

语言基础第二天：

笔记：

早期的计算机：算数(数字)

现在的计算机：算数(数字、文字、图片、音频、视频等)

计算机是离不开数据的

1. 变量：存数的，可以变化的数

- 声明：（在银行开了个帐户）

```
int a; //声明了一个整型变量，名为a
int b,c,d; //声明了三个整型变量，名为b,c,d
//int a; //编译错误，变量不能同名
```

- 初始化：（给帐户存钱）

```
int a = 250; //声明整型变量a并赋值为250
int b; //声明整型变量b
b = 250; //给变量b赋值为250
b = 360; //修改变量b的值为360
//赋值：存数的过程，初始化：第一次存数
```

- 使用：（使用的是帐户里面的钱）
 - 对变量的使用就是对它所存的那个数的使用

```
int a = 5;
int b = a + 10; //取出a的值5，加10后，再赋值给变量b
System.out.println(b); //输出变量b的值15
System.out.println("b"); //输出b，双引号中的原样输出
a = a + 10; //在a本身基础之上增10
System.out.println(a); //15
```

```
int a = 5, b = 10;
int c = a + b; //取出a的值5，加上b的值10，再赋值给c
System.out.println(c); //15
```

- 变量在使用之前必须声明并初始化

```
//System.out.println(m); //编译错误，变量m未声明

int m;
//System.out.println(m); //编译错误，变量m虽然声明了，但是未初始化
```

- 命名：
 - 只能包含字母、数字、下划线_和dollar符\$，不能以数字开头
 - 严格区分大小写

- 不能使用关键字（如public, static等）
- 允许中文命名，但不建议，建议"英文的见名知意"、"小驼峰命名法"

```
int a1, a_5$, _3c, $u2;
//int a*b; //编译错误，不能包含星号*等特殊符号
//int 1a; //编译错误，不能以数字开头
int aa = 5;
//System.out.println(Aa); //编译错误，严格区分大小写
//int class; //编译错误，不能使用class等关键字

int j; //不直观，不建议
int 年龄; //允许，但不建议用中文
int nianLing; //既不直观也不专业，必须杜绝
int age; //建议"英文的见名知意"
int score, myScore, myJavaScore; //建议"小驼峰命名法"
```

2. 八种基本数据类型: byte、short、int、long、float、double、boolean、char

- int: 整型, 4个字节, 负21多亿到正21多亿

- 整数直接量默认为int类型，但不能超出范围，若超范围则发生编译错误

```
int a = 250; //250为整数直接量，默认为int类型
//int b = 10000000000; //编译错误，100亿默认为int类型，但超出范围了
//int c = 35.678; //编译错误，int类型只能装整数
```

- 运算时若超出范围，则发生溢出，溢出不是错误，但需要避免

```
int d = 2147483647; //int的最大值
d = d+1;
System.out.println(d); //-2147483648(int的最小值)，发生溢出
```

- 两个整数相除，结果还是整数，小数位无条件舍弃(不会四舍五入)

```
System.out.println(5/2); //2
System.out.println(2/5); //0
System.out.println(5/2.0); //2.5
//除法前后只要有一个是小数，则结果是小数
```

- long: 长整型, 8个字节, 很大很大很大

- 长整型直接量需在数字后加L或l (l不建议，因为容易与1混淆，建议用L)

```
long a = 25L; //25L为长整型直接量
//long b = 10000000000; //编译错误，100亿默认为int类型，但超出int范围了
long c = 10000000000L; //100亿L为长整型直接量
//long d = 25.678; //编译错误，long类型只能装整数
```

- 运算时若有可能溢出，建议在第1个数后加L

```

long e = 1000000000L*2*10L;
System.out.println(e); //200亿
long f = 1000000000L*3*10L;
System.out.println(f); //不是300亿, 1000000000L*3已经溢出了
long g = 1000000000L*3*10;
System.out.println(g); //300亿

```

◦ double: 浮点型, 8个字节, 很大很大很大

- 浮点数直接量默认为double类型, 若想表示float, 需在数字后加F或f

```

double a = 3.14; //3.14为浮点数直接量, 默认为double型
float b = 3.14F; //3.14F为float型直接量

```

- double和float型数据参与运算时, 有可能会发生舍入误差, 精确场合不能使用

```

System.out.println(3.0 - 1.9); //0.1
System.out.println(2.0 - 1.9); //0.10000000000000009
System.out.println(1.4 - 1.0); //0.3999999999999999
//结果可能正好, 也可能发生舍入误差, 可能多一点, 也可能少一点

```

◦ boolean: 布尔型, 1个字节

- 只能取值为true或false

```

boolean b1 = true; //true为布尔型直接量
boolean b2 = false; //false为布尔型直接量
//boolean b3 = 25; //编译错误, 布尔型只能赋值为true或false

```

◦ char: 字符型, 2个字节

- 采用的是Unicode编码格式, 每个字符都对应一个码
表现的形式为字符char, 但本质上是码int(0到65535之间)
(背!) ASCII码: 'a': 97, 'A': 65, '0': 48
- 字符型直接量必须放在单引号中, 只能放1个字符

```

char c1 = '女'; //字符女
char c2 = 'f'; //字符f
char c3 = '6'; //字符6
char c4 = '*'; //字符*
//char c5 = 女; //编译错误, 字符直接量必须放在单引号中
//char c6 = ''; //编译错误, 必须有字符, 不能为空
//char c7 = '10'; //编译错误, 只能有一个字符, 这里有1和0两个字符

char c8 = 65; //相当于: char c8='A';
//系统会把'A'翻译为65再存储, 如果直接写65相当于不用翻译了, 直接存储就行
System.out.println(c8); //输出时会依据c8的类型来显示数据
//因为c8是char类型, 所以会以字符的形式显示

```

- 特殊符号需要反斜杠\来转义

```

char c9 = '\\'; //特殊符号需要通过\来转义
System.out.println(c9); //输出\

```

3. 类型间的转换:

基本数据类型从小到大依次为:

数的放一类: byte, short, int, long, float, double

char单独一类: char, int

boolean不能转

○ 两种方式:

- 自动/隐式类型转换: 小类型到大类型
- 强制类型转换: 大类型到小类型 (语法: (要转换成为的数据类型) 变量)

注意: 强转有可能会溢出或丢失精度

```
int a = 5;
long b = a; //自动/隐式类型转换
int c = (int) b; //强制类型转换

long d = 5; //自动类型转换
double e = 5; //自动类型转换

long f = 100000000000L;
int g = (int) f;
System.out.println(g); //1410065408, 强转有可能发生溢出
double h = 25.987;
int i = (int) h;
System.out.println(i); //25, 强转有可能丢失精度
```

○ 两点规则:

- 整数直接量可以直接赋值给byte,short,char, 但不能超出范围
- byte,short,char型数据参与运算时, 系统一律先将其转换为int再运算

```
byte b1 = 5;
byte b2 = 6;
byte b3 = (byte) (b1 + b2);

System.out.println(2 + 2); //4
System.out.println(2 + '2'); //52, 2加上'2'的码50
System.out.println('2' + '2'); //100, '2'的码50, 加上'2'的码50
```

补充:

1. 命名法:

- 小驼峰命名法: 第1个单词首字母小写, 其余单词首字母大写 (变量)

```
score, myScore, myJavaScore
```

- 大驼峰命名法/帕斯卡命名法: 所有单词首字母都大写 (类)

```
Score, MyScore, MyJavaScore
```

2. 数据类型分两种: 基本数据类型、引用数据类型

3. 内存单位换算:

```
1G=1024M(兆)
1M=1024KB(千字节)
1KB=1024B(字节)
1B=8bit(位)
```

4. 直接量: 直接写出来的数(5, 25, 3.14, "hello", true.....)

整数直接量: 直接写出来的整数(5, 25.....)

5. 直接量超范围为编译错误, 运算超范围为溢出

6. Unicode: 万国码、统一码、通用码, 是世界级通用的定长(16位)字符集

7. 标识符: 我们自己起的名字, 和其它名字做区分的

8.

所有变量都会存储在内存中, 内存是有大小的,

变量到底占用多大的内存, 是由它的数据类型来控制的

```
int a = 25;           //a占用4个字节
int b = 1000000000;   //b占用4个字节
```

9.

```
double b = 5.0 / 2;
System.out.println(b); //2.5
double c = 5 / 2; //5/2的结果为2, 然后将2赋值给double, 发生自动类型转换
System.out.println(c); //2.0
```

10. 明日单词:

```
1)name:姓名
2)number/num:数字
3)flag:标记
4)max:最大值
5)if:如果
6)price:价格
7)else:否则、其它的
8)operator/oper:运算符
```

11. java的8种基本数据类型是什么?(简述java的8种基本数据类型)

答:八种基本数据类型包括: byte,short,int,long,float,double,boolean,char

1)byte:字节型, 用于存储整数, 占用1个字节, 范围-128到127

2)short:短整型, 用于存储整数, 占用2个字节, 范围-32768到32767

3)int:整型, 用于存储整数, 占用4个字节, 范围-2³¹到2³¹-1

4)long:长整型, 用于存储整数, 占用8个字节, 范围-2⁶³到2⁶³-1

5)float:单精度浮点型, 用于存储小数, 占用4个字节, 不能表示精确的值

6)double:双精度浮点型, 用于存储小数, 占用8个字节, 不能表示精确的值

7)boolean:布尔型, 用于存储true或false, 占用1个字节

8)char:字符型，采用Unicode编码格式，用于存储单个字符，占用2个字节

(byte,short,int,long的范围规律: $2^7, 2^{15}, 2^{31}, 2^{63}$, 7、15、31、63有规律)