

Java程 序 设 计

第 5 讲：循环结构与算法控制 (for, while 与转向控制)

- 1 本讲目标
- 2 循环一:三种循环
- 3 循环二:break 与 continue
- 4 循环三:两种经典死循环(找茬专场)
- 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. 会选:三种循环各有各的场合,随手写得出;
2. 会追踪:带 `break / continue` 的循环,能逐轮预判输出;
3. 会防:识别并修复两种经典死循环。

课堂节奏

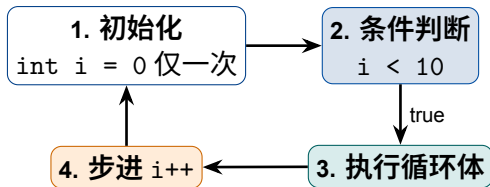
- 3 个循环,每个循环:讲 10 分钟 → 动手 5 分钟;
- 抢答先自己猜,再验证;实操限时完成,截图发课程群。

三种循环,一张表分工

语句	执行时序	适用场合
for	先判条件,再执行、再步进	次数已知:遍历、累加
while	先判条件,可能一次不跑	条件驱动:等连接、等数据
do-while	先跑一次再判条件	至少一次:菜单、输入重试

选型口诀:数得清用 for,等条件用 while,先干了再说用 do-while。

for 的四步舞:初始化、判断、执行、步进



while 与 do-while:先判后跑 vs 先跑一次

while:重试连网

```
int retry = 0;
while (retry < 3) {
    if (connect()) break;
    retry++;
}
```

do-while:主菜单

```
int choice;
do {
    showMenu();
    choice = sc.nextInt();
} while (choice != 0);
```

抢答 1: 条件永远为假, 输出几次?

【随堂抢答·概念检测】一共打印多少次"Java"?

```
int count = 0;  
do {  
    System.out.println("Java");  
    count++;  
} while (count < 0);
```

- A. 0 次
- B. 1 次
- C. 无限次
- D. 编译报错

【知识要点】先执行,后判断

- do-while 的循环体无条件先跑一次,然后才看条件;
- 跑完 `count = 1`, `1 < 0` 为假,收工——共 1 次;
- 对比:换成 `while (count < 0)` 则一次都不跑。

实操 1:猜数字游戏

【随堂实操·动手练笔】NumberGuessGame •限时 6 分钟

编写 NumberGuessGame.java:

1. 随机生成 1 ~ 100 目标数(上讲公式);
2. while 循环读入猜测,提示“大了 / 小了”;
3. 猜中后输出总次数。

完成标志:猜中并打印尝试次数 → 截图发课程群。

break 断整个, continue 跳本轮

break: 彻底收工

- 跳出当前整层循环;
- 找到即停的标配: 查找、命中退出。

```
outerLoop:
for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= 5; j++) {
        if (i * j > 10) break outerLoop; // 标签: 直接跳出外层
    }
}
```

continue: 这轮不要

- 跳过本轮剩余语句, 直接下一轮;
- 过滤的标配: 只要偶数、跳过非法行。

抢答 2: 逐轮追踪, 输出什么?

【随堂抢答·概念检测】在纸上画出每一轮 i 的去向

```
for (int i = 1; i <= 5; i++) {  
    if (i == 2) continue;  
    if (i == 4) break;  
    System.out.print(i + " ");  
}
```

- A. 1 2 3
- B. 1 3
- C. 1 3 4
- D. 1 3 5

【知识要点】逐轮回放

- $i = 1$:打印 1; $i = 2$:`continue` 跳过打印;
- $i = 3$:打印 3; $i = 4$:`break` 整层终止;
- $i = 5$ 根本没出场——输出 1 3;
- 心智模型:`continue` 想着“下一位”,`break` 想着“散会”。

找茬 1: 浮点计数器, 永远到不了 1.0

【随堂找茬·避坑挑错】这段“采样累加”为什么停不下来？

```
for (double d = 0.0; d != 1.0; d += 0.1) {  
    System.out.println("数据采样中: " + d);  
}
```

提示: 回忆上讲的 $0.1 + 0.2$ 。

【知识要点】根因与规范

- 0.1 在二进制里存不准,累加十次 $\approx 0.9999999999999999$;
- $d \neq 1.0$ 恒成立 $\rightarrow$ 永不终止;
- 规范:循环计数器只用整型;浮点终止用 $d < 1.0$ 加次数上限双保险。

【随堂找茬·避坑挑错】这段处理逻辑为什么卡死？

```
int i = 0;
while (i < 10) {
    System.out.println("处理中...");
    // 少了什么?
}
```

- A. 少了输出
- B. 少了 `i++`
- C. 少了分号
- D. 没问题

【知识要点】while 的三条体检项

- `i` 纹丝不动, `i < 10 万岁` 成立 → 死循环;
- 写 `while` 必查三件事: **初始值、推进语句、终止边界**;
- 主动的死循环另有其人: `for(;;)` 与 `while(true)`——服务器轮询的标准姿势,靠 `break` 或异常退出(第 19 讲细讲)。

【随堂实操·动手练笔】PrimeSieveDemo · 限时 5 分钟

编写 PrimeSieveDemo.java:

1. 写 isPrime(n): 试除到 $\sqrt{n}$ 即止 (剪枝);
2. 筛出 1 ~ 1000 全部素数, 每行打印 10 个;
3. 统计总个数。

完成标志: 总个数输出 168 → 截图发课程群。

1. 会选:菜单循环选哪种? 数得清次数选哪种?
2. 会追踪:1 3 那道题你预判对了吗?
3. 会防:浮点等值、漏步进——两种死循环的病根各一句话?

没做到的那条,就是课后任务的第一优先级。

工程视野

工业设备里死循环等于瘫痪:真实的通信服务器都会配看门狗(Watchdog)与最大重试次数,给失控的循环上锁。

1. **练习**: 嵌套循环打印倒直角三角形与对称菱形;
2. **AI 实战**: 让 AI 审查你的素数筛: 剪枝是否到位、嵌套是否有冗余计算。

下讲预告: 方法——把代码装进可复用的盒子(本篇压轴)。

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。