

Java 程 序 设 计

第 3 讲：控制台交互、格式化 I/O 与 Math 库



厦门大学嘉庚学院

XIAMEN UNIVERSITY TAN KAH KEE COLLEGE

- 1 Java 标准输入输出流体系
- 2 printf 格式化占位符深度解析
- 3 随堂概念抢答与避坑找茬
- 4 Scanner 扫描器与输入机制
- 5 常用数学函数库:java.lang.Math
- 6 随堂编程小练笔与实操任务
- 7 课程总结与工程视野

【知识要点】三大标准系统流 (java.lang.System)

Java 运行时默认预定义了 3 个与底层操作系统控制台绑定的标准流对象：

- `System.in`: 标准输入流 (InputStream), 默认绑定**键盘输入**;
- `System.out`: 标准输出流 (PrintStream), 默认绑定**控制台屏幕正常输出**;
- `System.err`: 标准错误输出流 (PrintStream), 默认绑定**控制台红色高亮报错**, 无缓冲立即刷盘。



System.out (标准输出)

- 带有行缓冲机制；
- 命令行重定向：`java App > out.log`；
- 用于常规业务信息与交互提示。

System.err (标准错误)

- 无缓冲, 立即输出；
- 命令行重定向：`java App 2> err.log`；
- 用于异常告警、致命错误与调试堆栈。

输出方法	功能特性	典型示例
print() println() printf()	输出内容后 不换行 输出内容后 自动追加换行符 格式化输出 , 精准控制宽度与精度	print(" 请输入: ") println(" 运行完毕") printf("%d: %.2f", id, v)

占位符	对应数据类型	格式化示例与输出效果
%d	十进制整型 (byte, short, int, long)	printf("%d", 100) → 100
%f	浮点型 (float, double)	printf("%.2f", 3.14159) → 3.14
%s	字符串类型 (String)	printf("%s", "Java") → Java
%c	字符型 (char)	printf("%c", 'A') → A
%x / %X	十六进制整数 (小写/大写)	printf("%02X", 0xeb) → EB
%b / %B	布尔类型 (boolean)	printf("%b", true) → true
%%	输出百分号字符 % 本身	printf("%d%%", 95) → 95%

【知识要点】格式化语法规范: %[flags][width][.precision]conversion

- %10s: 最小字段宽度为 10, 默认**右对齐**(左侧补空格);
- %-10s: 加入减号标志 -, 改为**左对齐**(右侧补空格);
- %04d: 数字宽度为 4, 不足 4 位时**前导补零**(如 0005);
- %,d: 加入逗号标志 ,, 按**千分位**分隔(如 1,000,000);
- %8.2f: 总宽度为 8(含小数点), 保留 2 位小数。



【随堂抢答·概念检测】问题:执行以下代码后,控制台输出的内容是什么?

```
1 int id = 7;  
2 double score = 92.568;  
3 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

【知识要点】正确答案:【B】

原理解析:

- %03d: 整型宽度为 3, 数字 7 不足 3 位, 前导补零为 "007";
- %-6.1f: 保留 1 位小数 (四舍五入为 92.6, 长度为 4), 减号 - 表示左对齐, 总宽为 6, 右侧补 2 个空格输出 "92.6 ";
- 最终完整拼接结果为: ID:007|Score:92.6 |END。

【知识要点】java.util.Scanner 原理与常用方法

Scanner 将输入流切分为由空白符(空格、Tab、换行符)分隔的 **Token**(词法标记):

- `nextInt()`:扫描并返回下一个 `int` 整型数;
- `nextDouble()`:扫描并返回下一个 `double` 浮点数;
- `next()`:扫描并返回下一个以空白符结束的单词字符串;
- `nextLine()`:**读取整行内容**(以回车换行符为结束标志,并消耗掉换行符)。

【随堂找茬·避坑挑错】看看以下代码为什么在输入年龄后,根本无法输入姓名?

```
1 Scanner scanner = new Scanner(System.in);  
2 System.out.print("请输入年龄: ");  
3 int age = scanner.nextInt(); // 用户输入 "20\n"  
4  
5 System.out.print("请输入姓名: ");  
6 String name = scanner.nextLine(); // 致命 Bug: 直接读到了残留的 "\n", 瞬间跳过!  
7 System.out.println("登记: " + name + ", 年龄: " + age);
```

'2'

'0'

'\n' (残留换行符)

nextInt() 仅消费数值"20"

nextLine() 立即读到换行并返回空串!

【代码演练】标准解法:在 nextInt() 后手动清空换行缓冲区

```
1 int age = scanner.nextInt();  
2 scanner.nextLine(); // 手动消费并吃掉残留在输入缓冲区中的回车换行符!  
3 String name = scanner.nextLine(); // 此时可以正常阻塞等待用户输入姓名!
```

【代码演练】输入守卫与循环重试 (RobustInputDemo.java)

```
1 Scanner sc = new Scanner(System.in);
2 int score = -1;
3 while (true) {
4     System.out.print("请输入有效考试成绩 (0~100): ");
5     if (sc.hasNextInt()) {
6         score = sc.nextInt();
7         if (score >= 0 && score <= 100) break; // 数值合法, 跳出循环
8         System.out.println("[警告] 成绩超出 0~100 合法区间, 请重新输入!");
9     } else {
10        String invalid = sc.next(); // 消费并丢弃非法字符 (如用户输入了 "abc")
11        System.out.printf("[错误] '%s' 不是合法整数, 请重新输入! \n", invalid);
12    }
13 }
```

数学方法	方法功能说明	示例与返回值
<code>Math.abs(x)</code>	计算绝对值 $ x $	<code>Math.abs(-5) → 5</code>
<code>Math.max(a, b)</code>	返回两数中的较大值	<code>Math.max(10, 20) → 20</code>
<code>Math.min(a, b)</code>	返回两数中的较小值	<code>Math.min(10, 20) → 10</code>
<code>Math.pow(a, b)</code>	计算幂运算 a^b (返回 double)	<code>Math.pow(2, 3) → 8.0</code>
<code>Math.sqrt(x)</code>	计算算术平方根 $\sqrt{x}$	<code>Math.sqrt(16) → 4.0</code>
<code>Math.round(x)</code>	四舍五入取整	<code>Math.round(3.6) → 4L</code>
<code>Math.random()</code>	生成 [0.0, 1.0) 均匀伪随机浮点数	生成 1 ~ 100 随机整数

【知识要点】生成 $[min, max]$ 闭区间内随机整数的标准通用公式

$$\text{randomInt} = (\text{int})(\text{Math.random()} \times (max - min + 1)) + min$$

```
1 // 生成 [10, 99] 之间的两位随机数
2 int code = (int)(Math.random() * (99 - 10 + 1)) + 10;
3 System.out.printf("生成随机验证码: %02d\n", code);
```



【随堂抢答·概念检测】问题：用户在控制台输入"Hello World" 并按下回车，调用 scanner.next() 会得到什么？

- A. "Hello World"
- B. "Hello"
- C. "World"
- D. 抛出异常



【知识要点】正确答案:【B】

原理解析:

- `next()` 默认以空白符(空格、Tab、换行符)作为 Token 分隔符;
- 遇到空格即停止扫描, 因此仅返回第一个单词 "Hello";
- 若要读取整行包含空格的完整字符串, 必须使用 `nextLine()`!

【随堂找茬·避坑挑错】看看以下代码在 IDE 中为什么会提示 Resource leak: 'sc' is never closed?

```
1 public static void main(String[] args) {  
2     Scanner sc = new Scanner(System.in);  
3     int val = sc.nextInt();  
4     // 遗漏了 sc.close(); 导致资源句柄无法及时释放  
5 }
```

【知识要点】规范指南

在标准应用程序中, 使用完毕的资源应显式调用 `close()` 关闭; 但注意: 关闭绑定了 `System.in` 的 `Scanner` 会同时关闭 `System.in`, 导致后续无法再次从控制台输入! 通常在 `main` 函数最末尾统一关闭。

【随堂实操·动手练笔】任务 1: 编写 GradeEvaluationSystem.java

编写 GradeEvaluationSystem.java, 要求:

1. 从键盘录入学生姓名、平时成绩(30% 权重)与期末成绩(70% 权重);
2. 计算综合总评成绩: $Total = Regular \times 0.3 + Final \times 0.7$;
3. 使用 printf 打印规范的居中对齐成绩报告单卡片。

【随堂实操·动手练笔】任务 2: 编写 RobustInputDemo.java

编写 RobustInputDemo.java, 要求:

1. 使用 hasNextDouble() 守卫连续读取用户输入的 3 个浮点数;
2. 若用户输入英文字符, 拦截报错并提示用户重新输入, 直至全部输入正确;
3. 调用 Math.max() 输出三者中的最大值。

【工程视野】以人为本与人机工程

命令行交互的友好度与容错能力体现了“以用户为中心”的工程伦理；对误操作的引导彰显人文关怀。

【实战技巧】代码审查与辅助开发

实战建议：利用 AI 生成输入校验正则表达式，配合 Scanner 实现严格格式化校验。

【随堂实操与课后思考】课后思考与巩固

- **练习：**完善成绩评定系统，增加最高分、最低分与标准差统计。
- **思考：**在百万并发的高性能通信服务器中，为什么严禁直接使用 `System.out.println()` 输出日志？

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>

思考: Questions & Answers

“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。