

文件又称为数据源

文件流：文件在程序中是使用文件流

如何判断输入流还是输出流：以内存为参照物，若数据是向内存流动，则是输入流，反之为输出流

java的俩种流：1，字节流，以字节方式读取的二进制文件以及任何类型的文件，byte  
2，字符流，可以用于读写文本文件，不能操作二进制文件

	字节流	字符流
输入	InputStream	Reader
输出	OutputStream	Writer

File是文件的本身，最基础的文件操作

File类的基本用法：操作空间有一个最小快，最小空间为4kb

## 一，文件的简单操作

```
File f=new File("E://newstartcontent//anqili.txt");
    f.getAbsolutePath();
    System.out.println(f.getName()); //获得文件名
    System.out.println(f.getAbsolutePath()); //获取文件的绝对路径
System.out.println(f.canRead()); //获得文件是否可读
    System.out.println(f.canWrite()); //判断文件是否可写
    System.out.println(f.length()); //获得文件的长度
    //创建文件盒创建文件夹
    File f1=new File("E://newstartcontent//linminjun.txt");
    if (!f1.exists()) { //判断文件是否存在
        //创建文件夹
        try {
            f1.createNewFile();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        System.out.println("文件创建成功");
```

```

    }
    else{
        System.out.println("这个文件已经存在");
    }
    //创建文件夹
    File f2=new File("E://newstartcontent//an");
    if (f2.isDirectory()) {
        System.out.println("该文件夹已经存在");
    }
    else{
        f2.mkdir();
        System.out.println("文件夹创建成功");
    }
    //列出一个文件夹下面的所有文件
    File f3=new File("E:\\MATLAB");
    if (f3.isDirectory()) {
        File lis[]=    f3.listFiles();//返回一个文件数组
        for(int i=0;i<lis.length;i++){
            System.out.println("文件名:"+lis[i].getName());
        }
    }
}

```

## 二，对InputStream类的使用

```

//得到一个文件对象
File f=new File("E://newstartcontent//anqili.txt");
//因为file没有读写的能力，所有需要使用一个InputStream
//文件流读出来后需要关闭
FileInputStream fileInputStream=null;
try {
    //为了缓存定义一个字节数组，相当于缓存
    byte[]bytes=new byte[1024];
    int m=0;//表示实际读取到的字节数
    fileInputStream=new FileInputStream(f);
}

```

```

while ((m=fileInputStream.read(bytes))!=-1) {
    String s=new String(bytes,0,m);
    System.out.println(s);

}
} catch (FileNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
finally {
    //关闭文件流必须放在finally中
    try {
        fileInputStream.close();
    } catch (Exception e2) {
        // TODO: handle exception
        e2.printStackTrace();
    }

}
}

```

### 三，//演示fileoutputstream流的使用

```

File f=new File("E://ss.txt");
FileOutputStream fileOutputStream=null;
try {
    fileOutputStream=new FileOutputStream(f);
    String s="anqili&linmingjun\r\n";//换行在需要换行的位置添加\r\n
    String s1="anqili&linmingjun";
    //定义一个字节数组
    byte b[]=new byte[1024];
    //如何把string转换成为一个byte数组

    fileOutputStream.write(s.getBytes());
    fileOutputStream.write(s1.getBytes());
} catch (Exception e) {
    e.printStackTrace();
} finally {

```

```

try {
    fileOutputStream.close();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

// TODO: handle finally clause
}

```

#### 四，//图片拷贝

```

//思路：先把图片读入内存，在想办法写入到某个文件，因为
//图片是一个二进制文件，因此只能使用字节流完成
File f=new File("e://q.jpg");
//输入流
FileInputStream fiStream=null;
//输出流
FileOutputStream fid=null;
try {
    fiStream=new FileInputStream("e://q.jpg");
    fid=new FileOutputStream("e:\\MATLAB\\q.jpg");
    byte b[]=new byte[1024];
    int m=0;
    try {
        while ((m=fiStream.read(b))!=-1) {
            //输出到指定文件
            fid.write(b);
        }
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
} catch (FileNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}

```

```

finally {
    try {
        fiStream.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    try {
        fid.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
}

```

## 五，拷贝文件

//读取一个文件并将文件写入另一个文件中

//文件取出字符流对象

FileReader fileReader=null;

//写入到文件

FileWriter fileWriter=null;

try {

    //创建filereader对象

    fileReader=new FileReader("e://ss.txt");

    //创建输出对象

    fileWriter=new FileWriter("e://newstartcontent/anqili.txt");

    //读入到内存

    char c[]=new char[100];

    int m=0;//记录实际读取到的字符数

    while((m=fileReader.read(c))!=-1) {

        fileWriter.write(c);

        System.out.println(c);

```

    }

    } catch (Exception e) {
        // TODO: handle exception
    }

    finally {
        try {
            fileReader.close();
        } catch (Exception e2) {
            // TODO: handle exception
            e2.printStackTrace();
        }

        try {
            fileWriter.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }
}

```

## 六，缓冲字符流

//效率很高

```

//缓冲字符流
//readline只是读取一行
BufferedReader bReader=null;
BufferedWriter bWriter=null;
try {
    //先创建一个FileReader对象
    FileReader fReader=new FileReader("e://anqili.txt");
    bReader=new BufferedReader(fReader);
    FileWriter fWriter=new
FileWriter("e://newstartcontent/anqili.txt");
    bWriter=new BufferedWriter(fWriter);
    //循环读取文件
    String s="";
}

```

```
while((s=bReader.readLine())!=null){
    System.out.println(s);
    //输出到e盘
    //打印一行就进行换行
    bWriter.write(s+"\r\n");
}
} catch (Exception e) {
    // TODO: handle exception
    e.printStackTrace();
}
finally {
    try {
        bWriter.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    try {
        bReader.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```