



请扫描二维码联系客服
webmaster@csdn.net
400-660-0108
QQ客服 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心



等级： 博客 5 访问量：15万+
积分：2249 排名：2万+

文章搜索

博客专栏

- CentOS
初体验

文章：25篇
阅读：4925
- Gradle
实战指南

文章：7篇
阅读：8764
- Fiddler
抓包指南

文章：11篇
阅读：15088
- redis
实战指南

文章：7篇
阅读：592

文章分类

- android (60)
android studio (42)
网站与服务器 (5)
运营与测试 (1)
Java (12)
设计模式 (1)
CVS(Git、SVN) (1)
工具 (12)

目录视图

摘要视图

Java IO流学习总结二：File

标签：File IO JAVA zhaoyanjun 赵彦军

2017年01月17日 11:26:32

735人阅读

评论(0)

收藏

分类： android (59) Java (11)

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/zhaoyanjun6/article/details/54581478>

目录(?)

[+]

转载请标明出处：<http://blog.csdn.net/zhaoyanjun6/article/details/54581478>
本文出自【赵彦军的博客】

Java File类的功能非常强大，利用java基本上可以对文件进行所有操作。
首先来看File类的构造函数的源码

```
1  /**
2   * Internal constructor for already-normalized pathname strings.
3   */
4  private File(String pathname, int prefixLength) {
5      this.path = pathname;
6      this.prefixLength = prefixLength;
7  }
8
9  /**
10   * Internal constructor for already-normalized pathname strings.
11   * The parameter order is used to disambiguate this method from the
12   * public(File, String) constructor.
13   */
14  private File(String child, File parent) {
15      assert parent.path != null;
16      assert (!parent.path.equals(""));
17      this.path = fs.resolve(parent.path, child);
18      this.prefixLength = parent.prefixLength;
19  }
20
21  /**
22   * Creates a new File instance by converting the given
23   * pathname string into an abstract pathname. If the given string is
24   * the empty string, then the result is the empty abstract pathname.
25   *
26   * @param  pathname A pathname string
27   * @throws NullPointerException
28   *         If the pathname argument is null
29   */
30  public File(String pathname) {
31      if (pathname == null) {
32          throw new NullPointerException();
33      }
34      this.path = fs.normalize(pathname);
35      this.prefixLength = fs.prefixLength(this.path);
36  }
37
38
39  /**
40   * @param  parent The parent pathname string
41   * @param  child  The child pathname string
```

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 🗨 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

Groovy 使用完全解析	(7358)
Java IO流学习总结一：输入...	(5844)
Android studio 如何查看模...	(5614)
Fiddler抓包使用教程-扫盲篇	(4699)
Jenkins实现Android自动化打...	(4546)
RxJava 2.x 使用最佳实践	(3695)
Java IO流学习总结七：Com...	(3642)
Android Gradle使用总结	(2895)

最新评论

- Java IO流学习总结七：Com...
君杉 : 辛苦lz
- Android Loader 异步...
qq_36768202 : good
- view.performClick...
爱吃腥的猫 : 多泄
- Android 路由框架ARout...
被时光移动的城 : 很不错
- Android 路由框架ARout...
Sunstorm_ : 感谢！
- Android 路由框架ARout...
赵彦军 : [reply]u010696783[/reply] 可以转发，最好注明出处。
- Java IO流学习总结七：Com...
果果爱吃苹果 : 很全的例子 收获很多
- Android 路由框架ARout...
键盘上de烟灰 : 我靠，没人评论？博主，我转发了哈，这么好的文章，天理不容
- Java IO流学习总结一：输入输...
jiangxxxz : 楼主第一张图里的ByteReader是不是错了，应该是CharArrayReader
- Java 中Comparator ...
赵彦军 : [reply]h18581124888[/reply] 这道题的需求就是按年龄排序。老板让你按年龄...

```

42  * @throws NullPointerException
43  *      If <code>child</code> is <code>>null</code>
44  */
45  public File(String parent, String child) {
46      if (child == null) {
47          throw new NullPointerException();
48      }
49      if (parent != null && !parent.isEmpty()) {
50          this.path = fs.resolve(fs.normalize(parent),
51                               fs.normalize(child));
52      } else {
53          this.path = fs.normalize(child);
54      }
55      this.prefixLength = fs.prefixLength(this.path);
56  }
57
58  /**
59   * @param parent The parent abstract pathname
60   * @param child The child pathname string
61   * @throws NullPointerException
62   *      If <code>child</code> is <code>>null</code>
63   */
64  public File(File parent, String child) {
65      if (child == null) {
66          throw new NullPointerException();
67      }
68      if (parent != null) {
69          if (parent.path.equals("")) {
70              this.path = fs.resolve(fs.getDefaultParent(),
71                                   fs.normalize(child));
72          } else {
73              this.path = fs.resolve(parent.path,
74                                   fs.normalize(child));
75          }
76      } else {
77          this.path = fs.normalize(child);
78      }
79      this.prefixLength = fs.prefixLength(this.path);
80  }
81
82  /**
83   * @param uri
84   *      An absolute, hierarchical URI with a scheme equal to
85   *      <tt>"file"</tt>, a non-empty path component, and undefined
86   *      authority, query, and fragment components
87   *
88   * @throws NullPointerException
89   *      If <tt>uri</tt> is <tt>>null</tt>
90   *
91   * @throws IllegalArgumentException
92   *      If the preconditions on the parameter do not hold
93   */
94  public File(URI uri) {
95
96      // Check our many preconditions
97      if (!uri.isAbsolute())
98          throw new IllegalArgumentException("URI is not absolute");
99      if (uri.isOpaque())
100          throw new IllegalArgumentException("URI is not hierarchical");
101      String scheme = uri.getScheme();
102      if ((scheme == null) || !scheme.equalsIgnoreCase("file"))
103          throw new IllegalArgumentException("URI scheme is not \"file\"");
104      if (uri.getAuthority() != null)
105          throw new IllegalArgumentException("URI has an authority component");
106      if (uri.getFragment() != null)
107          throw new IllegalArgumentException("URI has a fragment component");
108      if (uri.getQuery() != null)
109          throw new IllegalArgumentException("URI has a query component");
110      String p = uri.getPath();
111      if (p.equals(""))
112          throw new IllegalArgumentException("URI path component is empty");

```

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
👤 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

```
113
114     // Okay, now initialize
115     p = fs.fromURIPath(p);
116     if (File.separatorChar != '/')
117         p = p.replace('/', File.separatorChar);
118     this.path = fs.normalize(p);
119     this.prefixLength = fs.prefixLength(this.path);
120 }
```

从源码可以看出File类的构造函数有6个，精简如下

```
1 public File(String pathname) //文件的绝对路径
2 public File(URI uri) //文件的URI地址
3
4 public File(String parent, String child) //指定父文件绝对路径、子文件绝对路径
5 public File(File parent, String child) //指定父文件、子文件相对路径
6
7
8 //下面这两个是File类中私有的构造函数，外面不能调用
9 private File(String child, File parent)
10 private File(String pathname, int prefixLength)
```

现在就看的比较清楚了，6个构造函数，可以分为2类。4个公共构造函数，2个私有构造函数。

构造函数1：

```
1 //电脑d盘中的cat.png 图片的路径
2 String filePath1 = "D:/cat.png" ;
3 File file = new File( filePath1 ) ;
```

构造函数2：

```
1 String parentFilePath = "E:/cat" ;
2
3 String childFilePath = "small_cat.txt" ;
4
5 //创建parentFile文件
6 File parentFile = new File( parentFilePath ) ;
7 parentFile.mkdir() ;
8
9 //如果parentFile不存在，就会报异常
10 File file = new File( parentFilePath , childFilePath ) ;
11
12 try {
13     file.createNewFile() ;
14 } catch (IOException e) {
15     e.printStackTrace();
16 }
```

联系我们



请扫描二维码联系客服
webmaster@csdn.net
400-660-0108
QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

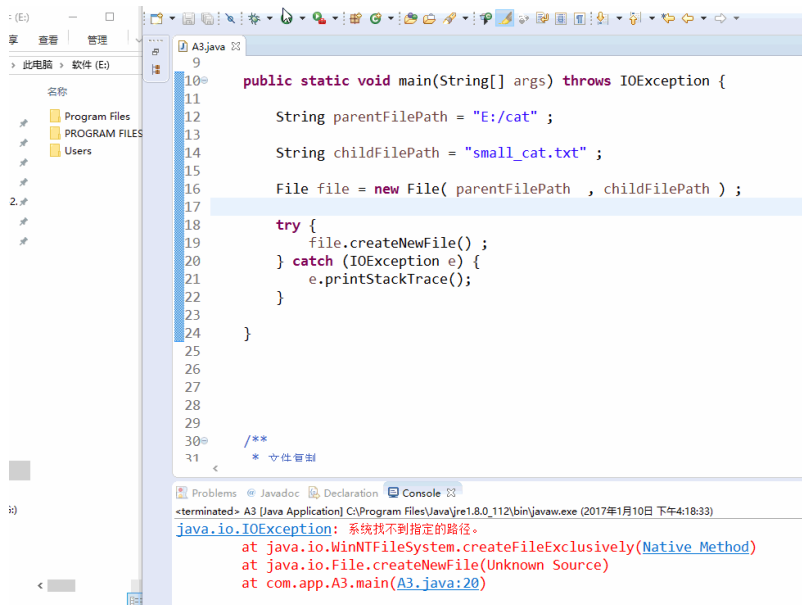
经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

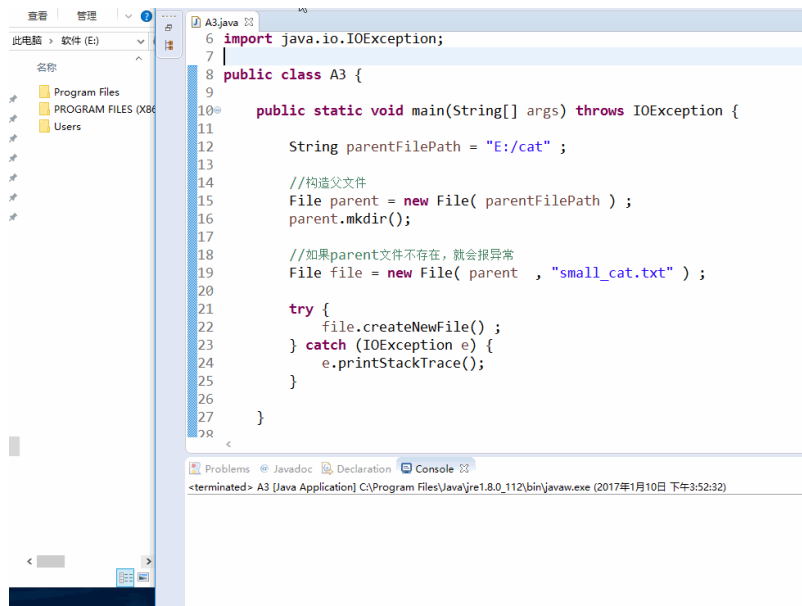
效果图：



构造函数3：

```
1 String parentFilePath = "E:/cat" ;
2
3 //构造父文件
4 File parent = new File( parentFilePath ) ;
5 parent.mkdir();
6
7 //如果parent文件不存在，就会报异常
8 File file = new File( parent , "small_cat.txt" ) ;
9
10 try {
11     file.createNewFile() ;
12 } catch (IOException e) {
13     e.printStackTrace();
14 }
```

效果图：



• 创建目录

```
1 boolean file.mkdir()
```

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
👤 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

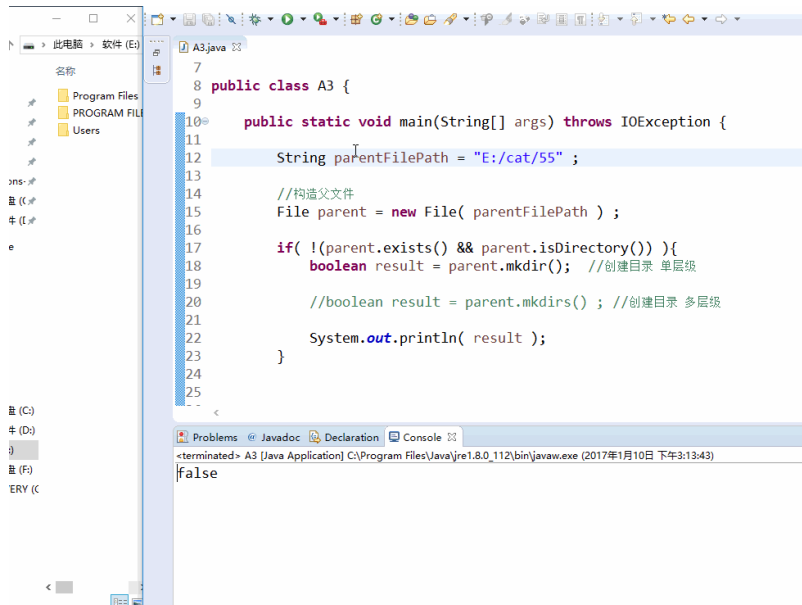
北京互联网违法和不良信息举报中心

如果创建成功,返回 `true`,创建失败, 返回 `false`。如果这个文件夹已经存在, 则返回 `false`。
只能创建一级目录, 如果父目录不存在, 返回 `false`。

• 创建多级目录

```
1 boolean file.mkdirs()
```

创建多级目录, 创建成功, 返回 `true`,创建失败, 返回 `false`。如果父目录不存在, 就创建, 并且返回 `true`



• 创建一个新的文件

```
1 boolean file.createNewFile() ;
```

如果文件不存在就创建该文件, 创建成功,返回 `true` ; 创建失败, 返回 `false`。如果这个文件已经存在
回 `false`。

• 判断方法

```
1 boolean file.exists() //文件是否存在
2
3 boolean file.isFile() //是否是文件
4
5 boolean file.isDirectory() //是否是目录
6
7 boolean file.isHidden() //是否隐藏 (windows上可以设置某个文件是否隐藏)
8
9 boolean file.isAbsolute() //是否为绝对路径
10
11 boolean file.canRead() //是否可读
12
13 boolean file.canWrite() //是否可写
14
15 boolean file.canExecute() //是否可执行
```

获取文件的信息

```
1 String file.getName() //获取文件的名称, 只是名称, 没有路径
2
3 String file.getParent() //获取父目录的绝对路径, 返回值是一个字符串。如果文件有父目录, 那么返回父目录
4
5 File file.getParentFile() //获取父文件, 返回值是一个File对象。
6
7 long time = file.lastModified() ; //返回文件最后一次修改的时间
```

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
👤 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

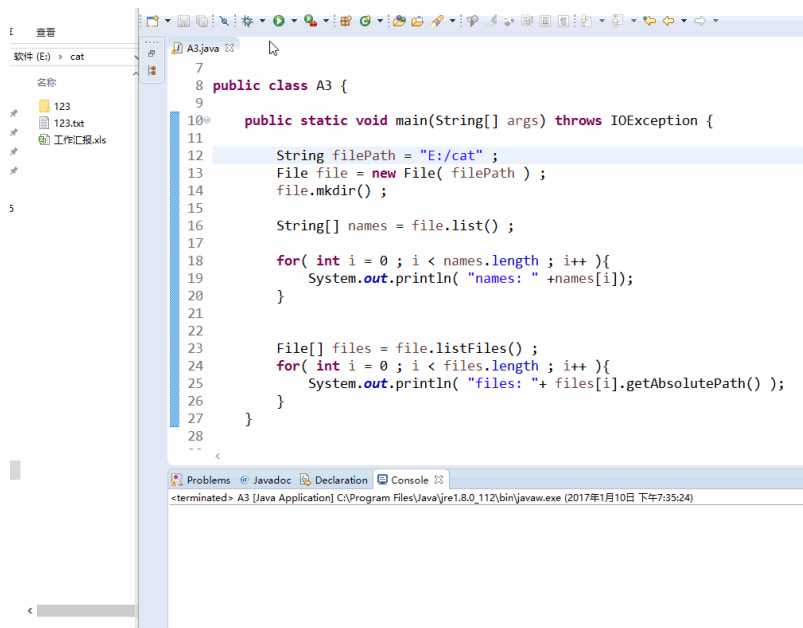
北京互联网违法和不良信息举报中心

```
8 Date dt = new Date(time);
9
10 boolean renameTo(File file) //文件命名
11
12 long file.length() //返回文件的大小, 单位字节
13
14 boolean file.delete() //删除文件
15
16 String[] file.list() //获取该目录下的所有的文件的名字。如果`file`为文件, 返回值为`null`, 在使用
17
18 File[] file.listFiles() //获取该目录下的所有的文件。如果`file`为文件, 返回值为`null`, 在使用时i
19
```

实战经验1：file.list() , file.listFiles()

```
1 String filePath = "E:/cat" ;
2 File file = new File( filePath ) ;
3 file.mkdir() ;
4
5 String[] names = file.list() ;
6
7 for( int i = 0 ; i < names.length ; i++ ){
8     System.out.println( "names: " +names[i]);
9 }
10
11 File[] files = file.listFiles() ;
12 for( int i = 0 ; i < files.length ; i++ ){
13     System.out.println( "files: " + files[i].getAbsolutePath() );
14 }
```

效果图：



实战经验2：扫描F盘所有的文件

```
1 public class A3 {
2
3     public static void main(String[] args) throws IOException {
4
5         String filePath = "F:/" ;
6         File file = new File( filePath ) ;
7         getFile(file);
8
9     }
10
11 }
```

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
💬 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息

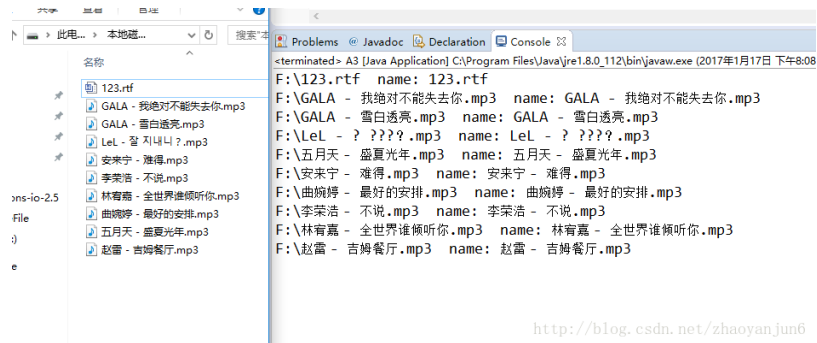
网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

```
12     private static void getFile( File file ){
13         File[] files = file.listFiles() ;
14         for( File f : files ){
15             if ( f.isHidden() ) continue ;
16
17             if(f.isDirectory() ){
18                 getFile( f );
19             }else{
20                 System.out.println( f.getAbsolutePath() + " " + f.getName() );
21             }
22         }
23     }
24 }
```

效果图：



<http://blog.csdn.net/zhaoyanjun6>

在上面的实战演练中用到了，`file.list()`，`file.listFiles()`。这是两个无参的方法，实际上还有两个方法，分别是

```
1 file.list(FilenameFilter filter) ;
2
3 file.listFiles( FilenameFilter filter) ;
4
5 file.listFiles(FileFilter filter)
```

FileFilter

FileFilter是io包里面的一个接口，从名字上可以看出，这个类是文件过滤功能的。

需要重写 `accept` 方法

比如：

```
1 static class MyFileFilter implements FileFilter {
2
3     MyFileFilter(){
4     }
5
6     //pathname: 文件的绝对路径+ 文件名 ， 比如： F:\安来宁 - 难得.mp3 ， 或者： F:\文件夹1
7     @Override
8     public boolean accept(File pathname) {
9         return false;
10    }
11 }
```

实战：获取指定目录的所有文件夹

```
1 package com.app;
2 import java.io.File;
3 import java.io.FileFilter;
4 import java.io.IOException;
5
6
7 public class A3 {
8
9     public static void main(String[] args) throws IOException {
```

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
👤 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

```
10
11     String filePath = "F:/" ;
12     File file = new File( filePath ) ;
13     getFile(file);
14
15 }
16
17
18 /**
19  * 获取指定目录的所有文件夹
20  * @param file
21  */
22 private static void getFile( File file ){
23     MyFileFilter myFileFilter = new MyFileFilter() ;
24
25     File[] files = file.listFiles( myFileFilter ) ;
26     for( File f : files ){
27         if ( f.isHidden() ) continue ;
28
29         System.out.println( f.getAbsolutePath() );
30     }
31 }
32
33
34 static class MyFileFilter implements FileFilter {
35
36     MyFileFilter(){
37
38     }
39
40     //pathname: 文件的绝对路径+ 文件名 , 比如: F:\安来宁 - 难得.mp3 , 或者: F:\文件夹1
41     @Override
42     public boolean accept(File pathname) {
43         if( pathname.isDirectory() ){
44             return true ;
45         }
46         return false;
47     }
48
49 }
50
51 }
```

FilenameFilter

FileFilter是io包里面的一个接口,从名字上可以看出,这个类是文件名字过滤功能的。需要重写里面的accept方法。

比如:

```
1 package com.app;
2
3 import java.io.File;
4 import java.io FilenameFilter;
5
6 public class MyFilenameFilter implements FilenameFilter {
7     //type为需要过滤的条件,比如如果type=".jpg",则只能返回后缀为jpg的文件
8     private String type;
9     MyFilenameFilter( String type){
10         this.type = type ;
11     }
12
13     @Override
14     public boolean accept(File dir, String name) {
15         //dir表示文件的当前目录, name表示文件名;
16         return name.endsWith( type ) ;
17     }
18
19 }
```


联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
🗣 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

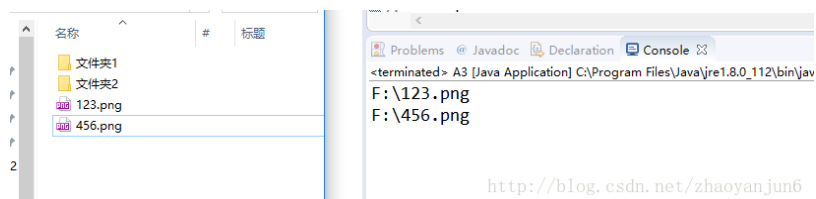
中国互联网举报中心

北京互联网违法和不良信息举报中心

实战：扫描出指定路径的所有图片

```
1 package com.app;
2 import java.io.File;
3 import java.io.FilenameFilter;
4 import java.io.IOException;
5
6
7 public class A3 {
8
9     public static void main(String[] args) throws IOException {
10
11         String filePath = "F:/";
12         File file = new File( filePath );
13         getFile(file);
14
15     }
16
17
18     /**
19      * 扫描出指定路径的所有图片
20      * @param file
21      */
22     private static void getFile( File file ){
23         MyFilenameFilter myFileFilter = new MyFilenameFilter( ".png" );
24
25         File[] files = file.listFiles( myFileFilter );
26         for( File f : files ){
27             if ( f.isHidden() ) continue ;
28
29             System.out.println( f.getAbsolutePath() );
30         }
31     }
32
33
34
35     static class MyFilenameFilter implements FilenameFilter {
36         //type为需要过滤的条件，比如如果type=".jpg"，则只能返回后缀为jpg的文件
37         private String type;
38         MyFilenameFilter( String type){
39             this.type = type ;
40         }
41
42         @Override
43         public boolean accept(File dir, String name) {
44             //dir表示文件的当前目录，name表示文件名：
45             return name.endsWith( type ) ;
46         }
47     }
48 }
49
50 }
```

运行结果：



<http://blog.csdn.net/zhaoyanjun6>

- 上一篇 Java IO流学习总结一：输入输出流
- 下一篇 Java IO流学习总结三：缓冲流-BufferedInputStream、BufferedOutputStream

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
👤 QQ客服 🗨 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

java IO流学习总结

du_minchao 2015年10月11日 12:35

Java中的流，可以从不同的角度进行分类。按流的方向分：输入流和输出流。按处理数据单位分：字节流和字符流
实现功能分：节点流和处理流。输出流： ...

Java中IO流学习总结

Yue_Chen 2017年05月26日 18:00

Java中IO流学习总结一.IO流中的结构 字符流：顾名思义，该流只能处理字符，但处理字符速度很快 字节流：可以处理所
为单位储存的文件，也就是说可以处理所有的文件，但是在处理字符上的速度不如...

JAVA?:使用 - CSDN博客

201

给初学者之一:浅谈java及应用 学java不知不觉也已经三年了 从不知java为何物到现在一个小小的j2ee项目经理 虽说不上
手,大概也算有点斤两了吧 每次上...

JAVA8 十大新特性详解 - CSDN博客

201

前言: Java 8 已经发布很久了,很多报道表明Java 8 是一次重大的版本升级。在Java Code Geeks上已经有很多介绍Java 8
文章,例如Playing with Java 8 -...

最完整的Java IO流学习总结

SilenceOO 2016年03月28日 00:37

一、File类 1、java.io.File 类是专门对文件进行操作的类。(只能对文件本身进行操作，不能对文件内容操作) 2、File类
和目录路径名的抽象表示”。而不是指文件的内容...

java视频下载 - CSDN博客

201

java (17) perl (1) linux (19) weblogic (3) eclipse (6) kettle (10) mysql (14) maven (6) svn (2) jsf (11) hadoop (18) .

Java中,&&与&,&&||与|的区别 - CSDN博客

201

原文地址: Java中,&&与&,&&||与|的区别 在java的逻辑运算符中,有这么四类:&&(短路与),&,&&|| (短路或)。&&和&都是表示
&&只要第一个条件...

Java中的IO流总结

u013309870 2016年08月03日 20:03

Java中的IO流总结 1. 流的继承关系，以及字节流和字符流。 2. 节点流FileOutputStream和FileInputStream和处理流Bu
putStream和Bu...

java IO体系的学习总结

nightcurtis 2016年05月05日 16:31

java IO体系的整理笔记1.Java Io流的概念，分类，类图。1.1 Java Io流的概念 java的io是实现输入和输出的基础，可
实现数据的输入和输出操作。在java中把不同...

Java历史简介 - CSDN博客

20

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
👤 QQ客服 🗨 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

*Java*是一种计算机编程语言,拥有跨平台、面向对象、泛型编程的特性,广泛应用于企业级Web应用开发和移动应用开发。
un公司的James Gosling等人开始开发名称为 Oak...

恭喜：一个公式教你秒懂天下英语

老司机教你一个数学公式秒懂天下英语



java实现文件下载的两种方式 - CSDN博客

201

jsp+*java*Bean+jdb 实现DAO设计模式 (1) Servlet (42) 第一个Servlet程序 HelloServlet (1) Servlet与表单 (1) 取得Http
实例化实例 (1) Servlet...

Java IO流知识点总结



wenzhi20102321 2016年09月18日 13:55

本主要对java的IO流知识点做比较全面的总结，但是没有很深入阐述。

毕向东Java视频学习笔记【day21-IO流<2>】



qq_24653023 2016年09月05日 17:25

23-IO流(演示键盘录入)一旦 System.in流 被关闭了，再创建一个对象in2没有用，会抛出异常。InputStream in = Syst
...

Java基础快速复习 - CSDN博客

201

本篇博客写的都是自己复习*Java*的时候所记录的一些笔记(还是比较篇概念的内容),所以可能内容不是很全面、或许也很杂
是从基础一点点往后增加的。1.数据类型...

undefined

Java IO流学习总结一：输入输出流



zhaoyan jun6 2017年01月09日 19:49

Java IO流学习总结一：输入输出流 转载请标明出处：http://blog.csdn.net/zhaoyan jun6/article/details/53761199 本
【赵彦军的...

java学习笔记之IO流总结(一)



qq_24235325 2016年10月14日 20:36

输入输出作为所有程序都必需的部分 对于我们的java学习有特别重要的意义，以至于博主辛苦看了三天的流分布，跟大
下一.首先我们说一下io流的概念 流是一组有顺序的，有起点和终点的字节集...

java中IO流的知识点总结



u011626964 2015年09月01日 10:39

1.流的概念流是一组有顺序的，有起点和终点的字节集合，是对数据传输的总称或抽象。即数据在两设备间的传输称为流
质是数据传输，根据数据传输特性将流抽象为各种类，方便更直观的进行数据操作。2.IO流...

【马士兵】笔记_Java IO流



na_simon2900086 2016年11月15日 05:41

有道云笔记 连接：http://note.youdao.com/share/?id=7a322ea0150fbc678da417d68f53c9f4&type=note -----..

程序员不会英语怎么行？

老司机教你一个数学公式秒懂天下英语



联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
🗣 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

Java基础-19总结IO流，异常try...catch，throws，File类

1:异常(理解) (1)程序出现的不正常的情况。(2)异常的体系 Throwable |--Error 严重问题，我们不处理。 |--Exception |--

s1547823103 2017年02月10日 17:33 1103

java IO流面试总结

baidu_37107022 2017年08月08日 11:48

java IO流面试总结

集合、IO流部分思维导图框架梳理

a910626 2015年04月23日 00:09

下图是学习集合过程自己梳理的脉络，希望对你有帮助。

Java基础学习之对IO流的理解

qq_32451699 2016年08月17日 14:56

1 什么是IO Java中I/O操作主要是指使用Java进行输入，输出操作。Java所有的I/O机制都是基于数据流进行输入输出。数据流表示了字符或者字节数据的流动序列。Java的I/O流提供...

Java中IO流思维导图

u010879420 2016年12月04日 22:34

JAVA NIO 学习总结(上)

lovejj1994 2017年01月23日 15:05

最近看到《java核心技术》的io部分，学习如何访问文件，如何以二进制格式以及文本格式来读写文件，以及jdk后续更新特性，这里主要总结nio的新内容。demo文件在：https://gith...

java IO流File常用的方法

wenzhi20102321 2016年09月18日 15:46

本文对java的IO流的File文件类展示基本的操作方法

Java_IO流_File类总结(扫描某个盘的全部内容)

longwoniu 2015年04月17日 09:47

在Java中，Everything is Object！所以在文件中，要不例外！在Java中，可以用File类来表示一个文件！！！（注意这些可以是所有文件，包括文件夹）下面的...

Java IO完全总结（转载）

baobeiSimple 2007年07月28日 17:34

Java IO江苏 无锡 缪小东 本篇主要讲述IO相关的内容，主要包括：与IO相关的简单的历史背景知识；Java IO的简单IO设计相关的两个模式；同时列举几个简单的例子；分析其中...