# Java第二课

## 回顾:

1. java发展历史,95 09 ,java平台:javaSE(标准版) javaEE(企业版) javaME(微型版)
2. Java语言的特点:面向对象,跨平台(字节码文件->不同的平台上有不同的虚拟机,虚拟机解释字节码文件),健壮性(强类型,取消指针,不用强制回收对象),分布式,多线程,动态性等
3. Jdk,jre jvm三者关系
4. Jdk的安装和环境配置
5. IDE工具Eclipse: 工具的配置与使用
6. Java基本语法:

package day01;

public class 类名{

public static void main(String[] args){

System.out.println(“内容~”);

}

}

1. java代码的执行过程: .java 编译 字节码文件.class 执行->结果

## 课后作业讲解:

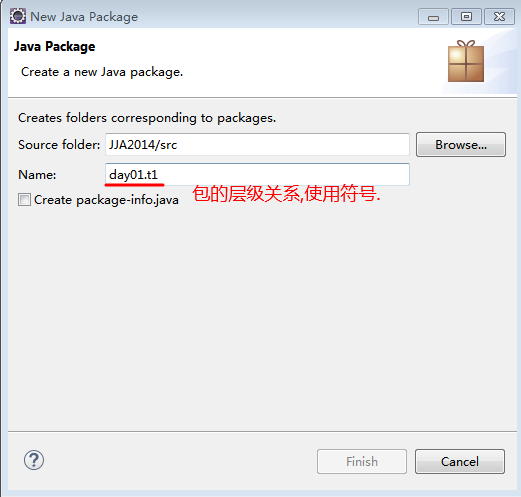
打印输出个人信息:

输出结果:

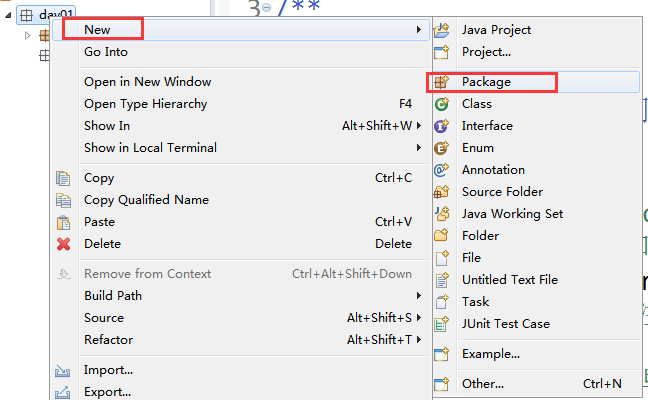
## Java基础结构

### 包 package 相当于目录

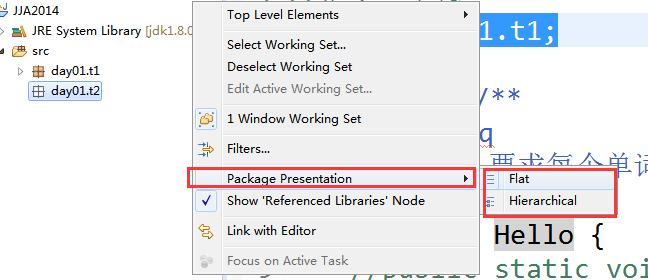
导包方式在类的第一行 : package 包路径;







切换包视图:



### 类

基本结构:

创建一个类:

class 类名{

开始写代码

}

1. class类:类是java程序的基本结构
2. class类名命名规则:首字母大写
3. 类里面包含多个方法.要看到运行结果必须有main方法
4. 方法用来存放要运行的语句
5. 每一个语句要以分号为结束
6. java程序创建类的规则:
7. java文件可以有多个class
8. 但是被public修饰的类只能有一个
9. 被public修饰的类的类名必须和java文件的文件名一致
10. 语句一定要写在方法里面
11. java程序的入口是main()
12. 包的概念:相当于一个文件夹,同一个包下java文件名不能重复,不同包下的java文件名可以重复

控制台的输出语句:

System.out.println(); 换行输出 ln-line

System.out.print(); 不换行输出

## Java中的关键字:

Java的关键字对Java的编译器有特殊的意义，他们用来表示一种数据类型，或者表示程序的结构等，关键字不能用作变量名、方法名、类名、包名和参数。如:package class void int等,这些都是java中已经定义好,我们自己定义变量,不能使用

## Java中的标识符，常量，变量:

小明有5块钱,小明和小红的钱加一起一共有12块,问小红有几块钱

解:假设小红有x块钱:

5+x=12

x=12-5; x=7;

标识符可以用来定义类，方法，变量等等，那么，这个标识符有哪些常见的规则：

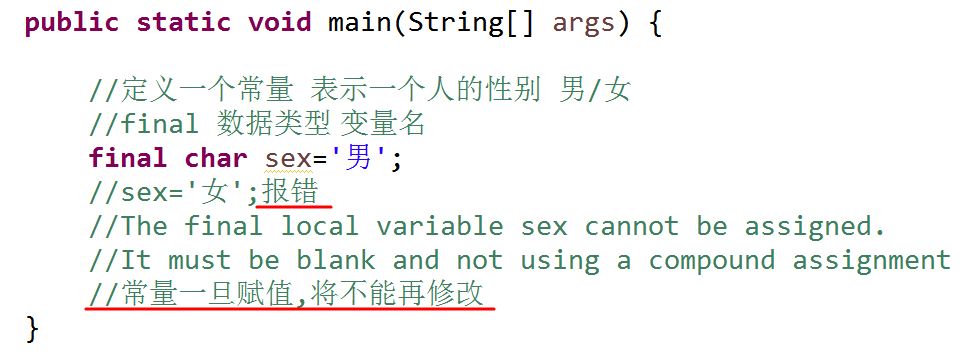
1. 标识符应该由字母 数字 下划线和$符号组成.
2. 不能以数字开头
3. 要有一定含义,尽量遵循骆驼命名的方法，第一个单词小写，后面每个单词首字母大写 studentNameBy/studentAge.
4. 在同一个作用域中，不能重复。

## 变量和常量：

常量: 在程序运行期，这个值应该一直保持不变

使用final修饰

final 数据类型 变量名



变量:通俗来说，就是和常量对应，程序运行期内是可以发生改变的，

本质来讲，变量应该是在内存中一个存储区域;

这个存储区域是有大小的，大小是由数据类型来确定的.

## java中两大数据类型

基本数据类型(8种)和引用数据类型

## 常见的数据类型：

int :整数 4字节长度

float :单精度浮点型 4字节 精确到6位

double :双精度浮点型 精确到15位 8个字节

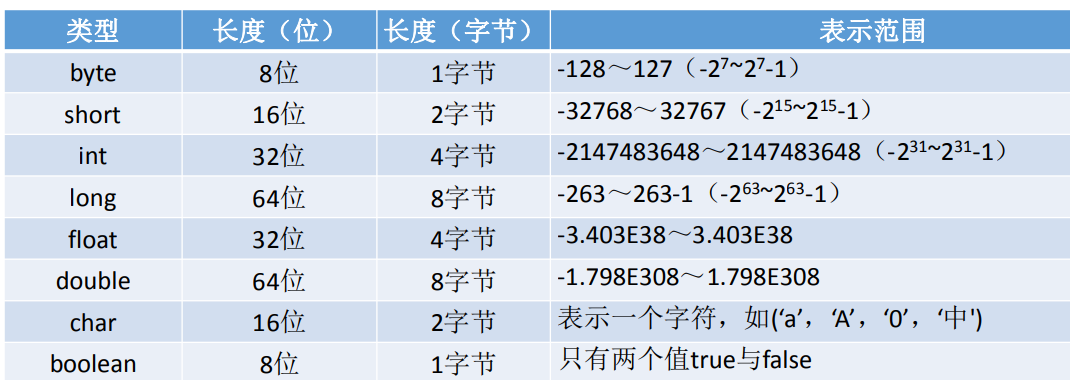
char : 字符 2字节

如何定义变量和常量:

数据类型 变量名 = 值; //将右侧的值 赋值给左侧的那个变量



**基本数据类型**:8种, 基本数据类型是存放在栈中.



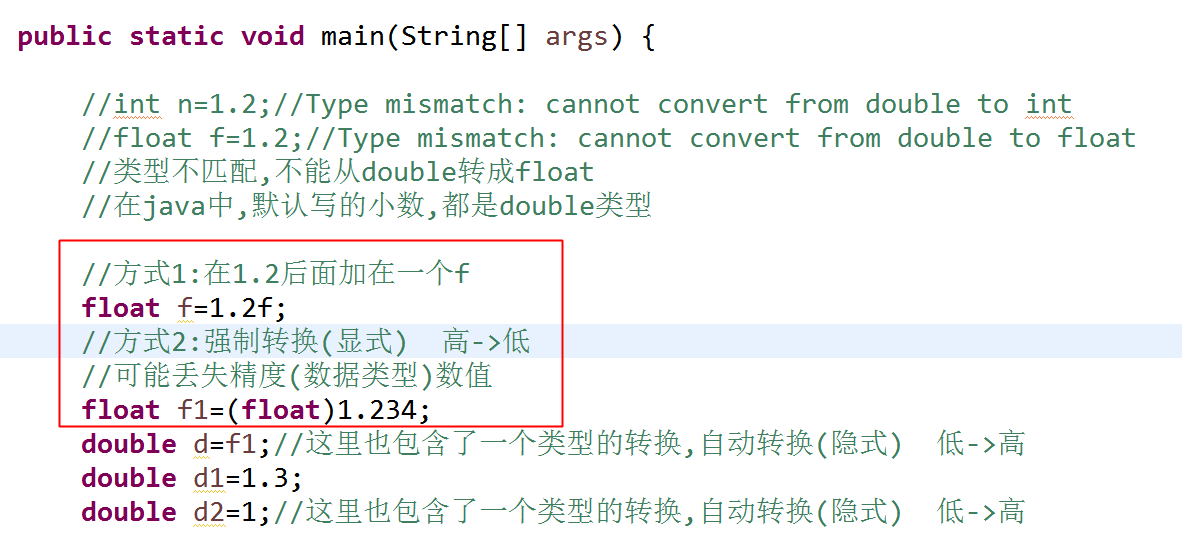
字节类型:byte

字符类型:char

布尔类型:boolean

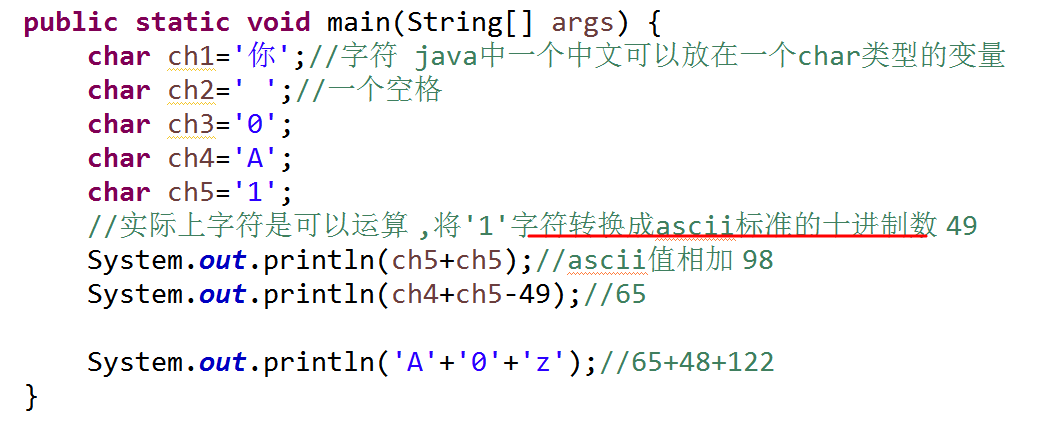
整型3种:short 短整型 int整型 long长整型 三者范围不同,由小到大

浮点型2种:float单精度 double双精度



### ASCII码表:



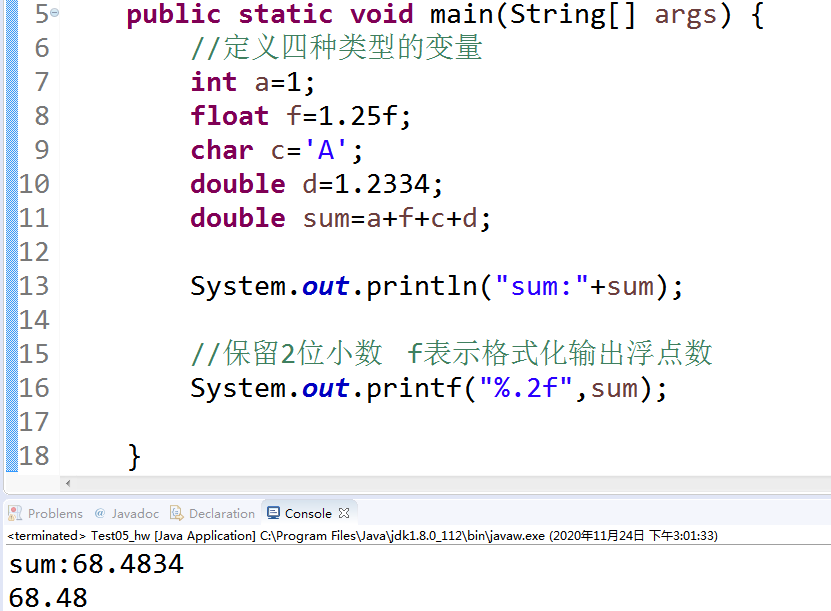


### 课堂练习:

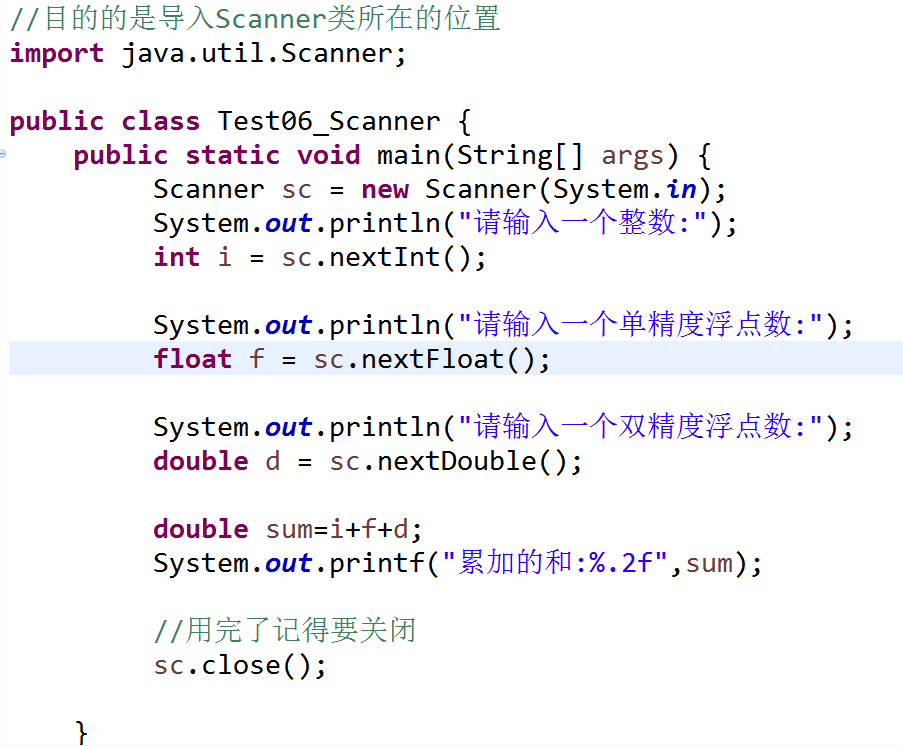
1. 实现int+char+float+double类型的求和操作;

变量定义： 数据类型 变量名 = 值;

分析: 多个数据类型累加，最终的结果是什么类型? double



1. 实现int+float+double类型的求和操作,但是要求int,float,double是从控制台输入进来的;

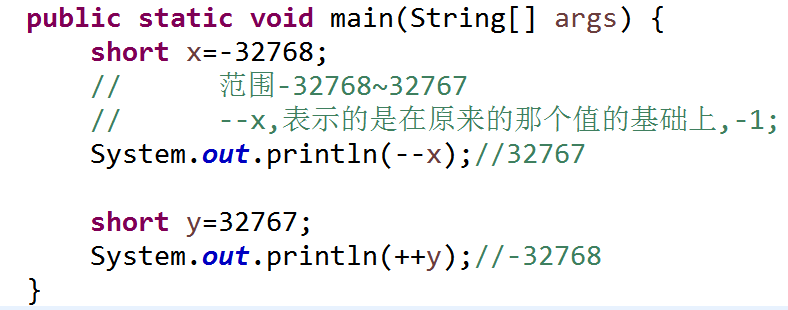


## 数据类型转换

1. 自动转换(隐式):低精度赋值给高精度
2. 强制转换(显式):高精度赋值给低精度 (数据类型)变量

## 有意思的代码

代码片段：

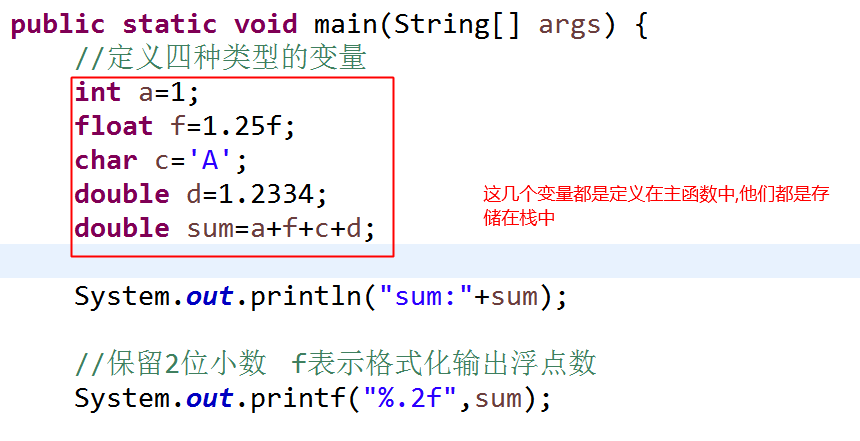


## 引用类型

引用类型: 类 接口 枚举 注解 数组等，

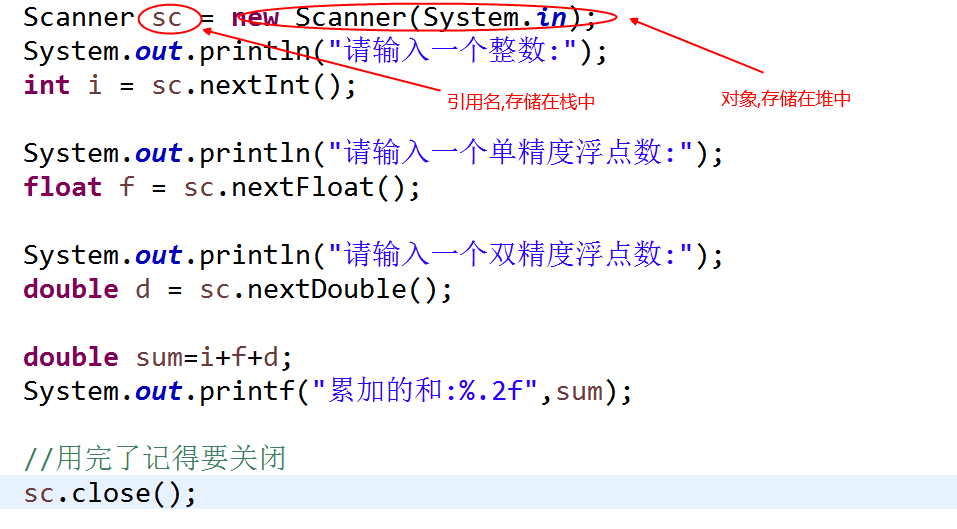
基本数据类型存储在**栈**中；~~引用类型存储在~~**~~堆~~**~~中；~~

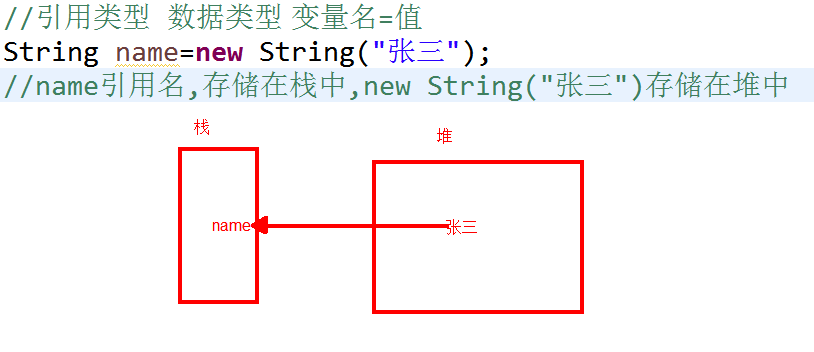
• 在函数（方法）中定义的基本数据类型变量存储在栈中；



• 引用类型实例的引用（reference）也是存储在栈中；

• 引用类型实例的成员变量，存储在堆中；

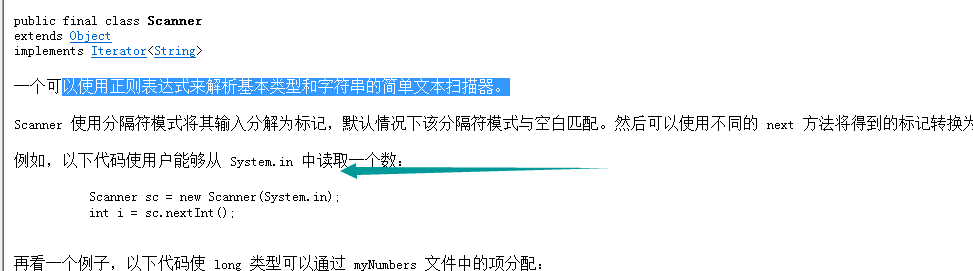




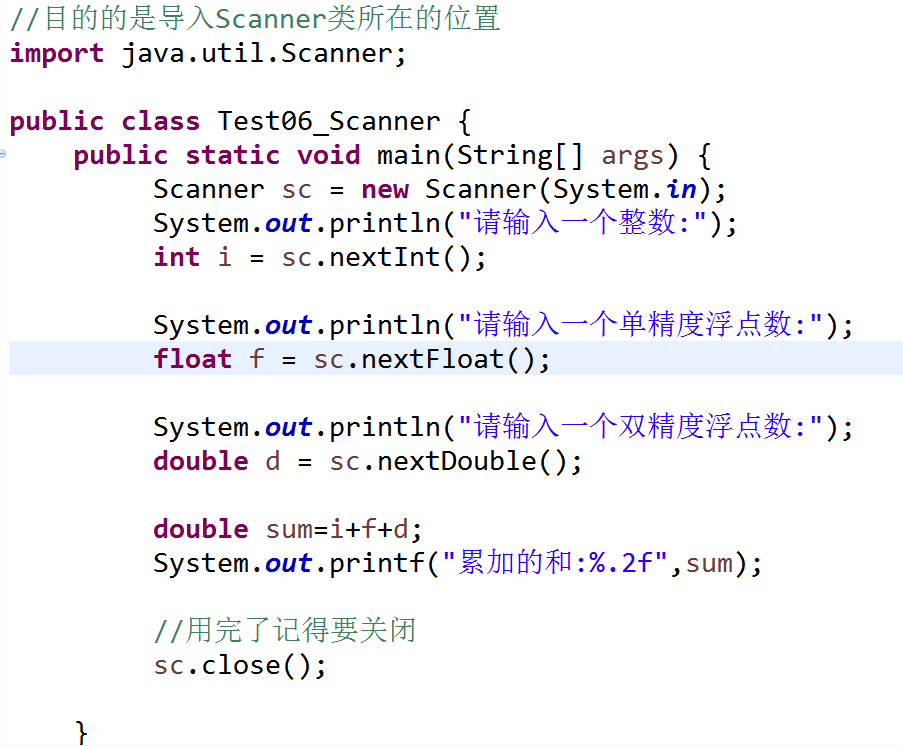
## 控制台输入:

控制台的输入:

Scanner



|  |  |
| --- | --- |
| double | **[nextDouble](mk:@MSITStore:E:\\开发十二\\API帮助文档\\JDK_API_1_6.CHM::/java/util/Scanner.html" \l "nextDouble())**()            将输入信息的下一个标记扫描为一个 double。 |
| float | **[nextFloat](mk:@MSITStore:E:\\开发十二\\API帮助文档\\JDK_API_1_6.CHM::/java/util/Scanner.html" \l "nextFloat())**()            将输入信息的下一个标记扫描为一个 float。 |
| int | **[nextInt](mk:@MSITStore:E:\\开发十二\\API帮助文档\\JDK_API_1_6.CHM::/java/util/Scanner.html" \l "nextInt())**()            将输入信息的下一个标记扫描为一个 int。 |



## 运算符:

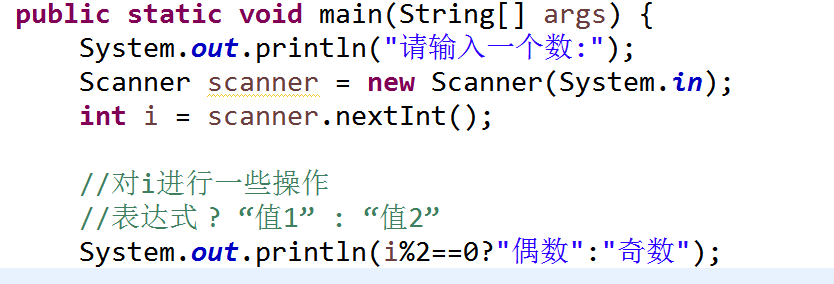
赋值运算符: = int a =1; += -= … a=a+b; a+=b

算术运算符: + - \* / %(取余) ++（自加） --（自减）

比较(关系) : == != > < >= <= 返回值是一个boolean类型的 true /false ,逻辑运算符：与&&（短路与） 或 ||（短路或） 非！取反

位运算符（了解） &

三目运算符：表达式 ?“值1”:“值2” 如果表达式为真，得到值1，否则值2



### 简单的比较问题:

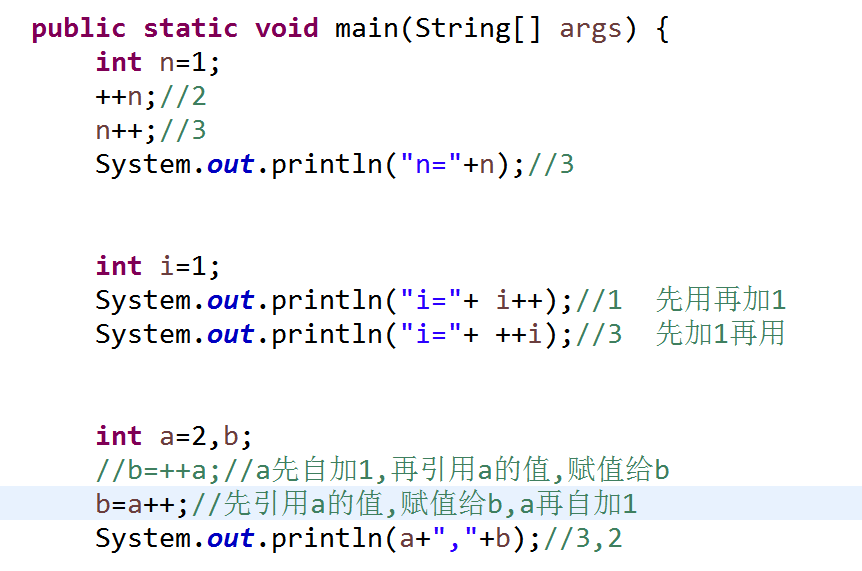


### ++和—：

（单目）来变量的基础上自加 1或者自减1 : a++ --a ；

无论是再前还是再后，**最终的结果**一定是+1或者-1;

|  |  |  |
| --- | --- | --- |
| 运算符 | 放在变量前 | 放在变量后 |
| ++ | ++a ; 先自加，再引用a的值 | a++ ; 先引用a的值，然后在自加 |
| -- | --a ; 先自减，再引用a的值 | a-- ;先引用a的值，然后再自减 |



### 关于位运算符：

#### 笔试题:

如何用比较快速的方法得到***2***的4次方?

16 8 4 2 1

1 0 0 0 0

1 0 0 0 0=16

2<<3 <<向左移动三位

00010 ->10000=>16

十进制的2转换为二进制10 > 位运算 10000 ->变成10进制

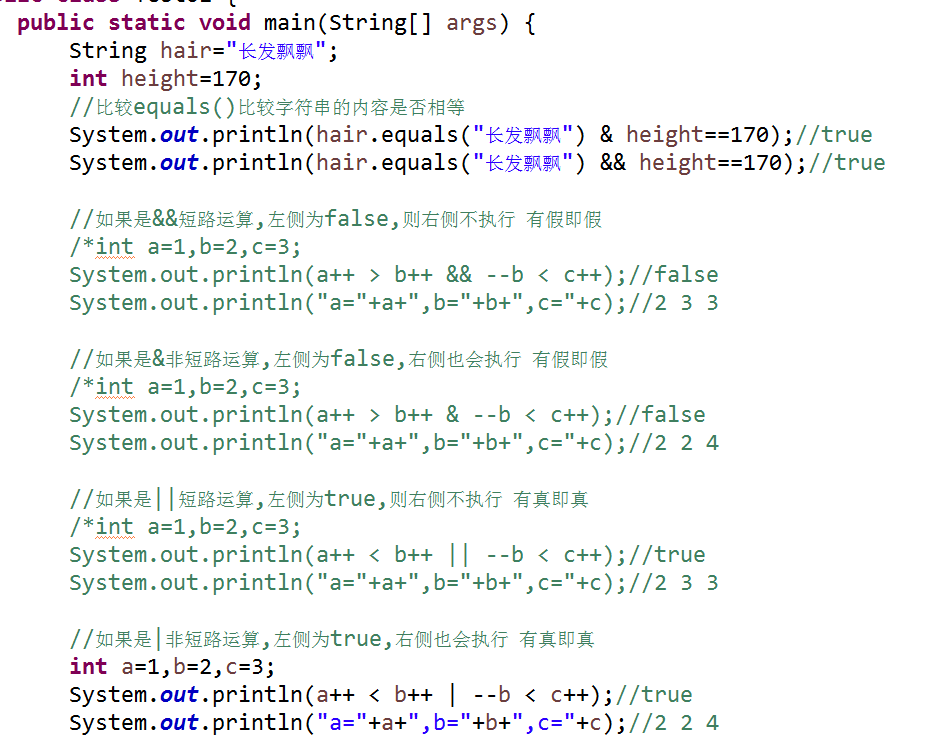
int n = 2;

System.out.println(2 << 3);

### 逻辑运算符

裴瑾: 女票

|  |  |  |  |
| --- | --- | --- | --- |
| 长发 | 170cm | 要么长发，要么170cm ,两条件都不成立，则false | 不管是否长发，两个判断都要有 |  |
| 长发 || 170cm | 要么长发，要么170cm ,两条件都不成立，则false | 如果长发，则170的条件就不判断了,直接结果true ,娶了她 |  |
| 长发 & 170cm | 长发和170要同时满足,结果才为true | &表示的非短路，长发要判断，170也要判断 |  |
| 长发 && 170cm | 长发和170要同时满足,结果true | &表示的短路，长发要判断如果为false，170就不判断 |  |



#### 面试题：&&与&的区别

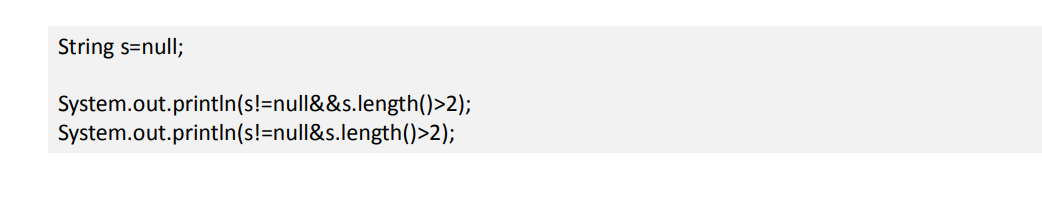
1.&&短路与逻辑运算符

2.&可以作为非短路与运算，也可以作为位运算

3. 如果是&&短路与运算,只要有一个条件为false,后面的条件都不执行

4.如果是&非短路与运算,如果有一个条件为false,后面的条件也都会执行

### 小题目(逻辑运算符)面试和考试出现的题目非常大

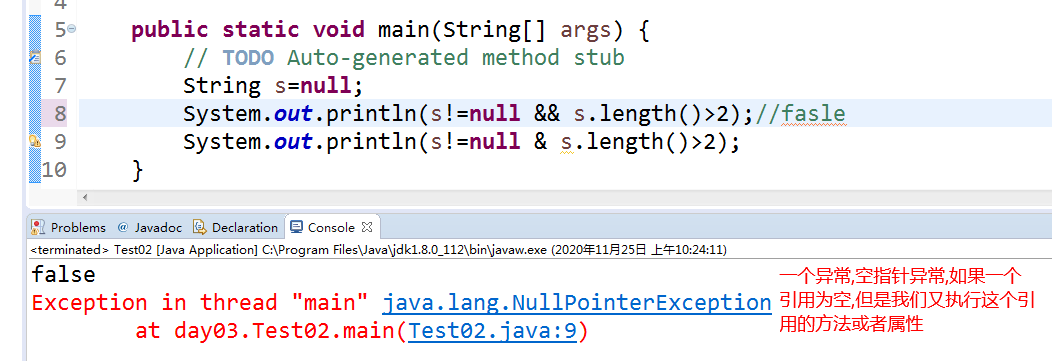


String 是一个引用类型的变量

如果引用的目标为null,那么java中是不允许出现对象名.属性或者方法的

&& :短路与 ,如果左侧是false,则右侧则不执行,false

&:非短路与,如果左侧是false,右侧也会执行,此时会出现错误



## 运算符小结:

类型分: 赋值运算符（= += -= ….）/算术(+ - \* / % ++ --) /逻辑(&& || ! & |)/关系(比较) (> < >= <= != ==) /位运算符(<<)

按照”目” :操作数 a+b => 双目/ a++ ,a – 单目/ 三目: 表达式1? 表达式2:表达式3 ;

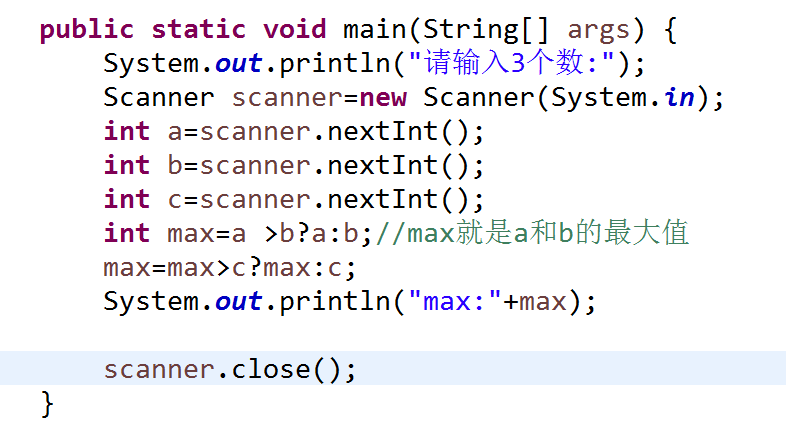
sum+=i sum=sum+i

## 运算符-优先级描述

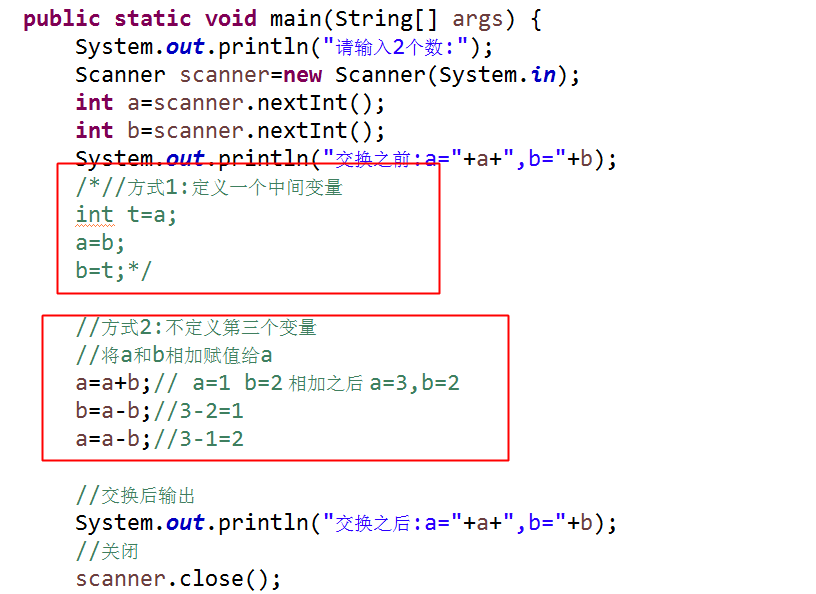
|  |
| --- |
| 1 括号()、[] |
| 2 正负号+、- |
| 3 自增自减，非++、--、! |
| 4 乘除，取余\*、/、% |
| 5 加减+、- |
| 6 移位运算<<、>>、>>> |
| 7 大小关系>、>=、<、<= |
| 8 相等关系==、!= |
| 9 按位与& |
| 10 按位异或^ |
| 11 按位或| |
| 12 逻辑与&& |
| 13 逻辑或|| |
| 14 条件运算?: |
| 15 赋值运算=、+=、-=、\*=、/=、%= |
| 16 位赋值运算&=、|=、<<=、>>=、>>>= |

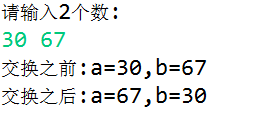
## 课堂练习:

1. 从控制台输入3个数,并求最大值; 三目



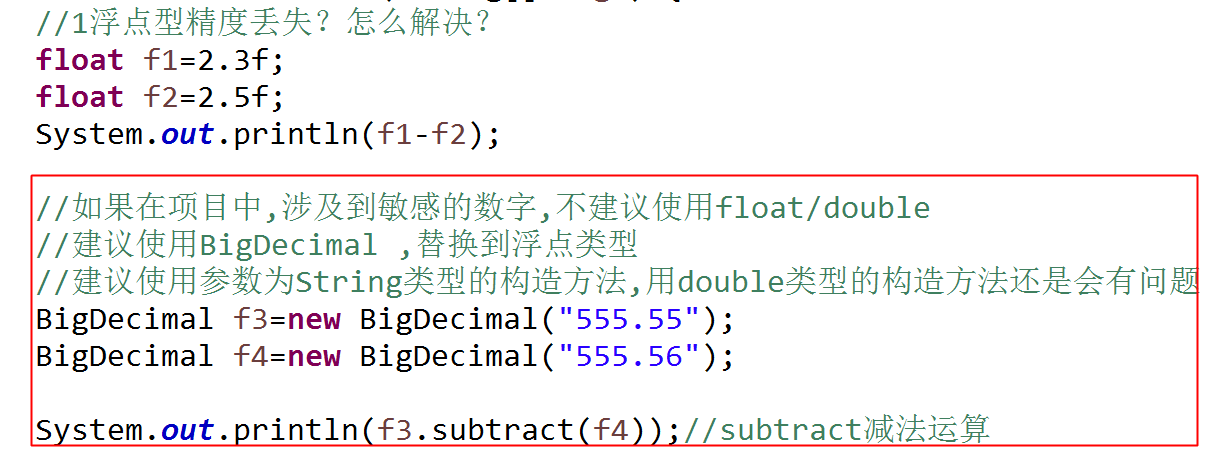
1. 从控制台输入两个数字,将这2个数字交换后输出





## 【面试题】

### 浮点型精度丢失？怎么解决？



参考:为什么浮点型精度会丢失? 计算机识别的是二进制的形式,对于小数使用二进制不能够精确表示,只能接近值.

解决方式:可以使用BigDecimal ,替换掉浮点类型,定义完浮点类型变量后再通过BigDecimal 类提供的算术方法进行操作,建议使用参数为String类型的构造方法.

### int的大小？

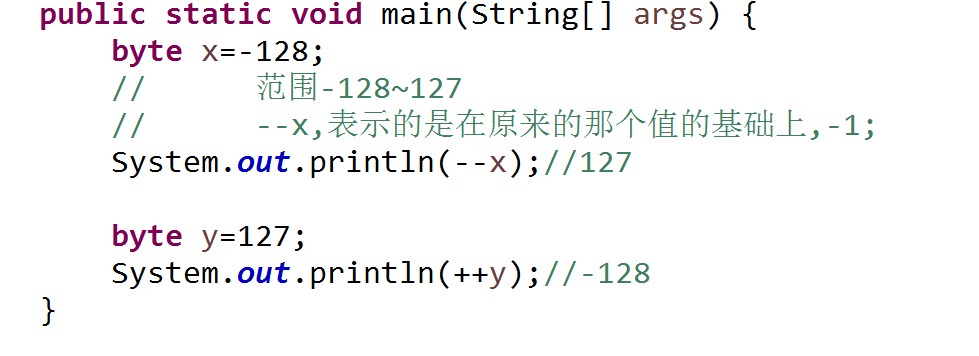
int占4个字节 32位(1字节=8位) 范围: -2的31次方到+2的31次方-1

int 类型变量数值最大是2147483647(10位)

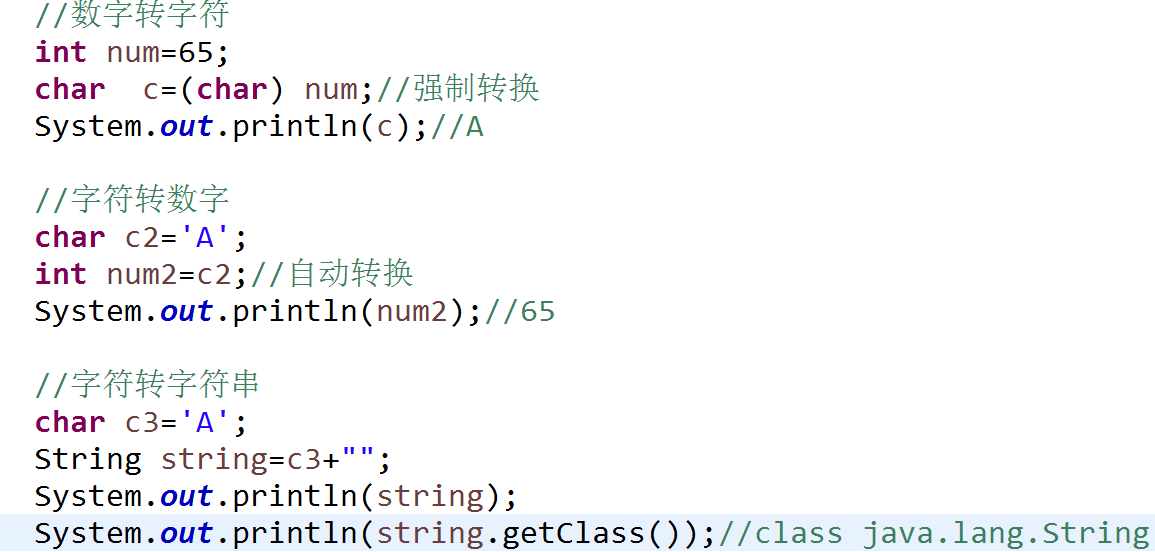


### byte的大小？最大值+1的结果会是怎样？

范围:-128~127



### int 和 char是否可以相互转换？



## 总结：

1. 类结构,变量,常量(final),声明，赋值，定义，命名规则等.
2. 数据类型(8个基本类型 byte .short.int.long.float.double.char.boolean),存储在栈中
3. 引用类型:引用名存储在栈中,对象(内容)存储在堆中
4. 运算符(= ==)
5. 控制台的输入(Scanner)

## 作业: