1. File类
   1. File类的使用非常的简单
      1. 只需要在实例化File类的时候指定路径就可

·File file = new File (“d:\\mldn.text”);

范例:

|  |
| --- |
| package com.company.JavaIo编程;  import java.io.File;  import java.io.IOException;  public class FileDemo {  public static void main(String[] args) throws IOException {  File file = new File("F:\\360MoveData\\Users\\john\\Desktop\\java学习\\java面向对象编程\\java\\java高级语言特性\\mldn.text");  if(file.exists()){  System.out.println("文件存在，删除文件");  System.out.println(file.delete());  }else{  System.out.println(file.createNewFile());  System.out.println("创建成功");  }  }  } |

* 1. 在实际的软件开发和运行过程中，我们往往在window系统上进行项目开发。而在项目部署的时候使用linux/Unix系统来进行项目的发布以保证生产环节的安全。

但是在不同的系统上分隔符号是不同的，因此file类提供了一个分隔符常量，File.separator

·File file = new File(“d:”+ File.separator+“mldn.text”)

注意:创建文件的时候我们应该确保父目录的存在。因此我们需要先判断父路径的存在！

·获取父路径:public File getPraentFile();

·创建目录:

- public Boolean mkdir() ：创建单级目录

-public Boolean mkdirs()：创建多级目录

|  |
| --- |
| if(!file.getParentFile().exists()){  file.getParentFile().mkdirs();  } |

1. 获取文件
   1. File类中也有可以获取内容的方法。
   2. 获取目录结构

范例：

|  |
| --- |
| package com.company.JavaIo编程;  import java.io.File;  import java.util.List;  /\*\*  \* @author JackWem  \*/  public class FileListDemo {  public static void main(String[] args) {  File file = new File("G:"+File.separator+"听力");  ListFiles(file);  }  public static void ListFiles(File file){  if (file.isDirectory()){  //列出目录中的所有内容  File[] result = file.listFiles();  if (result != null){  for (int i = 0; i < result.length; i++) {  ListFiles(result[i]);  }  }  }  System.out.println(file);  //file.delete();  }  } |

如果将打印文件名换成删除，那么就会批量删除该目录下的所有文件。

文件重命名:

file.renameTo(File dest)

范例:

|  |
| --- |
| package com.company.JavaIo编程;  import java.io.File;  public class ReFileNameDemo {  public static void main(String[] args) {  File file = new File("G:"+File.separator+"英雄时刻");  ReName(file);  }  private static void ReName(File file) {  if (file.isDirectory()){  File[] results = file.listFiles();  if (results != null){  for (int i = 0; i < results.length; i++) {  ReName(results[i]);  }  }  }else{  //获取文件名字（包含后缀）  String name = file.getName();  String fileName = null;  //判断是否有后缀  if (name.contains(".")) {  fileName = name.substring(0, name.lastIndexOf("."))+".txt";  }else {  //指定要更改的名字  fileName = name + ".txt";  }  File newfile = new File(file.getParentFile(), fileName);  file.renameTo(newfile);  System.out.println(newfile);  }  }  } |