

# Java程序设计基础

## 数据类型和运算符

# 数据类型和运算符主要内容

- 2.1 标识符和关键字
- 2.2 变量和常量
- **2.3 数据类型（重点）**
- **2.4 成员变量和局部变量（重点）**
- 2.5 从控制台输入数据
- **2.6 运算符（重点、难点）**
- 2.7 表达式

# 本章教学目标

- 能描述标识符规则（合法和合理），能辨别正确或错误的标识符
- 能为变量取符合规则、见名知意的名字
- 能读懂编译器对误用关键字作为变量名的错误提示并修正
- 能区分变量和常量这组概念，明确final关键字的含义
- 能准确描述Java数据类型的分类、基本数据类型的名称和特点
- 能准确描述Java 4种整形类型的表数范围及推理
- 能将其他进制数据转化为十进制（手算）
- 了解字符串的定义和基本使用
- 能描述基本数据类型相互转换的规则
- 能区分成员变量和局部变量这组概念
- 能运用Scanner接收控制台各种类型的数据，并将输入数据赋予合适的变量（类型和命名）
- 能准确描述各种运算符的作用，并能运用运算符解决实际的计算问题
- 基本清晰运算符的优先级
- 能推测表达式计算结果的数据类型
- 能综合运用上述知识解决实际问题（如温度转换）

# 标识符是什么

- Java 对各种变量、方法、类和包等要素命名时使用的字符序列称为标识符。
- 标识符的规则:
- 标识符由字母、数字、下划线 `_`、美元符号 `$` 组成，并且首字符不能是数字。
- 不能把 Java 关键字作为标识符。
- 标识符没有长度限制。
- 标识符对大小写敏感（在计算机领域，“对大小写敏感”就是指“会对大写或小写英文进行区分”）。

# 哪些是合法的标识符？

- lan\_qiao
- 520lzy
- LaoLuo-520
- luo zhao yong
- luoZhaoYong
- \$YanQun\$
- #lzy#
- ^\_^
- boolean

# 标识符的命名规范

- 变量和方法名：第一个单词应以小写字母作为开头，后面的每个单词则都要以大写字母开头，例如 stuAge、sendMessage，变量尽量用名词，方法尽量用动词。
- 类名：每一个单词的首字母都需要大写，如果类名称中包含单词缩写，则这个缩写词的每个字母均应大写，例如 XMLModule（XML 是几个单词的缩写）。另外，由于类是设计用来代表对象的，所以在命名类时应尽量选择名词。
- 常量：通常全部使用大写字母，并且指出该常量完整的含义。如果一个常量名称由多个单词组成，则应该用下划线来分割这些单词，例如 MAX\_VALUE。
- 包名：通常全部用小写。

# 哪些是规范的标识符

- 变量和方法名: HelloWorld、你好、Print、sayHello、\_maxNum
- 类名: hello、Person、STUDENT
- 常量: ID\_COUNT、maxAge、\$hiddenProperty
- 包名: Com、LANQIAO、dao

# 关键字是什么

- Java 关键字对 Java 编译器有特殊的意义，它们用来表示一种数据类型或者表示程序的结构，不能用作变量名、方法名、类名和包名。
- 以及下表未包含的 `var`

abstract	assert	boolean	break	byte
case	catch	char	class	continue
default	do	double	else	enum
extends	final	finally	float	for
if	implements	import	instanceof	int
interface	long	native	new	package
private	protected	public	return	strictfp
short	static	super	switch	synchronized
this	throw	throws	transient	try
void	volatile	while		

# 什么是变量

- 变量被用于存储程序中的值
- 变量的值可以被改变
- **变量中可以存放数值、字符串、日期和对象等不同类型的数据（重点）**
- **对变量的命名要符合以下要求（重点）：**

首字母只能是：字母/汉字、下划线、\$/¥

其余字母只能是：字母/汉字、下划线、\$/¥、数字

# 什么是变量

- 定义变量的核心要素是变量类型、变量名和变量值，其声明格式如下：

```
type varName [=value];
```

- 在使用变量时，要避免出现未赋值就使用的情况

- 变量使用示例

```
int num1 = 10 ;
```

```
int num2 ;
```

```
num2 = 20 ;
```

```
float area = 123.4f ;
```

```
double cube = 987.654 ;
```

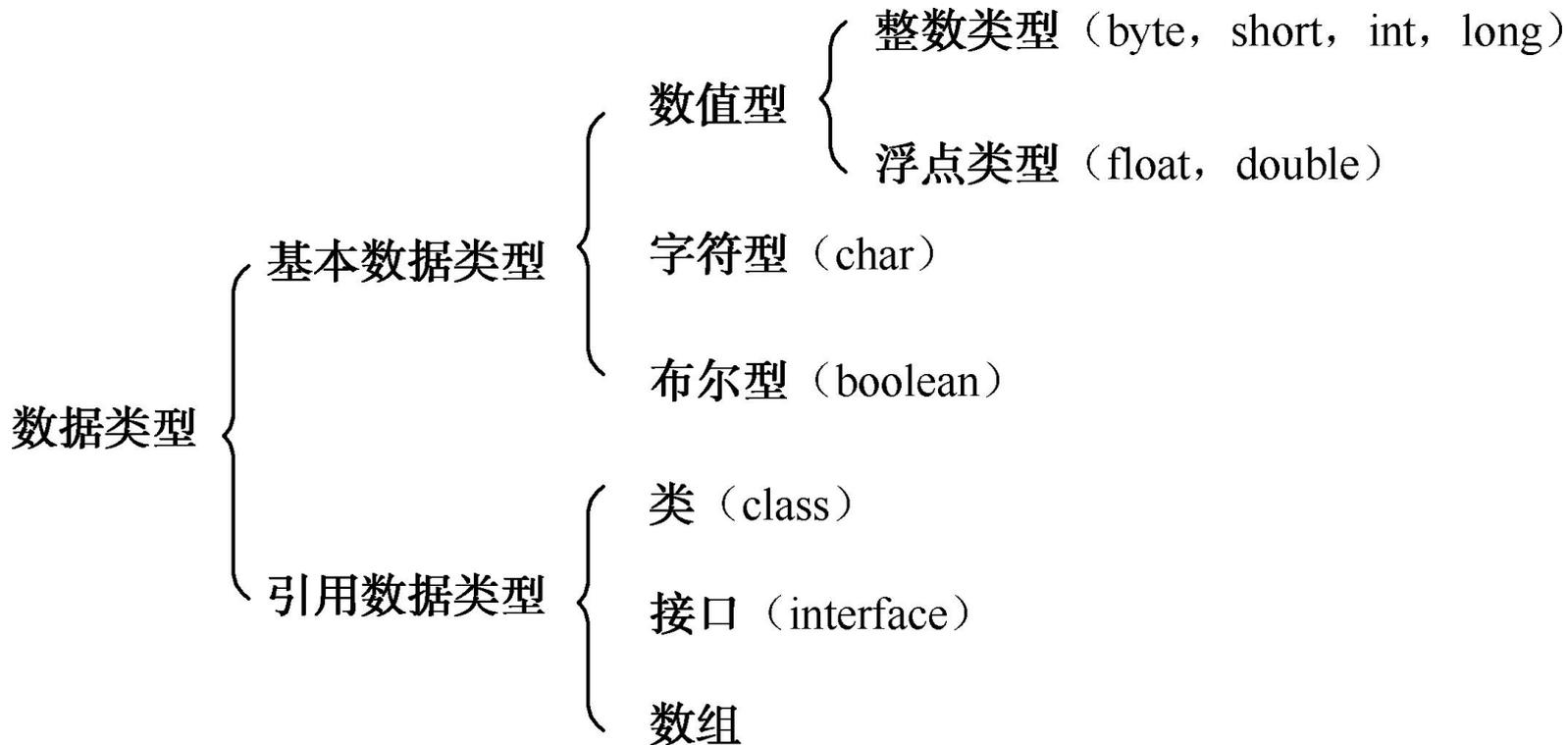
```
Student stu = new Student() ; // 定义一个学生对象 (了解)
```

# 什么是常量

- 变量的值可以被改变，常量的值不能被改变
- 在 Java 语言中，利用 `final` 关键字来定义 Java 常量  

```
final double PI = 3.141592654 ;
```
- 常量在声明的时候，要么直接进行初始化赋值，要么通过后续讲解的构造方法赋值
- 注意，直接书写的值如 “12”，“12.13” 我们称之为字面值常量，有时也简称为常量

# 变量都有哪些类型



# 各类型的特点

数据类型	字节数	位数	使用说明
byte	1	8	取值范围: $-2^7 \sim 2^7-1$
short	2	16	取值范围: $-2^{15} \sim 2^{15}-1$
int	4	32	取值范围: $-2^{31} \sim 2^{31}-1$
long	8	64	取值范围: $-2^{63} \sim 2^{63}-1$ , 直接赋值时必须在数字后加上 l 或 L
float	4	32	取值范围: $1.4E-45 \sim 3.4E38$ , 直接赋值时必须在数字后加上 f 或 F
double	8	64	取值范围: $4.9E-324 \sim 1.8E308$
char	2	16	使用 Unicode 编码 (2 个字节), 可存汉字
boolean	-	-	只有 true 和 false 两个取值

# 四种整型的特点

- 最大值：高位为 0 表示正数，其余7个位为1，十进制为127。
- 最小值：高位为 1 表示负数，其余7个位为0，十进制为-128。
- 所有的负数都是以补码的方式存储的。
- 以一个字节的-6为例：  
原码：1000 0110      反码：1111 1001      补码：1111 1010

数据类型	字节数	位数	使用说明
byte	1	8	取值范围: $-2^7 \sim 2^7-1$
short	2	16	取值范围: $-2^{15} \sim 2^{15}-1$
int	4	32	取值范围: $-2^{31} \sim 2^{31}-1$
long	8	64	取值范围: $-2^{63} \sim 2^{63}-1$ ，直接赋值时必须在数字后加上 l 或 L

# 四种整型的特点

- 两位可取的范围:
- 00
- 01
- 10
- 11
- 即 $2^2=4$ 个

# 四种整型的特点

- 127
- 01111111
- -127
- 原码: 11111111
- 反码: 10000000
- 补码: 10000001

# 计算机中常见的进制

- 二进制 (Binary) : Java中, 二进制数使用**0b**或**0B**开始, 如**0B110**表示十进制的6
- 八进制 (Octal) : Java中, 八进制数使用**0**开始, 如**014**表示十进制的12
- 十进制 (Decimal) : Java中, 默认使用10进制
- 十六进制 (Hexadecimal) : Java中, 二进制数使用**0x**或**0X**开始, 如**0XA5**表示十进制的165

# 二进制转十进制

• 1 1 0 0 1 0 1 1

• 7 6 5 4 3 2 1 0

•  $2^7+2^6+2^3+2^1+2^0=128+64+8+2+1=203$

# 十进制转二进制

- 23

11……1

5……1

2……1

1……0

0……1

从下往上，从左向右：10111

# 二进制转八进制

- 从右向左每3位按二进制转十进制的方式转换
- 1101110101
- 1565

# 八进制转二进制

- 每个数按十进制转二进制的方式转为3位二进制，不足时补齐3位
- 716
- 111 001 110

# 二进制转十六进制

- 从右向左每4位按二进制转十进制的方式转换
- 1101111101
- 37D

# 十六进制转二进制

- 每个数按十进制转二进制的方式转为4位二进制，不足时补齐4位
- 71A
- 111 0001 1010

# 浮点类型

- Java 语言浮点型字面值常量默认为 `double` 型，表示 `float` 型字面值常量需要在字面值常量后面加上 `f` 或 `F`，例如：`float floatNum = 3.14F;`
- Java 浮点类型字面值常量有以下两种表示形式。
- 十进制形式：例如 `3.14`，`314.0`，`.314`。
- 科学记数法形式：例如 `3.14e2`，`3.14E2`，`100E-2`。

<code>float</code>	4	32	取值范围：1.4E-45 ~ 3.4E38，直接赋值时必须在数字后加上 <code>f</code> 或 <code>F</code>
<code>double</code>	8	64	取值范围：4.9E-324 ~ 1.8E308

# 字符型有什么特点

- 字符字面值常量为用单引号括起来的单个字符。
- Java字符类型使用 Unicode 编码，一个 Unicode 编码占 2 个字节，字符型变量可以存放一个汉字，例如：`char eChar = 'q'; char cChar = '桥';`
- Java 字符型字面值常量有以下 3 种表示形式：
- 用英文单引号括起来的单个字符，例如'a'、'汉'。
- 用英文单引号括起来的十六进制代码值来表示单个字符，其格式为'\uXXXX'，其中 u 是约定的前缀（u 是 Unicode 的第一个字母），而后面的 XXXX 位是 4 位十六进制数，是该字符在 Unicode 字符集中的序号，例如'\u0061'等价于'a'。
- 某些特殊的字符可以采用转义符 '\ 来转换，转义符可以将其后面的字符转变为其他的含义，例如'\t'代表制表符，'\n'代表换行符，'\r'代表回车符等。

# 布尔类型

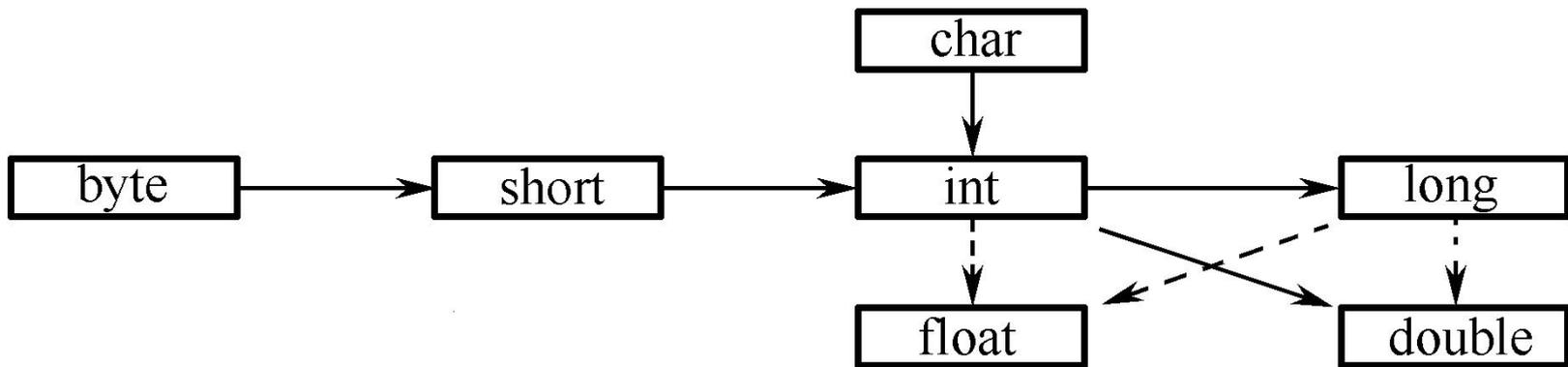
- Java 中的布尔型 (boolean) 可以表示真或假, 只允许取值 true 或 false (不可以用 0 或非 0 的整数替代 true 或 false, 这点和 C 语言不同), 例如: `boolean flag = true;`

# 字符串类型

- 字符串不属于基本数据类型。
- 字符串变量通过 `String` 来声明，字符串字面值常量则需要用半角双引号将内容包裹。字符串字面值常量的长度，可以是 0，也可以是任意数量的。
- 例如：
  - `String str1 = "";`
  - `String str2 = "Hello World" ;`

# 基本数据类型的转换

- 容量小的类型可自动转换成容量大的数据类型，如下图所示。
- byte、short、char 参与运算时会被自动转换为 int 类型。
- 容量大的数据类型转换成容量小的数据类型时，需要加上强制转换符，但可能造成精度降低或溢出，使用时需要格外注意。
- 有多种类型的数据混合运算时，系统首先自动地将参与运算的变量或字面值常量转换成容量最大的数据类型，然后再进行计算。



- 根据变量声明位置的不同，可以将变量分为成员变量和局部变量（重点）
- 成员变量定义在类的内部、方法的外部，其作用域从变量定义位置起到类结束（了解）

```
public class HelloWorld{  
    int num = 10; // num是成员变量  
    public static void main(String [] args){  
        ...  
    }  
}
```

- 类的成员有变量和方法，为示区分往往将它们称为成员变量和成员方法（了解）
- 所有成员方法可共享（使用）所有的成员变量

- 局部变量是在方法内部（含语句块）定义的变量（包括前面说到的参数），其作用域从变量定义位置起到方法（含语句块）结束（重点）

```
public static void main(String [] args) { // args是局部变量
    int num = 10 ; // num是局部变量
}
```

语句块是一对花括号“{}”包裹起来的若干代码，在语句块中也可以定义变量，也称之为局部变量

# 从控制台输入数据

- 引入 Scanner 类: `import java.util.Scanner;`
- 创建一个输入对象: `Scanner input = new Scanner(System.in);`
- 获取输入数据

# Java 中的运算符

- Java 语言支持如下运算符（重点）

算术运算符：+、-、\*、/、%、++、--、三目运算符

关系运算符：>、<、>=、<=、==、!=

赋值运算符：=、+=、-=、\*=、/= 等

逻辑运算符：!、&&、|| 等

位运算符：~、&、|、^、>>、<<、>>>（无符号右移）

- 从参与运算的因子的数量，可以将算术运算符分为以下 3 类。
  - 单目运算符：+（取正）、-（取负）、++（自增 1）、--（自减 1）。
  - 双目运算符：+、-、\*、/、%（取余）。
  - 三目运算符：(表达式 1)?(表达式 2):(表达式 3)，当表达式 1 的结果为真时，整个运算的结果为表达式 2，否则为表达式 3，该运算符是 Java 语言唯一一个三目运算符，常被使用，需要掌握。

# 关系运算符

- 关系运算符：>、<、>=、<=、==、!=
- 关系运算符的比较结果是布尔类型，例如：

```
boolean flag = 3 > 2 ; //比较结果： true
```

# 赋值运算符

- 赋值运算符：=、+=、-=、\*=、/= 等
- 赋值运算符 = 和关系运算符 == 看起来比较类似，但含义完全不同，== 用于判断两边是否相等，而 = 是将右边的值赋给左边。
- +=、-= 等是扩展的赋值运算符， $x += y$  等价于  $x = x + y$ ，程序员在实际的编程过程中，为了方便阅读，尽量不要使用这种扩展的赋值运算符。

# 逻辑运算符

- Java 语言中有 3 种逻辑运算符，它们是逻辑非（用符号 “!” 表示）、逻辑与（用符号 “&&” 表示）和逻辑或（用符号 “||” 表示）
- 逻辑非表示取反，其逻辑关系值表如下表所示

A	!A
true	false
false	true

# 逻辑运算符

- 逻辑与的运算规则为：有一个运算数为假，其值为假，两个运算数都为真，其值为真。逻辑与的关系值表如下表所示

A	B	A && B
false	false	false
true	false	false
false	true	false
true	true	true

- 逻辑或的运算规则为：有一个运算数为真，其值为真，两个运算数都为假，其值为假。逻辑或的关系值表如下表所示。

A	B	A    B
false	false	false
true	false	true
false	true	true
true	true	true

- 在计算机中，所有的信息都是以二进制形式存储的。可以用位运算符对整数的二进制位进行操作
- 位运算符主要包括按位非（用符号  $\sim$  表示）、按位与（用符号  $\&$  表示）、按位或（用符号  $|$  表示）、按位异或（用符号  $\wedge$  表示）和移位运算符（用符号  $\ll$ 、 $\gg$ 、 $\ggg$  表示）
- 按位非表示按位取反，其关系值表如下表所示

<b>A</b>	<b><math>\sim A</math></b>
1	0
0	1

# 位运算符

- 按位与是逐位逻辑与。按位与的关系值表如下表所示

A	B	A & B
1	1	1
1	0	0
0	1	0
0	0	0

# 位运算符

- 按位或是逐位逻辑或。按位或的关系值表如下表所示

A	B	A 或 B
1	1	1
0	1	1
1	0	1
0	0	0

# 位运算符

- 按位异或是当两个运算位不同时（重点在“异”字），结果为 1，否则为 0。按位异或的关系值表如下表所示

A	B	$A \oplus B$
1	1	0
0	1	1
1	0	1
0	0	0

# 什么是表达式

- 表达式是符合一定语法规则的运算符和操作数的组合。
- 例如:  $(a-b) * c - 4$        $(x>y)\&\&(m<=n)$

优先级	运算符	结合性
1	<code>()</code> , <code>[]</code>	从左向右
2	<code>!</code> , <code>+</code> (正), <code>-</code> (负), <code>~</code> , <code>++</code> , <code>--</code>	从右向左
3	<code>\</code> , <code>*</code> , <code>/</code> , <code>%</code>	从左向右
4	<code>+</code> (加), <code>-</code> (减)	从左向右
5	<code>&lt;&lt;</code> , <code>&gt;&gt;</code> , <code>&gt;&gt;&gt;</code>	从左向右
6	<code>&lt;</code> , <code>&lt;=</code> , <code>&gt;</code> , <code>&gt;=</code> , <code>instanceof</code>	从左向右
7	<code>==</code> , <code>!=</code>	从左向右
8	<code>&amp;</code> (按位与)	从左向右
9	<code>^</code>	从左向右
10	<code> </code>	从左向右
11	<code>&amp;&amp;</code>	从左向右
12	<code>  </code>	从左向右
13	<code>?:</code>	从右向左
14	<code>=</code> , <code>+=</code> , <code>-=</code> , <code>*=</code> , <code>/=</code> , <code>%=</code> , <code>=</code> , <code>&amp;=</code> , <code> =</code> , <code>^=</code> , <code>~=</code> , <code>&lt;&lt;=</code> , <code>&gt;&gt;=</code> , <code>&gt;&gt;&gt;=</code>	从右向左

# 思考以下表达式的值是什么

```
int i = 0;  
int j = 1;  
int k = j+++i;
```

i、j、k的值分别是什么？

表达式 $5 \ll 2 == 3/2$ 的值是什么？

- 以下哪些是合法的变量命名 ( ) ?

A.Hello-World    B.Hello\_World    C.HelloWorld

D.2HelloWorld    E.HelloWorld2    F.\$HelloWorld2

- 以下代码的运行结果是?

```
int a = 1>0?(3<2?4:5):(6==6?7:8);
```

- 以下代码的运行结果是?

- 以下代码的运行结果是？

```
int i = 10 ;  
System.out.print(i++);  
System.out.print(i++);  
System.out.print(i);  
i = 10 ;  
System.out.print(++i);  
System.out.print(++i);  
System.out.print(i);
```

- int表示的最大值和最小值分别是什么（     ）？

提示：根据数据类型所占的字节数思考。

- 请使用程序实现以下功能

某公司举办以下抽奖活动：员工按固定格式输入自己的入职日期（格式是“MMdd”如：0212就表示2月12日入职的），之后程序会自动计算此日期所表示数字的各个位数之和（如0212的各个位数之和就是 $0+2+1+2=5$ ）。如果计算出来的数字能被3整除，就表示此员工中奖。