

005.重定向标准输入或输出

Java的标准输入/输出分别通过System.in和System.out来代表，默认情况下它们分别代表键盘和显示器，

当程序通过System.in来获取输入时，实际上是从键盘读取输入，当程序试图通过System.out执行输出时，程序总是输出到屏幕。

System类里提供三个重定向标准输入/输出的方法：

- static void setErr(PrintStream err)：重定向“标准”错误输出流
- static void setIn(InputStream in)：重定向“标准”输入流
- static void setOut(PrintStream out)：重定向“标准”输出流

将System.out的输出重定向到文件，而不是在屏幕上输出

```
public class RedirectOut {
    public static void main(String[] args) {
        if (args == null || args.length == 0) {
            throw new RuntimeException("请输入路径");
        }
        PrintStream ps = null;
        try {
            String[] books = new String[] { "Java编程思想", "编译原理", "算
法" };
            ps = new PrintStream(new FileOutputStream(args[0]));
            System.setOut(ps);
            for (String book : books) {
                System.out.println(book);
            }
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } finally {
            if (ps != null) {
                ps.close();
            }
        }
    }
}
```

用FileInputStream创建了一个文件输入流，并使用System的setIn()方法将系统标准输入流重定向到该文件输入流，
运行下面的程序，程序不再等待用户输入，而是直接输入用户所给的路径所对应的文件内容

```
public class RedirectIn {

    public static void main(String[] args) {
        if (args == null || args.length == 0) {
```

```

        throw new RuntimeException("请输入路径");
    }
    FileInputStream fis = null;
    try {
        fis = new FileInputStream(args[0]);
        System.setIn(fis);
        Scanner sc = new Scanner(System.in);
        sc.useDelimiter("\n"); //把回车做为分隔符
        while (sc.hasNext()) {
            System.out.println(sc.next());
        }
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } finally {
        try {
            if (fis != null) {
                fis.close();
            }
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
}

```