

# Java程 序 设 计

第 3 讲：控制台交互、格式化 I/O 与 Math 库

- 1 本讲目标
- 2 循环一:输出——把信息排整齐
- 3 循环二:输入——揭晓 `nextLine` 之谜
- 4 循环三:健壮输入与 `Math` 工具
- 5 收束:自查与课后

# 本讲目标:下课前,你应该能做到

## 三件事

1. 会排:用 `printf` 打出一张对齐、补零、控精度的报表;
2. 会修:解释并修复 `nextLine` 吞掉回车的经典陷阱;
3. 会防:写出用户乱输也不崩的健壮输入。

## 课堂节奏

- 3 个循环,每个循环:讲 10 分钟 → 动手 5 分钟;
- 抢答先自己猜,再验证;实操限时完成,截图发课程群。

# 开场预测:输完年龄后,程序在等什么?

## 【随堂抢答·概念检测】用户输入 20 并回车之后——

```
Scanner scanner = new Scanner(System.in);  
System.out.print("请输入年龄: ");  
int age = scanner.nextInt();  
  
System.out.print("请输入姓名: ");  
String name = scanner.nextLine();  
System.out.println("登记: " + name + ", " + age);
```

- A. 安静地等你输入姓名
- B. 不等了,姓名直接是空
- C. 抛异常崩溃

记下你的答案,循环二现场揭晓。

# 三大标准流:程序出厂自带的三根管子

| 流对象        | 方向      | 用途                     |
|------------|---------|------------------------|
| System.in  | 键盘 → 程序 | 标准输入                   |
| System.out | 程序 → 屏幕 | 正常输出,可重定向 > out.log    |
| System.err | 程序 → 屏幕 | 报错专用,无缓冲立即出,2> err.log |

本讲聚焦 out(怎么写好看)和 in(怎么读可靠)。

## printf: 模板里挖坑, 按位填数

```
int id = 7;
double temp = 26.5;
System.out.printf("ID:%03d|Temp:%6.1f|END%n", id, temp);
// 输出:      ID:007|Temp: 26.5|END
```

- 语法: %[宽度][.精度] 类型, 常用 %d 整数、%f 浮点、%s 字符串;
- %03d: 不足 3 位补零; %6.1f: 宽 6、1 位小数、右对齐;
- 加 - 变左对齐: %-6.1f。

## 抢答 1:这行 printf 输出什么?

1/20

【随堂抢答·概念检测】先在纸上画出输出,再对答案

```
int id = 7;  
double score = 92.568;  
System.out.printf("ID:%03d|Score:%-6.1f|END", id, score);
```

- A. ID:7 |Score:92.6 |END
- B. ID:007|Score:92.6 |END
- C. ID:007|Score: 92.6 |END
- D. 编译报错

### 【知识点】逐位拆解

---

- %03d: 宽 3, 7 不足补零 → 007;
- %-6.1f: 92.568 保留 1 位四舍五入得 92.6, 减号 = 左对齐, 宽 6 右补 2 空格;
- 对齐的艺术: 数字右对齐个位对齐, 文字左对齐——报表立刻专业。

## 实操 1: 打一张设备状态报表

**【随堂实操·动手练笔】DeviceReport · 限时 4 分钟**

用 `printf` 打印 3 台设备 (数据自己定):

1. 每行: 设备名 (`%-8s`)、温度 (`%6.1f`)、编号 (`%04d`);
2. 三列必须上下对齐。

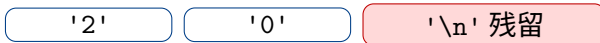
**完成标志:**      三行报表列对齐 → 截图发课程群。      参考:  
`code/lec03/DeviceReportDemo.java`

## Scanner:把输入流切成一个个词

- `new Scanner(System.in)`:接管键盘输入;
- 按空白符切分:`nextInt()` / `nextDouble()` / `next()` 各取一个词;
- 唯独 `nextLine()`:读整行,以回车为界,并把回车吃掉。

记住这句:`nextInt` 只拿数字,回车还留在管道里——谜底已经在话里了。

## 开场谜底:回车卡在了管道里



`nextInt()` 只取走"20"

`nextLine()` 一读就撞见回车,立即返回空串

```
int age = scanner.nextInt();  
scanner.nextLine();           // 解法:加一句,把残留的回车吃掉  
String name = scanner.nextLine(); // 这才真正等用户输入
```

## 抢答 2:next() 读"Hello World" 得到什么?

【随堂抢答·概念检测】用户输入 Hello World 并回车,scanner.next() 返回?

- A. "Hello World"
- B. "Hello"
- C. "World"
- D. 抛异常

### 【知识要点】next vs nextLine

---

- `next()` 以空白符分词,遇到空格就交货 → "Hello";
- "World" 还留在管道里,喂给下一个读取;
- 要带空格的整行,用 `nextLine()`;
- 经验:数值用 `nextXxx`,人名地址用 `nextLine`,中间记得清回车。

## 实操 2: 复现并修复“吞回车”

**【随堂实操·动手练笔】ScannerTrapDemo ·限时 4 分钟**

code/lec03/ScannerTrapDemo.java 已埋好陷阱：

1. 先原样运行, 亲眼看姓名变空;
2. 加一行修复, 让姓名正常输入。

**完成标志:** 修复后姓名能正常录入 → 截图发课程群。

# 用户乱输怎么办:先探后读

```
Scanner sc = new Scanner(System.in);
System.out.print("请输入成绩(0~100): ");
if (sc.hasNextInt()) {           // 先探:是整数吗?
    int score = sc.nextInt();    // 是,才读
} else {
    String bad = sc.next();      // 不是,丢弃这个词
    System.out.println(bad + " 不是合法整数");
}
```

- **hasNextXxx 先探路, nextXxx 再取货**——非法输入不会炸程序;
- 配合循环 = 输错就重来的交互体验。

## 常用方法

- `abs` 绝对值、`max / min`;
- `pow(2,3)=8.0`、`sqrt(16)=4.0`;
- `round(3.6)=4`(四舍五入)。

## 随机整数万能公式

- `Math.random()`  $\in [0.0, 1.0)$ ;
- 取  $[min, max]$  整数:  
`(int)(Math.random() * (max-min+1)) + min`;
- 例:[10, 99] 验证码。

细节:Scanner 用完在 main 末尾 `close()`——但它连着 `System.in`,关了就再也读不了输入。

## 实操 3: 输错的用户也要伺候好

### 【随堂实操·动手练笔】RobustInputDemo · 限时 5 分钟

编写 RobustInputDemo.java:

1. 用 hasNextDouble 守卫, 连读 3 个浮点数;
2. 输入字母时拦截提示并要求重输;
3. 用 Math.max 打印三者最大值。

**完成标志:** 输 abc 被拦、3 个数出最大值 → 截图发课程群。

1. 会排：`%03d`、`%-6.1f` 各是什么效果？
2. 会修：`nextLine` 吞回车——一句话说清成因与修法？
3. 会防：先探后读是哪两个方法的配合？

没做到的那条，就是课后任务的第一优先级。

### 工程视野

交互的容错就是人的容错：把每一次误输入都当成正常场景来设计，是“以人为本”的工程伦理。

1. **练习:**编写 `GradeEvaluationSystem`——录入姓名与平时/期末成绩,按  $30\% + 70\%$  算总评, `printf` 打印成绩单卡片;
2. **AI 实战:**让 AI 解释“为什么高性能服务器严禁 `System.out.println` 刷日志”,把它的理由与 `System.err` 的无缓冲特性对照。

**下讲预告:**流程控制——写一个会做选择的程序。

## 在线主页与课程资源

### ■ 个人主页:

<https://matnoble.top>

### ■ 课程专栏:

<https://matnoble.top/courses/>

### ■ GitHub 仓库:

<https://github.com/MatNoble>

## 自由交流与提问 (Q&A)

### “学贵善疑,逻辑筑基”

感谢大家的专注聆听!

课件最新版本与补充资料将持续同步至课程主页。

**现在进入自由提问与交流时间。**