

编程入门基础

—— 文本文件

主讲教师：耿宇航

串序列的观点

- 如果把数据看待为有序的串，则为文本观点
- 串间如何分隔?
 - 不同的操作系统，提供不同的规则。
 - Mac os 回车符 (ASCII码=13)
 - Linux/Unix 换行符 (ASCII码=10)
 - Windows 回车符+换行符
- 数据本身只是1和0组成的信息，本身无所谓文本、照片、声音等
- 所谓文本观点是我们处理数据时所持有的观点，不是数据本身特质

更方便的文本操作

- **InputStream, OutputStream** 字节流
- **InputStreamReader, OutputStreamWriter** 字符流
- **BufferedReader, BufferedWriter** 串流
- **PrintWriter** 更丰富的功能，与向屏幕输出一致

以文本观点从键盘输入

- **System.in**
 - 提供了字节输入
- **InputStreamReader**
 - 提供了字符输入
- **BufferedReader**
 - 提供了整行字符的输入（串输入）

```
BufferedReader bs =
```

```
    new BufferedReader(new InputStreamReader(System.in));
```

```
String s = bs.readLine();
```

```
System.out.println(s);
```

java中有哪些预定义流？

- **System.in** 标准输入流 默认键盘
- **System.out** 标准输出流 默认屏幕
- **System.err** 标准错误流 默认屏幕
- 这些预定义的标准流可以重定向
 - `System.setIn(...)`
 - `System.setOut(...)`
 - `System.setErr(...)`

多样的服务

- **InputStream**
 - 只提供基本的字节服务。虽然也可使用，但有的场和很不方便。
- **InputStreamReader**
 - 可以提供面向字符的服务。
- **BufferedReader**
 - 提供面向串的服务。

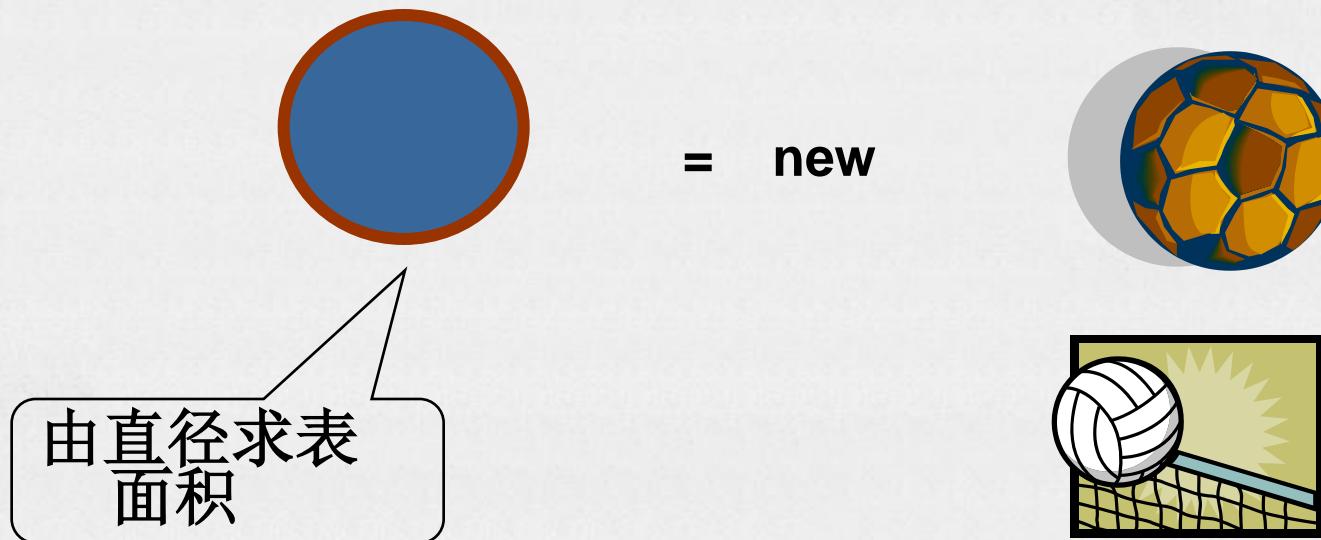
键盘输入原理示意

```
BufferedReader br =  
    new BufferedReader(  
        new InputStreamReader(  
            System.in)));
```



复习-泛化

- `InputStream ins = new FileInputStream("a.txt")`
- 这个就足够了，并不需要太具体！！



I/O类的层次

- **InputStream**
 - FileInputStream 字节流
 - FilterInputStream
 - BufferedInputStream
 - DataInputStream 读取基础数据类型 平台无关
- **OutputStream**
 - FileOutputStream
 - FilterOutputStream
 - BufferedOutputStream
 - DataOutputStream 写入基础数据类型 平台无关

I/O类的层次 (续)

- **Reader**
 - BufferedReader 提供读一行
 - InputStreamReader 提供读字符而非字节
 - FileReader 直接使用文件名, 无需组合
- **Writer**
 - BufferedWriter
 - OutputStreamWriter
 - FileWriter直接使用文件名, 无需组合
 - PrintWriter 如屏幕输出一样方便
- **File**

文本文件与二进制文件

- **文件本身没有强制性的区别**
 - 一般文本文件中都是可识别的字符（供人阅读）
 - 二进制文件高度结构化，供机器识别
- **区别在于我们的观点**
 - 观点：行的序列，文本文件
 - 文本文件无法随机访问，因为行的长度不确定
 - 观点：字节的序列，二进制文件

文本文件的读写

- **PrintWriter(FileOutputStream)**
 - 提供了丰富的输出格式化。默认编码方式
- **FileWriter**
 - 需显式换行。默认编码
- **BufferedWriter(OutputStreamWriter)**
 - 可指定编码
- **BufferedReader(InputStreamReader)**
 - 可指定编码

如何添加到文件尾

- 默认的情况是，新内容覆盖旧内容
- **添加到尾的两种方式**
 - 构造成器中指示定参数
 - 使用随机文件类

尾添加-指定参数

- **InputStream**
- **把第二个参数设为true**
 - new FileOutputStream("a.txt",true)
- **这样的弱点是：不能在旧的内容上修改，只能添加**

谢 谢 !