的类：java.io.IOException和java.io.BufferedReader和import java.io.FileReader;；

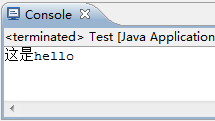
（2）构造BufferedReader对象和FileReader对象，比如：

Reader fr=new FileReader("C:\\myTest.txt ");

BufferedReader br=new BufferedReader(fr);；

（3）调用readLine()方法读取数据，br.readLine();；

（4）关闭文件流对象：br.close();fr.close();；



**7、使用FileWriter类向文本文件写数据**

需求说明：与字节流FileOutputStream类实现向文本文件写入数据步骤类似。按照如下的四个步骤，使用FileWriter类向文本文件写入“我热爱我的团队！”,程序执行后应当在“简介.txt”文件中包含如下图所示的内容。

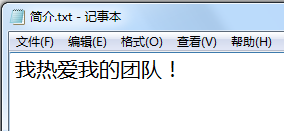
（1）引入相关的类：java.io.IOException和java.io.Reader和import java.io. FileWriter;；

（2）创建FileReader对象，比如：

Writer fw= new FileWriter("D:\\myDoc\\简介.txt");

（3）写文本文件，fw.write();；

（4）关闭相关的流对象：fw.close();；



**8、使用BufferedWriter和FileWriter类写文本文件**

需求说明：按照如下的四个步骤，使用BufferedWriter和FileWriter类写文本文件,程序执行后，读取该文件的效果如下图所示。

（1）引入相关的类：

import java.io.FileWriter;

import java.io.BufferedWriter ;

import java.io.IOException;；

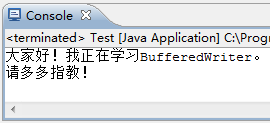
（2）构造BufferedWriter对象和FileWriter对象，比如：

FileWriter fw=new FileWriter("C:\\myTest.txt");

BufferedWriter bw=new BufferedWriter(fw);；

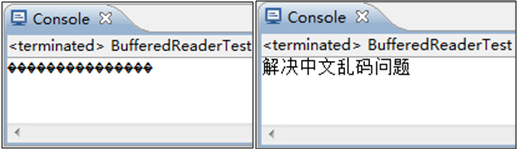
（3）调用write()方法写数据据，bw.write("hello");；

（4）流对象的清空和关闭flush()和close()：bw.flush();fw.close();；



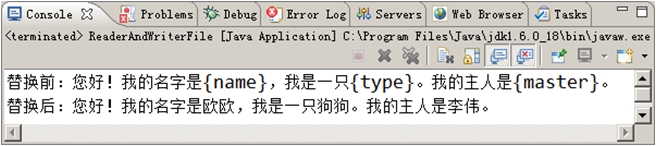
**9、解决中文乱码**

需求说明：读取包含中文的文本文件时，可能出现中文乱码，使用InputStreamReader并设置编码格式解决中文乱码问题，效果如图所示。



**10、替换文本文件内容**

需求说明：读取模板文件C:\pet.template的内容，按照pet.template的模板格式保存宠物数据到文本文件，即把{name}、{type}、{master}替换为具体的宠物信息，将替换后的内容写入到C:\myDoc\pet.txt中，程序执行效果如图所示。



**11、二进制文件的读写**

需求说明：与字节流FileOutputStream类实现文本文件读取步骤极其相似。按照如下的四个步骤，使用 DataOutputStream写二进制文件,程序执行后，两个文件在硬盘的信息如下图所示。

（1）引入相关的类：

import java.io.FileOutputStream;

import java.io.DataOutputStream;

（2）构造数据输出流对象，比如：

FileOutputStream outFile=

new FileOutputStream("C:\\temp.class");

DataOutputStream out=

new DataOutputStream(outFile);

（3）调用write()方法写二进制文件的数据，out.write();；

（4）关闭数据输出流：out.close();；