* File类
* java.io.File类：文件和目录路径名的抽象表示形式，**与平台无关**。
* File能新建、删除、重命名文件和目录，但**File不能访问文件内容本身**。如果需要访问文件内容本身，则需要使用输入/输出流。
* File对象可以**作为参数传递给流的构造函数**。
* File是一个类，可以有构造器创建其对象。此对象对应着一个文件(.txt .avi .doc .ppt .mp3 .jpg)或文件目录。
* **File类常见构造方法：**
* **public File(String pathname)**
* 以pathname为路径创建File对象，可以是绝对路径或者相对路径，如果pathname是相对路径，则默认的当前路径在系统属性user.dir中存储。
* **public File(String parent,String child)**
* 以parent为父路径，child为子路径创建File对象。

注：File的静态属性String separator存储了当前系统的路径分隔符。

在UNIX中，此字段为‘/’，在Windows中，为‘\\’

* **File类常用方法：**
* 访问文件名：
* getName()
* getPath()
* getAbsoluteFile()
* getAbsolutePath()
* getParent()
* renameTo(File newName)

System.***out***.println(file1.getName());  
System.***out***.println(file1.getPath());  
System.***out***.println(file1.getAbsoluteFile());  
System.***out***.println(file1.getAbsolutePath());  
System.***out***.println(file1.getParent());

// file1重命名为file2的要求：file1一定存在，file2一定不存在  
boolean b = file1.renameTo(file2);  
System.***out***.println(b);

* 文件检测：
* exists()
* canWrite()
* canRead()
* isFile()
* isDirectory()

System.***out***.println(file2.exists());  
System.***out***.println(file2.canRead());  
System.***out***.println(file2.canWrite());  
System.***out***.println(file2.isFile());  
System.***out***.println(file2.isDirectory());

* 获取常规文件信息：
* lastModified()
* length()

System.***out***.println(file2.lastModified());  
System.***out***.println(file2.length());

* 文件操作相关：
* createNewFile()
* delete()

File file1 = new File("d:/io/helloworld.txt");  
System.***out***.println(file1.delete());  
  
if (!file1.exists()) {  
 boolean b = file1.createNewFile();  
 System.***out***.println(b);  
}

* 目录操作相关：
* mkDir()——创建一个文件目录，只有在上层文件目录存在的情况下，才能返回true。
* mkDirs()——创建一个文件目录，若上层文件目录不存在，一并创建。
* list()
* listFiles()

File file2 = new File("d:\\io\\io2");  
if (!file2.exists()) {  
 boolean b = file2.mkdir();  
 System.***out***.println(b);  
}

File file3 = new File("d:\\io");  
String[] strs = file3.list();  
for (int i = 0; i < strs.length; i++) {  
 System.***out***.println(strs[i]);  
}

File[] files = file3.listFiles();  
for (int i = 0; i < files.length; i++) {  
 System.***out***.println(files[i].getName());  
}