第4章 运算符

4.1 运算符概述

运算符是用于数据运算、赋值和比较的特殊符号。Java包含以下六类运算符：

1. 算术运算符
2. 赋值运算符
3. 关系运算符（比较运算符）
4. 逻辑运算符
5. 位运算符
6. 三元运算符

4.2 算术运算符

4.2.1 基本语法与分类

运算符	描述	示例
`+`	加法	`5 + 3`→8
`-`	减法	`10 - 4`→6
`*`	乘法	`2 * 3`→6
`/`	除法	`10 / 3`→3
`%`	取模	`10 % 3`→1
`++`	自增	`i++`
`--`	自减	`j--`

4.2.2 核心知识点

运算特性

1. **整数除法**：结果只保留整数部分

```
System.out.println(10 / 4);        // 输出2（非2.5）
System.out.println(10.0 / 4);      // 输出2.5
```

示例：

```
System.out.println(-10 % 3);   // -1
System.out.println(10 % -3);   // 1
```

自增/自减陷阱

- **前++**：先自增后赋值（`++i`）
- **后++**：先赋值后自增（`i++`）

案例演示：

```
int j = 8;
int k = j++;      // k=8, j=9
int m = ++j;      // m=10, j=10
```

4.3 关系运算符（比较运算符）

4.3.1 运算符列表

运算符	描述	示例	结果
`>`	大于	`5 > 3`	true
`<`	小于	`2 < 1`	false
`==`	等于	`4 == 4`	true
`!=`	不等于	`5 != 3`	true
`>=`	大于等于	`7 >= 7`	true
`<=`	小于等于	`3 <= 2`	false

4.3.2 使用场景

主要用于条件判断（如`if`语句）和循环控制（如`while`循环）。

示例：

```
int a = 9, b = 8;
boolean flag = (a > b);  // flag=true
```

4.4 逻辑运算符

4.4.1 分类与规则

运算符	名称	规则
`&&`	短路与	两个条件均为`true`时结果为`true`
`||`	短路或	至少一个条件为`true`时结果为`true`
`!`	逻辑非	取反操作：`!true → false`
`&`	逻辑与	同`&&`，但会计算全部条件

运算符	名称	规则
`|`	逻辑或	同`||`，但会计算全部条件
`^`	异或	两个条件不同时结果为`true`（`true ^ false → true`）

4.4.2 短路机制对比

运算符	特点
`&&`	若第一个条件为`false`，跳过第二个条件判断（效率更高）
`&`	无论第一个条件是否为`false`，均执行第二个条件判断

示例：

```
if (false && ++i < 10) {}  // ++i未被执行
if (false & ++i < 10) {}   // ++i被执行
```

4.5 赋值运算符

4.5.1 基本与复合赋值

运算符	描述	等价形式	示例（设a=3）
`=`	赋值	`a = 5`	`a`变为5
`+=`	加后赋值	`a = a + 2`	`a += 2` → `a=5`
`-=`	减后赋值	`a = a - 1`	`a -= 1` → `a=2`
`*=`	乘后赋值	`a = a * 3`	`a *= 3` → `a=9`
`/=`	除后赋值	`a = a / 2`	`a /= 2` → `a=1`
`%=`	取模后赋值	`a = a % 2`	`a %= 2` → `a=1`

类型转换特性：

```
byte b = 3;
b += 2;    // 等价于 b = (byte)(b + 2)，避免类型错误
```

4.6 三元运算符

4.6.1 语法与使用

格式：
`条件表达式 ? 表达式1 : 表达式2`

规则：

- 若条件为`true`，结果取表达式1的值
- 若条件为`false`，结果取表达式2的值

示例：

```
int max = (a > b) ? a : b;  // 取a和b中的较大值
```

4.6.2 类型兼容性

要求表达式1和表达式2的类型可兼容。支持自动类型转换，例如：

```
double d = (5 > 3) ? 3 : 3.14;  // int自动转double
```

4.7 位运算符（需二进制基础）

4.7.1 基本运算符

运算符	描述	示例（二进制运算）
`&`	按位与	`0b1010 & 0b1100 → 0b1000`
`|`	按位或	`0b1010 | 0b1100 → 0b1110`
`^`	按位异或	`0b1010 ^ 0b1100 → 0b0110`
`~`	按位取反	`~0b1010 → 0b0101`（补码运算）
`<<`	左移（低位补0）	`0b1010 << 2 → 0b101000`
`>>`	右移（符号位填充）	`0b1010 >> 2 → 0b0010`
`>>>`	无符号右移（高位补0）	`0b1010 >>> 2 → 0b0010`

案例：位移运算本质

```
int a = 1 >> 2;     // 1 / 4 → 0
int c = 1 << 2;     // 1 * 4 →4
System.out.println(8 << 3);  // 64
```

4.8 运算符优先级与结合性

4.8.1 优先级表（由高到低）

优先级	运算符	结合方向
1	`()`、`[]`、`.`	从左到右
2	`!`、`~`、`++`、`--`	从右到左
3	`*`、`/`、`%`	从左到右
4	`+`、`-`	从左到右
5	`<<`、`>>`、`>>>`	从左到右
6	`<`、`>`、`<=`、`>=`、`instanceof`	从左到右

> **口诀**：单目 > 算术 > 位移 > 关系 > 逻辑 > 三元 > 赋值

4.9 标识符命名规范

4.9.1 规则（必须遵守）

1. **组成字符**：字母、数字、`_`、`$`
2. **首字符**：不能是数字
3. **避免关键字**：如`class`、`public`

4.9.2 规范（推荐遵守）

类型	命名规范	示例
包名	全小写，点分隔	`com.project.util`
类/接口	首字母大写的驼峰式	`StudentManager`
变量/方法	首字母小写的驼峰式	`userName`
常量	全大写，下划线分隔	`MAX_SIZE`

4.10 进制与转换

4.10.1 进制表示法

- **二进制**：`0b1010`
- **八进制**：`0123`
- **十六进制**：`0x1A3F`

4.10.2 进制转换方法

十进制 → 二进制

步骤:
将该数不断除以2，记录余数，倒序排列。
示例:
34 → `0b100010`

二进制 → 八进制

规则:
每3位二进制转换为1位八进制。
示例:
`0b11010101` → `0325`

十六进制 → 二进制

规则:
每1位十六进制转换为4位二进制。
示例:
`0x23B` → `0b0010 0011 1011`

4.11 键盘输入实践

4.11.1 使用步骤

1. **导入包**: `import java.util.Scanner;`
2. **创建对象**: `Scanner scanner = new Scanner(System.in);`
3. **读入数据**:

```
System.out.print("请输入姓名：");
String name = scanner.next();
System.out.print("请输入年龄：");
int age = scanner.nextInt();
```

本章习题

一、 选择题（每题2分，共20分）

1. 以下哪个表达式的结果是**整数除法**?
   A) `10.0 / 4`
   B) `10 / 4`
   C) `(double)10 / 4`
   D) `10 % 4`
2. 以下代码段的输出是?

```
int i = 5;
int j = i++ + ++i;
System.out.println(j);
```

A) 11
B) 12
C) 13
D) 14

3. 以下哪个符号代表**逻辑与**且支持短路特性?
   A) `&`
   B) `|`
   C) `&&`
   D) `||`
4. 表达式 `5 > 3 && 4 < 2` 的结果是?
   A) `true`
   B) `false`
5. 以下哪项可以正确简化代码 `a = a * 2 + 3;`?
   A) `a += 3 * 2`
   B) `a = +2 * a + 3`
   C) `a = a * (2 + 3)`
   D) **没有正确选项**
6. `int a = 0b1010;` 的值是? (十进制)
   A) 8
   B) 10
   C) 12
   D) 16
7. 以下哪个标识符是合法的?
   A) `2user`
   B) `user-name`
   C) `_age`
   D) `public`
8. `int result = (10 > 5) ? 100 : 200;` 的 `result` 值是?
   A) 100
   B) 200
9. 以下代码的输出是?

```
int a = 10, b = 4;
System.out.println(a % b + ", " + (a / (double)b));
```

A) `2, 2.5`
B) `2.5, 2`
C) `2, 2`
D) `2.5, 2.5`

10. 八进制数 `0123` 对应的十进制是？
    A) 83
    B) 123
    C) 19
    D) 91

二、 填空题（每空2分，共20分）

1. 赋值运算符 `%=` 的运算顺序是：先求余再__。
2. 逻辑异或运算符是 __，其规则是两操作数不同时结果为 `true`。
3. 表达式 `(3 == 3) ^ (5 < 2)` 的结果是 __。
4. Java中二进制数字字面量以 __ 开头。
5. 标识符命名规范中，类的命名应为__风格（如 `StudentInfo`）。
6. 将十进制数 25 转换为二进制的结果是 __。
7. Java中右移运算符 `>>` 的高位填充由__位决定。
8. 表达式 `!true || false` 的结果是 __。
9. 运算符 `++` 若在变量前（如 `++i`）称为__增。
10. Java的单一实例输入类 `Scanner` 位于 __ 包。

三、 判断题（每题1分，共10分）

1. `int a = 10++;` 是合法的语法。 （ ）
2. `0x1A3` 是合法的十六进制数。 （ ）
3. 逻辑与运算符 `&` 和短路与 `&&` 的功能完全相同。 （ ）
4. `"abc" == "abc"` 的结果是 `true`。 （ ）
5. 表达式 `10 / 0` 会抛出算术异常。 （ ）
6. `int a = 3; long b = a;` 是隐式类型转换。 （ ）
7. `~5` 的按位取反结果是 `-6`（假设使用8位二进制）。 （ ）
8. 三元运算符的条件部分必须是布尔表达式。 （ ）
9. `byte b = 256;` 能正常编译。 （ ）
10. 标识符 `goto` 是合法的变量名。 （ ）

四、 简答题（每题5分，共20分）

1. 简述 `i++` 和 `++i` 的区别，并举例说明。
2. 逻辑运算符 `&&` 和 `&` 的主要区别是什么？在什么场景下优先使用 `&&`？
3. 请写出十进制数 `-7` 的原码、反码、补码（假设8位二进制）。
4. 列举3种正确的变量命名方式，并说明Java标识符命名规范中的一个重要规则。

五、 编程题（每题10分，共30分）

1. **日期转换**：编写程序，给定总天数 `days`，输出对应的周数和剩余天数。
    - 示例输入：`17` → 输出：`2周零3天`
2. **温度转换**：编写程序，将华氏温度转换为摄氏温度。
    - 公式：`摄氏度 = 5.0/9*(华氏度-100)`
    - 示例输入：`123.45` → 输出对应摄氏度。

3. **闰年判断**：输入一个年份，判断是否为闰年。
    - 闰年条件：能被4整除但不能被100整除，或能被400整除。
    - 示例输入：`2024` → 输出：`是闰年`