

数据流将“基本数据类型与字符串类型”作为数据源，从而允许程序以与机器无关的方式从底层输入输出流中操作Java基本数据类型与字符串类型。

DataInputStream和DataOutputStream处理流，提供了可以存取与机器无关的所有Java基础类型数据(如：int、double、String等)的方法。

DataInputStream和DataOutputStream的使用

```
import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class TestDataStream {
    public static void main(String[] args) {
        DataOutputStream dos = null;
        DataInputStream dis = null;
        FileOutputStream fos = null;
        FileInputStream fis = null;
        try {
            fos = new FileOutputStream("D:/data.txt");
            fis = new FileInputStream("D:/data.txt");
            //使用数据流对缓冲流进行包装，新增缓冲功能
            dos = new DataOutputStream(new BufferedOutputStream(fos));
            dis = new DataInputStream(new BufferedInputStream(fis));
            //将如下数据写入到文件中
            dos.writeChar('a');
            dos.writeInt(10);
            dos.writeDouble(Math.random());
            dos.writeBoolean(true);
            dos.writeUTF("北京尚学堂");
            //手动刷新缓冲区：将流中数据写入到文件中
```

```
dos.flush();
//直接读取数据：读取的顺序要与写入的顺序一致，否则不能正确读取数据。
System.out.println("char: " + dis.readChar());
System.out.println("int: " + dis.readInt());
System.out.println("double: " + dis.readDouble());
System.out.println("boolean: " + dis.readBoolean());
System.out.println("String: " + dis.readUTF());
} catch (IOException e) {
    e.printStackTrace();
} finally {
    try {
        if(dos!=null){
            dos.close();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    try {
        if(dis!=null){
            dis.close();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    try {
        if(fos!=null){
            fos.close();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    try {
        if(fis!=null){
            fis.close();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
```

