

Java 程 序 设 计

第 2 讲：数据类型、变量、常量与运算表达式



厦门大学嘉庚学院

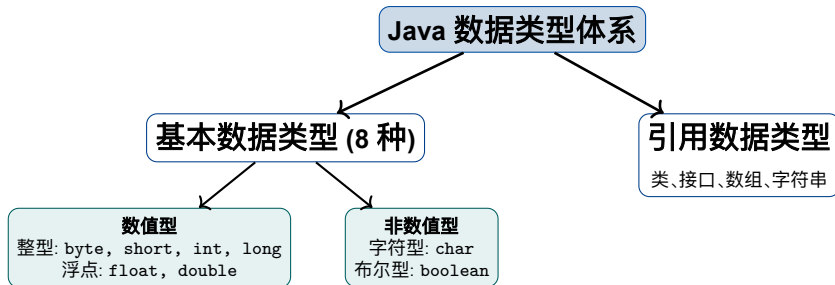
XIAMEN UNIVERSITY TAN KAH KEE COLLEGE

- 1 强类型系统与数据分类
- 2 8 大基本数据类型深度剖析
- 3 随堂概念抢答与避坑找茬
- 4 类型转换与溢出回绕
- 5 运算符、表达式与短路逻辑
- 6 常量与 final 关键字
- 7 随堂编程小练笔与实操任务
- 8 课程总结与工程视野

【知识要点】变量的物理本质

变量是内存中一块带名称的连续存储单元, 用来存放可变的数据; Java 是一门强类型静态语言:

- 所有变量必须先声明数据类型, 后使用;
- 变量一经声明, 其内存类型即固定, 不可存储不兼容类型的数值;
- 语法格式: 数据类型变量名 [= 初始值];。



整型类型	字节/比特	数学表示取值范围	常数字面量写法
byte	1 字节 (8 bit)	$-128 \sim 127 (-2^7 \sim 2^7 - 1)$	(byte) 100
short	2 字节 (16 bit)	$-32768 \sim 32767 (-2^{15} \sim 2^{15} - 1)$	(short) 1000
int	4 字节 (32 bit)	约 -21.47 亿 $\sim +21.47$ 亿	100 (整型默认)
long	8 字节 (64 bit)	$-2^{63} \sim 2^{63} - 1$	100000000000L (加 L)

float (单精度浮点 - 4 字节)

- 有效数字约 6 ~ 7 位;
- 书写字面量必须强制加 F 后缀;
- 示例:`float f = 3.14F;`。

double (双精度浮点 - 8 字节)

- 有效数字约 15 ~ 16 位;
- Java 中所有浮点字面量默认即为 `double`;
- 示例:`double d = 3.1415926535;`。

【知识要点】char 的本质是无符号 16 位整数

- char 占用 2 个字节 (16 bit), 取值范围 0 ~ 65535;
- 采用 Unicode 字符集, 可直接存储单个英文字符、数字或汉字 (如 '中');
- 支持转义字符: '\n' (换行)、'\t' (制表符)、'\' (反斜杠)。

【知识要点】boolean 类型的唯一合法取值

- boolean 只有两个字面量值: true(真) 与 false(假);
- **注意:** 在 Java 中, boolean 与整数 0 或 1 **绝不兼容**, 不能相互强转!

【随堂抢答·概念检测】问题: 以下 4 行声明中, 哪一行会直接发生编译报错?

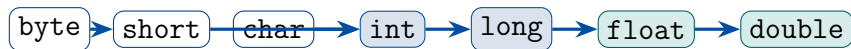
```
1 long a = 100;           // 行 1
2 long b = 2147483648;     // 行 2
3 long c = 2147483648L;    // 行 3
4 float d = 3.14;          // 行 4
```

- A. 仅行 2 报错
- B. 仅行 4 报错
- C. 行 2 和行 4 均报错
- D. 全部正常通过编译

【知识要点】正确答案:【c】

原理解析:

- 行 2: 未加 L 后缀的整数字面量默认按 `int` 处理, 而 2147483648 超出了 `int` 最大上限 (2147483647), 编译器报错: `integer number too large`;
- 行 4: 3.14 默认是 8 字节 `double` 类型, 赋给 4 字节 `float` 属于向下窄化, 报错: `lossy conversion!`



【知识要点】强转语法:(目标类型) 变量

当将大范围数据强转为小范围类型时, JVM 直接将高位二进制截断舍弃, 极易发生正负号翻转与数值失真!

```
1 int x = 258; // 二进制: 0000 0000 0000 0000 0000 0001 0000 0010
2 byte b = (byte) x; // 截取最低 8 位: 0000 0010 (结果为 2)
3 System.out.println("b = " + b); // 输出: 2
```

【随堂找茬·避坑挑错】看看以下代码输出是多少? 为什么不是 128?

```
1 byte b = 127;  
2 b++; // 自增运算  
3 System.out.println("b 的值: " + b); // 输出: -128
```

【知识要点】二进制补码原理解析

127 补码为 0111 1111, 加 1 产生进位变为 1000 0000, 最高位变为 1 (负数符号位), 真值变为 -128!

【知识要点】整型除法陷阱

在 Java 中, 当参与除法 / 的两个操作数都是整型时, 执行的是整除运算, 直接丢弃小数部分!

```
1 int a = 5, b = 2;  
2 System.out.println(a / b);           // 输出 2 (发生了截断!)  
3 System.out.println((double)a / b);    // 输出 2.5 (浮点除法)  
4 System.out.println(1.0 / 2 * 10.0);   // 输出 5.0
```

&& (短路与)

- 只要左侧为 `false`, 整个表达式必为假;
- 右侧表达式被立即跳过, 根本不执行!

|| (短路或)

- 只要左侧为 `true`, 整个表达式必为真;
- 右侧表达式同样被立即跳过!

【随堂抢答·概念检测】问题: 执行以下代码后, 变量 x 和 y 的值分别是?

```
1 int x = 5, y = 10;  
2 boolean res = (x > 10) && (++y > 10);
```

- A. $x = 5, y = 11$
- B. $x = 5, y = 10$
- C. $x = 6, y = 11$
- D. 编译报错

【知识要点】正确答案:【B】

原理解析:

- 左边表达式 ($x > 10$) 为 false;
- 触发短路保护, JVM 立即判定结果为 false, 直接跳过右侧;
- ++y 完全没有被执行, y 仍保持初始值 10!

【代码演练】常量定义三要素

```
1 public class ConstantsDemo {  
2     // 1. final 修饰不可二次修改  
3     // 2. static 全局唯一共享  
4     // 3. 变量名全大写加下划线  
5     public static final double PI = 3.141592653589793;  
6     public static final int DEFAULT_BAUD_RATE = 115200;  
7 }
```

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

编写 TemperatureConverter.java, 实现:

1. 定义摄氏度转华氏度公式: $F = C \times \frac{9}{5} + 32$;
2. 确保 $\frac{9}{5}$ 写为 9.0 / 5.0 避免整除截断;
3. 将摄氏度 26.5°C 换算为华氏度并输出。

【随堂实操·动手练笔】任务 2: 编写 TypeCastDemo.java

编写 TypeCastDemo.java, 实现:

1. 将一个大于 127 的整数 (如 258) 强制转换为 byte;
2. 观察并打印截断后的数值, 从二进制补码角度解释其输出原因。

【工程视野】数字底座与工匠精神

类型定义与精度边界是软件大厦的基石；“差之毫厘，谬以千里”，严谨对待字节取值范围，是通信工程师保障系统可靠的专业底线。

【实战技巧】代码审查与辅助开发

实战建议：利用 AI 检查整型溢出与浮点精度漂移，提示 AI 生成针对极端边界值的单元测试用例。

【随堂实操与课后思考】课后思考与巩固

- **练习：**编写三目条件运算符 $(a > b ? a : b)$ 实现三数取最大值。
- **思考：**为什么在金融结算与高精度通信计量中，推荐使用 `BigDecimal` 而不是 `double`？

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。