Java 语言基础

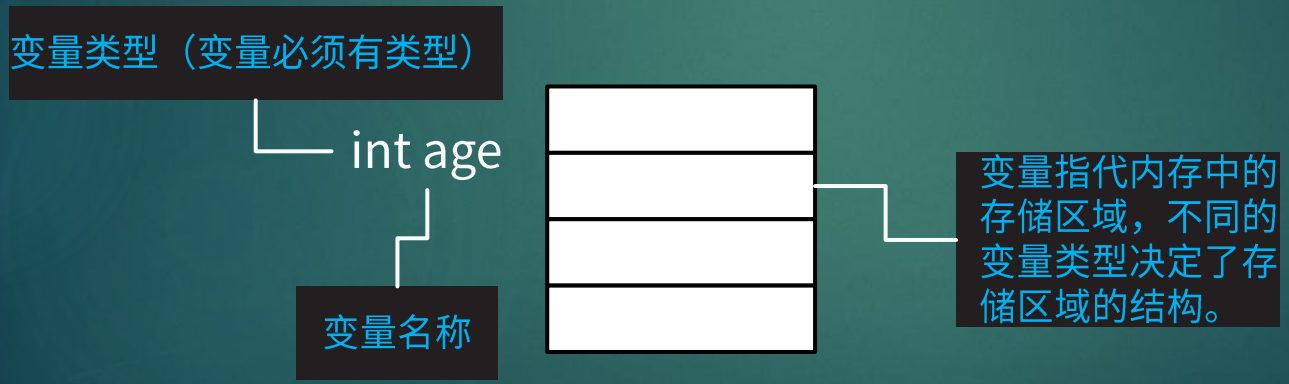
DAY02

Java 变量

Variable

什么是变量?

- 用于指代内存中的一块存储区域。
- 变量必须有特定的数据类型，不同的数据类型表示不同的数据存储结构。
- 每个变量都有自己的作用范围，叫作用域。



变量的关键词—类型

- JAVA是强类型语言，变量在使用前必须声明，指明其数据类型。编译器会根据变量的类型检测对变量的操作是否合法。

```
int a = 100;
System.out.println(a);
System.out.println(b);
a = 123.456;
```

变量没有被声明。

对变量的赋值与变量的类型不匹配。

变量的关键词—声明和初始化

- JAVA变量在使用前必须声明和初始化，及赋以确定的初值。这点和C语言不同。(属性系统会自动初始化)

```
int a;
a = 100;
System.out.println(a);
```

在第一次使用前必须赋以确定值。

```
int a = 100;
```

通常声明和赋初始值结合在一起。

```
int b;
System.out.println(b);
```

编译错误，未赋初值。

变量的关键词—标识符

- 变量名必须先是标识符，标识符命名的一些规则：
- 1 必须是字母、数字、下划线、$等，不能是随意的字符。
- 2 数字不能是第一个字符。
- 3 不能是JAVA关键字，比如：public static class ….
- 4 可以是汉字，但是不推荐使用。

- 标识符可以给类/变量/属性/方法/包 起名字。

Java关键字

- Java关键字全部都是小写。

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

Java 数据类型

data type

Java数据类型

- Java分为基本数据类型和引用数据类型。
- 基本类型有8种。
- 引用数据类型是更广泛的，包括所有的：类、接口、数组、枚举和标注。
- 枚举可以看成特殊的类，标注可以看成特殊的接口。

8种基本数据类型

类型名称	字节空间	类型说明
byte	1字节（8位）	存储1个字节数据
short	2字节（16位）	兼容性考虑，一般不用
int	4字节（32位）	存储整数（常用）
long	8字节（64位）	存储长整数（常用）
float	4字节（32位）	存储浮点数
double	8字节（64位）	存储双精度浮点数（常用）
char	2字节（16位）	存储一个字符
boolean	1字节（8位）	存储逻辑变量（true、false）

变量的输入和输出。

输入人名和年龄，打印输出。

整数类型—int

- int（32位）的最大表示范围为：$-2^{31} \sim 2^{31}-1$，即 -2147483648 ~2147483647。如果表示更大的整数可以使用long。
- 直接量(literal)， 即直接写出的常量。整数的直接量的类型默认是int类型，整数直接量也经常写16进制的形式（以0X或0x开头）。

```
int a = 100000;
int b = 0x186a0;
int c = 0303240;
int d = 10000000000;
```

16进制以0X或0x开头，8进制以0开头。

直接量超过的整数的表达范围。

进制转换

- 整数有4种进制，二、八、十六进制 和 十进制。其中，十六进制用a到f代表10到15。八进制和十六进制其实都是二进制的简写。
- 正数的二进制和十进制转换只需要按照权重相加即可。
- 正数的十进制转二进制除2取余数，反向输出。
- 负数的需要补码：按位取反，再加1 。

整数类型—long

- long（64位）的最大表示范围为：$-2^{63} \sim 2^{63}-1$（9223372036854775807）。
- 如果要表示long直接量，需要以L或l结尾。

```
long timeMillis =
System.currentTimeMillis();
System.out.println(timeMillis);
```

System.currentTimeMillis();方法返回1970年1月1日零点到此时此刻所经历的毫秒数，该方法经常用于计时操作。

```
long d1 = 10000000000;
long d2 = 10000000000l
```

直接写出的即为int类型，但超过的整数的表达范围。

浮点类型

- 浮点数，就是小数，包括： FLOAT和 DOUBLE
- DOUBLE的精度要大于FLOAT，因此，一般只使用DOUBLE计算浮点数。默认的浮点数字面量是DOUBLE类型。
- 由于舍入误差的原因，浮点数不能精确运算。

```
double money = 3.0;
System.out.println(money - 2.9);
//0.10000000000000009
```

注意舍入误差的问题，如果需要精确计算，可以使用BigDecimal。

计算自由落体运动中，物体的位移。

注：自由落体位移公式为： 0.5*9.8*T*T

$S = 1/2 \times G \times T^2$

S —位移（M）

T —时间（S）

G—重力加速度（9.8M/S）；

字符类型

- 字符类型事实上是一个16位无符号整数，这个值是对应字符的编码,
- JAVA字符类型采用UNICODE字符集编码。UNICODE是世界通用的定长字符集，所有的字符都是16位
- 字符直接量可以采用诸 ‘中’ 的形式，也可以采用16进制的表示形式，例如： ‘\u4e2d’

```
char c1 = '中';
char c2 = '\u4e2d';
```

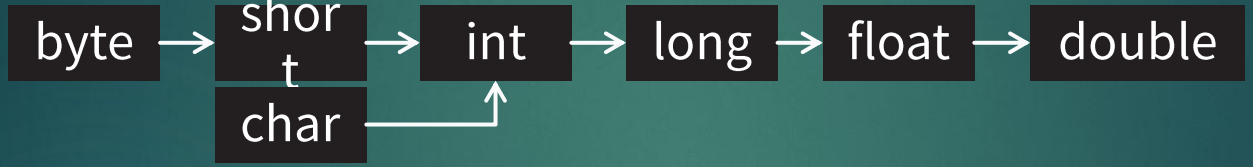
‘4e2d’ 为 ‘中’ 所对应的16位Unicode编码的16进制表示形式。

字符类型(续)

- 对于不方便输入的字符采用转义字符表示, 如: '\N' '\T' '\\' '\B' '\R' '\'' '\"' '\U4E2D‘
- 数字编码: '0':48, '1':49...
- 英文编码: 'A':65, 'B':66... ‘a’ :97
- 字符是整数, 可以参与运算
  - char c = 'A'+1
  - int n = '6'-'0’
- 整数字面量(不超范围)可以给char 变量赋值
- 控制字符是没有显示效果
- 只有本地字符集中存在的字符才能显示

基本类型的转换

- 自动类型转换（隐式类型转换）：从小类型到大类型可以自动完成：



- 强制类型转换：从大类型到小类型需要强制转换符，会造成精度损失或者溢出。

```
long l = 1024L * 1024 * 1024 * 4;
int i = (int) l;   会造成溢出！
double pi = 3.14159265358979323384;
float f = (float) pi;    会造成精度损失
```

Java 运算符及表达式

operator　　expression

算数运算

- 加（+）、减法（-）、乘（*）、除（/）、 取余（%）
- 整数相除，只能取整数部分，小数部分被舍弃。
- 整数运算时，0不能做除数；浮点运算时，0.0可以，但结果是无穷。
- 算数运算看起来简单，其实有难度。

```
int a = 120;
int b = 15;
int c = a / b;
```

结果是8，
整数除法会取整

```
int a = 5;
System.out.println(a % 2);
```

计算a除以2取余数，结果是1。

计算倒计时。

注：输入余下的秒数，打印出X小时X分X秒。

比如：输入7199，输出1小时59分59秒。

关系运算

- JAVA关系运算用于判断数据之间的大小关系。
- JAVA提供如下关系运算符：“>”（大于），“<”（小于），“>=”（大于等于），“<=”（小于等于），“==”（等于），“!=”（不等于）。
- 关系表达式的值为boolean类型（true或false）

```
int a = 100;
boolean b1 = a > 100;   // false
boolean b2 = (a + 1) >= 100;   // true
```

++和--运算

- 自增（++）、自减（--）。
- 只能用于变量，常数不可以。

```
int i = 10;
int m = i++;    ++(--)在后时，为先用后加（减）。
System.out.println(m + "," + i); // 10, 11
int n = ++i;    ++(--)在前时，为先加（减）后用。
System.out.println(n + "," + i); // 12, 12
```

今日重点

1、变量的使用，重点是各种类型变量输入输出的代码。

2、数据类型

3、整数的进制转换。

4、倒计时的代码。