链滴

Java IO 流、属性操作

作者：whoms
原文链接：https://ld246.com/article/1638527598755
来源网站：链滴

1.IO流

根据流向分类：

- 输入流：把数据从输入设备读取到内存中的流
- 输出流：把数据从内存中写出到输出设备的流

根据操作数据单位不同分类：

- 字节流：以字节为单位读写数据的流
- 字符流：以字符为单位（主要操作文本数据）读写数据的流

在Java中描述的底层父类（JavaIO流共涉及40多个类，都是从如下4个底层父类派生的，由这4个类生出来的子类名称都是以其父类作为子类名后缀，4个底层父类都已实现AutoCloseable接口）：

- 字节流
  - 字节输入流：InputStream
  - 字节输出流：OutputStream
- 字符流
  - 字符输入流：Reader
  - 字符输出流：Writer

- 读取流

```
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;

public class Test4 {
    public static void main(String[] args) throws IOException {
      //循环单个字节读取
      InputStream is = new FileInputStream("D:/aaa/b.txt");
      int len;
      while((len=is.read())!=-1){
          System.out.print((char) len);
      }
      is.close();

      //字节数组读取
      InputStream is = new FileInputStream("D:/aaa/b.txt");
      byte[] b = new byte[1024];
      int len;
      while((len=is.read(b))!=-1){
        //在转换时不能全部转换，而是只转换有效字节len（实际读取到的字节个数）
        //System.out.println(new String(b,0,1024));
        System.out.println(new String(b,0,len));
      }
      is.close();
```

```
    }
}
```

- 写入流

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;

public class Test2 {
    public static void main(String[] args) throws IOException {
      OutputStream os = new FileOutputStream("D:/aaa/b.txt");
      byte[] bytes = "hello world".getBytes();
      os.write(bytes);
      //写出一个换行
      os.wirte("\r\n".getBytes());
      os.close();
    }
}
```

回车符\r和换行符\n：

- 回车符\r：回到一行的开头，ASCII为13
- 换行符\n：下一行，ASCII为10

系统中的换行：

- Windows系统中，每行结尾是 回车+换行，即\r\n
- Unix系统中，每行结尾只有 换行，即\n
- Mac系统中，每行结尾是 回车，即\r。（从Mac OS x开始与Linux统一）

通过System.getProperty("line.separator")可以获取操作系统的换行符

2.IO异常的处理

- JDK7以前

```
try{
  //创建流
}catch(Exception e){
  e.printStackTrace()
}finally{
  //关闭流
}
```

- JDK7的处理

JDK7提供的try-with-resources语句，是异常处理的一大利器，该语句确保了每个资源在语句结束时闭。

```
try(创建流对象语句){
  //读写数据
```

```
}catch(Exception e){
  e.printStackTrace();
}
```

声明在try()中的类必须实现AutoCloseable接口，在资源处理完毕时，将自动的调用 AutoCloseable 口中的close方法，如果没有实现AutoCloseable接口的类将不能写在try()中。

```
public class Demo{
    public static void main(String[] args){
        try(
            TestAutoCloseable closeable = new TestAutoCloseable();
        ){
            int num = 1/0;
            closeable.test();
        }catch(Exception e){
            e.printStackTrace();
        }
    }
}
class TestAutoCloseable implements AutoCloseable{
    public void close() throws Exception{
        System.out.println("Close方法被调用");
    }
}
```

```
D:\Data\jdk1.8.0_152\bin\java.exe ...
Close方法被调用
java.lang.ArithmeticException: / by zero
    at com.dfbz.demo1.Test1.main(Test1.java:20)

Process finished with exit code 0
```

3. 属性集

java.util.Properties继承于Hashtable，来表示一个持久的属性集。它使用键值对结构存储数据，每键其对应的值都是一个字符串。

- public Object setProperty(String key,String value)：保存一对属性值
- public String getProperty(String key)：使用此key搜索值
- public Set<String> stringPropertyNames()：所有键的名称的集合
- public void load(InputStream inStream)：从字节输入流中读取键值对
- public vodi store(OutputStream out,@Nullable String comments)：将字节输入流写入到一个Properties文件中，参数二为一个注释的字符串