

# Java 算法

作者

2019



# 目录

<b>第一章 基础</b>	<b>5</b>
1.1 基础编程模型 . . . . .	5
1.1.1 Java 程序的基本结构 . . . . .	5
1.1.2 原始数组类型与表达式 . . . . .	5
1.1.3 方法的性质 . . . . .	6
1.1.4 问题 . . . . .	6
1.1.5 作业 . . . . .	6
1.2 数据抽象 . . . . .	10



# 第一章 基础

## 1.1 基础编程模型

尽量减少对 Java 语言的依赖描述算法。

- 程序是算法精确、优雅和完全的描述；

### 1.1.1 Java 程序的基本结构

Java 的基本结构：

1. 原始数据类型：浮点数、整数、布尔值、字符型；
2. 语句：声明、赋值、条件、循环、调用、返回；
3. 数组：多个同种数据类型的集合；
4. 静态方法：封装重用代码；
5. 字符串：一连串的字符；
6. 标准输入/输出；
7. 数据抽象：数据抽象和重用代码。

### 1.1.2 原始数组类型与表达式

- 标识符：有字母、数字、下划线和 \$ 组成的字符串，首字母不能是数字；
- 逻辑运算符：优先级  $! > \&\& > ||$ ；
- 强制转换 double 转 int 为截断，而不是四舍五入；
- 别名：引用；

```
1  int[] a = new int[N];  
2  a[i] = 1234;  
3  int[] b = a;  
4  b[i] = 5678; //a[i] 也变化
```

### 1.1.3 方法的性质

1. 方法的参数按值传递：方法中改变参数的值，而不是参数本身；
2. 方法名可以被重载；
3. 方法只能返回一个值；
4. 方法可以产生副作用：void 类型的静态方法会产生副作用。

### 1.1.4 问题

1. Java 的字节码：Java 程序的低级表示，可以运行于 JVM。
2. Java 不会自动检测溢出，但是 JDK1.8 中封装了安全的四则运算。
3. Math.abs(-2147483648)=2147483648
4. 无穷大：Double.POSITIVE\_INFINITY 和 Double.NEGATIVE\_INFINITY
5. Java 未初始化使用变量抛出异常；
6. 1/0 抛出除零异常，1.0/0.0 为无穷大；
7. if <expr1> if <expr2> <stmtA> else <stmtB> 有二义性；
8. Java 不能重新读入标准输入中的值；
9. 标准输入为空后，会得到一个错误，StdIn.isEmpty() 能够帮助检查是否还有可用的输入以避免错误。

### 1.1.5 作业

1. 7; 200.0000002; true
2. double, 1.118; double, 10.0; boolean, true; String, 33
3. 算法如下：

```

1 public static void main(String[] args) {
2     if (args.length != 3) {
3         System.err.println("not equal");
4         System.exit(1);
5     }
6     if (args[0].equals(args[1]) && args[0].equals(args[2])) {
7         System.out.println("equal");
8     } else {
9         System.out.println("not equal");
10    }
11 }

```

4. then 非关键字;  $a > b$  没有括号; 正确;  $c = 0$  分号分割

5. 算法如下:

```

1 double x, y;
2 x = StdIn.readDouble();
3 y = StdIn.readDouble();
4 if ( x < 1 && x > 0 && y < 1 && y > 0) {
5     StdOut.print(true);
6 } else {
7     StdOut.print(false);
8 }

```

6. 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610

7. 结果如下:

- 3.00009(这里使用  $a_{i+1} = \frac{a_i + \frac{n}{a_i}}{2}$ )
- 499500
- 999000

8. b; 197; e

9. 代码如下:

```
1 String s = "";
2 for (int n = N; n > 0; n /= 2)
3     s = (n % 2) + s
```

10. 编译错误

```
1 java.lang.Error: Unresolved compilation problem:
2     The local variable a may not have been initialized
```

11. 代码如下:

```
1 boolean[][] dim = { {true, true}, {false, true}};
2 for (int i = 0; i < dim.length; i++) {
3     for (int j = 0; j < dim[i].length; j++) {
4         StdOut.print(dim[i][j] ? '*' : " ");
5     }
6     StdOut.print("\n");
7 }
```

12. 打印 a[i]: 0 1 2 3 4 4 3 2 1 0;

打印 i: 0 1 2 3 4 5 6 7 8 9

13. 算法如下:

```
1 int[][] b = new int[N][M];
2 for (int i = 0; i < M; i++) {
3     for (int j = 0; j < N; j++) {
4         b[j][i] = a[i][j];
5     }
6 }
```

14. 算法如下:

---



```
1 public static int lg(int N) throws Exception {  
2     boolean isPositive = true;  
3     if (N == 0) {  
4         throw new Exception("Error: N is 0");  
5     }  
6     if (N < 0) {  
7         isPositive = false;  
8         N = -N;  
9     }  
10    int result = 0;  
11    int sum = 2;  
12    while (sum <= N) {  
13        sum *= 2;  
14        result++;  
15    }  
16    return result;  
17 }
```

15. 代码如下:

```
1 public static int[] histogram(int[] a, int M) {  
2     int[] r = new int[M];  
3     for (int i:a) {  
4         if (i <= M && i > 0) {  
5             r[i - 1]++;  
6         }  
7     }  
8     return r;  
9 }
```

16. 311361142246

17. 错误。

18. 50,33,  $a*b$ ,  $2^{25}$  和  $3^{11}$

19. 算法如下:

```
1 public static BigInteger[] a = new BigInteger[100];
2 public static void main(String[] args) {
3     a[0] = BigInteger.valueOf(0);
4     a[1] = BigInteger.valueOf(1);
5     for (int i = 2; i < 100; i++) {
6         a[i] = a[i-2].add(a[i-1]);
7     }
8     for (int N = 0; N < 100; N++) {
9         StdOut.println(N + 1 + "\t" + a[N]);
10    }
11 }
```

20. 算法如下:

```
1 public static double getLn(int n) {
2     if (n == 1) {
3         return 0;
4     }
5     return Math.log(n) + getLn(n-1);
6 }
```

## 1.2 数据抽象