

第五章 程序控制结构

5.1 程序流程控制概述

程序的执行顺序由流程控制语句决定，主要分为三大类：

1. 顺序控制
2. 分支控制
3. 循环控制

5.2 顺序控制

特点：代码按顺序逐行执行，无跳转和分支。

示例代码：

```
int a = 10;
System.out.println(a); // 输出10
a = 20;
System.out.println(a); // 输出20
```

5.3 分支控制：if-else

5.3.1 分支控制分类

1. 单分支结构 `if`
2. 双分支结构 `if-else`
3. 多分支结构 `if-else if-...-else`

5.3.2 单分支应用与代码示例

需求：根据年龄判断是否需要负法律责任。

```
Scanner scanner = new Scanner(System.in);
System.out.print("请输入年龄：");
int age = scanner.nextInt();
if (age > 18) {
    System.out.println("需对行为负责");
}
```

5.3.3 双分支应用与代码示例

需求：完善年龄判断逻辑，加入容错处理。

```
if (age > 18) {
    System.out.println("需对行为负责");
} else {
    System.out.println("未成年免于处罚");
}
```

5.3.4 多分支应用与代码示例

需求：信用评分分级系统。

```
System.out.print("输入信用分(0-100): ");
int score = scanner.nextInt();
if (score == 100) {
    System.out.println("信用极好");
} else if (score > 80) {
    System.out.println("信用优秀");
} else if (score >= 60) {
    System.out.println("信用一般");
} else {
    System.out.println("信用不良");
}
```

5.4 嵌套分支

语法结构：在分支结构中嵌套另一个分支结构。

应用案例：歌手比赛性别分组。

```
System.out.print("输入比赛成绩: ");
double score = scanner.nextDouble();
if (score >= 8.0) {
    System.out.print("输入性别: ");
    char gender = scanner.next().charAt(0);
    if (gender == '男') System.out.println("进入男子组");
    else if (gender == '女') System.out.println("进入女子组");
} else {
    System.out.println("淘汰");
}
```

5.5 switch分支控制

5.5.1 基本语法与穿透特性

```
char c = scanner.next().charAt(0);
switch (c) {
    case 'a':
        System.out.println("周一");
        break;
    case 'b':
        System.out.println("周二");
        break;
    default:
        System.out.println("无效输入");
}
```

5.5.2 典型应用：季度判断

```
System.out.print("输入月份: ");
int month = scanner.nextInt();
switch (month) {
    case 3: case 4: case 5:
        System.out.println("春季");
        break;
    case 6: case 7: case 8:
        System.out.println("夏季");
        break;
    case 9: case 10: case 11:
        System.out.println("秋季");
        break;
    case 12: case 1: case 2:
        System.out.println("冬季");
        break;
    default:
        System.out.println("月份无效");
}
```

5.6 循环控制：for与while

5.6.1 for循环四要素与执行流程

```
for (初始化语句; 循环条件; 迭代语句) {
    // 循环体
}

// 示例：等比数列输出
for (int i = 1; i <= 5; i *= 2) {
    System.out.println(i); // 输出1,2,4
}
```

5.6.2 while循环与案例剖析

```
int count = 0;
while (count < 5) {
    System.out.println("执行次数: " + count);
    count++;
}
```

5.6.3 do-while循环本质区别

```
// 至少执行一次的输入校验
char answer;
do {
    System.out.print("确认操作? (y/n): ");
    answer = scanner.next().charAt(0);
} while (answer != 'y' && answer != 'n');
```

5.7 多重循环与算法设计

5.7.1 矩阵输出与排列组合

```
// 3阶矩阵输出
for (int i = 1; i <= 3; i++) {
    for (int j = 1; j <= 3; j++) {
        System.out.print(i * j + "\t");
    }
    System.out.println();
}
```

5.7.2 立体图形输出：金字塔代码实现

```
int totalLevel = 5;
for (int i = 1; i <= totalLevel; i++) {
    // 打印空格
    for (int k = totalLevel - i; k > 0; k--) {
        System.out.print(" ");
    }
    // 打印星号
    for (int j = 1; j <= 2 * i - 1; j++) {
        System.out.print("*");
    }
    System.out.println();
}
```

5.8 流程跳转语句

5.8.1 循环中断语句break

```
int sum = 0;
for (int i = 1; i <= 100; i++) {
    sum += i;
    if (sum > 1000) {
        System.out.println("循环终止时的i=" + i);
        break;
    }
}
```

5.8.2 循环继续语句continue

```
// 打印奇数
for (int i = 1; i <= 10; i++) {
    if (i % 2 == 0) continue;
    System.out.println(i); // 输出1,3,5,7,9
}
```

5.8.3 方法退出语句return

```
public static void main(String[] args) {
    System.out.println("程序启动");
    if (Math.random() > 0.5) {
        System.out.println("异常终止");
        return;
    }
    System.out.println("正常结束");
}
```

5.9 综合实战案例

案例1: 成绩统计系统

```
Scanner scanner = new Scanner(System.in);
double totalScore = 0;
int passCount = 0;

for (int cls = 1; cls <= 3; cls++) {
    double classSum = 0;
    System.out.println("=== 班级 " + cls + " ===");
    for (int stu = 1; stu <= 5; stu++) {
        System.out.print("输入学生成绩: ");
        double score = scanner.nextDouble();
        classSum += score;
        if (score >= 60) passCount++;
    }
    System.out.println("班级平均分: " + (classSum / 5));
    totalScore += classSum;
}
```

```
}

System.out.println("总平均分: " + (totalScore / 15));
System.out.println("及格人数: " + passCount);
```

案例2: 登录系统验证

```
String name, pwd;
int attempts = 3;

for (int i = attempts; i > 0; i--) {
    System.out.print("用户名: ");
    name = scanner.next();
    System.out.print("密码: ");
    pwd = scanner.next();
    if ("admin".equals(name) && "123456".equals(pwd)) {
        System.out.println("登录成功");
        break;
    }
    System.out.println("剩余尝试次数: " + (i-1));
}
```

本章配套练习

理论检验

- 1. 选择题
 - Q: 哪种循环至少执行一次循环体?
A) for B) do-while C) while D) switch-case
- 2. 填空题
 - Q: `int a = 10 / 3;`, a的值是_; `10.0 / 3`的结果类型是_。

编程实战

- 1. 古典问题: 斐波那契数列前20项输出
- 2. 文本处理: 实现文字版乘法口诀表
- 3. 算法设计: 统计用户输入文本中的元音字母频率
- 4. 游戏框架: 猜数字游戏实现 (数字范围1-100)

本章技术要素总结

层级结构	核心语法	开发应用场景
单分支结构	<code>if (condition) { ... }</code>	权限校验、数据过滤
多路分支结构	<code>switch (var)</code>	菜单选择、状态机处理
计数型循环	<code>for (初始化;条件;迭代)</code>	数组遍历、数学级数计算

层级结构	核心语法	开发应用场景
条件型循环	<code>while (condition)</code>	文件读取、网络请求重试机制
后验型循环	<code>do { ... } while (...)</code>	输入合法性验证、服务自检
流程中断控制	<code>break, continue</code>	循环优化、异常处理
嵌套循环设计	多重for循环嵌套	矩阵运算、游戏地图生成

题型与题目

一、选择题（每题2分，共20分）

1. 以下Java语句中，能够正确执行且打印"Adult"的是？
- A. `if (age >= 18) { System.out.print("Adult"); }`
 - B. `if age >= 18 { System.out.print("Adult"); }`
 - C. `if (age >= 18); { System.out.print("Adult"); }`
 - D. `if (age >= 18) System.out.print("Adult");`
2. switch 语句中的表达式允许的数据类型是？
- A. `float`
 - B. `String`
 - C. `boolean`
 - D. `double`
3. 下列循环代码的输出结果是什么？

```
int i = 0;
do {
    System.out.print(i + " ");
    i += 2;
} while (i < 5);
```

- A. `0 2`
 - B. `0 2 4`
 - C. `0 2 4 6`
 - D. 无输出
4. 以下代码段执行后，输出结果是？

```
for (int i = 3; i > 0; i--) {
    if (i % 2 == 0) break;
    System.out.print(i + " ");
}
```

- A. `3 2 1`
- B. `3 1`
- C. `3`
- D. `3 1 0`

5. 下列哪个运算符的优先级最高?

- A. +
- B. &&
- C. >
- D. %

6. 以下代码的作用是?

```
int count = 0;
while (count++ < 5) {
    if (count == 3) continue;
    System.out.print(count + " ");
}
```

- A. 打印 1 2 4 5
- B. 打印 1 2 3 4 5
- C. 打印 1 2 4
- D. 打印 1 2 4 5 6

7. 关于 break 语句, 正确的是?

- A. 只能跳出 switch 或一层循环
- B. 可用在 if 中终止代码块
- C. 可以结束所有嵌套循环
- D. 与 continue 功能完全相同

8. 下列代码的输出结果是?

```
for (int i = 1; i <= 2; i++) {
    for (int j = 1; j <= i; j++) {
        System.out.print(i * j + " ");
    }
}
```

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

9. 以下代码的正确打印次数是?

```
for (int i = 0; i < 10; i += 3) {
    System.out.println("Hello");
}
```

- A. 3次
- B. 4次
- C. 10次
- D. 5次

10. 以下代码的输出结果是?

```
int x = 5;
System.out.print(x > 2 ? "Yes" : "No");
```


- A. Yes
- B. No
- C. 编译错误
- D. true

二、填空题（每题2分，共20分）

1. 在Java中，终止当前循环并跳至下一次循环的语句是__。
2. do-while 循环的循环体至少执行__次。
3. 表达式 (Math.random() * 50) 的取值范围是__。
4. if (a != 0 && 100 / a > 10) 中可以避免__异常。
5. switch 语句中，若没有 break，将发生__现象。
6. for (; ;) 循环表示__循环。
7. 若 i 的初始值为10，当执行 while (i-- > 8) 时，循环体执行的次数是__。
8. 下列代码的输出结果： System.out.print(5 + 3 + "Java"); → __。
9. 如果想在循环中结束当前方法并返回值，应使用__语句。
10. 以下代码段的功能等价于一个__循环：

```
int i = 0;
for ( ; i < 5; ) { i++; }
```

三、判断题（每题1分，共10分）

1. if (a = 5) 在Java中是合法的语法。()
2. switch 语句的 case 标签可以是变量。()
3. while (true); 会导致无限循环。()
4. continue 语句可以用于跳过当前迭代的剩余代码。()
5. do-while 循环中，条件判断在循环体之前执行。()
6. for (int i = 0, j = 0; ;) 是合法的语法。()
7. 多个 if 语句的顺序不影响程序输出结果。()
8. int[] arr = {1,2,3}; for (int num : arr) 中的 num 可直接修改数组原值。()
9. return 语句只能返回一个值且必须写在方法末尾。()
10. i++ 和 ++i 在 for 循环的迭代部分中功能完全相同。()

四、简答题（每题5分，共25分）

1. 简述 switch 语句与多重 if-else 的区别及适用场景。
2. 解释 break 和 continue 的作用，并举例说明。
3. 什么是循环变量的迭代？给出 for 和 while 中的示例。
4. 如何处理用户输入非整数时的异常？（如 Scanner 输入非数字）
5. 描述如何实现一个空心金字塔的打印逻辑（无需代码，文字描述步骤）。

五、编程题（每题5分，共25分）

1. **登录验证**: 用户有3次机会输入用户名和密码 (正确为 `admin/123456`) , 超过次数提示"锁定"。
2. **闰年判断**: 输入年份, 判断是否为闰年。
3. **成绩统计**: 输入10个学生成绩, 计算平均分及高于平均分的人数。
4. **奇数累加**: 求1~100内所有奇数的和 (用 `for` 循环) 。
5. **图形打印**: 输入层数 `n` , 输出由 `*` 组成的n层等腰直角三角形。