

java源文件基本组成为类；源文件取名规则：public类名与文件名等等；java应用程序执行入口是main()方法；固定格式：**Public static void main(String args[]){.....}**

注解方式：单行；多行；

标识符：

以字母、下划线、\$ 开头；由上述三种字符加数字组成；大小写敏感。

命名规范：

包的命名全部小写；类或者接口每个字符的首字母都大写，如：TaoBao；方法或者变量第一个字母全小写，第二个字母以后首字母大写，如：taoBao；常量全部大写，且每个单词之间使用_（下划线）分隔，如：TAO_BAO；

关键字：

全部都是小写；goto和const不使用但仍作为保留字。

常量：

字符常量用单引号；字符串常量用双引号。

程序执行过程？代码加载到内存中后？

基本数据类型：

4类8种：整型（byte、short、int、long）（1/2/4/8字节）、布尔类型、浮点型（float、double）（4/8字节）、字符类型（2字节）布尔类型的内存占用大小，与虚拟机有关：虚拟机中内有提供布尔值专用的字节码指令，布尔类型数据在编译后用int类型来代替，布尔数组会被编码成byte数组，所以单独使用是4个字节，数组中是1个字节。

浮点数的精度

1. java浮点数有固定的表数范围和字段长度，与机器无关，遵循IEEE754原则；
2. 浮点型数据的取值范围和精度：浮点型数据存储在内存中时，分为：符号位S，指数位E、尾数位M；
 - float型数据：符号位(S)：最高位(31位)为符号位，表示整个浮点数的正负，0为正，1为负 指数位(E)：23-30位共8位为指数位，这里指数的底数规定为2。并且指数位是以补码的形式来划分的(最高位为指数位的符号位，0为正，1为负)。另外，标准中还规定了，当指数位8位全0或全1的时候，浮点数为非正规形式(0.M)，所以指数位真正范围为：-126~127。尾数位(M)：0-22位共23位为尾数位，表示小数部分的尾数，即形式为1.M或0.M，至于什么时候是1 什么时候是0，则由指数和尾数共同决定。小数部分最高有效位是1的数被称为正规(规格化)形式。小数部分最高有效位是0的数被称为非正规(非规格化)形式，其他情况是特殊值。说明：float型数据 = $(-1)^s * 2^{(e-127)} * (1.m)$ 或者 $(0.m)$ ，即非正规和正规，当指数位8位全0或全1的时候，浮点数为非正规形式(0.M)。
 - float取值范围：由上述等式，尾数位正规式(1.m)中是大于1小于2的，非正规式(0.m)中是大于0小于1的，所以，float型数最大值一定小于 $(2^{127}) * 2$ ，即 2^{128} （8位指数位，最高位为0，其余都是1，这时适合规范式的标准，所以乘以规范式的最大值2）；同理，最小值大于 -2^{128} ；所以取值范围为： $(-2^{128}, 2^{128})$ 。

- float的精度：23位的尾数位，所代表的的数值为 $2^{23}=8\ 388\ 608$,共7位，因为1.m或者0.m；所以6位小数必精确，7位部分精确，精确范围为6到7位。
- double是11位指数位，52位尾数位，所以精度为15-16位小数。
- java默认是double；如果希望使用float类型就要在浮点数后加f或者F；与long类型同理

基础类型转换 分为自动类型转换和强制类型转换。数值类型（整型、字符型、浮点型）不能与布尔类型之间进行转换

- 自动类型转换

1. 系统会自动把表数范围小的直接赋值给表数范围大的（小->大）；
2. [char、(byte->short)] -> int -> long -> float -> double（整型->浮点）；
3. char、(byte->short)之间不会相互转换，因为计算时会先自动转换成int；
4. 如果想把基本数据类型转换成字符串类型，就用+"空字符串。

- 强制类型转换

书写格式：（数据类型）value;如果大变小，则是从低位开始截，而且是要补码转原码才能知道取值；
小知识：随机生成的验证字符串，设计原理：先随机生成一个指定范围的int数字（ASCII码字母大小写的取值范围），然后强制转换成char类型，再将多次生成的字符串起来。包装类：每个包装类都提供一个parseXXX(String str)的静态方法将字符串转换成基本数据类型。boolean -> Boolean / byte -> Byte / short -> Short / **int -> Integer** / long -> Long / **char -> Character** / float -> Float / double -> Double
表达式运算中，会把所有数据类型转换成表达式中最高的那个在进行运算，其计算值也是表数范围最高数据的类型；如：Short S = 1; S = S + 2; 这样是错误的，表达式(s+2),因为java默认整型数据是int类型，所以运算值为int类型，转换成short类型会出错。注意整型运算和浮点运算中的默认类型这个设定

引用类型

1. 所有基础类型之外的变量类型都是引用类型；
2. 占用两块内存：栈空间（变量名存对象的地址信息）和堆空间（对象）

运算符

1. ++（--）：在前时先进行+1（或者-1）运算后取值；在后时先取值后进行运算。
2. 逻辑与、或（&和|），短路与、或（&&和||）运算结果相同，但是计算过程不同：短路的话可能第二个表达式不计算

三目运算符 x? y:z

switch语句 `switch(){ case xx: break; case yy: break; default: ; }`

只能监测四种整型类型、枚举类型和String类型（不能是StringBuffer或者StringBuilder）；

break与continue

标签：对循环体进行命名，在循环体前加 标签名+，使用时，是break 标签名；。

增强for循环