# File

文件. 使用一个路径代表一个文件/文件夹.

## 构造方法

创建文件对象.

|  |
| --- |
| **[File](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/io/File.html" \l "File(java.io.File, java.lang.String))**([File](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/io/File.html" \o "java.io 中的类) parent, [String](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html" \o "java.lang 中的类) child)            根据 parent 抽象路径名和 child 路径名字符串创建一个新 File 实例。 |
| **[File](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/io/File.html" \l "File(java.lang.String))**([String](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html" \o "java.lang 中的类) pathname)            通过将给定路径名字符串转换为抽象路径名来创建一个新 File 实例。 |
| **[File](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/io/File.html" \l "File(java.lang.String, java.lang.String))**([String](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html" \o "java.lang 中的类) parent, [String](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html" \o "java.lang 中的类) child)            根据 parent 路径名字符串和 child 路径名字符串创建一个新 File 实例。 |

|  |
| --- |
| **package** com.zhiyou100;  **import** java.io.File;  */\*\*  \** ***@author*** *QiuShiju  \** ***@date*** *2021/1/5  \** ***@desc*** *File类的构造方法  \*/* **public class** Demo1 {   **public static void** main(String[] args) {  *constructor3*( );  }   **private static void** constructor3() {  *// 父级和子级路径拼接* File parent = **new** File(**"E:\\A"**);  File file = **new** File(parent, **"a.txt"**);   System.***out***.println(file.exists() );  }   **private static void** constructor2() {  *// File(String parent, String child)  // 根据 parent 路径名字符串和 child 路径名字符串创建一个新 File 实例。  // 即父级路径和子级路径拼接成完整路径,代表一个文件对象  // E:\A\a.txt* File f1 = **new** File(**"E:\\A"**, **"a.txt"**);   *// 判断文件是否存在* System.***out***.println(f1.exists() );  }   */\*\*  \* 构造方法1  \*/* **private static void** constructor1() {  *// File(String pathname)  // 绝对路径* File f1 = **new** File(**"E:\\A\\a.txt"**);   *// 相对路径:当前项目的根路径下.即E:\IdeaWorkSpace\day26\a.txt* File f2 = **new** File(**"a.txt"**);   *// ps: 路径代表的文件,不一定存在.* }  } |

## 创建功能

|  |
| --- |
| **package** com.zhiyou100;  **import** java.io.File; **import** java.io.IOException;  */\*\*  \** ***@author*** *QiuShiju  \** ***@date*** *2021/1/5  \** ***@desc*** *创建功能  \*/* **public class** Demo2 {   **public static void** main(String[] args) **throws** IOException{   File f1 = **new** File(**"E:\\A\\a.txt"**);  File f2 = **new** File(**"E:\\A\\b.txt"**);  */\*\*  \* boolean createNewFile()  当且仅当不存在具有此抽象路径名指定名称的文件时，不可分地创建一个新的空文件。  \*/* **boolean** result1 = f1.createNewFile( );  **boolean** result2 = f2.createNewFile( );  System.***out***.println(**"是否创建文件 ? "**+result1 );  System.***out***.println(**"是否创建文件 ? "**+result2 );   */\*\*  \* boolean mkdir() ==> 只能创建单层文件夹  创建此抽象路径名指定的目录。  \*/* File f3 = **new** File(**"E:\\A\\B"**);  **boolean** result3 = f3.mkdir( );  System.***out***.println(**"是否创建文件夹B ? "**+result3 );    */\*\*  \* boolean mkdirs() ==> 创建多层级文件夹  创建此抽象路径名指定的目录，包括所有必需但不存在的父目录。  \*/* File f4 = **new** File(**"E:\\A\\C\\D"**);  **boolean** result4 = f4.mkdirs( );  System.***out***.println(**"是否创建文件夹C\\D ? "**+result4 );  } } |

## 获取功能- getXxx

|  |  |
| --- | --- |
| [File](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/io/File.html" \o "java.io 中的类) | **[getAbsoluteFile](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/io/File.html" \l "getAbsoluteFile())**()            返回此抽象路径名的绝对路径名形式。 |
| [String](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html" \o "java.lang 中的类) | **[getAbsolutePath](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/io/File.html" \l "getAbsolutePath())**()            返回此抽象路径名的绝对路径名字符串。 |
| [String](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html" \o "java.lang 中的类) | **[getName](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/io/File.html" \l "getName())()            返回由此抽象路径名表示的文件或目录的名称。** |

|  |
| --- |
| **package** com.zhiyou100;  **import** java.io.File; **import** java.io.IOException;  */\*\*  \** ***@author*** *QiuShiju  \** ***@date*** *2021/1/5  \** ***@desc*** *获取功能 - getXxxx  \*/* **public class** Demo3 {   **public static void** main(String[] args) **throws** IOException {   File f1 = **new** File(**"E:\\A\\a.txt"**);  File f2 = **new** File(**"a.txt"**); *// f2.createNewFile(); // System.out.println("f2是否存在 ? "+f2.exists() ); // System.out.println(f1 );  /\*\*  \* File getAbsoluteFile()  返回此抽象路径名的绝对路径名形式。  String getAbsolutePath()  返回此抽象路径名的绝对路径名字符串。  \*/* File absoluteFile = f1.getAbsoluteFile( );  File absoluteFile2 = f2.getAbsoluteFile( ); *// System.out.println(absoluteFile ); // System.out.println(absoluteFile2 );* String absolutePath = f2.getAbsolutePath( ); *// System.out.println(absolutePath );    /\*\*  \* String getName()  返回由此抽象路径名表示的文件或目录的名称。  \*/* System.***out***.println(f1.getName() );   */\*\*  \* String getPath() ==> 只是获得构造方法中的路径  将此抽象路径名转换为一个路径名字符串。  \*/* }  } |

## 获取功能 – list

|  |  |
| --- | --- |
| [String](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html" \o "java.lang 中的类)[] | **[list](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/io/File.html" \l "list())**()            返回一个字符串数组，这些字符串指定此抽象路径名表示的目录中的文件和目录。 |
| [File](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/io/File.html" \o "java.io 中的类)[] | **[listFiles](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/io/File.html" \l "listFiles())()            返回一个抽象路径名数组，这些路径名表示此抽象路径名表示的目录中的文件。** |

|  |
| --- |
| **package** com.zhiyou100;  **import** java.io.File; **import** java.util.Arrays;  */\*\*  \** ***@author*** *QiuShiju  \** ***@date*** *2021/1/5  \** ***@desc*** *list功能  \*/* **public class** Demo4 {   **public static void** main(String[] args) {  */\*\*  \* list方法是针对文件夹设计的,目的是为了获得文件夹内部的文件或文件夹  \* 如果使用文件对象调用list方法,返回null  \*/ // File f1 = new File("E:\\A\\a.txt"); // String[] list = f1.list( );// 返回null // System.out.println(list.length );// 空指针异常   /\*\*  \* String[] list()  返回一个字符串数组，这些字符串指定此抽象路径名表示的目录中的文件和目录。  ps: 如果是空文件夹,返回的数组长度是0  场景: 列出A文件夹下的所有记事本文件 (判断字符串后缀)  \*/* File a = **new** File(**"E:\\A"**);  String[] list = a.list( ); *// System.out.println(list.length );* **for** (String str : list) {  *// System.out.println(str );* }   */\*\*  \* File[] listFiles()  返回一个抽象路径名数组，这些路径名表示此抽象路径名表示的目录中的文件。  场景: 删除A文件夹下的所以文件  \*/* File[] files = a.listFiles( );  **for** (File file : files) {  System.***out***.println(file );  }  } } |

## 判断

|  |  |
| --- | --- |
| boolean | **[isDirectory](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/io/File.html" \l "isDirectory())**()            测试此抽象路径名表示的文件是否是一个目录。 |
| boolean | **[isFile](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/io/File.html" \l "isFile())**()            测试此抽象路径名表示的文件是否是一个标准文件。 |
| boolean | **[isHidden](mk:@MSITStore:E:\\000_J_A_V_A\\ZhiyouWorkSpace\\讲课\\00%20总\\邱世举_Java\\API\\JDK_API_1_6_zh_CN.CHM::/java/io/File.html" \l "isHidden())**()            测试此抽象路径名指定的文件是否是一个隐藏文件。 |

|  |
| --- |
| **public static void** main(String[] args) {  File file = **new** File(**"E:\\A\\a.txt"**);  System.***out***.println(**"是否是文件夹 ? "**+file.isDirectory() );  System.***out***.println(**"是否是文件 ? "**+file.isFile() );  System.***out***.println(**"是否是隐藏文件 ? "**+file.isHidden() ); } |

## 删除和重命名

|  |
| --- |
| **package** com.zhiyou100;  **import** java.io.File;  */\*\*  \** ***@author*** *QiuShiju  \** ***@date*** *2021/1/5  \** ***@desc*** *\*/* **public class** Demo6 {   **public static void** main(String[] args) {   *// 平级路径,只是改名 // File file = new File("E:\\A\\b.txt"); // File fileNew = new File("E:\\A\\b2.txt");   // 如果是不同路径,那就是移动并重命名,即剪切重命名* File file = **new** File(**"E:\\A\\b2.txt"**);  File fileNew = **new** File(**"E:\\A\\C\\D\\b3.txt"**);  **boolean** b = file.renameTo(fileNew);  System.***out***.println(b );    }   **private static void** delete() {  File file = **new** File(**"E:\\A\\a.txt"**);  File file2 = **new** File(**"E:\\A\\B"**);   */\*\*  \* delete()  \* 只能删除文件和空文件夹  \* 思考:  \* 删除指定文件夹  \*/* System.***out***.println(file2.delete() );  }  } |

# 作业

设计方法:

1列出A文件夹下的所有记事本文件 (判断字符串后缀)

2删除A文件夹下的所有文件

3删除指定文件夹